AN INVESTIGATION OF THE SOLUTION
TO THE OPTIMAL POWER FLOW PROBLEM
INCORPORATING CONTINUATION METHODS

by

© Maurice Huneault

B.Sc.A (Université Laval)
M.Sc. (Université Laval)

A thesis submitted to the faculty of Graduate
Studies and Research in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Department of Electrical Engineering
McGill University
Montréal, Canada
August, 1988

ABSTRACT

This thesis analyzes and tests some new solution techniques for the optimal power flow problem. This new methodology exploits a parametric technique, called the continuation method, which is applied to different tasks in the solution procedure. In a first application, the continuation method solves the quadratic subproblems generated sequentially by the optimal power flow's nonlinear program. It first creates a simple subproblem, which is easy to solve, and then links it to the subproblem we wish to solve. Starting at the solution of the simple problem, it generates optimal solution trajectories for the intermediate problems, leading to the desired optimal solution. Solution times are often advantageous, because this technique avoids the lengthy combinatorial search required in conventional methods to locate the set of active constraints. Furthermore, the solution trajectories are often very useful in themselves. In a second application, the algorithm tracks optimal solutions trajectories of the nonlinear problem when the load is slowly varied. This constitutes an example of "incremental loading', a technique already used for real power dispatch, but in this case a complete network model is used. The flexibility of the algorithm at various levels allows for some excellent computation times in this load-tracking mode: we have observed reductions in computation times for new solutions of the order of 70%, compared to the computation time of the initial load.

This thesis first presents an analysis of the various structures used in optimal power flow algorithms. Then, having chosen and presented the structure of our algorithm, we analyze the quadratic subproblems generated by this algorithm for some of its more important tasks: minimum cost, minimum losses and load shedding. New rules are proposed to link the solutions of successive subproblems to ensure the convergence of the nonlinear problem. Then, as a final contribution to the theory, some extensions are suggested for the subproblems: among them are ramp constraints, bus incremental costs, and provisions for redispatching.

Numerical simulations of the proposed optimal power flow algorithm using the minimum fuel cost task were performed on four test systems, with sizes

ranging from 6 to 118 buses. The results are documented in detail, and results for the 30 bus test are compared to those reported by other authors. All in all, our results demonstrate quite well the potential of this technique.

RESUME

Cette thèse fait l'analyse et l'essai de nouvelles techniques de solution pour le problème de l'écoulement optimal de puissance. Cette nouvelle méthodologie exploite une technique paramétrique, appelée la méthode de continuation, pour résoudre plusieurs des tâches du problème. Dans un premier temps, cette technique résoud les programmes quadratiques générés sequentiellement par le programme nonlinéaire qu'est l'ecoulement optimal de puissance. Elle crée dabord un problème simple, plus facile à résoudre, et ensuite elle le relie au problème à résoudre. A partir de la solution du problème simple, elle crée des trajectoires de solutions optimales intermédiaires, se terminant à la solution désirée. Les temps de calcul utilisant cette methode sont souvent avantageux, car elle évite les longues recherches combinatoires; ces dernières sont requises dans les méthodes conventionnelles, pour trouver les contraintes actives. De plus, les trajectoires sont souvent très utiles en soi. Dans un deuxième temps, l'algorithme suit la trajectoire des solutions optimales du problème nonlinéaire lorsque la charge est variée lentement. Cela constitue effectivement une application du "chargement incrémental", dont le principe est déjà utilisé dans l'exploitation du réseau, mais ici on profite d'un modèle complet du réseau. La flexibilité de l'algorithme à tous les niveaux permet d'obtenir d'excellents temps de calcul à cette étape: nous observons des reductions des temps de calcul des nouvelles solutions de l'ordre des deux tiers, par rapport au temps de calcul pour la première charge.

Cette thèse présente dabord une analyse des structures des algorithmes de solution pour l'écoulement optimal de puissance. Ayant choisi et présenté la structure de notre algorithme, nous analysons ensuite les programmes quadratiques générés par cette methode pour quelques tâches importantes: le coût minimum, les pertes minimum, et le délestage de charge. Nous proposons aussi de nouvelles règles pour relier les programmes quadratiques de façon à assurer la convergence du problème nonlinéaire. Enfin, nous formulons plusieurs extensions au programmes quadratiques: entre autres, on traite les contraintes dynamiques sur les variations de génération, les coûts incrémentaux des charges individuelles, et le dispatching rapide suite à un changement dans le réseau.

De nombreux essais numériques ont étés effectués avec notre algorithme d'écoulement optimal de puissance ayant comme tâche le coût minimum des generations. Les données ont étées tirées de quatre réseaux tests, allant de 6 à 118 barres. Les résultats sont documentés en détail, et ceux du système à 30 barres sont comparés aux resultats publiés par d'autres auteurs. Dans l'ensemble, nos resultats démontrent assez bien le potentiel de cette technique.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to Prof. F.D. Galiana for his expert guidance, encouragement, friendship and generosity during the course of this research.

The warmth and friendship of Prof. B.T. Ooi is also greatly appreciated.

Thanks are also due to Prof. Š. Zlobec from the Dept. of Mathematics, for the valuable consultations and interesting discussions on the theoretical and the "philosophical" questions in optimization.

I would like to express my appreciation to Prof. C.C.Paige, Prof. P.R. Bélanger, and Dr. A. Moshref for the use of various software packages.

I am particularly indebted to my many friends and colleagues in the power group for their help, their good-natured company and their enthusiasm. Many thanks to the "OPF people ", Alireza Vojdani, Mariam Juman, and especially Ranendra Ponrajah for sharing their knowledge. Special thanks also to the "quasi-OPF people" Danny Fok and Fulin Zhuang, and in particular to my good friend Rodolfo Calderon. Sincere appreciation is also extended to John Cheng, Juan Dixon, Huang Jinan and Marco Tremblay for their friendship.

The assistance and friendship of Mrs. P. Hyland, Mrs. P. Menon and Mrs. R. Pinzarrone of the department staff are also appreciated.

The financial support from the provincial FCAR fund, the National Sciences and Engineering Research Council of Canada, and the McGill Graduate Faculty awards are gratefully acknowledged.

Finally, I would like to thank my family for their continuous encouragement during the course of my graduate studies.

TABLE OF CONTENTS

LIST OF SYMBOLS

a	Vector of linear terms of the objective function C(P)
a_0	Intermediate value for a scalar step size
a^A	Subvector of a corresponding to active real power generations
a'	Vector of linear terms for a general quadratic objective
A	Matrix which regroups the left-hand-sides of all the functional constraints
A_l	Matrix of LHS coefficients corresponding to inactive loads, in load shedding
A_l^A	Matrix of LHS coefficients corresponding to active loads, in load shedding
b	Vector of excursions of the independent variables in the subproblem
B	Diagonal matrix of quadratic terms of the objective function C(P)
B^A	Subbmatrix of B corresponding to active generations
B'	Matrix of quadratic terms for a general quadratic objective
b_c	Vector of reactive admittances of the shunt compensation devices
b_c^m	Vector of lower bounds on shunt compensation admittances (in p.u.)
b_c^M	Vector of upper bounds on shunt compensation admittances (in p.u.)
b_D	Vector of real and reactive power demands (in p.u.)
$b_D(\Theta)$	Vector of varying demand, in the perturbation function
b_{D0}	Vector of last satisfiable demands, in the varying demand method
Δb_D	Vector of predicted demand variation, in the varying demand method
b_e	Vector of expansion point values of independent variables
b_g	Vector of independent variables (in p.u.)
b_g^A	Subvector of b_g corresponding to active independent variables
b_g^{lim}	Vector of limits on active independent variables
b_g^m	Vector of lower bounds on independent variables (in p.u.)
b_g^M	Vector of upper bounds on independent variables (in p.u.)
b_{g0}	Vector set points for independent variables, in continuation method
Δb_g	Vector of variations for independent variables, in continuation method
BIC	Bus incremental cost
BIC_0	Vector of set points of BIC
ΔBIC	Vector of variations of BIC
b_l	Vector of loads

b_l^M	Vector of upper bounds on loads
b_{l0}	Vector of set points for the loads, in a load perturbation function
Δb_l	Vector of load variation terms, in a load perturbation function
b_l^A	Subvector of b_l with active loads, in load shedding
C	Expansion point candidate vector
C(P)	Quadratic objective function in OPF
c_0	scalar term of the objective function C(P)
d_g	Vector of dependent variables (in p.u.)
d_{gi}^{lim}	Scalar, the original limit on variable d_{gi}
Δd	Scalar value of the largest violation, in the varying limits method
ΔD	Matrix of variations for b^l in the continuation process
d_g^I	Vector of inactive dependent variables
$d_g^{lim}(\theta)$	Vector of varying limits on active dependent variables, in the perturbation function
d_{g0}^{lim}	Vector of original limits on active dependent variables, in the perturbation function
d_g^m	Vector of lower bounds on dependent variables (in p.u.)
d_g^M	Vector of upper bounds on dependent variables (in p.u.)
d_{g0}	Vector of set points for dependent variables, in continuation method
Δd_g	Vector of variations of dependent variables, in continuation method
d_0	Vector of set points in the dependent variable functions
$D_\lambda(\theta)$	Scalar denominator of the solution trajectory for λ
$D_p(\theta)$	Scalar denominator of the solution trajectory for P_g
e	The unit vector
e_p	A unit vector for real power generations concatenated with a nul vector
e_q	A Unit vector for reactive power generations concatenated with a nul vector
E	Expansion point vector
f	An objective function (general notation used in Ch. 1)
F	The set of load flow equations
F_p	The set of real power equations
F_q	The set of reactive power equations
F_J	The set of line current magnitudes squared equations
g	Equality constraint functions (general notation used in Ch. 1)
g_0	Vector of coefficients of the generalized power balance equation

g_0^A	Subvector of g_0 corresponding to active independent variables
g_{0p}	Subvector of g_0 corresponding to the real power generations
g_{0t}	Subvector of g_0 corresponding to the transparent variables
G	Matrix of coefficients made by concatenating g_0 and H
G_p	Submatrix of G corresponding to real power generations
G_t	Submatrix of G corresponding to transparent variables
G_1	Matrix of coefficients of the functional dependent constraints
h	Inequality constraint functions (general notation used in Ch. 1
H	Matrix of coefficients of the active functional constraints multiplied by the appropriate r_i indices
H_p	Submatrix of H corresponding to real power generations
H_t	Submatrix of H corresponding to transparent variables
H^A	Submatrix of H corresponding to active independent variables
I	The identity matrix
J	The load flow Jacobian matrix
J_{bd}, J_{bb}, J_{md}, J_{mb}, J_{dd}, J_{db}	Partitions of the J matrix
J_t	Vector of transmission line current magnitudes squared (in p.u.)
J_t^M	Vector of upper limits on the line currents squared (in p.u.)
k	Right-hand-side vector of coefficients of active constraints
$\hat{k}$	Vector formed in the solution process of the subproblem
k_0	Vector of set points of k in the perturbation function
Δk	Vector of variations of k in the perturbation function
ΔK	Matrix of variations for b_1 in the perturbation function
k^{lim}	Right-hand-side vector of active functional constraints, made by concatenating k_0^{lim} and k_1^{lim}
k_0^{lim}	Right-hand-side coefficient of the generalized power balance equation
k_1^{lim}	Right-hand-side vector of active dependent constraints
k'	Right-hand-side vector made by concatenating k with other vectors
k_0'	Vector of set points of k' in the perturbation function
$\Delta k'$	Vector of variations of k' in the perturbation function
$\Delta K'$	Matrix of variations for b_1 in the perturbation function
K	Matrix formed in the solution process of the subproblem
L	Matrix formed in the solution process of the subproblem

M	Matrix formed in the solution process of the subproblem
n	Vector formed in the solution process of the subproblem
n_0	Vector of set points for n in the perturbation function
nd	Number of dependent variables
nf	Number of phase shifters
ng	Number of real power generations
nj	Number of transmission lines
nl	Number of loads
nt	number of transparent variables
NT	Number of time intervals
$N_\lambda(\theta)$	Vector numerator of the solution trajectory for λ
$N_p(\theta)$	Vector numerator of the solution trajectory of P_g
p	System parameters
p_i	Initial values of the parameters p
p_f	Final values of the parameters p
P	General notation for real power (generation or load)
P_D	Scalar system real power load (sum of real loads)
P_{D0}	Scalar set point of P_D
ΔP_D	Scalar variation of P_D
PF	Participation factor of the loads
P_g	Vector of real power generations (in p.u.)
P_g^A	Subvector of P_g with active real generations
P_g^{lim}	Vector of limits on active real power generations
P_{g0}^{lim}	Vector of limits on active real power generations, used in a perturbation function for contingency studies
P_g^m	Vector of lower bounds on real power generations (in p.u.)
P_g^M	Vector of upper bounds on real power generations (in p.u.)
P_{g0}	Vector of set points for P_g in the continuation process
ΔP_g	Vector of variations for P_g in the continuation process
P_l	Vector of real power loads (in p.u.)
Q_D	Scalar system reactive power load (sum of reactive loads)
Q_{D0}	Scalar set point of Q_D
ΔQ_D	Scalar variation of Q_D
Q_g	Vector of reactive power generations (in p.u.)
Q_g^m	Vector of lower bounds on reactive power generations (in p.u.)
Q_g^M	Vector of upper bounds on reactive power generations (in p.u.)

Q_l	Vector of reactive power loads (in p.u.)
r	Vector of limits on ramp constraints
R_b	Vector of active/inactive status indices for independent variables
R_d	Vector of active/inactive status indices for dependent variables
R_l	Vector of active/inactive status indices for loads
R_p	Vector of active/inactive status indices for real power generations
R_r	Vector of active/inactive status indices for ramp constraints
R_t	Vector of active/inactive status indices for transparent variables
S	Candidate expansion point vector (in Ch. 3 only)
S_i	Scalar, i^{th} apparent power (in Appendix 3.1 only)
SIC(P)	Scalar system incremental cost of real power
SIC(Q)	Scalar system incremental cost of reactive power
$S(b_l)$	The quadratic objective function of load shedding
s_0	Scalar term of the objective $S(b_l)$
S_0	Matrix used to express the objective function $S(b_l)$
s_1	Vector of linear terms in the objective $S(b_l)$
$s_1(\Theta)$	Perturbation function vector of s_1 for load shedding
s_{10}	Vector of set points of s_1 in the perturbation function
Δs_1	Vector of variations of demand, in the perturbation function
S_2	Matrix of quadratic terms in the objective $S(b_l)$
t	Vector of voltage ratios of variable tap transformers (in p.u.)
t_g	Vector of transparent variables
t_g^A	Subvector of t_g with active transparent variables
t_g^{lim}	Vector of limits on active transparent variables
t_{g0}	Vector of set points for t_g in the continuation process
Δt_g	Vector of variations for t_g in the continuation process
t^m	Vector of lower bounds on variable transformer tap ratios (in p.u.)
t^M	Vector of upper bounds on variable transformer tap ratios (in p.u.)
T	Matrix linking line flow to voltage phase angles in the DC load flow model
T_1	Matrix linking line flows to phase shifter angles in the DC load flow model
U_K	Upper triangular matrix computed in the continuation procedure
U_L	Upper triangular matrix computed in the continuation procedure
$v_i^{(n)}$	Notation for variable i at time interval n
V	Vector of bus voltage magnitudes (in p.u.)

V^m	Vector of lower bounds on bus voltage magnitudes (in p.u.)
V^M	Vector of upper bounds on bus voltage magnitudes (in p.u.)
x	Vector of load flow states
x_b	Vector of independent states
x_d	Vector of dependent states
x_r	Scalar reference state
x^m	Vector of lower bounds on certain states
x^M	Vector of upper bounds on certain states
y	Vector of load flow injections
y_b	Vector of independent injections
y_d	Vector of dependent injections
y_m	Scalar manifold variable
y^m	Vector of lower bounds on injections
y^M	Vector of upper bounds on injections
Y	Matrix linking real power generations to voltage phase angles in the DC load flow model
Y_1	Matrix linking the real power generations to phase shifter angles in the DC load flow model
α	Scalar step size
α'	Cumulative scalar step size
δ	Vector of bus voltage phase angles (in rad.)
λ	Vector of Lagrange multipliers corresponding to active functional constraints, made by concatenating λ_0 and λ_1
λ_0	Scalar Lagrange multiplier corresponding to the generalized power balance equation
λ_1	Vector of Lagrange multipliers corresponding to active dependent constraints
λ^0	Vector of set points for λ in the continuation process
$\Delta\lambda$	Vector of variatiions for λ in the continuation process
$\Delta\lambda_1$	Vector of Lagrange multipliers corresponding to active functional constraints in the load shedding problem
$\Delta\lambda_{1-g}$	Subvector of $\Delta\lambda_1$ corresponding to dispatchable variables, in load shedding
$\Delta\lambda_{1-\lambda}$	Subvector of $\Delta\lambda_1$ corresponding to Lagrange multipliers of the dispatching problem, in load shedding

Λ	Vector of all the Lagrange multipliers
Λ_0	Vector of set points for Lagrange multipliers, in continuation method
$\Delta\Lambda$	Vector of variations of Lagrange multipliers, in continuation method
ϕ	Vector of phase shifter angles (in rad.)
ϕ^m	Vector of lower bounds on phase shifter angle (in rad.)
ϕ^M	Vector of upper bounds on phase shifter angle (in rad.)
ρ	Vector of Lagrange multipliers corresponding to ramp constraints
θ	Scalar continuation parameter
μ	Vector of Lagrange multipliers corresponding to active independent variables
μ_l	Vector of Lagrange multipliers corresponding to active loads
μ_p	Vector of Lagrange multipliers corresponding to active real power generations
μ_{p0}	Vector of set points for μ_p in the continuation process
$\Delta\mu_p$	Vector of variations of μ_p in the continuation process
μ_t	Vector of Lagrange multipliers corresponding to active transparent variables
μ_{t0}	Vector of set points for μ_t in the continuation process
$\Delta\mu_t$	Vector of variations of μ_t in the continuation process
$\mathcal{L}$	Lagrangian function

Superscripts A and I pertain to active and inactive variables

Superscripts m and M pertain to lower and upper bounds

ABBREVIATIONS

BIC	Bus incremental cost
ED	Economic dispatch
GPBE	Generalized power balance equation
LHS	Left-hand-side
LP	Linear programming
LS	Load shedding
MAL	Minimum aggregate load
ML	Minimum loss
QP	Quadratic programming
RHS	Right-hand-side
SIC	System incremental cost
TL	Threshold load

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 General

Ever since their inception, the electric power utilities have strived to keep pace with the energy needs of the population they serve. The demand for increasing amounts of energy has spurred the construction of large, complex power systems, comprising of many generating plants and intricate, widespread networks for transmission and distribution.

Over the years, some notable changes have occurred in the general power system structure. Generation plants have been moved away from the load centers, for various reasons. In the province of Quebec, where hydroelectric power is abundant, the large energy sources being tapped are becoming more and more remote from the major load centers. Elsewhere in North America, conventional thermal and nuclear plants predominate. Their energy resources can often be transported more easily to the load centers, but in the last couple of decades plants have been built away from urban areas as a result of concerns for pollution or radiation hazards. Intricate transmission networks have been built to link these generation plants among themselves and to the load centers, as well as to neighboring utilities. The overall combination is advantageous, in that the construction and the operation of large generation plants provides an economy of scale, and their interconnection to the entire network of loads ensures a higher level of reliability of supply for each load. As a result though, the complexity of power systems and of the controls needed to operate them have increased.

Up until the oil crisis of the mid-1970's, power utilities were expanding mostly to meet the increases in their own internal demands, which in North America was doubling roughly every ten years. Since then and until the mid-80's, the dramatic rise in fuel costs forced consumers to make a more efficient use of energy and to adopt conservation measures. The effects of these measures on the power industry have been mixed. On one hand, importers of energy came to view it as a limited and expensive resource, and many have since implemented load management practices to reduce its use wherever possible, rather than increase their production capacity [Seelke 1982, Chan 1986]. On the other hand, the completion of projects planned before the oil crisis left many utilities with excess generating capacity. That was the case for Hydro-Quebec, even though 99% of its production is hydro. Efficient energy practices considered essential elsewhere also proved attractive in Quebec, so that the demand for electricity lagged behind expectations. This general situation in the power system industry prevailed until recently. Since the mid-80's, fuel costs have plummeted, and in the United States the electric power industry has become deregulated [Fischetti 1986]. Already there have emerged some large-scale importers of electric energy (for example, the Northeastern U.S and southern California) and some large-scale exporters (for example, the central Canadian provinces, British Columbia and the American Northwest). To enhance the trade of electricity, much of the Hydro-Quebec's (and other utilities') recent planning efforts have gone towards strengthening the interties with their neighbors. Already in some american power utilities, free market practices are taking over in the every day operation of the system, with numerous energy transactions being proposed and accepted as opportunities occur. These practices are no doubt making a more efficient use of overall energy resources, but they increase once again the complexity of the controls needed to operate the system.

Since the early 1970's, many utilities have built computerized control centres to aid in the operation of their systems [Dy Liacco 1974, Dy Liacco 1977, Scheidt 1979]. Supervisory control and data acquisition functions (SCADA) were the first to be implemented in these centers. Measurements from the power system are continuously channeled to a central location in real-time and compared to estimated values from a state estimator program. The verified quantities are then checked for system security, for the reliability of the network configuration to supply the load, and in some cases for stability margins. Other on-line functions presently available on most systems are economic dispatch and load-frequency control. The former computes the most economical distribution of generations, given the list of available generators. The latter supervises energy interchanges and controls the system frequency in response to imbalances between the system's generation and its load. Due to their complexity, some other useful functions for power systems operation have yet to be implemented on-line, but are used off-line as tools for analysis. Three such functions are security evaluation and security control [Debs & Benson 1975], and the subject of this thesis, optimal power flow.

The general problem of energy management which faces the power utilities - to satisfy customer demand in a safe, reliable and cost effective manner- is a very complex one. It requires much insight into the workings of the power system, for sure, but also a good working knowledge of mathematical optimization theory. Many problems of power system management and control have been formulated, covering the whole spectrum of mathematical programming disciplines, ranging from very long term (ten to fifteen years for generation and transmission planning) to very short term (a few minutes for dispatching). Due to their complexities, each problem is usually treated separately. They are usually performed in a hierarchy, from long term to short term, with the output of the long term tasks serving as targets for the short term tasks.

The short term functions are grouped under the category of power system operation. Some of the more important tasks in this group are economic dispatch, minimum loss dispatching, minimum load shedding and minimum deviation from an operating point. These problems and others, which are subject to the load flow equations as constraints, share a common nonlinear programming formulation called optimal load flow or optimal power flow, denoted OPF.

The full OPF serves two purposes. In operations, it periodically sets optimal target values for the electrical variables of the power network, in following the system's varying load. Based on the OPF's optimal values, the variables can then be dispatched every few minutes to follow small variations in loads, using simpler algorithms. The OPF would be an ideal dispatching tool if it could be made to compute much faster. A second application for OPF is in system planning, where it is used to study the effects of parameter variations (changes in equipment) on the system's optimal operation.

The OPF is a complex tool, but its subsets are often simple enough to be used as dispatching tools. These subsets use approximations of the basic problem, with a linearization replacing the nonlinear load flow equations, or else they neglect some variables in the formulation, or neglect limits on some variables. This thesis will be concerned for the most part with the study of the OPF and its subsets.

1.2 The Spectrum of Power System Control Functions

At this point it is worthwhile to look at the various control functions required in power system control and the place occupied by the optimal power flow. Table 1.1 displays some of the major functions, starting with the long term functions at the top and moving downward towards the shorter term functions.

The long term functions deal with planning. Their main purposes are (1) to predict future electrical energy needs, (2) to assure adequate supplies of energy in bulk over a fairly long time period, and (3) to provide an adequate infrastructure to deliver that energy reliably and economically to the load centers. They are generally formulated as optimization problems, to ensure the most efficient use of the new resources.

Planning functions dealing with the power system [Fischl 1975, Sullivan 1977] are generally split into two groups, generation planning and transmission planning . Generation planning [Rutz et.al. 1983, Caramanis et.al. 1984, Desrochers et.al. 1986] studies the various alternatives for the addition of new generating capacity: the timing of the addition, the type, size, location and cost of the new plants, their integration into the existing network, and in recent years, their environmental impact. Transmission planning [Kaltenbach et.al. 1970, Lee et.al. 1974] studies the alternatives for the addition of equipment to the transmission network, using the same criteria, to meet the requirements of added generation, or of a changing load profile, or to improve the system's reliability in supplying the load. The time horizon for these functions is typically from 10 to 15 years.

TABLE 1.1 A LIST OF POWER SYSTEM OPERATIONS AND PLANNING FUNCTIONS

Functions	Time frame
MARKET AND INVESTMENT ANALYSIS - Long term demand forecasting - Load management policies - Financing POWER SYSTEM PLANNING - Generation planning - Transmission planning MANAGEMENT OF RESOURCES - Reservoir management (hydro systems) - Fuel purchasing and allocation policies	LONG TERM
MEDIUM TERM ENERGY MANAGEMENT - Maintenance scheduling - Fuel scheduling - Hydro & hydro-thermal scheduling	MEDIUM TERM
SHORT TERM ENERGY MANAGEMENT - Load forecasting - Unit Commitment - Hydro & hydro-thermal coordination OPERATIONS, OFF-LINE STUDIES - Static coordination problems - Contingency analysis - OPF	SHORT TERM
SHORT TERM OPERATIONS - Dispatching (optimal) - Emergency redispatching (non-optimal) AUTOMATIC GENERATION CONTROL - Automatic load-frequency control - Automatic generation control	VERY SHORT TERM
MONITORING - SCADA - Security monitoring - State estimation PROTECTION - Various protection schemes - Coordination between protection devices	INSTANT
RESTORATION	

Planning functions dealing with the energy supply study, the efficient procurement and management of the limited energy resources. In hydro systems, reservoir management is concerned with the storage of water energy for use at the most opportune times [El-Hawary & Christensen 1979, Ihura & Gross 1984]. For example, in Quebec most reservoirs are filled during the spring and the summer to provide for the peak demand in the winter. The water management is complicated by the presence of various reservoir types, with storage cycles from zero (run of river plants) to one year (large storage plants). Typical planning horizons for reservoir management can be of the order of five years. In thermal systems, the scheduling of thermal generating plants is influenced by the availability of certain fuel types, and by the commitments of long term fuel-purchasing contracts [Kondragunta & Walker 1984, Levin & Zahavi 1984]. In recent years, planning strategies for resource management have incorporated stochastic models, to take into account the random nature of such things as yearly precipitation levels and fluctuating fuel prices [Dodu & Merlin 1979].

The functions classified here as medium term form a transition between planning and operations functions. Given the typical load distribution and energy production targets over a period of one year, these functions schedule the prolonged use/non-use of the various components of the power system [Turgeon 1981, Vemuri 1984]. For example, in prolonged periods of weak demand, the more expensive generation plants can be turned off. Also in these periods, system components are scheduled to be shut off for maintenance [Yamayee 1982]. Then in periods of strong demand, most of the system components would be made available.

The short term and very short term functions are operations functions. They can be characterized by the presence of power requirements, as opposed to energy requirements in the previous functions. The short term functions provide the decisions typically needed to meet daily power requirements. Unit commitment [Gruhl et.al. 1975, Pang et.al. 1981, Lauer & Bertsekas 1982], hydro and hydro-thermal coordination [Calderon & Galiana 1987] determine the on/off timing sequence and the general production levels of generating units to satisfy most economically the varying power load, plus reserve and ramping constraints. Present algorithms used for these functions only incorporate forecasted daily loads at intervals of one hour or so, and usually little or no power network information is used. Hence as output from these functions the commitment schedules are firm, but the production levels only serve as guidelines.

Using these commitment schedules, optimal values of real power generations and other electrical variables can be computed, incorporating the difficult network constraints and equipment limitations, using the optimal power flow. Besides dispatching real power, this program supplies targets for reactive powers, voltages and passive controls on the system, which can be updated at various times in the day. In some OPF implementations, especially the more recent ones, the so-called security constraints have also been included to the formulation [Carpentier 1975, Stott et.al. 1987]. These further restrict the operation of the system, such that following the removal of any one of the system's components, all the system variables remain feasible. The security constraints are fed to the OPF by the contingency analysis function. This studies plausible contingencies in the present operating conditions, and from them formulates a set of constraints in the likely post-contingency states. Presently, because of their long computation times, implementations of the OPF and of contingency analysis are computed off-line. They would be of great use in dispatching if they could be updated much faster.

The very short term operations functions are split into optimal dispatching and non-optimal redispatching. The dispatching functions are meant to quickly satisfy the power demand as it varies. As pointed out earlier, dispatching algorithms optimize the values of the electrical variables using a simplified network model. Such algorithms are presently available for real-time control, with solutions being updated in the order of minutes. The other operations function, also available in real-time, is emergency redispatching [Krogh et.al 1983]. It is used when the system finds itself operating with some quantities outside their limits, usually following a contingency. The redispatching forsakes optimality to quickly find a feasible operating point towards which the system can easily be moved. Ideally, implementations of redispatching should be faster than those for dispatching.

The next group of functions, designated as instantaneous in Table 1.1, are on a much smaller time scale than previous functions, and occupy a large field on their own in power system studies. Hence they are not classified as operations functions.

Automatic generation control [Wood & Wollenberg 1984] operates in the order of seconds. It is a closed-loop control which monitors and maintains (1) the prescribed tie-line power flows, and (2) the system's nominal frequency, to satisfy the internal system load. The control action is performed by constantly modifying the real power generations to minimize a norm of the discrepancies between the scheduled frequency and tie-line flows and their measured values. The changes in the individual generations are given by their participation factors, which are computed from a perturbation analysis of the last dispatching solution [Wood & Wollenberg 1984].

Two more very fast functions are presently implemented in real-time. They are (1) the SCADA monitoring functions, which measure and verify all the system variables every few seconds [Miller 1983], and (2) protection, which detects and isolates faulted network elements within hundredths of a second after the occurrence of the fault [Warrington 1968 & 1974].

The remaining function, restoration, has not yet found its way into its ideal time slot. This function deals with restoring full use of the power system following a partial or a total shutdown [Peach 1984]. Ideally this would be an operations function, with appropriate strategies being computed for various partial system shutdown situations. Unfortunately, most restoration plans presently available stem from simulation studies, which require complex analysis and numerous runs of programs with very long computation times. As a result, only a few pre-computed restoration strategies are ever available to a utility.

1.3 A First Look at the Optimal Power Flow Problem

A comprehensive description of the optimal power flow problem - its formulation, its history and its solution methodologies - will be provided in the following chapters. In this section, we take a quick first look at the OPF.

The OPF is an operations function whose role is to find the optimal settings of all the electrical variables in the network, for a given load. The optimization involves all the electrical variables which are available, as decided by the scheduling and commitment functions placed higher up in the control hierarchy. Optimality is established according to some particular criterion, expressed as an algebraic objective function. In most OPF studies the objective is to minimize fuel costs, although other objectives are available for various tasks. Engineering and system limitations are expressed as algebraic equality and inequality constraints. These constraints include the nonlinear equations which model the network, called the load flow equations, and upper and lower bounds on most of the variables. This is the basic description of the OPF problem.

Additional constraints have been considered for the OPF problem. The security constraints described earlier have been incorporated in simplified form into some OPF packages, but according to Carpentier [Carpentier 1987], few algorithms are presently efficient for security-constrained problems. Spinning reserve and ramp constraints on real power generation have also been mentioned as potential constraints for the OPF, but these are probably best handled in the scheduling and unit commitment functions. Present implementations which incorporate these constraints are dispatching algorithms, with simplified network models in place of the load flow equations. Tie-line power flows to the neighboring utilities have often been considered as separate constraints, but their modelling can be incorporated into the load flow equations. Hence, the constraints described in this paragraph will not be considered in the OPF formulation and in the subsequent solution methodology presented in this thesis.

Using the nomenclature developed for the variables in Chapter 3 and its appendices, the optimal power flow problem is expressed mathematically as the following nonlinear program (see next page):

$$\min_{b_g} \quad f(b_g, d_g, b_D, p)$$

$$\text{s.t.} \quad g(b_g, d_g, b_D, p) = 0$$

$$h(b_g, d_g, b_D, p) \le 0$$

where

f, g and h	are the objective function, the equality and the inequality constraints, respectively.
b_g	is the vector of independent variables.
d_g	is the vector of dependent variables.
b_D	is the vector of loads. In most tasks, it is a fixed parameter for which a new solution is required.
p	is the set of fixed system parameters.

This formulation of the OPF problem dates back to the late 1950's- early 1960's. At that time, mathematical optimization theory had just formulated the tools needed to solve the problem. Since then OPF research has looked for better ways to solve this difficult problem. Power systems researchers have been quick to apply the latest emerging numerical optimization techniques, and in some cases have instigated the development of successful techniques [Abadie & Carpentier 1969].

In this thesis, dispatching is referred to as a subset of the optimal power flow. Historically, dispatching algorithms preceded the OPF; the arrival of the OPF marked the end of the "classical" period of economic dispatch, which had developed over almost 30 years [Kirchmayer 1958]. The OPF was a radical departure from the earlier dispatching, although now those dispatch algorithms can be seen as crude simplifications of the OPF. Basically the two solve for the same minimum cost objective, but in the classical dispatching algorithms, only real power generations were considered, and the load flow equations were represented by a single equality constraint, called the power balance equation [Wood and Wollenberg 1984]. By the late 1960's, more sophisticated dispatching algorithms were developed as an outgrowth of the OPF. Here the nonlinear load flow equations are replaced by a linearized model, but all the variables and their bounds were kept, as in the OPF. Since the early 70's, some implementations have gone one step further, in separating the components of the weakly coupled real and reactive power problems in the linearized model. The use of the real and the reactive dispatch in tandem provides fairly good results, especially for real power dispatch, and is much faster than the full OPF. Also since the early 70's, these recent dispatching algorithms have served as subproblems in the OPF solution methodology. The subproblem is used as a block within an iterative scheme in which the nonlinear information is updated. The nonlinear optimal solution is reached when the subproblem solution coincides with the load flow feasible expansion point from which it was generated. That is the approach used in the popular sequential quadratic programming strategy, and it will be used in the work described in this thesis.

1.4 The Use of the OPF and its Subsets in Higher Order Functions

Pieces of the optimal power flow and of its dispatching subsets have been used as tools in system expansion planning and in the time-related operations functions. Transmission planning has become more complex in recent years, with the load flow constraints appearing in the formulation, along with the more usual power capacity constraints. The usual objectives of planning functions are also related to the objectives in OPF. Many of these are formed at least in part by the integral over time of the typical OPF objectives. The most common of these is the minimum costs objective. Particularly in reactive power planning, the standard formulations have integrated the load flow constraints, in implementations ranging in complexity from linear programming [Kishore & Hill 1971, El Shibini & Dayeh 1975] to nonlinear programming [Hughes et.al. 1981, Lee et.al. 1986] and integer programming [Kohli & Kohli 1975]. Robust commercial OPF programs have been made available over the last decade, and they are now being used as building blocks for the larger planning problems. For example, researchers at General Electric have used their OPF package to solve reactive power planning problems [Fernandes et.al. 1983]. Typically, the OPF can be used for long term planning functions, where computation times are not a limiting factor.

Few efforts have been made to incorporate any model of the load flow constraints into the time-related operations functions. Recently, El-Hawary and Tsang [1986] formulated and solved a hydro-thermal coordination problem with the nonlinear load flow constraints. Predictably, they reported very long computation times for realistic sized problems. That being the general case, usually much simpler and faster dispatching algorithms are used in scheduling and unit commitment. An example in generation scheduling is the work by Waight and colleagues [Waight et.al. 1981a]. There the simplest of economic dispatch algorithms, using the power balance equation to represent the network, is integrated to the larger scheduling algorithm, along with ramp and reserve constraints on the generations. In unit commitment also, the simple dispatching model is integrated into the larger algorithm. Examples for three solution techniques of unit commitment are mentioned: in branch and bound [Ohuchi & Kaji 1975] and in dynamic programming [Snyder et.al. 1987] techniques, the values of the nodes being compared are the solutions of an economic dispatch algorithm for the various combinations of generators; in the recent Lagrangian relaxation methods [Zhuang & Galiana 1987], the economic dispatch constitutes the primal subproblem.

1.5 The Continuation Method

This section describes a little-known mathematical technique which serves as the basis for the work presented in this thesis. The continuation method serves in solving (nonlinear) sets of equations, but is used in conjunction with standard numerical techniques. It is also well-suited for optimization. The literature in numerical mathematics actually presents the continuation method from two viewpoints. In earlier publications, it was seen as a method for improving the convergence of the standard methods, by generating sequences of more easily solved intermediate problems, leading to the desired problem. In more recent implementations, the intermediate problems have taken on some physical significance, and as a result the solutions form some useful trajectories.

The basic idea behind the continuation method is quite simple. Figure 1.1 illustrates this idea for the solution of nonlinear equations. A problem

$F_1(x)=0$ has a solution x_1, which can be computed using a standard iterative technique as long as the initial guess is in the region R_1. Unfortunately this region is usually unknown a priori, and in some cases can be very small. In this example, with an initial guess of x_0 the standard technique would fail to converge to the desired solution. In a first step of the continuation approach, some simply resolved problem $F_0(x)=0$ is built, for which the solution is x_0. F_0 and F_1 are linked through a parametric relationship $F_\theta(x) = F(x,\theta) = 0$, where θ is a scalar contained in the unit interval, called the continuation parameter. Given F_1, the most general conditions for the choice of the relationship $F(x,\theta)$ and for a suitable initial problem F_0 are covered in the difficult homotopy theory [Hu 1959]. However, for most applications so far and particularly for polynomial functions, simple methods exist to validate these choices [Garcia & Zangwill 1981, Morgan 1987].

The second step of the continuation method consists in increasing the value of θ from zero to one. This can be done incrementally for systems which allow analytical solutions, or discretely for systems which only allow numerical solutions. The solution of the desired problem F_1 in the former case would be obtained by integrating $F(x,d\theta)$ over the interval $\theta \in [0,1]$. This has been the basis for the theoretical explanations of this method [Davidenko 1953], and has been used in some applications. Most applications however apply numerical solution techniques, either in conjunction with standard numerical nonlinear solution techniques, or in numerical integration schemes. In these cases, the problem F is perturbed by small amounts starting from $F_0(x_0)=0$. If the perturbation is small enough, the solution to the new problem $F(x,\theta_i)=0$ is easily found using the previous solution x_{i-1} as an initial guess. This is illustrated in figure 1.1, with the x_{i-1} situated in the regions of convergence R_i, and usually x_{i-1} is close to the x_i. The solution to the desired problem is achieved when the continuation parameter reaches one.

Some of the prominent references for the numerical solution of nonlinear equations by the continuation method are now mentioned. Historically, the first papers on continuation methods are attributed to Schauder [Schauder 1934] and to Lahaye [Lahaye 1934, Lahaye 1948], although their work was very limited in scope. The theoretical basis of the method was established in more general terms in the early 1950's, particularly by Friedrichs [Friedrichs 1950] and by Ficken [Ficken 1951]. In 1953, Davidenko [Davidenko 1953] developed the first systematic numerical continuation algorithm for solving nonlinear equations, based on the integration of the differential equation, as mentioned above. Over the next fifteen years, advances in this approach were reported most notably by [Freudenstein & Roth 1963, Deist & Sefor 1967, and Meyer 1968]. More recently, computer implementations based on Davidenko's method were written by Kubicek [Kubicek 1976] and by Rheinboldt and Burkardt [Rheinboldt & Burkardt 1983b], and made publicly available in the ACM software library. In all of these applications of the Davidenko approach, there remained a major unresolved problem: in some instances, the Jacobian matrix built at one stage of the process can become singular for some value of the parameter, and the process bogs down. Scarf [Scarf 1967] avoided the Davidenko approach altogether in his solution technique, which is based on the more difficult simplicial techniques (see [Garcia & Zangwill 1981]). His work was continued, amongst others in [Eaves 1972, Eaves 1976, Saigal 1977, Saigal & Todd 1978, Saigal 1983]. This line was summarized in [Allgower & Georg 1980]. The singularity problem in Davidenko's approach was resolved by Chow and colleagues [Chow et.al. 1978] and by Keller [Keller 1978], in algorithms which were specifically designed to eliminate the possibility of singular points. Their improvements have made the continuation method a robust numerical tool for general use. Some recent computer implementations based on these improvements are reported by [Garcia & Zangwill 1979, Garcia & Zangwill 1981, Watson & Fenner 1980, and Morgan 1987]. The last three references are of particular interest: two are excellent textbooks [Garcia & Zangwill 1981 and Morgan 1987], and Watson's program is available in the ACM software library.

The continuation method has also been used successfully for the solution of optimization problems. Parametric linear and quadratic programming using single parameter variations are in fact examples of the continuation method in optimization [Boot 1964, Van de Panne 1975]. One application of parametric quadratic programming to be given a name of its own is Houthakker's Capacity Method [Houthakker 1960]. Its separate treatment is justified in that this method's solution techniques are different from the simplex-type methods used at the time. Implementations of parametric linear and quadratic programming have been relatively fast and reliable, because their solutions procedures avoid nonlinear equations and they easily identify changes in the active set as the continuation parameter increases. In this thesis, the application of the continuation method in optimization is a form of parametric quadratic programming. It is used as a subproblem for the more general nonlinear program.

Figure 1.2 illustrates the use of the continuation method for a small quadratic program. The axes of the graph represent the two variables x_1 and x_2 and the continuation parameter θ. The feasible region of the problem we wish to solve is the polygon on the front face $\theta=1$ of the polytope in (x,θ) space. Because of its complexity, the resulting solution process might be lengthy. Hence the continuation method is tried in the hope of simplifying the process. The simple problem, at $\theta=0$, is chosen so that its feasible region is the rectangular box at the back surface of the polytope in (x,θ) space. The solution procedure easily finds the optimal solution x_0^* to the simple problem. Then as θ is increased, the feasible region is deformed back towards its original shape. In the process, the optimal solution trajectory leads from x_0^* to the desired solution x_1^*, when $\theta=1$. Little computational effort was required, because the optimal solution trajectory changed direction only once. This is an example of the "varying limits strategy", used in power systems dispatching [Galiana et.al. 1983].

The solution to the more general and more difficult parametric nonlinear programming problem has been tackled only in the last few years. In addition to the usual problems in dealing with the convergence of nonlinear programs, the major difficulties in these programs arise in trying to accurately track the nonlinear solution trajectories, and in locating the "breakpoints", or values of θ for which the active set changes. The main contributions so far in this fledgling field are possibly those of Guddat, Bank and colleagues at Humboldt University in East Germany, and Gfrerer and Wacker at Johannes Kepler University in Austria [Bank et.al. 1982, Gfrerer et.al. 1983, Guddat et.al. 1984]. Other important contributions have been presented in compilations and proceedings of specialized conferences, for example [Fiacco 1982 and Fiacco 1984].

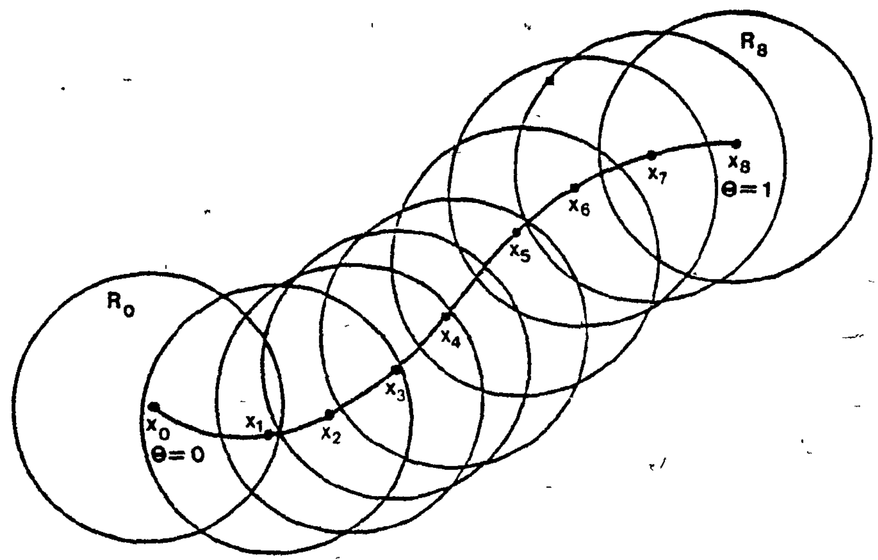


Figure 1.1. An illustration for the solution of nonlinear equations using the continuation method.

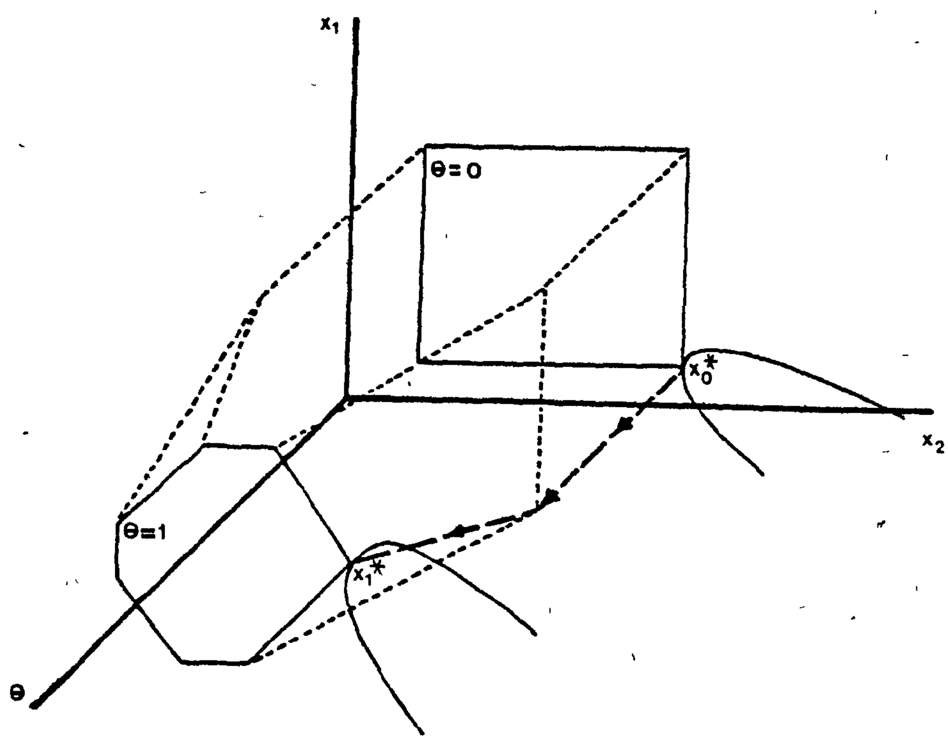


Figure 1.2. An illustration for the solution of a quadratic program using the continuation method.

In power systems, Ponrajah recently tackled the nonlinear OPF problem for a single input load using continuation methods in a manner similar in scope to Guddat's [Ponrajah 1987]; this application will be discussed a little further.

From all these applications, there emerge four major advantages in using the continuation method:

- For some problems, the construction of solution trajectories leads to the solution of the desired solution faster or more reliably (or both) than the standard techniques.

- The solution trajectories are useful in their own right, in cases where the continuation parameter is actually some physical parameter which varies in the system. One example in various engineering fields is the so-called incremental loading technique.

- This method is not restricted in its choice of an initial guess.

- This method can be made very robust. That allows to reduce the occurrence of numerical instabilities. Presently with many numerical techniques, when a computer program ends abnormally it is difficult to establish whether the cause is numerical instability or the infeasibility of the problem to be solved. In a recent power system application, the continuation method has shown the ability to detect feasibility limits [Famideh-Vojdani & Galiana 1983].

1.6 The Continuation Method in Power Systems

Parametric programming and continuation methods have been suggested for power system dispatching by a few research groups over the last decade. The most common application so far has been the tracking the optimal solution in dispatching problems, as a function of the varying system load. This is called the "load tracking strategy" in this thesis. Other applications have also been suggested for dispatching, and recently some researchers have turned their attention to the optimal power flow. There are still many untried areas in operations and in operations planning for using continuation techniques.

To illustrate the general idea of the continuation approach, we present Figure 1.3, which portrays in its simplest form the input-output structure of a power system operation problem. In conventional optimization techniques, the input is a single load; in the continuation approach using the varying load strategy, the input is a load trajectory. The resulting output is an optimal generation trajectory. Implementations of an optimization box for real power dispatch have proven very attractive - the computations of solution trajectories by the continuation method are as fast as the solution for one load by most conventional methods. Note that the varying parameter need not be limited to the load. One example is the varying limits strategy described earlier. That strategy solves for a given load, by varying the electrical parameters "inside the big box" from some relaxed positions to their intended positions. Other (as yet untried) parameter variations affecting the system performance could be useful for planners. Two of these are suggested inside the big box in fig. 1.3: (1) for expansion planning, the electrical parameters (device capacities, admittances, etc.) are varied and their effects on the optimal operating costs are readily obtained; and (2) for economic planning, the effect of an external parameter (such as varying fuel costs) on the optimal operating costs can be studied.

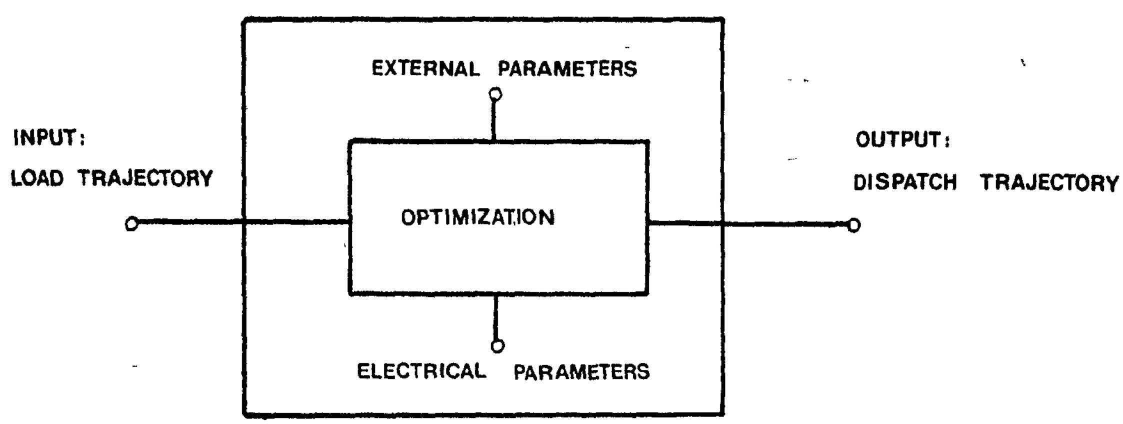


Figure 1.3. A schematic diagram of the different positions of parameter variations for power system optimization.

The first mention of parametric programming for power system dispatching was made by Dillon [1981]. His paper suggests with some detail the general idea of parameter variations in the right-hand-side of the optimality equations. (These are the only applications to have been tried so far.) However, he did not pursue the issue any further.

Prof. Galiana and graduate students in the Power Systems Group at McGill University have been studying continuation methods for the solution of power systems operation problems since the beginning of the 1980's. For his Ph.D. thesis, Vojdani produced a computer program implementing economic dispatch with the varying load strategy, using a DC load flow network model [Fahmideh-Vojdani & Galiana 1983]. He shrewdly noted that his work is an application of the incremental loading concept, developed using a better network model than in the standard formulation. About a year later, Juman's Master's thesis reported on a potentially rapid solution technique for quadratic programming, based on the varying limits strategy illustrated earlier [Galiana et.al. 1983]. A summary of results from those two projects and of some new ideas for parameter variations in economic dispatch were published by this author and his colleagues in 1985 [Huneault et.al. 1985]. By then, a second wave of applications suggested the use of the continuation method for other power system problems. In his Master's thesis, Ponrajah [Ponrajah & Galiana 1985] used Vojdani's program to compute optimal incremental bus costs. Calderon, in his Ph.D. thesis, produced a program for the time-dependent hydro-thermal coordination problem [Calderon & Galiana 1987]. In that work, he suggested time-dependent parameter variations which succeeded in identifying and handling the difficult active dynamic constraints. More recently, Ponrajah's Ph.D. thesis applied continuation methods to solve the OPF problem for a single load [Ponrajah & Galiana 1988]. In a first step, he solves a simplified nonlinear OPF problem, with dependent variables neglected, using parametric techniques. In a second step, he reintegrates the neglected nonlinear constraints into the problem. Violated constraints are handled by the varying limits strategy, but here, as in Guddat's work mentioned earlier, the variables are tracked along nonlinear trajectories. This author is also proposing an OPF solution methodology in this thesis, but with a much different solution strategy.

Starting also in the early 1980's, The French national power utility Electricité de France (EDF) has been developing its own real and reactive power dispatch algorithms, based on parametric quadratic programming and the varying load strategy. Carpentier first reported on their real power dispatch in 1983 [Carpentier et.al. 1983] , and Blanchon followed shortly after for the reactive [Blanchon et.al. 1983]. More recent papers report that the utility's results with these programs are very promising for use in real-time dispatching [Carpentier 1985, Carpentier 1986, Carpentier 1987].

A third group, from the Italian power utilities research institute ENEL, has been exploring the use of parametric methods for real power dispatching with a look-ahead capability. Franchi and colleagues [Franchi et.al 1980] proposed an automatic generation control using parametric linear programming, and then updated it to handle network constraints and ramp constraints [Innorta & Marannino 1985]. They have recently reported on work in this direction with parametric quadratic programming [Innorta et.al. 1987].

1.7 The Present Thesis

1.7.1 General Comments

The work in this thesis is in part an extension of the McGill group's previous dispatching studies using continuation methods. It proposes a solution methodology for the optimal power flow problem, incorporating continuation methods at different levels.

A first application of the continuation method is built into the optimization solver. We propose to solve the OPF using the well-known sequential quadratic programming strategy. Here however, the quadratic subproblem at the heart of the process exploits the varying limits strategy. The solution process for the OPF subproblem, containing all the electrical variables, is more complex than that for real generation dispatching. New concepts have been introduced, such as transparent variables and fold lines in the load flow manifold, in order to adapt the methodology to this problem.

In a second application of the continuation principle, a sequence of closely spaced loads is fed to the nonlinear OPF optimization, as portrayed previously in fig. 1.3. This produces as output a discrete OPF solution trajectory. Once this load-tracking procedure is initiated, the solution times for the individual OPF solutions are greatly reduced. This is because the proposed algorithm is designed to execute its tasks quickly at every level when presented with a good initial guess from the previous optimal solution.

The merits of the OPF algorithm proposed above have been investigated in a computer implementation for the economic dispatch task. The OPF algorithm is modular, with the simpler components feeding the more complex ones. Some of these modules can be used on their own for the simpler dispatching functions. Two of these modules are the real power dispatch module and the continuation method - quadratic subproblem itself, which is basically a voltage - reactive power dispatch. Results from the program were closely monitored throughout the computation, and provide much insight into the internal workings of the algorithm.

1.7.2 Outline of the Thesis

The chapters of this thesis are organized as follows:

CHAPTER II - SURVEY AND ANALYSIS OF THE OPF LITERATURE

This chapter presents a comprehensive survey and analysis of the optimal power flow literature. In a first part, a compilation of some three hundred publications on OPF and dispatching is organized chronologically and according to the optimization techniques used to solve the problem. The more important contributions are signaled out in the discussion, but all the publications are listed in an appendix. The general trends in solution techniques for the OPF over the last 15 years or so are also examined. In a second part, the basic OPF solution methodology is broken down into its basic components, and options for each component are enumerated. Example from the literature of the uses of the various components and solution structures are enumerated in what could be used as a classification scheme for OPF.

CHAPTER III - DESCRIPTION OF A NEW OPTIMAL POWER FLOW ALGORITHM

This chapter presents our OPF algorithm, featuring applications of the continuation method. It also presents many numerical techniques used to aid in the solution. The algorithm is first presented in general, in a descriptive manner, to give the reader a better feel for what is to follow. The OPF and its quadratic subproblem are then formulated mathematically, and all the system variables and parameters are introduced. Then the OPF algorithm is presented in detail, covering each component of the program, and in some cases, the alternatives which were discarded. Among the important details are the subproblem solution procedure, the various homotopy strategies for solving the subproblem, the Newton-Raphson solution procedure incorporating a step size control, the explanation of the inherent numerical difficulties which require an "anti-zigzagging" device, and the load-tracking step.

CHAPTER IV - SOLUTION OF THE OPF SUBPROBLEM USING THE CONTINUATION METHOD

In this chapter, the mathematical details of the subproblem solutions are presented. The quadratic subproblems for three tasks are analyzed: economic dispatch, minimum loss and minimum load shedding. The first two tasks are solved in two ways, using either the varying limits strategy or the varying load strategy. Details are available for the solution of economic dispatch by the varying limits strategy, so they are presented here. The minimum load shedding subproblem is then formulated for the first time in this thesis, and a solution procedure is suggested using the varying demand strategy. For each task, we provide the subproblem formulation, the resulting optimality conditions and the solution trajectories, and a suggested initial, simple problem to start the continuation process.

CHAPTER V - OTHER APPLICATIONS OF THE CONTINUATION METHOD IN OPF

A few more applications of continuation methods in power system operations are proposed in this chapter. They are: (1) a formulation and a solution technique based on the continuation method, for the incorporation of the time-dependent ramp constraints into the quadratic subproblem of economic dispatch; (2) a strategy to vary transmission and generation parameters by the continuation approach, in solving for post-contingency redispatch; (3) a look at possible additions to the real power dispatch using the DC load flow model; and (4) the computation of optimal bus incremental costs, based on the solution of economic dispatch by the varying load strategy.

CHAPTER VI - DETAILS OF THE NUMERICAL IMPLEMENTATION OF AN ECONOMIC DISPATCH - OPF ALGORITHM

This chapter documents the main procedures of a computer program which implements our OPF algorithm for the economic dispatch task using the varying limits strategy. First some general comments are made concerning the basic building blocks of the program: data structures, linear equation solvers, and matrix-vector products. Then the details of the subproblem solution are presented. Special attention is given to the real power dispatch solver, which could be used on its own. Other important sections of the subproblem computation are also described: the quick updating schemes for the optimality conditions following changes in the active set, the computation of solution trajectories, and determining the next breakpoint. Tests for resolving certain cases of degeneracy were also implemented in the program.

CHAPTER VII - DESCRIPTION AND ANALYSIS OF THE NUMERICAL SIMULATIONS

This chapter documents and analyzes the numerical results obtained from our OPF program. Tests were carried out on four test systems, ranging in size from 6 to 118 buses. In a first section, the results for the 6 bus system are presented. Here the series of graphs and tables are commented in detail. Then for the remaining tests, the same set of results are presented in the graphs and tables, but only the highlights are displayed in the discussion. The results, which cover the whole range of computations, are presented from the most general to the most detailed. They include an analysis of the global performance of the solution procedure, and the values of the variables at various stages of the computation. Following this presentation, a discussion reviews and analyzes the main results. Our results for the 30-bus test are then compared to those of other programs. The chapter closes with a discussion on numerical difficulties encountered in the program and possible remedies.

1.7.3 Important Sections for a First Reading

This being a fairly large thesis, the author suggests a limited list of important sections for a quick first reading. They should give the reader a good overview of the work before delving into the details.

Chapter II, sections 2.1, 2.2, 2.3 and 2.5.1
Chapter III, sections 3.1, 3.2, and 3.4.4.5
Chapter IV, section 4.1 and the opening descriptive paragraphs of the formulation sections 4.2.1, 4.3.1 and 4.4.1
Chapter VII, sections 7.1 and 7.6 to 7.9

1.8 Claim of Originality

To the best of the author's judgement, the following are original contributions to the study of the optimal power flow problem:

(1) A comprehensive classification scheme for the description of optimization algorithms used in OPF and dispatching. [Huneault & Galiana 1988]

(2) An algorithm for solving the OPF problem based on the sequential quadratic programming strategy and involving continuation methods at different stages. This includes:

(a) A new, more general formulation of the OPF quadratic subproblem:

i. It is formulated in a space of real power injections and other variables, so that it is useful on its own as a dispatching tool.

ii. The set of independent variables can be varied dynamically, using simple rules, to simplify the solution process.

iii. The notion of transparent variables and their efficient exploitation are introduced for the first time in an OPF algorithm.

iv. Restrictions usually placed on the slack injection in the formulation are eliminated.

v. The resulting optimality equations form a very simple and attractive structure from a numerical point of view: a bordered block with a diagonal main submatrix and a small border.

(b) A solution methodology for the quadratic subproblem based on the continuation method.

i. The varying limits strategy, used successfully in real power dispatching, has been adapted for use in the OPF subproblem.

ii. A more difficult "initial simple problem", based on that in real power dispatching, is completely analyzed and implemented. Certain theoretical points concerning the dispatching of compensation devices have emerged from the analysis.

iii. An efficient algorithm has been implemented to track the optimal solution of the subproblem using the varying limits strategy.

iv. Quick numerical updating schemes are developed for the optimality conditions following a change in the active set.

(c) A set of rules for ensuring convergence of the nonlinear OPF optimization. This includes:

i. The use of a step size as a means to control nonlinear convergence of the OPF algorithm. This has been used in the quasi-Newton Han-Powell method for OPF, but not in a Newton method.

ii. The development of simple heuristics for this step size control.

iii. The use of a non-standard load flow solver to ensure descent of the objective function (used, albeit, as a backup for a more commonly used technique).

iv. The development of simple heuristics for step size control in the Newton-Raphson solver.

v. In the context of the OPF problem, a theoretical explanation of a problem inherent to the linearization of nonlinear equations, which causes a numerical problem called zigzagging.

vi. The development of simple heuristics used to reduce zigzagging.

(d) The addition of a load-tracking loop to the OPF algorithm, as a means to produce quick OPF solutions for subsequent loads.

(3) A theoretical analysis for applications of the continuation method to the subproblems of other operations tasks.

(a) An analysis of the OPF subproblem for economic dispatch based on the varying load strategy.

(b) An analysis of the minimum loss subproblem.

i. The linear formulation of real power losses $P_L = e^T(P_g - P_d)$ is proposed for use as the objective function in minimum loss optimization problems.

ii. The optimality conditions resulting from the use of the linear objective are formulated for the minimum loss problem.

iii. Theory is provided for the solution of this minimum loss problem by the varying limits strategy.

iv. Theory is provided for the solution of this minimum loss problem by the varying load strategy.

(c) An analysis of the load shedding problem.

i. A formulation of the load shedding problem in which the objective is clearly a norm of the unsatisfied load.

ii. Inclusion of the active dispatching constraints in the formulation of the load shedding constraints.

iii. An analysis and a proposed algorithm which links the optimal loads suggested by the load shedding to its corresponding unique optimal dispatch of generations.

iv. The optimality conditions are formulated for this minimum loss problem.

v. Theory is provided for the solution of the load shedding problem using the "varying demand" strategy. This strategy is also new to this thesis.

(d) Theory for the inclusion of ramp constraints to the OPF subproblem developed in this thesis and a suggested method of solution using continuation methods.

(e) Strategies are suggested for parameter variations of the continuation method associated with contingencies in the power system.

(f) A solution methodology based on the continuation approach is proposed for real power dispatch based on the DC load flow model augmented with phase shifter variables.

(g) Theory is provided to compute bus incremental costs for real and reactive power loads, based on the solution of economic dispatch using the varying load strategy.

(4) A fast, hybrid, real power dispatch algorithm, implemented in the OPF algorithm, but which can be used on its own.

(5) A computer program implementing the OPF algorithm for the economic dispatch task, using the varying limits strategy in the subproblem and a load-tracking loop.

(a) A large set of numerical results from the program, illustrating the behaviour of the algorithm on four test systems, ranging in size from 6 to 118 buses.

(b) A discussion which provides a fair amount of insight into the mechanisms of the computation.

(c) The confirmation of predicted advantages of the method: (1) in tracking the subproblem solutions with relatively few breakpoints, (2) in obtaining descent of the objective function at each iteration, and (3) in computing quick solutions in the load-tracking mode.

(d) A description and an analysis of future improvements to the present program.

CHAPTER II

SURVEY AND ANALYSIS OF THE OPTIMAL POWER FLOW LITERATURE

2.1 Introduction

The history of optimal power flow research can be characterized as the application of more and more powerful optimization tools to a problem which basically has been well-defined since the 1950's. Modern optimization theory dates back to about that time, and advances in numerical optimization has followed the theory closely. Both have made great strides since their infancy. Optimal power flow (OPF) has been quick to profit from these advances and from speccacular advances in computer technology. Steady progress has manifested itself in the solutions of larger and more complex problems in a suitable time frame.

The first part of this chapter proposes an exhaustive survey of the OPF literature, organized with a view on optimization techniques. Preparatory material on basic optimization theory and on numerical methods has been relegated to Appendices 2.1 and 2.2, respectively. The chapter begins with a general overview of tasks performed by OPF, and the evolution of solution techniques. There follows a discussion on recent trends. A detailed survey of the literature then lists the major contributions in each branch of activity; these and other references have been compiled in an exhaustive list, presented in Appendix 2.3.

The second part of this chapter proposes an analysis of numerical optimization methods used in OPF and its subsets. This covers formulations and solution procedures. Basic elements are described, along with the many available options, and structures of numerical computation are identified. A summary of the analysis, suitable for classification of OPF algorithms, is presented in Appendix 2.4.

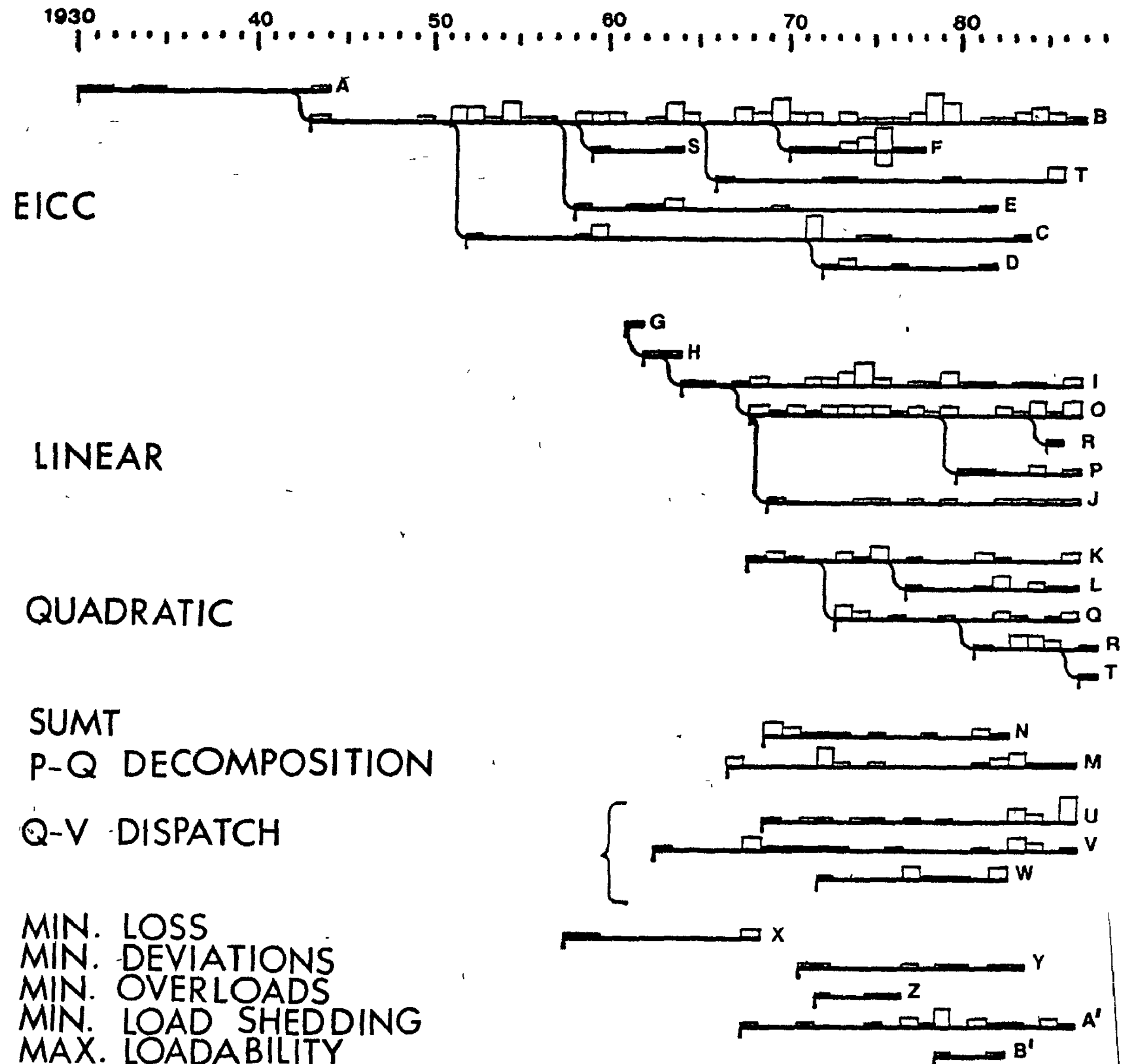
2.2 A General Overview

Figure 2.1 traces the evolution of the OPF literature, from the early stages to the present. It shows many things: the arrival and the disappearance of methods, the relationship between methods, and the interest shown in each method. The structure of the figure is based on the presentation of numerical optimization methods of Appendix 2.2. Its actual filling-in is based on the compilation of some three hundred publications on OPF and its subsets. The chronology of each method is clearly depicted, and the interest shown in each method, as measured by the number of yearly publications, is represented by the height of the blocks. The literature survey will consist more or less in filling in the details of fig. 2.1.

The main groupings are according to the performed tasks. By far the most studied is minimum fuel costs, or economic dispatch. The first systematic efforts, dating back to the early 1930's, produced the "incremental loading" methods (branch A in fig. 2.1). The addition of losses to the network model, starting in 1943, resulted in the "classical" economic dispatch (branch B). Optimality conditions for these methods constitute the equal incremental cost criterion ,and thus lends its name to this first group of methods. Other branches in this group consider additional problems, such as valve point loading (branch E), dynamic constraints (branches T) and interties (branches C and D).

Equal incremental cost methods constitute the most popular economic dispatch tool. Loss models have been improved over the years, and in recent proposals, classical economic dispatch has served as a building block for more complete algorithms. These methods are simple and fast, because they limit the network modelling to its simplest expression.

A second group, dating from the early 1960's, occupies the other end of the spectrum. The true OPF techniques consider all variables, nonlinearities, and bounds. Their development was made possible by the appearance of powerful optimality conditions in the 1950's. The various applications have followed the evolution in numerical optimization, from inefficient successive approximation (early 60's), to gradient and penalty methods (60's and early



LEGEND

A	Incremental loading	O	ED-Linear progr
B	Classical EICC/loss formulae	P	ED-Linear network techniques
C	Interconnected systems & EICC	Q	ED-Quadratic programming
D	Interconnected systems & other	R	ED-Parametric Programming
E	Valve point loading	S	ED-Dynamic programming
F	Environmental dispatch	T	Dynamic dispatch
G	OPF-Lagrangian techniques	U	QV dispatch, other than V & W
H	OPF-Successive approx	V	QV dispatch by minimum loss
I	OPF-Gradient method	W	QV dispatch by min slack generation
J	OPF-Successive linear progr	X	Minimum loss
K	OPF-Newton Strategy	Y	Minimum deviations
L	OPF-Successive quadratic progr	Z	Minimum overloads
M	OPF-PQ decomposition	A'	Minimum load shedding
N	OPF-SUMT	B'	Maximum loadability

Figure 2.1. A chart of OPF publications

70's), to sequential linear and quadratic programming (mid 70's - present), and parametric programming (80's - present). In figure 2.1, OPF methods have been split into three groups. Gradient (branch I) and successive linear programming (branch J) are placed in the group of linear methods. Newton strategy (branch K) and Projected Lagrangian (branch L) are in the group of quadratic methods. Penalty methods (branch N) are classified on their own for their formulation; their solution, in fact, usually follows a quadratic technique.

A third group bridges the gap between the equal incremental cost methods and the full OPF techniques. First appearing in the late 1960's - early 70's, methods based on linear (branch O) or quadratic (branch Q) programming offer some of the advantages of both previous groups. While retaining a simpler linearized model of the network, all variables and bounds can be represented. Chronologically, in fig. 2.1, these methods are portrayed as spin-offs from the more general nonlinear methods. More recent sub-branches explore the use of network techniques for linear programming (branch P), parametric programming (branches R), and dynamic constraints (branches T).

A remaining group for economic dispatch is based on the decomposition of the OPF problem into two subproblems, for real and reactive powers (branch N). Each subproblem holds constant variables of the other subproblem. The subproblems are solved alternatively, with one feeding new values to the other. Each subproblem can then fall into one of the previous categories.

Tasks other than economic dispatch have received relatively little attention. In fig. 2.1, only the reactive power-voltage control task has enough entries to warrant some kind of classification. It is based on objective functions though, and not on optimization methods. These other tasks will be discussed later.

2.3 Recent Trends in OPF

Recent trends in OPF can be traced in its more important review papers. Going back to 1974, Sasson and Merril [Sasson an Merrill 1974] discuss the different uses of OPF, linking it to planning, medium-term operation problems (unit commitment, reserves and scheduling), and special static problems (reactive dispatch, environmental dispatch). Optimization methods considered promising at the time were unconstrained Newton and quasi-Newton techniques associated with penalty methods, and gradient methods aimed at solving the Kuhn-Tucker conditions.

A few years later, Happ's review [Happ 1977] put the emphasis on equal incremental cost methods for economic dispatch, with various improvements (valve point loading, multi-area dispatch, considerations for automatic generation control (AGC), and environmental dispatch). He also presents the more general nonlinear problem and cites linear programming and gradient solution techniques.

In the two previous reviews, the authors basically reported on their areas of expertise. Stott and his colleagues [Stott et.al. 1980] give a more comprehensive picture of the state of the art at the beginning of the 1980's. They look at the different elements (objectives, constraints, modelling) of the problem, and qualities of a good numerical solver. Solution techniques considered promising are primal and dual-based gradient methods, penalty methods, and successive linear programming. The benefits of decoupling real and reactive subproblems are also discussed. Finally, they describe the idea of using the OPF as a simulation tool for contingency-constrained OPF.

Talukdar and Wu [Talukdar & Wu 1981] cover many areas of power system operation in their review. Methods reviewed for OPF are classical economic dispatch, gradient methods, and for the first time in a review, a (non-penalty) quadratic method, the Han-Powell quasi-Newton method.

Dillon [Dillon 1982] presented a study of penalty methods at about the same time, and concluded that they present serious weaknesses for thermal power dispatch. Interest has waned of late in these methods.

Much progress was achieved in quadratic methods by the time the review by Carpentier was published [Carpentier 1985]. He first suggests an interesting classification of methods, based on properties of the algorithmic structure. Solution techniques using gradient methods and the new projected Lagrangian methods are described and compared for reliability and computation time on typical problems. He singles out one of the quadratic methods as being very promising [Sun et.al. 1984]. He also notes that despite recent advances in software and hardware, none of these methods is suitable for on-line use. He then discusses recent parametric quadratic methods applied to decoupled real and reactive subproblems. These methods were developed amongst others by Carpentier and his colleagues. In this and in a follow-up paper [Carpentier 1987], he considers the possibility of using these algorithms on-line and in contingency-constrained OPF.

The latest important review on OPF is that of Stott and colleagues [Stott et.al. 1987]. As in their 1980 review, these authors present the state of the art in the field, as well as their own insights into the different aspects of the OPF problem, and the directions for future research. They classify the latest OPF implementations in much the same way as in Carpentier's 1985 review, and comment on the different structures. The emphasis is placed however on the development of new security-constrained OPF algorithms, with different levels of complexity for the security control strategies.

Present trends for OPF are geared towards dispatching in real-time. Most recent implementations seem to be going towards the decoupling idea [Stott & Alsac 1983, Sun et.al. 1984, Carpentier 1985, Innorta & Marannino 1985, Contaxis et.al. 1986]. The real power problem having been extensively studied, interest has shifted of late towards voltage - reactive power dispatch. The nonlinearity of the latter problem makes it difficult to resolve quickly [Stott & Alsac 1983], but reasonable computation times have been reported for large systems [El-Kady et.al. 1986]. The real and reactive subproblems are being solved mostly using quadratic methods.

2.4 Detailed Review of the Literature

Full lists of publications for each branch of figure 2.1 can be found in Appendix 2.3, giving author and year of publication. The complete information can then be found in a bibliography, separate from the references. This section points out only the main contributions in each branch of the OPF literature.

2.4.1 Economic Dispatch

a) EICC Group

The early work on incremental loading (branch A) is summarized in the book by Steinberg and Smith [Steinberg & Smith 1943]. Fuel cost curves are accurately represented, including "bumps" due to valve points. The network is represented by a lossless power balance equation. Using this model, equations are derived to characterize optimal operating conditions; they are called coordination equations. The ensuing natural solution strategy is called the equal incremental cost criterion. The implementation at the time was by graph or by a dedicated slide-rule.

The addition of a model for real power transmission losses to the incremental loading problem lead to classical economic dispatch. A first loss model was proposed by George [George 1943]. Improvements from the late 1940's - early 50's produced the B coefficient model [Kirchmayer & Stagg 1951, Glimn et.al. 1952, Hale 1952]. It is based on certain assumptions [Tudor & Lewis 1963] which allow real power losses to be expressed as a quadratic function of real power generations. Despite being a rough approximation, it has remained popular [for example, Aoki & Satoh 1984, Mansour et.al. 1984, Wenyuan 1985]. Coordination equations were developed [George et.al. 1949, Kirchmayer & Stagg 1952, Glimn et.al. 1954] to incorporate the losses. Since then, a quadratic approximation of the objective function has usually been chosen for implementations, on analogue computers [George et.al. 1949] and on digital computers [most publications from the early 50's]. Kirchmayer's book [Kirchmayer 1958] summarizes the work on classical economic dispatch.

At about the same time, improved loss models were proposed [Brownlee 1954, Cahn 1955] based on network equations. Linear models were built from first order differential information [Tudor & Lewis 1963, Van Ness 1963], and later quadratic models using second order information [Hill & Stevenson 1968]. In these, all independent variables are involved. The linear models have remained more popular, since they are more easily updated in iterative algorithms. These iterative algorithms appeared in the 1970's [Happ 1974, Wollenberg & Stadlin 1974, Shoults 1977]. Basically, they use the EICC method as a subproblem, updating nonlinear information and handling dependent constraints at each iteration.

Branches C and D look at interconnected systems. The coordination equation approach (branch C) was first proposed with two or three interties [Glimn et.al. 1958, Kirchmayer 1959, Kerr & Kirchmayer 1959]. Methods for systems with any number of interties appeared some ten years later [Aldritch et.al. 1971, Happ 1975]. In all these proposals, equality constraints are added to the basic formulation to enforce intertie power flows. Some other approaches (branch D) consider nonlinear programming [Peschon 1972] and linear programming [Deo 1973].

Computer implementations of economic dispatch incorporating the effects of valve points on the cost function are rare. That is because the modelling of nonlinearities and the ensuing optimization problem are difficult, and the benefits rather small. Reported benefits range from 0.1% [Ringlee & Williams 1963] to 1.8% [Decker & Brooks 1958] over dispatches which neglect valve points. Network models in these implementations are limited to their simplest expression to avoid complicated nonlinear programming. Another difficulty, discussed by most authors and analyzed by Vojdani [Famideh-Vojdani 1982], is the discontinuity of optimal solution trajectories.

Concerns with the effect of emissions from fuel-burning power plants on the environment (branch F) attracted attention over a period of some 7 years, starting in 1970. Early proposals for minimum emission dispatch [Gent & Lamont 1971] or reduced area-wide emissions [Delson 1974] were usually rather simple, adding a single equality constraint to the problem. The paper by Cadogan and Eisenberg [Cadogan and Eisenberg 1977] collects many of the ideas in the field, proposing different formulations, with single-area or multiple-location emission constraints. Note that the proposed constraints have always been static, with no build-up or displacements of emissions being considered.

Solution of the classical economic dispatch by dynamic programming (branch S) was proposed to avoid modelling valve point nonlinearities explicitly [Ringlee & Williams 1963]. This branch has long since been terminated.

In the remaining branch T, dynamic dispatching proposes the addition of dynamic ramp constraints to the usual static economic dispatch. This is different from the case with static ramp constraints [Isoda 1982], in that it offers a look ahead capability , to avoid infeasible operation following large jumps in system load. A disadvantage is that many coupled static problems must be solved in tandem, but proposals based on classical economic dispatch [Bechert & Kwatny 1972, Ross & Kim 1980] are relatively easy to solve.

b) Linear Methods Group

This regroups the various dispatching and true OPF methods based on linear approximations forming the search direction, plus two precursors. Squires [Squires 1961] used classical Lagrangian techniques to formulate the optimality conditions for OPF (branch G). These incorporate the load flow equations, but neglect bounds on variables. A year later, Carpentier [Carpentier 1962] presented the optimality conditions for OPF, including bounds, based on the Kuhn-Tucker conditions. This is generally considered the first publication on OPF. The proposed solution technique of successive approximation was inefficient though, and it was never implemented in a production code.

The first efficient solution of OPF was accomplished using gradient methods. Basically two variants dominate the literature. The Carpentier approach [Carpentier 1968 & 1972] solves the OPF by the primal method. The Dommel and Tinney approach [Dommel & Tinney 1968] solves the Kuhn-Tucker equations using a combination of the gradient method for a fixed set of independent variables and penalty functions for violated dependent constraints. The latter has the advantage of a fixed formulation, but is hampered by the problems associated with penalty factors. It has been the more popular of the two gradient methods, and indeed one of the more popular in the entire literature. Important improvements were proposed to exploit sparsity [Peschon 1971], or the fast decoupled load flow model [Alsac & Stott 1975]. Many features were included to the basic algorithm in an implementation by researchers from General Electric [Burchett et.al. 1980].

Two other applications of gradient methods can be mentioned. Wu and his colleagues [Wu et.al. 1979] solve the OPF in two stages, both by the gradient method. The first stage ignores dependent constraints; after adjustments to the solution, the second stage adds possibly violated voltages to the objective via penalty functions. This program has the capability of handling very large problems, but often infeasible values remain upon completion. In the other application, Burchett and colleagues [Burchett et.al. 1981] applied the general purpose nonlinear programming package MINOS [Murtaugh & Saunders 1980 & 1983] to solve the OPF problem. It builds internally a sequence of projected Lagrangian subproblems, which it solves by any one of a family of gradient methods.

Successive linear programming (branch J) has been used in a few OPF applications to date. Khan [Khan & Kuppusamy 1979] suggests the use of a linear programming subproblem in a nonlinear iterative loop, with special considerations to avoid oscillation of the iterates. Stott and Alsac [Stott & Alsac 1983] present the results of their experience with SLP, following extensive work with linear programming. Researchers from Control Data Corporation [Van Meeteren et.al. 1986] reported recently on their work with successive linear programming, but details are lacking.

Linear programming applications (branch O) abound in real power dispatching. Early papers [Wells 1968, Shen & Laughton 1970] were already quite complete, with piecewise-linear objectives and constraints on all variables. A first major production code based on LP was developed by EDF [Merlin 1972] to handle most operating tasks. Many publications have been presented since then in the 1970's - 80's, the best known being those of Stott and colleagues [Stott & Marinho 1979]. Many features, including piecewise-linear objectives and the latest sparse methods are included in their program for real power dispatch. Very fast network techniques (branch P) have been proposed during the 1980's [Lee et.al 1980 &1981, Hobson 1984] to solve network-structured linear programs.

Parametric linear programming (branch R) is the newest variant of the group [Innorta & Marannino 1985]. It efficiently computes solution trajectories for real power dispatch, given a load forecast. Look-ahead times of as much as one half hour have been considered. This branch might be short-lived, since the authors have recently converted to parametric quadratic programming.

c) Quadratic Methods Group

Many of the tailor-made OPF algorithms proposed in the late 1960's-early 70's can be classified as Newton strategy methods (branch K) [Peschon et.al. 1968, El-Abiad & Jaimes 1969, Shen & Laughton 1969]. Typically these methods solve the Kuhn-Tucker equations using the Newton-Raphson solver for nonlinear equations, with added controls for active constraints. The standard successive quadratic programming, which developed in the early 70's, offers a more flexible solution algorithm, made up of reliable parts. The first applications in power systems came before the popularization of the added exact penalty functions. In 1973, two papers [Nabona & Freris 1973, Nicholson & Sterling 1973] proposed quadratic subproblems to drive the optimization, followed by a Newton-Raphson solver to maintain feasibility. In both cases the subproblems were solved using Beale's method, a simplex-type technique.

Since the mid 70's, quadratic programming methods based on the Kuhn-Tucker equations have taken over. Methods using this formulation and applying the Newton search direction in the solution process are called Lagrange-Newton methods. Dillon [1975 & 1981] has investigated these methods, and proposed parametric programming extensions. This branch has evolved, with the advent of exact penalty functions, into projected Lagrangian methods.

The first projected Lagrangian programs for OPF (branch L) were proposed in the late 70's - early 80's [Biggs & Laughton 1977, Lipowski & Charalambous 1981]. They have been overshadowed by some well-publicized commercial OPF packages. The General Electric package has evolved along with MINOS from a gradient solver [Burchett et.al. 1981] to a quasi-Newton solver [Burchett et.al. 1982] to a Newton solver [Burchett et.al. 1984, Maria & Findlay 1986]. The ESCA package [Sun et.al 1984] implements the Lagrange-Newton solver to sparse decoupled subproblems, with penalty terms added to handle violated dependent constraints. The PCA package [PCA 1985] also uses the Lagrange-Newton solver and decoupling, but little else is known about it. Researchers from Control Data Corporation [Van Meeteren et.al 1986] have reported recently on their package, including both successive linear and successive quadratic programming.

Quadratic programming applications for dispatching first appeared in a flurry, in 1973 - 74 [Nabona & Freris 1973, Nicholson & Sterling 1973, Reid & Hasdorff 1973, Podmore 1974, Wollenberg & Stadlin 1974]. Some of these were proposed as subproblems and some on their own, for real power dispatch and reactive power dispatch. Podmore formulated a quadratic program, but inappropriately solved it by the gradient method. Another good paper on the subject is that of Dayal [Dayal et.al 1976], which also includes discussions on sparsity and degeneracy. All those papers used primal simplex-type solution techniques. Since then few implementations have been reported. Applications based on the Kuhn-Tucker conditions appeared in 1982 [Bottero et.al. 1982, Quintana & Lipowski 1982].

The two remaining branches in this group are related. Parametric quadratic programming (branch R) was proposed by Dillon [Dillon 1981], but he did not pursue the issue. Since the early 1980's three research groups have been active in the field. The McGill University group, of which this author is a member, has reported good results for real power dispatch using the continuation method, in two different applications [Famideh-Vojdani & Galiana 1983, Galiana et.al 1983]. A recent paper reported on various new parameter variations for redispatching [Huneault et.al. 1985]. The EDF group in France has solved decoupled real power [Carpentier & Cotto 1983] and reactive power [Blanchon et. al. 1984] problems. They report that the very fast computation times make these methods promising for real-time dispatching. A third group, from the Italian power utility ENEL, have applied parametric linear programming [Innorta & Marannino 1985] and parametric quadratic programming [Innorta et.al. 1987] to real power dispatch. In the second publication, they also include dynamic costs and ramp constraints (thus constituting branch T); the combination of parametric and dynamic techniques seems well-suited.

d) Penalty Methods Group

This group is based mainly on the work of Sasson and colleagues during the late 1960's - early 70's. They applied the newly developed SUMT method to OPF. Their first implementations used quasi-Newton techniques [Sasson 1969a]. These were soon considered inadequate for large power system problems, and were replaced by Newton methods [Sasson et.al 1971b & 1973]. They proposed applications for economic dispatch, minimum loss [Sasson 1969b] and even load flow [Sasson et.al. 1971a]. Since then, similar applications have appeared sporadically.

e) P-Q Decomposition Group

The splitting of OPF into subproblems has been quite common. Today many publications propose real or reactive dispatch algorithms which could be inserted as subproblems into a decoupled OPF. Publications placed in this group (branch M), however, consider the entire nonlinear optimization. Dopazo was the first to solve a P-Q decomposition [Dopazo et.al. 1967]. The proposed solution process used classical economic dispatch for real power and a minimum loss objective for reactive power. The latter was solved using a gradient solver. Other objectives for reactive power subproblems are discussed a little further. Later publications, grouped mostly in the early 1970's and early 1980's, use the same solvers for both subproblems. Solvers include linear programming [Chamorel & Germond 1982], quadratic programming [Nicholson & Stirling 1973, Contaxis et.al. 1983], gradient [Talukdar et.al. 1983, Lee et.al. 1984], and SUMT - Newton [Housos & Irisarri 1983]. Recall that several of the projected Lagrangian methods reviewed earlier also use decoupled formulations.

2.4.2 Reactive Power and Voltage Control

Reactive power and voltage are often dispatched alone, after real power, using the remaining degrees of freedom offered by the network. The benefits are "reduced production costs, unloading of equipment, and an improved voltage profile" [Fernandes et.al. 1978]. Typically some norm of reactive power deviations [Kishore & Hill 1971, Stott & Alsac 1983], or a closely related function such as real power losses [Peschon et.al. 1968, Billinton & Sachdeva 1972, Franchi et.al. 1983] or dependent "slack" generation [Adielson 1972, Fernandes et.al. 1978, Contaxis et.al. 1983] is minimized, subject to voltage, reactive power, shunt, and tap constraints. Three branches of fig. 2.1, U,V, and W respectively, correspond to the three objective functions. In some of the earlier cases only voltages were controlled [Kumai & Ode 1968, Hano et.al. 1969, Narita & Hamman 1971, Savulescu 1976]. In a recent paper [El-Kady et.al. 1986], the objective is a combination of minimum loss and minimum deviations.

Solution methods for this problem include linear programming [Kishore & Hill 1971, El-Shibini & Dayeh 1975, Zhang 1986a & b], parametric linear programming [Blanchon et.al. 1984], and successive linear programming [Elangovan 1983, Stott & Alsac 1983, Mota-Palomino & Quintana 1986]. Stott and Alsac note that due to the high nonlinearity of the problem, a single pass of LP is inadequate; in fact, usually many iterations are required. Other solution methods are quadratic programming [Nicholson & Sterling 1973], gradient [Fernandes et.al. 1978], and nonlinear quadratic methods [Horton & Grigsby 1984, Padiyar 1986]. A sequence of papers on the subject by researchers at General Electric [Fernandes et.al 1978,Happ & Wirgau 1978, Wirgau 1979, Aldrich et.al 1980] and in collaboration with researchers at Ontario-Hydro [El-Kady et.al 1986] closely parallel their work reported earlier on economic dispatch.

2.4.3 Other OPF Tasks

Minimum loss (branch X) solved as a function of all dispatchable variables has received little attention. It is of some use only where dispatchable power comes mostly from hydro-generation. Two early papers [Calvert & Sze 1958, Sze et.al 1959] apply classical Lagrangian techniques to obtain the optimality conditions of the problem. A later application [Sasson 1969b] solves the problem using SUMT.

Post-contingency dispatching problems make up the next two branches. The minimum deviations task (branch Y) is invoked following a contingency, to find a feasible operating point as close as possible to the pre-contingency operating point. The deviations can be considered for some variables [Hobson 1980, Zhang & Brameller 1984] or all of them [Kaltenbach & Hajdu 1971]. Linear programming is the most common solution method.

The minimum violations task (branch Z) is also invoked following a contingency, and seeks only a feasible operating point. Two proposed solution methods are least squares (quadratic objective) [Shoults & Chen 1976] and SUMT, using a gradient solver [Sachdev & Ibrahim 1975]. Some algorithms described in the previous pages include something similar, a "phase one" of linear programming [Murty 1983], to find a feasible operating point when needed [for example, Horton & Grigsby 1984].

Load shedding (branch A') is performed only in the emergency state, when the demand cannot be met without violating constraints. In this thesis, only steady-state load shedding, also called load curtailment, will be considered. Up to now, there has been no clear-cut methodology for load shedding. On the one hand, researchers are still striving to formulate "the least objectionable" solution [Zaborsky et.al 1985]. On the other hand, it is felt that certain numerical difficulties have impeded the development; they are reported a little further.

The literature can be split into two groups: those algorithms which minimize a norm of shed load, and those which control loads in trying to achieve some other objective. The algorithms in the first group will express load as a function of generation and line flows [Palaniswamy et.al. 1981], or include other independent variables in the objective [Subramanian 1971]. Those in the second group usually minimize a combination of load shedding and some other function, such as minimum deviation [Ejebe et.al. 1977, Ghoneim et.al. 1977, Chan & Schweppe 1979] or minimum violation [Krogh 1983]. Others have suggested simply minimum deviations, with some control on loads [Medicherla et.al. 1979]. In most cases, priorities can be attributed to the loads by using weighting coefficients.

The complications in the formulation of the objective were probably meant to avoid the forementioned numerical problems. Chan and Yip [Chan & Yip 1979] tackled this problem when they proposed a decoupled algorithm, with loads forming the objective for one subproblem and generations for the other. They note that in the first subproblem, the reallocation of generation is non-unique. What's more, dependent load flow constraints cannot be modelled in this case. In the Chan and Yip paper, the first subproblem is solved completely, and then dependent constraints are handled in the reallocation of generation. The first important paper on load shedding, by Hajdu and colleagues [Hajdu et.al. 1968] also proposes a load-only objective, subject to a full set of constraints, but they did not seem to notice the problem of non-uniqueness.

Solution methods for load shedding algorithms range from Newton strategy [Hajdu et.al. 1968], to quasi-Newton [Palaniswamy 1981], and quadratic programming [Subramanial 1971], but mostly linear programming [Subramanial 1971, Ejebe et.al 1977, Ghoneim 1977, Krogh et.al. 1983]. Piecewise-linear objective functions are used for linear programming in the two papers co-authored by Chan [Chan & Schweppe 1979, Chan & Yip 1979].

The last branch is maximum loadability (branch B'). It finds the largest sum of real power generations which can safely satisfy the demand, given a load distribution [Garver et.al. 1979]. It is useful as a planning task, in determining the influence of varying parameters in expansion planning. The analytical approach has also been proposed [Dersin & Levis 1982], in trying to establish a description of the loadability region.

2.5 Analysis of Numerical Optimization Algorithms Used in OPF

This section undertakes the analysis of numerical optimization methods in general. Some basic building blocks and their various modes of interconnection are proposed. The role of each block is explained, along with different available options in OPF. Then the algorithmic structures identified in the OPF literature are presented.

A purpose of this section is to show that numerical optimization algorithms are basically an assembly of parts. The parts are often interchangeable, and can be chosen to suit the problem. The assembly need not fit exactly into any recognizable category; parts can be added or deleted if necessary. That, in fact, is the case of many proposed applications.

2.5.1 The Elements of a Nonlinear Program

Despite the diversity of numerical optimization algorithms, it is felt that they share relatively few, fairly simple, common elements. Conceptually, the solution process of an iterative optimization algorithm can be broken down into seven parts: formulation, initialization, projection, choice of subproblem, solution of subproblem, test for convergence, and rules for starting a new iteration. The formulation and the choice of subproblem are fixed at the outset; the other parts form the numerical procedure per se. Each part is now described, with references made to its use in OPF or its subsets.

Formulation: This is the choice of the problem to solve. The constraints can be modelled with different degrees of accuracy, depending on the available computation time. We consider three main levels of complexity, to which we give the following names:

- The OPF level, which retains all the nonlinearities.

- The dispatch level, in which the network model is replaced by a linear approximation, but not reduced in dimension. There are different linear models, varying in complexity. Three commonly used are the load flow Jacobian model [Carpentier 1962], the fast decoupled load flow model [Alsac & Stott 1975], and for real power dispatch, the DC load flow model [Wood & Wollenberg 1984].

- The lumped network level, in which all network considerations are reduced to a single real power balance equation. This model is used in equal incremental cost methods. Different models exist, varying in accuracy and complexity.

We will say that problems formulated at the latter two levels are dispatching problems. Until now, only problems with these simplified formulations can be solved on large systems within the time limitations of dispatching (order of minutes).

Completing the description of the load flow constraints, upper and lower bounds are placed on most variables. Bounds are not placed on voltage phase angles per se, but occasionally real power line flows are expressed as phase angle differences, to which bounds are assigned [Merlin 1972]. Another load flow constraint proposed in some lumped network formulations is a second equality constraint for reactive power balance [Moskalev 1963, El-Hawary & Feehan 1978], but that is rarely used.

Constraints other than the static load flow constraints have appeared in various formulations, mostly in dispatching problems. The most important are environmental constraints [Cadogan & Eisenberg 1977], frequency constraints [Somuah & Schweppe 1981, Palaniswamy et.al. 1985] and reserve [Waight et.al. 1981b, Farghal et.al. 1984] and ramp constraints on real power generations. The latter have been proposed as static [Isoda 1982] or dynamic [Ross & Kim 1980] constraints.

Simplifications which can occur at any level are the omission of certain variables or of certain constraints on some variables.

<u>Initialization</u>: An initial estimate of the solution is proposed, which serves as an expansion point for numerical approximations. Usually a feasible value

is required, although in some recent techniques, as mentioned in Appendix 2.1, the problem is bypassed. The description of the initial estimate also includes the corresponding active set and a choice of independent variables.

Projection: This step consists in choosing a subset of the system variables as independent variables. The choice can be fixed at the outset, as in many applications, or can be allowed to vary with each iteration, as in reduced gradient. Advantages of a proper choice are as follows:

- The formulation can be simplified or maintained in some naturally simple form, when expressed as a function of certain variables.

- Functional constraints can be avoided by making active variables independent.

Actions which exploit these two advantages often conflict. Hence, certain rules are needed to manage the choice of independent variables. In the OPF algorithm proposed in this thesis, some simple rules will be provided to strike a compromise between both objectives above.

Choice of subproblem: Nonlinear programs are solved, whether explicitly or implicitly, by generating a sequence of simpler problems. These subproblems are usually very reliable, and usually chosen to terminate. The subproblems are constituted of the following:

- A linear or quadratic approximation of the objective function.

- Usually, a linear approximation of the constraints.

- Usually, constraints on independent variables. These can be handled with exact bounds or penalty terms. Dependent constraints are sometimes omitted from the subproblem and treated elsewhere.

The operations needed to solve the optimization problem described above, for a given approximation (a given expansion point), delimit the subproblem.

In problems with a single equality constraint, a quadratic approximation is sometimes used for the power balance equation, in lumped-network-level methods [Aoki & Satoh 1982]. Using the above definition, each linearization of the approximation constitutes a subproblem, even though the entire process is quite fast.

Solution of the subproblem: Many solution techniques are available to solve any given subproblem. They differ in their choice of search direction and step size, and in the way they handle constraints. Different methods presently used in OPF are as follows:

- For search direction:
    - Quadratic objective functions: Newton step, quasi-Newton step, conjugate gradient step, simplex-type step, solution trajectories from the continuation method.
    - Linear objective functions: gradient step, simplex step, and solution trajectories from the continuation method.

- For active constraint management:
    - Exact constraint verification: primal approach (as in gradient and simplex methods), Lagrange multiplier approach, active set methods.
    - Penalty function methods: exact penalty terms, interior and exterior penalty methods; on independent and/or dependent constraints.

The subproblem can be single-staged or multi-staged. In the single staged subproblem, a single search direction is computed for each approximation. That is the case in gradient and penalty methods. In the multi-staged subproblem, the optimization is carried out with a given approximation until it terminates; examples are linear and quadratic programming.

Test for convergence: If the difference between two quantities is smaller than a certain tolerance, the latest iterate is retained as the optimal solution. The monitored quantities can be:

- Values of the objective function
- Values of a norm of the vector of variables

The comparison of active sets should not be considered, because optimal solutions are sometimes situated at a degenerate vertex (see Appendix 4.2). The quantities can be evaluated in one of these combinations:

- At successive expansion points (usually, in primal methods).
- At an expansion point and the ensuing subproblem's optimal solution (usually, in dual methods)

Rules for starting a new iteration: If convergence has not been achieved, the expansion point and possibly the set of independent variables are updated, to feed the next subproblem. In future in this thesis, this step will be referred to as the "Rules". This is probably the most complex and the most obscure step of the process, since it deals with the nonlinearities. In designing this step the following items must be considered:

ITEM 1. The solution from the subproblem is always infeasible. Solution values are retained for a subset of subproblem variables, and the remaining variables are recomputed using the load flow equations. There are two possibilities:

- The state variables are retained. In that case a simple evaluation of the load flow equations suffices.

- Injections, or a combination of injections and states are retained. An iterative solver is needed.

The first choice is always made in projected Lagrangian methods. It is fast, but since conditions are usually placed on injections, not states, these methods satisfy functional constraints only at the final, optimal solution.

The second choice requires much more computation, but allows some control over the choice of injection values. A possible disadvantage is that the proposed injections could be infeasible [Jarjis 1980]. In that case, for this step to be successful, the process must be able to suggest an alternative. Projected gradient techniques [Rosen 1961] and Newton-Raphson solvers with step size control [Gross & Luini 1975, PCA 1985] offer such capabilities.

ITEM 2. The loadflow feasible point of ITEM 1 could be out of bounds. For methods which require a feasible expansion point, the following actions can be taken:

- Violated values can be set to their bounds and the loadflow equations recomputed. This method is unreliable when more than a few violations occur, and the result may not be useful (see ITEM 3).

- A new value is found along the search segment, established between the previous expansion point and the load flow feasible candidate of ITEM 1.

For methods with penalty terms, excursions from the feasible region are restrained, but only in the next iteration, by updating penalty coefficients.

ITEM 3. The new candidate from ITEM 2 might not be an improvement, in that its objective value might be higher than that of the previous expansion point. In fact, in many methods no precaution is taken to aid descent. One way to ensure descent is to design the search segment between the previous expansion point and the ITEM 1 point to be a descent direction. Then by applying the appropriate step size from the previous expansion point, a better ITEM 3 point can be found [Han 1977]. Another way, for methods using trust regions or exact quadratic penalty terms, is to adjust coefficients to reduce excursions in the next subproblem.

ITEM 4. The dependent/independent partition can be updated in the Rules for the next subproblem, independently of the choice of independent variables made in ITEMS 1, 2, and 3 of this iteration.

The degree to which these items are satisfied and the precise implementation vary greatly from one method to another. That is because (1) there are many choices available in the solution strategies, and (2) there are different levels of simplification, relinquishing accuracy or completeness for shorter computation times.

2.5.2 The Interconnection of Elements

The mode of interconnection of the previously described elements determines the structure of the solution algorithm, or solution strategy. The possible structures differ in two respects: the position held by the processing of dependent functional constraints and the position of the nonlinear iteration loop.

The processing of functional constraints includes their formulation, optimization, verification of feasibility, and remedial actions if necessary. They have been handled in three ways:

- They are processed mainly within the subproblem, and then verified for feasibility by the Rules.
- They are processed at the end of a subproblem, within the nonlinear iteration loop.
- They are processed outside the nonlinear iteration loop.

The first position is common in the OPF literature. It is possibly the most reliable, since excursions of dependent variables are closely monitored in different steps. The second position has been used as an extension to the simpler subproblem with functional constraints neglected. It could be quite fast in dealing with problems with few changes in the active set. The third position is an extension for the case of a nonlinear programming package which cannot handle functional constraints. This position might be awkward for OPF level problems, because an entire nonlinear program must be repeated each time a violated functional constraint occurs.

The nonlinear iteration loop, or major iteration loop, regroups all the operations between the computation of two expansion points. As described earlier, a new expansion point can be computed in one of two circumstances:

- After computing a single search direction and step size, in a single-staged subproblem.

- After termination of a multi-staged subproblem.

The first position is used in gradient and penalty methods. It requires many time-consuming updates of the function, and for gradient methods the Jacobian, although in practice the latter need only be computed every few iterations. The second position processes more information for a given approximation, so it is usually faster overall. The validity of the approximation can be doubtful as the computation leads away from the expansion point, but with the proper precautions that is rarely the case. The second approach has received more attention in recent methods.

The relative positions of the constraint processing and the nonlinear iteration loop are illustrated in a tree structure, in Figure 2.2. Five combinations can be found in the OPF literature, and two of these can be simplified for dispatch level. Three of the ensuing algorithmic structures are displayed in Figures 2.3 - 2.5.

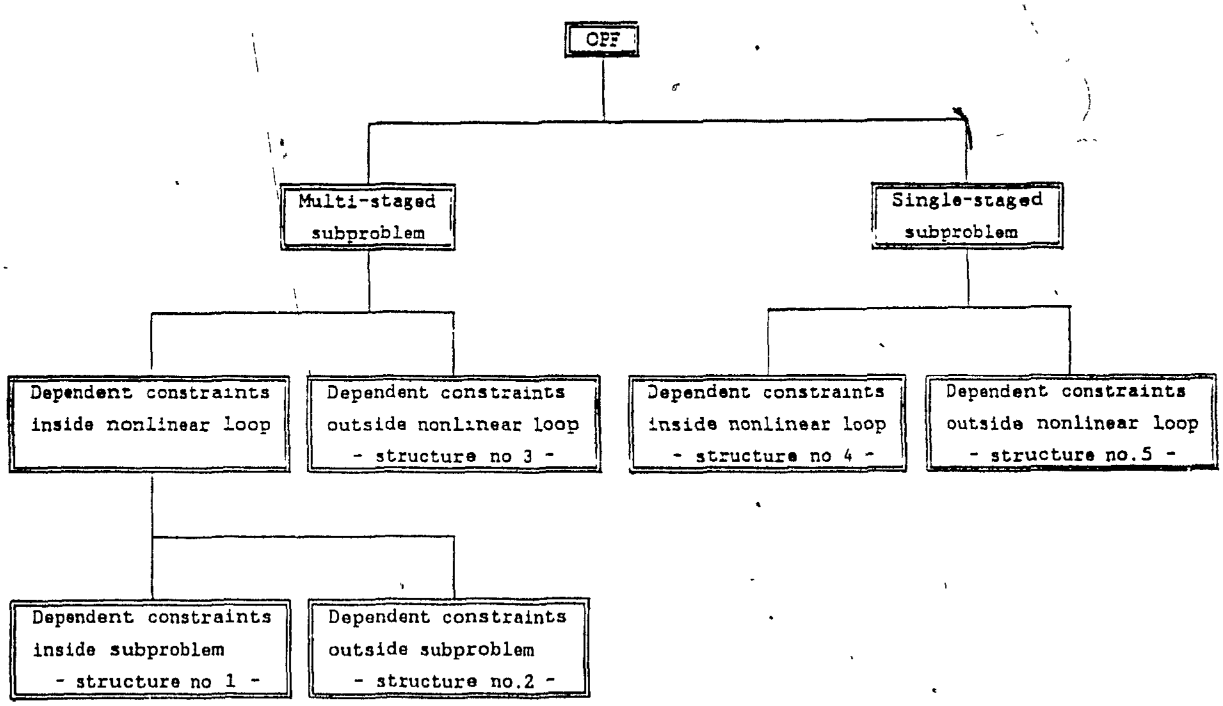


Figure 2.2. Relationships between structures for OPF algorithms.

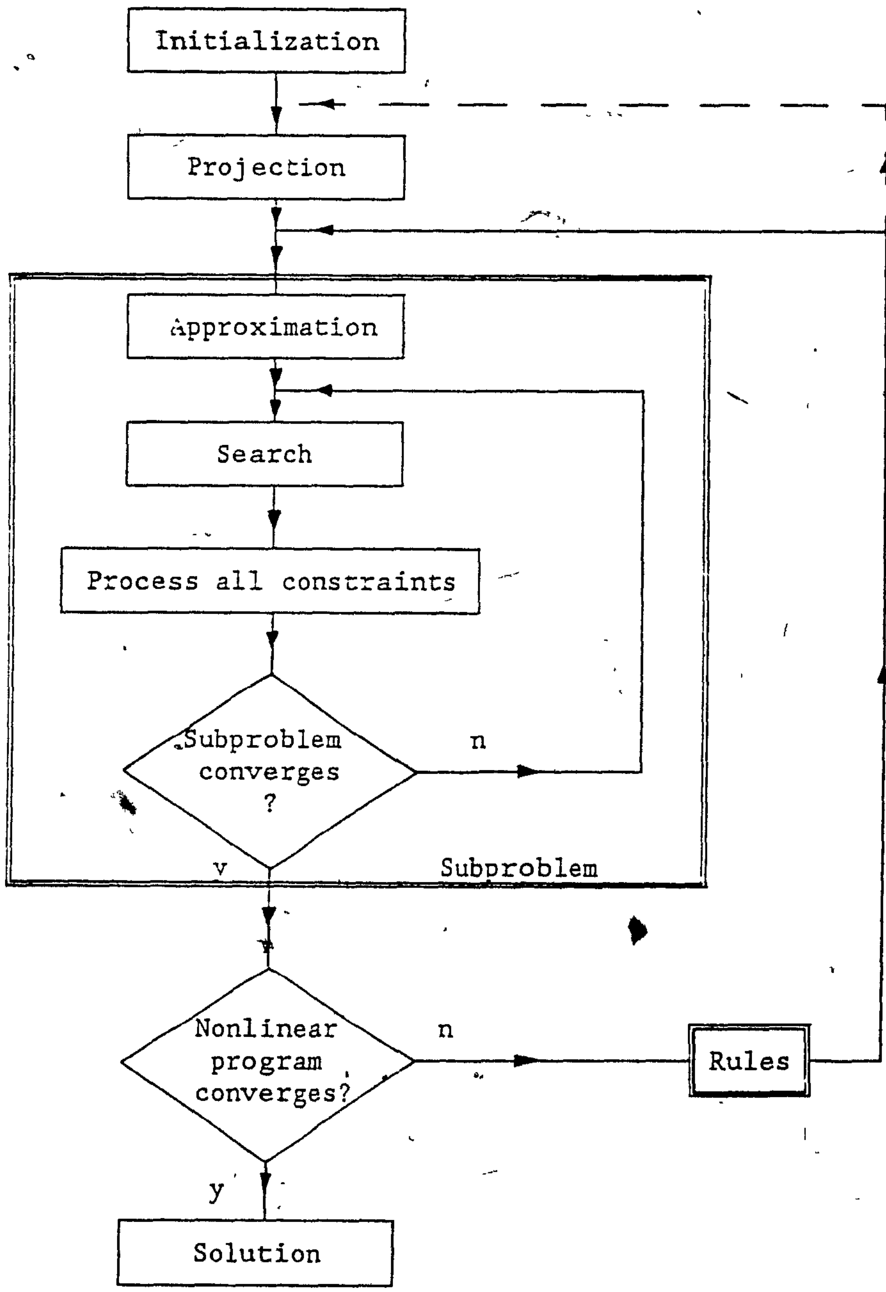


Figure 2.3. Structure no. 1
- Multistage subproblem
- Dependent constraints in subproblem

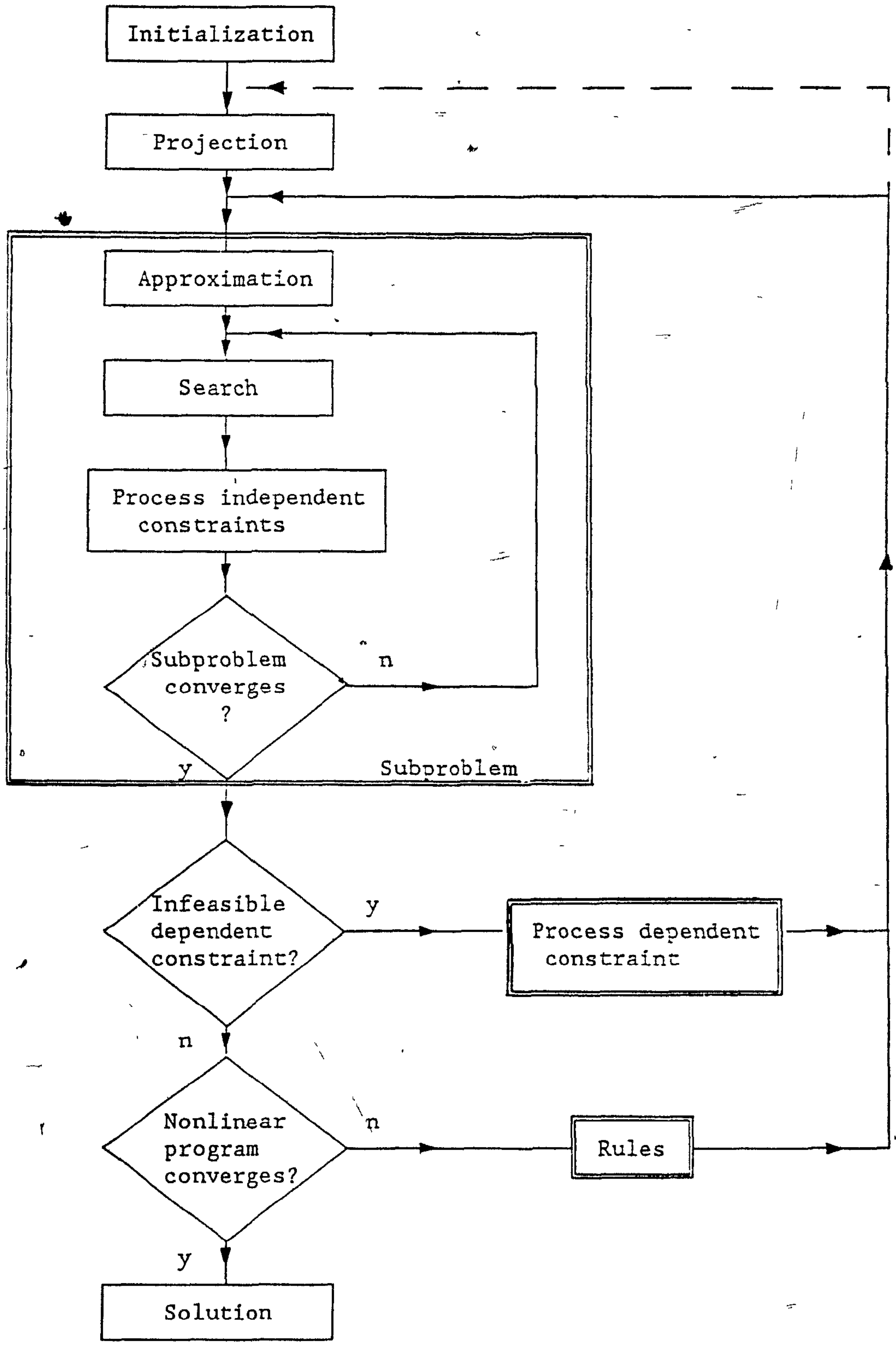


Figure 2.4. Structure no.2.
- Multistage subproblem
- Dependent constraints outside subproblem

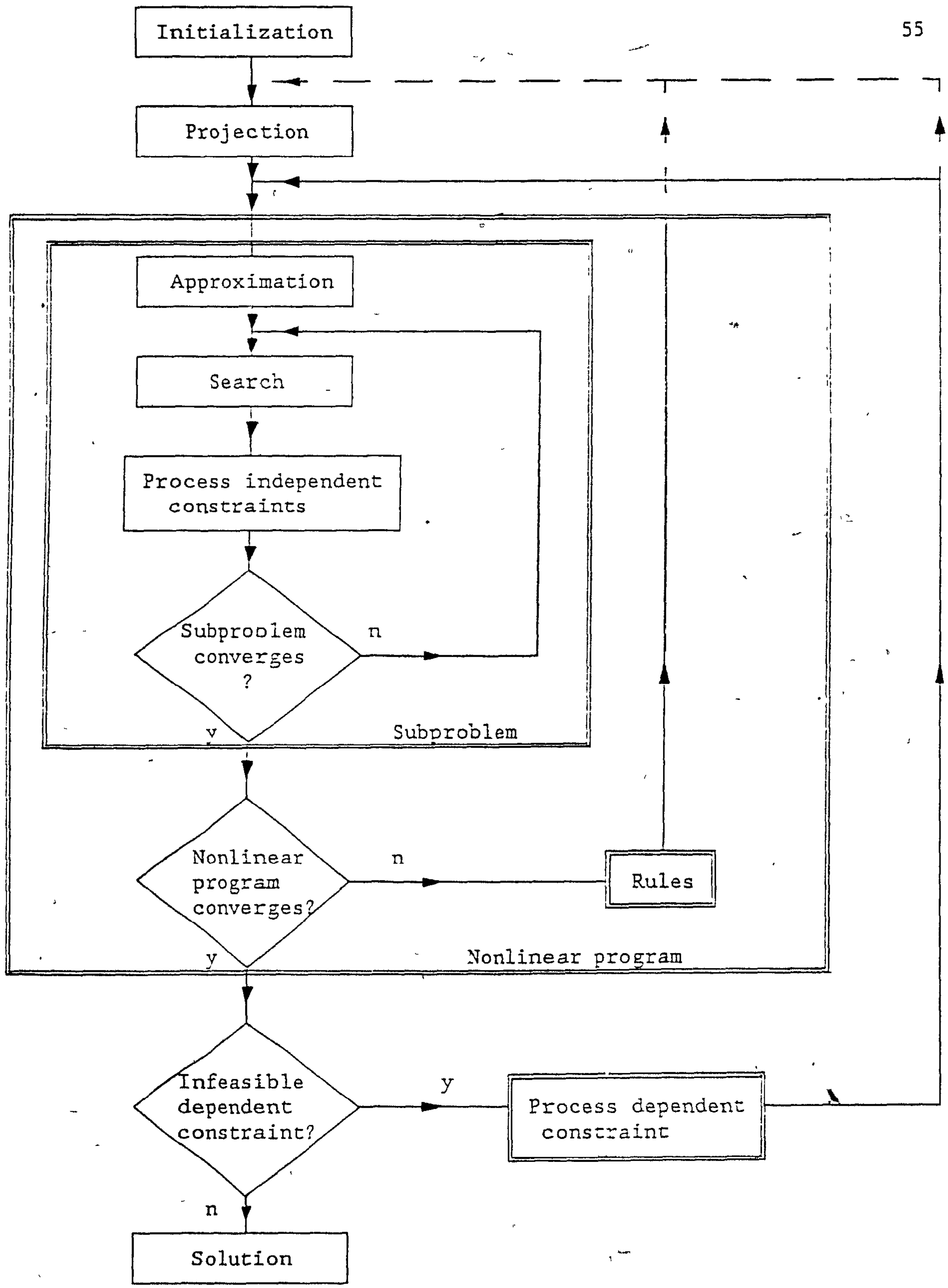


Figure 2.5. Structure no.3.
- Multistage problem
- Dependent constraints outside nonlinear program

Here are some examples of OPF program structures with references from the literature.

Structure no. 1 handles functional constraints inside the multi-staged subproblem. This structure is the most common. It is used in successive linear [Stott & Alsac 1983] and quadratic programming [Burchett et.al. 1984], including Newton strategy methods [Nicholson & Stirling 1973], and in gradient-based projected Lagrangian methods [Burchett et. al. 1982a,b]. The structure of the OPF algorithm developed in this thesis is a based on this structure.

Structure no. 2 handles dependent constraints outside the multi-staged subproblem. Two examples are Algorithm I by Sun et.al. [Sun et. al 1984], which handles dependent constraints by penalty functions, and the algorithm by Contaxis et.al. [Contaxis et.al 1986] which adds violated dependent constraints in the formulation of the next subproblem.

Structure no 3 handles the dependent constraints outside a standard nonlinear programming package. It is used mostly in conjunction with lumped-network-level methods. A proposed optimal solution is scrutinized for possible violations on previously neglected variables. Violated variables would then be processed in the next pass of the nonlinear programming package. A very recent application by Ponrajah [Ponrajah 1987] uses the simple nonlinear programming package once, and then processes violated dependent variables by the continuation method.

Structure no. 4 is similar to structure no. 1, except for its single-staged subproblem (remove the loop in the subproblem of structure 1). An example is Carpentier's "Differential Injection Method" solved by the generalized reduced gradient method [Carpentier 1972].

Structure no. 5 is similar to structure no. 2, except for its single-staged subproblem. An example is Dommel and Tinney's method [Dommel & Tinney 1968], solved by a combination of gradient method for independent variables and penalty method for dependent variables. Penalty methods such as SUMT [Sasson 1969a,b] would also fall into this category.

Dispatch level formulations drop the nonlinear loop. Two possible structures are described in the following.

Structure no. 6 is extracted from structure no. 1. Upon exiting the subproblem, the solution is reached. This is the most common structure for dispatch level problems. Examples of linear [Stott & Marinho 1979] and quadratic [Bottero 1982] programming abound.

Structure no. 7 is extracted from structure no. 2. Here the step of Rules preparing the next iteration are removed. Upon exiting the conditions on dependent constraints with a "no" reply, the solution is reached. This structure is more common in security dispatch or redispatch [Nicholson & Sterling 1973, Hobson 1980].

2.5.3 Formulations of the Tasks Performed by OPF

The objective function is the only part of the optimization process left to describe. It defines the task to be performed. We have already identified seven main tasks in section 2.3, and their purposes have already been discussed. In this section their mathematical formulations are briefly described.

Economic dispatch is usually expressed as a quadratic function of the real power generations [for example, Huneault et.al. 1985], or as a piecewise-linear function, for LP [Stott & Marinho 1979] or SLP [Romano et.al 1981]. More complicated expressions in terms of real power generation [Dillon et.al. 1975], or in terms of state variables [Dhar & Mukherjee 1973] have been proposed, but have attracted little interest. More accurate models of the cost functions, including the effects of valve points [Decker & Brooks 1958], are never expressed analytically; they are formed graphically or, in computer implementations, stored as discrete points.

Reactive power - voltage control is performed in different ways. The choice of objectives has already been discussed. They are usually expressed as a quadratic approximation of reactive power and voltage [Contaxis et.al. 1983], or as a piecewise-linear [Hobson 1980] function of those variables, or in the earlier studies, as a function of voltage alone [Hano et.al. 1969]. Taps and shunts are also included in many implementations.

Minimum loss is usually expressed as a quadratic function of states [Calvert & Sze 1958], or as a quadratic approximation of a combination of injections and states [Sasson 1969b, Horton & Grigsby 1984], or as a linear approximation of injections and states [Stott & Alsac 1983]. It has been expressed in analysis as an exact linear function of real power injections [Alvarado 1978, Elangovan 1983], but so far that form has not been used as an objective function.

The minimum deviations task is usually solved by linear programming.

The minimum violations task has been proposed using least squares and SUMT (quadratic objectives).

Minimum load shedding is performed using linear or piecewise-linear [Chan & Schweppe 1979, Krogh et.al 1983] and quadratic [Hajdu et.al. 1968, Palaniswamy et.al. 1981] functions, of the loads alone [Chan & Yip 1979], or the loads and the controllable variables [Palaniswamy et.al. 1981].

Maximum loadability has been solved by linear programming.

The term redispatch is often mentioned in conjunction with the normal operating tasks. It refers to the organization of the algorithm, and not to the task as such. Redispatch algorithms are usually proposed following a contingency, and profit from pre-contingency information. References made directly to redispatching have been compiled and added to the list in Appendix 2.3.

2.5.4 Enumeration of Problems and Solution Techniques used in OPF and its Subsets

A final, detailed summary of the analysis is presented. It is a compilation containing all formulations, subproblems, and solution techniques encountered in the OPF literature. This list is presented without comment in Appendix 2.4. Some added information, not yet covered, are the choice of coordinates, available variables and specific subsets of allowable independent variables, and exact details of the subproblem structure. The ordering system forms a classification which can be used to accurately describe any OPF algorithm.

CHAPTER III

DESCRIPTION OF A NEW OPTIMAL POWER FLOW ALGORITHM

3.1 Introduction

This chapter presents the description of a new OPF algorithm. Its main features are the use of continuation methods at different levels of the optimization, and the implementation of strong rules to enhance robustness. These rules assure that from one iteration to another, the objective function always decreases and the proposed solution is always feasible.

The structure of the OPF algorithm is presented in general, in a first section, and then in detail, following the formulation of the OPF problem. The detailed description will be carried out following the framework of the analysis of Chapter 2. For the most part, discussion in this chapter is limited to the choice of elements and to the logic of the algorithm.

The description is intended to be as general as possible, to be applied to any OPF task. However reference is made occasionally to numerical tests. They were carried out on an OPF package developed for the economic dispatch task; details of its implementation will be presented later. In a final section, the dictinction is made between modules in the algorithm which are common to all OPF tasks and modules specialized for particular tasks.

3.2 A General Overview of the New OPF Algorithm

3.2.1 Introductory Remarks

The highlights of the OPF algorithm developed in this thesis are presented in this section. Its general structure is illustrated in figure 3.1. It is made up of two main elements: a nonlinear program, in the conventional sense, and an outer load-tracking loop. This loop, new to OPF, supplies a sequence of loads to the nonlinear program, which responds with a sequence of optimal dispatches. With loads in the sequence chosen close together, the OPF convergence is facilitated and accelerated for each individual load. This idea, which has been used successfully in dispatch-level problems ([Fahmideh-Vojdani & Galiana 1983, Carpentier 1983] for real power dispatch, [Blanchon 1983 & 1984] for reactive power dispatch) will be presented first.

3.2.2 The Outer Load-Tracking Loop

The outer load-tracking loop exploits the continuation principle, by varying the load from some initial load to the desired value. At the initial load, which can be arbitrary, the solution to the OPF problem is known. When small discrete steps in load are input to the nonlinear program, the algorithm generates the sequence of corresponding OPF solutions. This is done with relatively little computational effort, because the solution to one problem serves as the initial guess for the next problem. The sequence of input loads leads to the desired load, at which point the nonlinear program generates the desired optimal solution.

This differs from the usual approach, in which the nonlinear program is used alone, in searching for the solution to a single load. The major difficulties in the usual approach are to find the active set, and to solve the corresponding nonlinear optimality equations from arbitrary starting points. Thanks to the proximity of adjacent solutions in the continuation process, these difficulties are avoided. The continuation approach will be advantageous if the computation of several relatively simple optimizations is easier than the one difficult optimization it replaces.

In this continuation approach, by making the step size in load small enough, the following two advantages are observed:

- Few changes, if any, are needed in the active set, from the solution of one input load to the next. Changes in the active set are then quickly implemented.

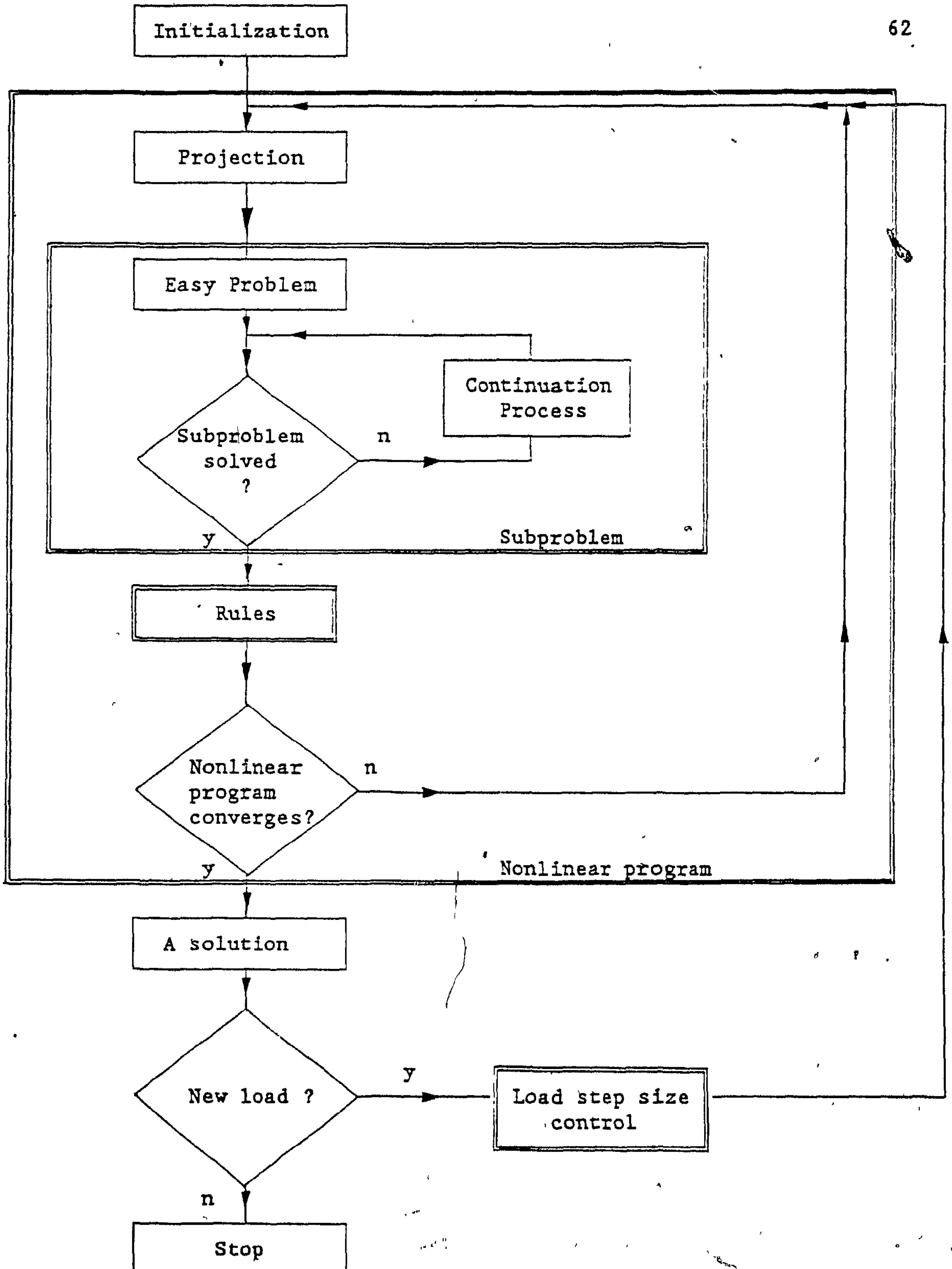


Figure 3.1. Structure of the new OPF algorithm.

- The linear model of the network used in the first subproblem of the nonlinear program is accurate enough to produce a very good estimate of the nonlinear solution. Then very few iterations are required in each nonlinear program.

If the initial load is "close" to the desired load, the entire solution process is quite fast. Alternatively, if the desired load is far from the initial load, the interval separating the two can be split up into a sequence of more closely spaced loads. A step size control can be implemented, before each change of load, to improve overall speed. It would weigh the advantages of smaller step sizes, described above, against the necessity to compute solutions for a greater number of intermediate loads.

The OPF based on the load-tracking outer loop is robust and systematic. The solution of each nonlinear program is simplified by the continuation approach. The chain of solutions from the nonlinear program constitutes an optimal solution trajectory (albeit discrete). If this solution strategy does not lead to the desired solution, then the solution trajectory will clearly be seen to move to a feasibility boundary.

This approach is of great interest in power systems operation, because fairly good trajectories of forecasted loads are usually available. An optimal solution can be found for a first load in the forecasted trajectory, using the process outlined above, or by solving a nonlinear program directly for that load. Then the optimal solution trajectory is initiated. If such a scheme can produce solutions fast enough to handle the incoming loads, then the solution tracking process can continue indefinitely on-line. Dispatching-level algorithms being proposed for real-time use by researchers at EDF [Carpentier 1987] and ENEL [Innorta et.al. 1987] are based on this idea.

The idea just described is typical of incremental loading. In power systems, implementations of incremental loading have been limited to cases with the simplest of network models. In other fields such as structural mechanics, [Watson et.al. 1983, Rheinboldt & Burkardt 1983a], incremental loading of much more sophisticated problems has been solved by continuation methods. The implementation proposed here can be seen as an incremental loading process based on a complete network model.

3.2.3 The Nonlinear Program

The sequential quadratic programming strategy has been chosen for the nonlinear program in this thesis. This method is well-suited to most OPF tasks, with their quadratic objective functions; also, see Appendix 2.2 for more general advantages of the method. The standard modules have been replaced however, by some new proposals. These stand well on their own, to form a nonlinear programming solver, but also take advantage of the presence of the outer load-tracking loop for convergence control.

The structure of the nonlinear program proposed here is basically similar to structure no.1 of Chapter 2. The most important innovations are situated in two areas, the subproblem and the Rules. These will be described briefly in this section.

The subproblem handles all constraints, as in structure no.1, but in a manner different from previous implementations. As shown in fig. 3.1, here the subproblem is solved in two stages, using the continuation approach. In a first stage, an initial simple subproblem, related to the desired subproblem, is quickly solved using standard optimization techniques. Simple tests verify whether this solution satisfies the optimality conditions of the desired subproblem. If so, the subproblem is solved; if not the continuation process is invoked.

In the continuation process, system parameters are initially relaxed so that the solution to the simple problem is also the solution to the relaxed problem. As shown later, the required values of the relaxed parameters are quite easy to find. Then, the relaxed parameters are moved together along a one-dimensional continuous trajectory back to their original values. To each intermediate value of the parameters there corresponds an intermediate optimal solution. Only certain kinds of parameter perturbations are considered, that is the so-called "right-hand-side" perturbations. Hence, since the subproblem is a quadratic program, the optimal solution trajectory can be tracked analytically as a piecewise-linear function of the parameter perturbation. The optimal solution of the subproblem is obtained when the relaxed parameters have reintegrated their original values [Huneault et.al. 1985].

This subproblem solution method is an extension of the real power dispatch problems solved in earlier work by the McGill group [Fahmideh-Vojdani 1982, Juman 1983]. It can be used on its own as a dispatching-level algorithm, with all network variables considered. The solution trajectories are computed very efficiently, and offer insight into the workings of the system. Nonlinear information can be updated when deemed necessary.

There are many advantages to this subproblem solution strategy:

- It is very fast when the solutions to the initial simple problem and to the desired subproblem coincide; indeed, the simple problems have been designed to exploit this property.

- When it has been invoked, the continuation process has usually shown itself to be very fast anyway.

- The method is also very robust in quadratic programs, since the tracking process supplies analytic trajectories which lead either to the desired solution or to a recognizable feasibility limit.

Referring again to fig. 3.1, the subproblem solution is checked for convergence and then sent to the block "Rules". They provide at their output a load flow and bounds feasible point which satisfies all the loads. If convergence of the nonlinear program has not yet been achieved (the right-side branch following the convergence check in Figure 3.1), they also provide a point with a lower value of the objective, compared to the present expansion point. That point serves as the expansion point for the next subproblem. Hence, these rules satisfy all the criteria put forward in section 2.5.1.

The actual rules which achieve this are described briefly. The independent variables of the subproblem solution form a candidate for the next expansion point. They are fed to a Newton-Raphson solver, which computes the values of the remaining variables, thereby completing the load flow feasible point. This point is then checked for bound feasibility and, if convergence has not been achieved, for lower cost. If the point is unsatisfactory for either criterion, a new candidate is sought on the line segment between the subproblem solution and the present expansion point. The step size along the segment is determined by a set of heuristics, to be described later. If the new candidate remains unacceptable, the step size is reduced further. Eventually the candidate will either be acceptable or it will move towards the present expansion point. The latter case is an indication that the present expansion point is the optimum.

Theoretically, the present set of rules does not assure descent, although in almost every iteration we have observed it did provide descent in the objective function. The load flow solver converges well to a feasible point because it allows the slack real power generation to take on a value which is not controlled by the algorithm. In some cases that computation does not reduce the cost no matter how much the step size is reduced. An alternative procedure which does assure descent has been included in the algorithm to be used in such cases. This is based on a slackless load flow. Contrary to the standard procedure, it provides points of lower cost but takes time in finding a feasible point. Convergence for the non-standard load flow is generally more difficult than for the standard load flow, but safeguards are provided by the use of a step size control in the Newton-Raphson solver. The entire procedure will be provided in detail in section 3.4.4.

These Rules are designed for robustness; they ensure descent at each iteration. The computation can be lengthy in difficult cases, where several iterations of the Rules are applied, but it is felt that the extra effort in this step is worthwhile. In algorithms where Rules do not assure descent, convergence is doubtful in difficult cases, and even when convergence is achieved, a larger number of subproblems is required. Again, due to the close tracking of the outer load-tracking loop, this new approach seldom sees convergence difficulties within the Rules step.

3.3 Formulations of the OPF Problem and its Subproblem

Formulations for the new OPF algorithm are presented in this section. The formulations of the problem and of the subproblem are summarized, with details relegated to the appendices. We then present a note we feel is relevant, concerning some discrepancies between terminology used in this thesis and that in current use in the OPF literature.

3.3.1 Summary of the OPF Formulation

The OPF formulation adopted in this thesis is basically similar to others used in many recent implementations. An OPF task is optimized, subject to operating constraints in the form of load flow equations, and bounds on most load flow variables.

The load flow equations are a set of nonlinear equations linking the power-related variables, or injections, to the voltage-related variables and passive network controls, called states. They are written as:

$$y = \begin{bmatrix} P_g - P_l \\ Q_g - Q_l \\ J_t \end{bmatrix} = \begin{bmatrix} F_p\ (V,\delta,b_c,t,\phi) \\ F_q\ (V,\delta,b_c,t,\phi) \\ F_j\ (V,\delta,b_c,t,\phi) \end{bmatrix} = F(x) \tag{3.1}$$

where y is the vector of injections. Its components are:

- P_g the real power generations
- Q_g the reactive power generations
- P_l the real power loads
- Q_l the reactive power loads
- J_t the transmission line current magnitudes squared

and x is the vector of states. Its components are:

- V the voltage magnitudes
- δ the voltage phase angles
- b_c the reactive admittances of shunt compensation devices
- t voltage ratios of variable tap transformers
- ϕ phase shifter angles

The exact formulation of eq. 3.1 is relegated to Appendix 3.1. However it can be noted that they are quadratic in V and t, trigonometric in δ and ϕ, and linear in b_c. They cannot be attributed "nice" properties, such as convexity, but they do have some useful properties for optimization; these are also presented in Appendix 3.1.

In practice the passive network controls (b_c, t, ϕ) can only take on certain discrete values, but as in most other formulations, they will be considered as continuous variables.

The treatment of the loads P_l and Q_l depends on the task to be performed. In economic dispatch and minimum loss tasks, they are treated as parameters, which are fixed in a given problem. In minimum load shedding, for which only the subproblem will be analyzed in this thesis, the loads are the variables of interest.

The line current magnitudes squared (J_L) have been retained in this formulation, instead of the more usual real power line flows. The latter are of limited use here, because generally both real and reactive components of line flows must be considered. We shall call J_L the line current injections.

Note that polar coordinates are used to describe the complex voltage. An equivalent set of equations could be written with rectangular coordinates, but that formulation will not be considered in this thesis.

Upper and lower bounds are placed on all load flow variables except voltage phase angles. These will be written as

$$\begin{bmatrix} y^m \\ x^m \end{bmatrix} = \begin{bmatrix} P_g^m \\ Q_g^m \\ 0 \\ \bar{V}^m \\ b_c^m \\ t^m \\ \phi^m \end{bmatrix} \leq \begin{bmatrix} P_g \\ Q_g \\ J_L \\ \bar{V} \\ b_c \\ t \\ \phi \end{bmatrix} \leq \begin{bmatrix} P_g^M \\ Q_g^M \\ J^M \\ \bar{V}^M \\ b_c^M \\ t^M \\ \phi^M \end{bmatrix} = \begin{bmatrix} y^M \\ x^M \end{bmatrix} \tag{3.2}$$

where superscripts m and M designate lower and upper bounds respectively, and $\underline{x}$ designates all bounded states.

In one application to be studied later, the minimum load shedding task, bounds will also be added to the loads.

In the OPF applications studied in this thesis, the objectives are functions of real power generations or of real power loads. They are modelled as either quadratic or linear functions. From now on, in order to simplify the presentation, the linear function will be considered as a particular case of the quadratic function, with quadratic terms identically nil. Without yet specifying generations or loads, a general objective function is written as

$$C(P) = c_0 + a^T P + \frac{1}{2} P^T B P \tag{3.3}$$

where

- c_0 is a scalar of fixed costs,
- a is a vector of the linear terms,
- B is a square, positive diagonal matrix of the quadratic terms.

Specific objectives for economic dispatch, minimum loss and minimum load shedding will be discussed later.

Having defined all these quantities, the Optimal Power Flow problem can now be formulated mathematically as an optimization problem:

$$\min_{y,x} \quad C(P)$$

$$\text{s.t.} \quad y = F(x) \tag{OPF}$$

$$\begin{bmatrix} y^m \\ x^m \end{bmatrix} \leq \begin{bmatrix} y \\ x \end{bmatrix} \leq \begin{bmatrix} y^M \\ x^M \end{bmatrix}$$

The optimization is carried out over all the variables of (y,x). As stated earlier, the loads P_l and Q_l are fixed to specified values in normal dispatching tasks, but they do become variables in load shedding.

3.3.2 The Subproblem Formulation

The subproblem retains the original objective function from the OPF formulation. The load flow equations 3.1 are replaced by a linearization, the load flow Jacobian model, considering all injections and all states. Its derivation can be found in Appendix 3.2. Bounds from the OPF formulation are all retained in the subproblem.

Starting from the loadflow model described above, the subproblem is reorganized into what Carpentier refers to as the "compact model" [Carpentier 1987]. This takes advantage of: (i) the natural sparsity of the objective function, and (ii) variables known to be at a bound. The set of independent variables, or algebraic basis, is modified from the states, x, in the natural formulation, to a combination of injections and states. All but one real power injection are included in the basis. The one remaining is expressed as a dependent variable, and the equation which links it to the independent variables is called a generalized power balance equation. In this scheme, the objective function can be expressed in the optimization using its natural sparse form, as in real power dispatch. Voltage phase angles are removed from the basis, because they are unbounded. The remaining basis variables are chosen amongst those suspected of going to their bounds, to keep in their simplest form as many of the active constraints as possible. The transformation of the load flow Jacobian model implemented here is described in Appendix 3.3.

Nomenclature for the reorganized subproblem, including various partitions for coefficients and variables, have been relegated to Appendix 3.4. Two items in the notation are needed before the subproblem formulation can be presented. The vector of dependent variables is denoted d, and a vector made up of the independent variables and the remaining dependent real power injection is denoted b. Without subscripts, the notation designates excursions of the variables from the expansion point of the linearization. The subscript g pertains to the dispatchable variables, on which bounds are imposed.

The formulation of the subproblem, denoted S, can now expressed mathematically:

$$
\begin{array}{ll}
\min\limits_{b} & C(P) \\
\text{s.t.} & \\
& g_0^T b = 0 \\
& d_g^m \le d_0 + G_1 b \le d_g^M \\
& b_g^m \le b_g \le b_g^M
\end{array}
\qquad (S)
$$

The three constraints are the generalized power balance equation, constraints on dependent variables expressed versus independent variables, and bounds on dispatchable independent variables.

This formulation has the following advantages:

- The Hessian of the objective function is a diagonal, positive matrix. This assures that the search process for an optimum leads to a minimum.

- The partition of independent/dependent variables can be chosen to reduce as much as possible the number of active functional constraints. However, the assignment of priorities to real power generations, as described above, assures that the previous point is always respected.

- Many independent variables are absent from the objective function. These variables, called transparent variables, are particularly easy to handle in the solution process to be developed.

- The numerical structure of the ensuing optimality conditions is a single bordered block [George & Liu 1981], which is easy to handle.

A disadvantage is that functional constraints are no longer sparse, as is the case where the independent variables are the states. The computation of the sensitivity coefficients which make up the constraints is quite efficient however, especially with the advent of sparse-vector techniques [Tinney et.al 1985].

It is felt that the advantages of this formulation far outweigh the disadvantage. If the independent/dependent partition is well chosen, then few dependent constraints will be at their bounds. Then the predominant element in the optimization is the very sparse Hessian of the objective function.

3.3.3 A Note on Terminology

It is felt that some of the terminology in use in the OPF literature is incorrect or misleading. In this section these terms are pointed out, and a "clean" set of terms is proposed.

This author feels that the terms equality and inequality constraints, as used in most papers in the OPF literature (for example, in such important papers as [Peschon et.al. 1971, Alsac & Stott 1974, Burchett et.al 1982b]), can be confusing. In all these presentations of the OPF formulation, injections from the standard load flow have come to be known as equality constraints, and the other injections as inequality constraints. This partition actually refers to the independent/dependent variables, and not to the equality/inequality constraints. The present practice has come about because the standard load flow injections are usually fixed as independent variables (equality), while the other injections are only monitored to remain between bounds (inequality). Even though this association is understood, it should be pointed out that this usual terminology is strictly incorrect. In the more general case the usual terminology is misleading, because other implementations do not necessarily partition variables the same way. The true equality constraints are all the load flow equations, $y=F(x)$ of eq. 3.1, and the inequality constraints are the upper and lower bounds on the variables.

The terms control and state have taken on two meanings in OPF. From a power systems point of view, the states have been defined in the formulation in section 3.3.1, while controllable variables are those on which actual controls have been placed. The second meaning comes from numerical optimization, where control and state correspond to independent and dependent variables, respectively. The "state - control formulation" of the optimality conditions considers derivatives of the Lagrangian with respect to all variables, but then uses equality relations to eliminate dependent variables. In this thesis the power system definitions prevail, although the term state-control formulation will also be retained. The terms independent and dependent variables will be used in the study of numerical optimization.

The term "slack" real power injection is used the same way in OPF as in load flow computations. In the latter, a single real power injection, the slack, is neglected in nonlinear solvers. It is computed after the solver has arrived at a solution, as a dependent variable. That is done to avoid difficult convergence or even infeasibility in the solver.

This author feels that the role of a dependent real power injection is somewhat different in OPF. In our algorithm, a real power injection is made dependent, but because it is cost-related, its value is directly controlled in the computation. Schemes to provide this control are presented in various parts of this chapter. Because of the differences, it is felt that a new term is desirable for this variable in OPF. In this thesis, the term "manifold" variable will be used, because the expression of the load flow manifold is provided (implicitly) by the equation linking the dependent real power injection to the independent variables.

Based on these rectifications and on discussions in Appendix 3.4 and in the previous section, Table 3.1 forms a summary of the recommended terminology. It proposes partitions of the set of variables and of the set of constraints based on some physical or mathematical aspects. The three columns of the table contain the following information:

1) The proposed terminology, regrouped within some particular classification. For example, all the variables can be classified into two basic groups, injections or states. Independently of this classification, the variables can also be placed into the other classifications.

TABLE 3.1 - PROPOSED OPF TERMINOLOGY

Terminology	Context	Status
Terminology for Variables		
injection - state	basic load flow definitions of the variables, given in this chapter after eq. 3.1.	fixed classification of the set of all variables.
cost variable or transparent var.	new partition for independent variables in optimization, depending on whether they are cost-related. This is defined in Appendix 3.4.	fixed classification of the set of independent variables, within a subproblem.
independent var. or manifold var. or dependent var.	part of the mathematical manipulations in optimization, as defined in this chapter and in Appendices 3.3 and 3.4.	in this study, this classification of all variables can be modified before each subproblem.
unbounded var. or bounded var.	pertains to the absence or the presence of constraints on a given variable.	in our formulation, only voltage phase angles are unbounded.
controllable variable	from a power systems point of view, actual controls act upon these quantities.	fixed set of variables, not related to OPF.
Terminology for Constraints		
equality constr. or inequality constr.	basic definitions from optimization.	fixed classification of the set of constraints.
active constr. or inactive constr.	the status of inequality constraints, determined during computation, in the optimization.	classification can be modified in the subproblem or in the Rules.

2) The context from which the chosen definitions are taken. For example, the definitions chosen for injections and states are those of section 3.3.1, following equation 3.1.

3) The status pertains to whether the classification of the variables or of the constraints is fixed at the outset or if it can be varied as the computation progresses.

The main advantages of this terminology are that it covers all the important notions, and that all the terms are uniquely defined.

3.4 A Detailed Description of the New OPF Algorithm

In this section, the new OPF algorithm is presented in detail, using the analysis of Chapter 2 as a framework. In particular, we describe our implementations of the elements which constitute the structure of fig. 3.1 This will not include the initialization step, which has nothing new. The subproblem solution will be introduced here, but due to its complexity, its mathematics will be presented on its own in the next chapter.

3.4.1 Projection

Motivations for the projection step (i.e., the choice of the set of independent variables) were described briefly, while presenting the subproblem in the previous section. In our algorithm, if the number of functional constraints in a subproblem surpasses a small integer tolerance, the set of independent variables is updated in order to reduce the number of functional constraints in the next subproblem.

The partition of independent/dependent variables for the $k+1^{th}$ subproblem is prepared following the solution of the k^{th} subproblem. To simplify the presentation in fig. 3.1, the projection step was placed before the subproblem. However, in our algorithm, portions of this step are actually situated on either side of the Rules block. The sequence of instructions is basically as follows.

First, a list of active dependent variables from the subproblem solution is drawn up for swapping. In this list, states are placed before injections. This is a heuristic meant to keep as many useful states as possible in the basis, to reduce the dimension of the set of nonlinear equations in the Rules. Other schemes could be tried, but it is doubtful that any smart choice could be found without some kind of lengthy combinatorial analysis. That would defeat the purpose of the change in partition, which is meant to speed up computations.

A second list is that of the inactive transparent variables. Some of these variables can be suspected of being inactive in the next subproblem, when their coefficients in the generalized power balance equation are very close to zero. This result will be explained shortly. Hence, these variables are given priority in this list.

The independent/dependent status of variables in each list are swapped one for one. The process is stopped when one of the lists is depleted. Once these operations have been performed, the algorithm proceeds with the Rules.

When the slackless loadflow must be used in the Rules, this process has an extra step. Before the load flow is solved, the preassigned dependent real power generation (the manifold variable) is made independent by swapping its status with a transparent variable. Then when leaving the Rules, its status is changed back to the dependent manifold variable, and a dependent variable is made independent. If the first list described above (active dependent variables) was not empty, the first entry has its status changed to independent. If the list was empty, any dependent variable (preferably an active state) can be used. The algorithm then proceeds with the next subproblem.

A possible disadvantage of this adaptive partitioning scheme is that it can unknowingly propose a basis where the block of the load flow Jacobian formed by the rows of the independent injections and the columns of the dependent states is very ill-conditioned or even singular. Then computations bog down. One change of basis known to exhibit this problem is that where all the real power injections replace all the voltage phase angles, so it is avoided. There exist other combinations which leave the Jacobian block singular, such as those with a zero row or column. Combinations which leave the Jacobian block ill-conditioned are very difficult to detect a priori.

The problem of serious ill-conditioning has been observed in the testing of the algorithm. The remedy implemented in the program is to perform one more independent/ dependent swap. This could look to break up possible combinations in the basis, where the variables from one bus are either all present or all absent. Another remedy would be to recall the previous partition.

3.4.2 The Subproblem Solution

In the subproblem, the quadratic program is solved using the continuation method. The solution technique has been described in a general manner in section 3.2; here, a more rigourous discussion on the method will be presented. Details on the subproblem solutions for specific tasks are developed in Chapter 4.

The continuation method is used to create a continuous family of quadratic programs. One member, an easily-solved quadratic program, is linked to the original, desired program S, defined in section 3.3.2. We define a homotopy strategy as being a choice of a simple problem coupled with a rule for linking it with the desired problem. This rule allows some physical (or system) parameters to vary over a one-dimensional trajectory. The rule is called a perturbation function, and the mathematical parameter which defines the position along the trajectory is the continuation parameter, denoted θ.

To each value of θ there corresponds a quadratic program and its optimal solution. The family of quadratic programs leading to S will be called the perturbed model of S, and denoted (S,θ). In its most general form, all the parameters vary. The perturbed model is written as follows.

$$\begin{array}{llll} \min_{b} & C(P,\theta) & & \\ \text{s.t.} & & g_0(\theta)^T b = 0 & \\ & d_g(\theta)^m \le & d_0(\theta) + G_1(\theta)\, b \le d_g(\theta)^M & \\ & b_g(\theta)^m \le & b_g \le b_g(\theta)^M & \end{array} \qquad (S,\theta)$$

The optimality conditions for problem S form a set of linear equations. The corresponding perturbed optimality conditions for (S,θ) are still linear in the variables and the Lagrange multipliers, but its coefficients are functions of θ. These optimality conditions will be developed in detail in the next chapter. For now, the optimality conditions are simply stated in a general form for the OPF subproblem:

$$\begin{bmatrix} B'(\theta) & -G(\theta)^T \\ G(\theta) & 0 \end{bmatrix} \begin{bmatrix} b_g \\ \Lambda \end{bmatrix} = \begin{bmatrix} -a'(\theta) \\ k(\theta) \end{bmatrix} \qquad (3.4)$$

where

$B'(\theta)$ and $a'(\theta)$	are arrays of varying cost-related parameters,
$G(\theta)$ and $k(\theta)$	are arrays related to varying network parameters and varying bounds,
b_g and Λ	are the unknown independent variables and Lagrange multipliers.

Solutions of eq. 3.4 render all variables functions of θ. The optimal solution as a function of θ is called the optimal solution homotopy, or optimal solution trajectory. Problem (S,θ) is formulated such that $(S,\theta=0)$ is the simple problem and $(S,\theta=1)$ is the desired problem. By following the solution trajectory to $\theta=1$, the optimal solution of problem S is obtained. When following this trajectory from $\theta=0$ to $\theta=1$, we say that we are tracking the optimal solution trajectory.

Details of the various homotopy strategies and of the computation algorithm for tracking the solution trajectories are now presented.

3.4.2.1 Homotopy strategies

Two different homotopy strategies have already been investigated. They have been tested successfully in real power dispatch problems [Fahmideh-Vojdani & Galiana 1983, Galiana et.al. 1983], and are now expanded for use in the larger load flow model. A third homotopy strategy will be introduced for load shedding applications. They affect only parameters of the right-hand-side vector in eq. 3.4. That is important, since singularity of the left-hand-side matrix as a function of θ need not be envisaged. Also, continuous variations of the right-hand-side vector result in continuous optimal solution trajectories for the load flow variables.

a) The varying load strategy

The varying load strategy starts with a problem where the load is set to a minimum. That value is determined by setting the real power generations to their lower bounds, and by sending transparent variables to the bounds which minimize their effects. To that minimum (optimal) dispatch there corresponds the minimum load which can be computed from the generalized power balance equation. Simple rules are available to resolve an initial degeneracy, and decide which real power injection comes off its bound first, to satisfy the next increment of load. From its minimum value, the load is increased along some trajectory and the corresponding optimal dispatch is tracked. The load could be sent to some desired value, but better still it could follow a load prediction curve. For the latter, the perturbation would be unrestricted in range. The perturbation function is written as

$$b_1(\theta) = b_{10} + \Delta b_1 \theta \qquad (3.5)$$

The perturbation trajectory $b_1(\theta)$ need not be limited to a straight line segment. A set of b_{10} and Δb_1 vectors as in eq. 3.5 could be furnished, each member being used over an interval of θ, to create a piecewise-linear perturbation function. It is stressed that this strategy efficiently generates trajectories of optimal dispatches, given trajectories of input load.

In this strategy variables for which there are loads are best handled as independent variables or the manifold variable. From previous arguments, the real power generations are sure to be included in this group; the reactive power loads should also be included. Otherwise the dependent demands of the perturbation function will be expressed as functions of the independent variables. The choice of the remaining independent variables, for the remaining degrees of freedom, is unrestricted.

b) The varying limits strategy

The varying limits strategy starts with a simple problem which ignores constraints on dependent variables. This is similar to the subproblem of structure no.2 in Chapter 2. The problem can easily be solved, for a given load, by sending transparent variables to the appropriate bounds, and conducting an optimization over real power generation using standard techniques. The generalized power balance equation stands as the only functional constraint in the problem. Dependent variables are then computed as a function of the newly-determined independent variables. Bound violations for dependent variables are checked. If no violation occurs, the solution of the simple problem is also the desired solution. Hence a "good" choice of the independent/dependent partition could greatly simplify the solution procedure, by avoiding suspected violations of dependent variables. If such violations are discovered, their bounds are relaxed by the amount of the largest violation. As a result, all violations are removed, except for one previously violated constraint which is "just" active. The relaxed problem has the same solution as the simple problem, with an added Lagrange multiplier, identically zero, for the newly activated constraint. The perturbation function moves the values of the violated limits from their relaxed values to their original values, and the corresponding optimal dispatch is tracked. The perturbation function for violated constraint i is written as follows:

$$d_{gi}^{lim}(\theta) = (d_{gi0}^{lim} + \Delta d) - \Delta d_i \theta \qquad (3.6)$$

where

$d_{gi}^{lim}(\theta)$	is the perturbation function, applied to limit d_{gi}^{lim},
d_{gi0}^{lim}	is the original limit on d_{gi},
Δd	is the amount of the largest violation, among all violated dependent constraints,
i	spans all violated constraints.

The original problem is solved when $\theta=1$. This strategy is deemed useful for dispatching a single, given load. In tests on real power dispatch problems, it reached a solution quite efficiently, because the simple problem is often close to the given problem. Its efficiency has also been observed in this research, in solving the subproblem.

The physical significance of the perturbation function is that intermediate solutions on the solution homotopy, with $0 < \theta < 1$, are the optimal dispatches for problems with the intermediate bounds. That approach can be useful in operations planning, when deciding on increasing transmission capacity.

c) The varying demand strategy (for load shedding)

A third homotopy strategy is introduced for load shedding, somewhat similar to the varying load strategy. When a dispatch algorithm based on a normal operating task reaches a loadability limit, load shedding is invoked. The latter task minimizes a norm of the mismatch between the customer demand and a load offered by the utility which maintains system feasibility. Its constraints are the same as those used for dispatching, plus constraints on the loads.

At the loadability limit both the load shedding and the dispatch are feasible, so that the initial optimum for load shedding is furnished by the final optimum for the dispatch. At that point, the discrepancy between customer demand and the supplied load is nil. From there, as the customer demand is modified, the load shedding objective value increases, indicating that load shedding must take place. Load shedding is enforced as long as the value of its objective function is positive. The perturbation function of customer demand considered here contains both real and reactive power loads. It is written as

$$b_D(\theta) = b_{D0} + \Delta b_D \theta \tag{3.7}$$

where

- b_D is the demand. (This is not to be confused with dependent variables.)
- $b_D(\theta)$ is the perturbation function applied to b_D.
- b_{D0} is the last demand for which a feasible dispatch exists, or for which a load shedding optimum is known.
- Δb_D is the predicted demand variation.

There are two important similarities between this and the varying load strategy for dispatching. The demand can be made to follow a piecewise linear trajectory of forecasted demand. Also, the variables for which there are demands are best handled as independent variables.

3.4.2.2 General considerations for the tracking step

With the strategies described above, the solution homotopies for the load flow variables are continuous piecewise-linear functions of θ. Specific solution homotopies will be worked out in the analysis of Chapter 4, but a general form for these solutions is written as

$$b_g(\theta) = b_{g0} + \Delta b_g \theta \tag{3.8.a}$$

$$\Lambda(\theta) = \Lambda_0 + \Delta\Lambda . \theta \tag{3.8.b}$$

while monitoring values of the inactive dependent variables

$$d_g(\theta) = d_{g0} + \Delta d_g \theta \tag{3.8.c}$$

For $\theta=0$, the problem has a known optimal solution. Then as θ is increased, the optimal solution is continuously monitored so as to verify the Kuhn-Tucker optimality conditions. Occasionally, at discrete values of θ called breakpoints, these conditions are on the verge of being violated, i.e. one of the two following situations occurs:

- A Lagrange multiplier reduces to zero. As soon as that happens, the corresponding constraint must be released from the active set.

- The optimal solution trajectory meets a new active constraint. It must be added to the active set and its Lagrange multiplier is activated.

In both cases the active set is updated, and the new active set is maintained over an interval of θ until optimality conditions are once again on the verge of being violated. To each interval of θ there corresponds a segment of the optimal solution trajectory, obtained analytically. These expressions need only be modified at each breakpoint. The solution segments placed end to end form a global continuous optimal solution trajectory.

An important observation concerns the search for the active set. The determination of the active set is performed in the continuation process by linking the known active set of the simple problem to that of the desired problem. Hence no combinatorial search procedure need be implemented.

3.4.2.3 Tracking the optimal solution trajectories

All that is left to determine is a procedure to update the active set. The details of these updates will be presented in a later chapter. In this section, we take a quick look at the updates, but more importantly, we present a general algorithm for updating and keeping the continuation process moving.

The effect of an update is to repartition the variables and the corresponding sets of coefficients in eq. 3.4. This causes a particular addition, removal, or displacement of rows and columns in the optimality conditions. There are six different updates. Some of the updates are referred to in the upcoming algorithm, so for easier reference they are numbered, as follows:

1. An inactive dependent variable hits a bound.
2. An active dependent variable is freed when its Lagrange multiplier reduces to zero.
3. An inactive transparent variable goes to a bound.
4. An active transparent variable is freed.
5. An inactive real power goes to a bound.
6. An active real power is freed.

Here then is the general algorithm for tracking the optimality conditions, i.e. computing the optimality conditions over a range of the continuation parameter.

STEP 1. Set breakpoint counter, i=0.

STEP 2. Solve the initial, simple problem.

STEP 3.

a) For the varying limits strategy, apply the perturbation to the violated limits. Designate the most violated constraint as the next to enter the active set. Invoke update 1.

b) For the varying load strategy or load shedding, resolve the initial degeneracy. Designate the freed real power as the next to leave the active set. Invoke update 6.

STEP 4. Check for degeneracy. If none, or if resolved, go to STEP 5. If unresolved, notify the user or some control outside the subproblem. STOP.

STEP 5. Implement the appropriate update.

STEP 6. Compute new optimal solution trajectory coefficients, eq. 3.8, effective in interval $[\theta_i, \theta_{i+1}]$.

STEP 7. Set $i=i+1$. Compute the value of the next breakpoint, θ_i, as follows:
Compute values of θ for which all variables hit a bound and for which all Lagrange multipliers reduce to zero. Pick the smallest of these values as next breakpoint. Recall which condition causes the new breakpoint and invoke its particular update scheme.

STEP 8. If θ_i is greater than one, go to STEP 10. If not, compute values of the variables and Lagrange multipliers at θ_i. Store them - they are the endpoints of the linear segments of the solution trajectory.

STEP 9. Go to STEP 4.

STEP 10. Compute values of the solution trajectories at $\theta=1$. The solution is found. STOP.

The algorithm terminates normally in STEP 10 or abnormally in STEP 4. The intricate STEPS 2, 5 and 6 form the heart of the process; they will be explained in detail in the next chapters. Causes and remedies for degeneracy (STEP 4) are also discussed in Appendix 4.2. The remaining steps are quite straightforward.

No provisions have yet been implemented following an abnormal end to the subproblem, due to infeasibility. One suggestion is to go to the load shedding mode upon detection of infeasibility. A load shedding solution would be sought for the set of desired parameters. Using the theory expounded in Chapter 4, this idea can be implemented in a dispatching-level algorithm. However, it is not clear in a nonlinear programming algorithm when to switch from one task to the other, because the subproblem can be infeasible even though the nonlinear problem is feasible. This is an interesting subject for future research.

3.4.3 The Convergence Test

The description of the convergence test in Chapter 2 is sufficient. In our program we monitor three quantities: the reduction in the objective function from one iteration to the next, and the difference in the objective function and in the independent variables between the expansion point and the subproblem solution. An important observation in numerical testing is that a very tight tolerance is preferable for the load tracking process to be successful. It seems that if the optimal solution to the initial problem is known with great precision, the subsequent problems move quickly to their optimal solutions. The increased effort in solving the initial problem is then worthwhile; this point is verified in the numerical results described in Chapter 7.

3.4.4 The Rules

3.4.4.1 Introductory remarks

The purpose of the Rules, as stated in section 3.2, is to find a feasible load flow point from the subproblem solution. When this point is to serve as the expansion point for a new iteration, it is chosen also to be of lower cost than the previous expansion point. In the OPF problem, this guarantees global convergence to a local optimum.

The Rules will now be described in depth. First in an illustrative section we present the rules implemented in our algorithm as well as alternatives which were tried and discarded. Then a flow chart of the Rules is drawn and each element is described. Finally, two of the more important elements of the Rules - the Newton Raphson solver and the anti-zigzagging device - are described in detail on their own.

3.4.4.2 Illustration of the Rules strategies with different alternatives

A nonlinear optimization problem is illustrated in fig. 3.2. The feasible region is the intersection of the hexagon and the region to the right of the nonlinear boundary B. The point E is the present (feasible) expansion point and point S_0 is the new subproblem solution. They are seen in a space of states (x) and the set of independent injections (y_b). The cost of S_0, denoted cost(S_0), is always smaller than or equal to cost(E), because the two points lie within the same feasible region used in the subproblem optimization, for which S_0 is the optimal solution. If the two points coincide (to within a tolerance), then $E=S_0$ is the optimum of the nonlinear problem. If they do not coincide, then point S_0 is not load flow feasible, but a load flow feasible candidate for the expansion point can be generated from S_0. Graphically, the process associates to point S_0 a point to the right of the nonlinear boundary B.

The independent variables of S_0 are fixed and the others are computed using a standard Newton-Raphson solver. The resulting load flow feasible point is denoted C_0. In the process all the y_b maintain the feasible values found in S_0. The slack generation and the other dependent variables take on the required values to be load flow feasible. In fig. 3.2 the point C_0 is drawn out of bounds and its cost is higher than cost(E), so it is rejected as a candidate for the next expansion point.

The candidate emanating from S_0 being rejected, a new candidate for the expansion point is sought. A new point S_1 is chosen on the straight line segment linking E to S_0, from which a feasible point C_1 is generated. The step size which determines the choice of S_1 will be described later. For now the important point is that the cost of any S_1 along the interval [E-S_0] is lower than cost(E), because the value of the objective along segment E-S_0 is monotonically decreasing from E to S_0. Hence that line segment is particularly useful in the search for a cheaper expansion point. Also, with S_1 closer to the feasible region than S_0, the discrepancies between S_1 and C_1 are smaller, so that cost(C_1) is more likely to be acceptable. In this illustration, point C_1 is both bounds feasible and of lower cost than cost(E), so that it is accepted as the next expansion point.

We now describe three other approaches which were tested in our algorithm. One approach we studied before adopting the more standard approach described above was that of a slackless load flow. In this strategy all the real power generations at a point S_1 on the line segment [E-S_0] are made to be scheduled injections in the solver, to take advantage of the known lower cost at that point. Hence if the resulting C_1 is feasible, it is acceptable

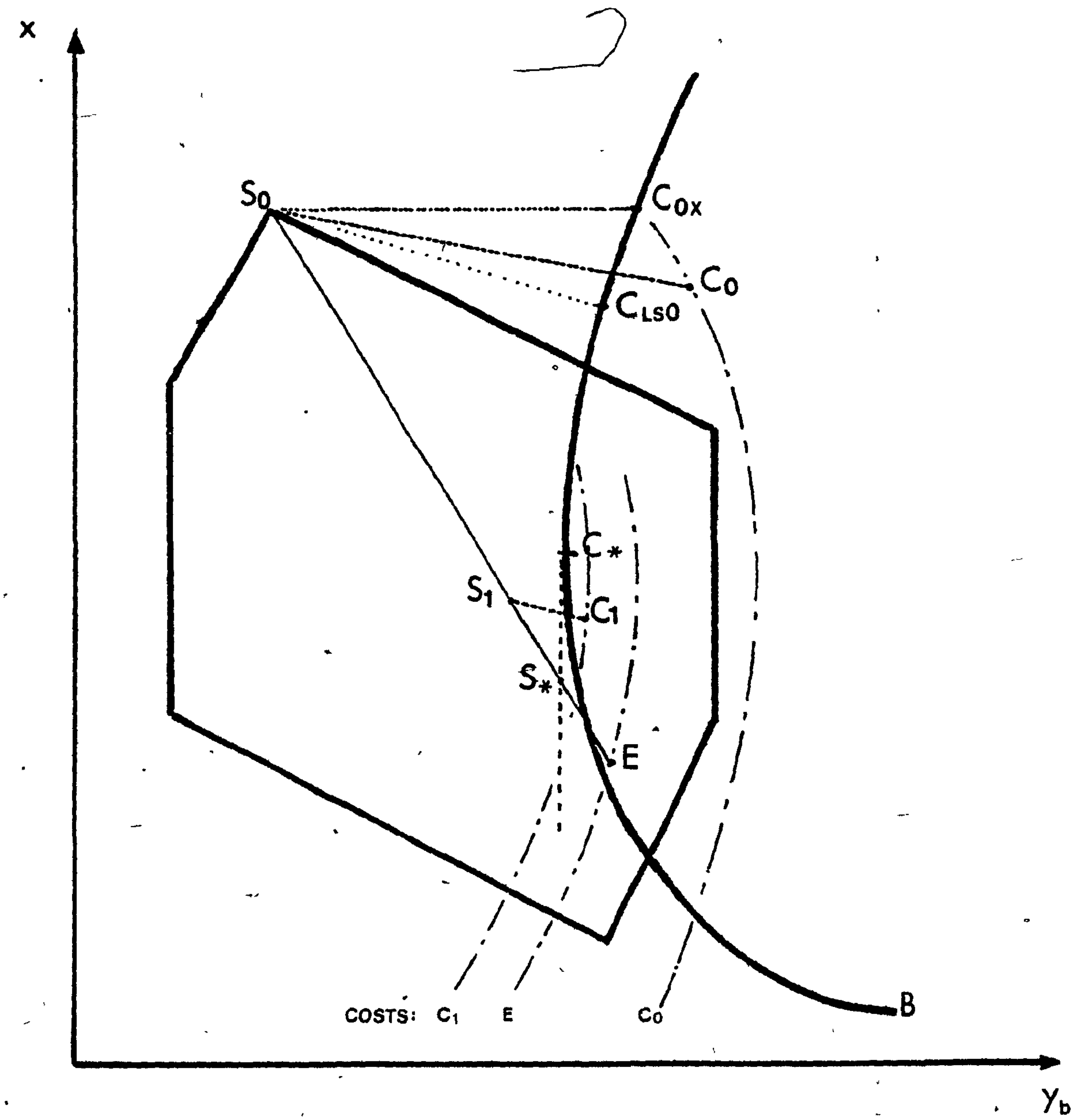


Figure 3.2. Illustration for the Rules step.

because it has the same cost as S_i. In the illustration, the value S_1 would be found to be infeasible by the Newton-Raphson solver. Our solver would detect the infeasibility, and give the least-squares solution to the problem. To obtain a feasible point whose generations lie on the E-S_0 line segment, the step size would have to be reduced until the candidate point lies between E and S_*.

In practice, there are two problems with this approach. The first is that the step size leading from E to a feasible point denoted C_* in fig. 3.2 is usually rather small. To reach C_* the number of Rules iterations can be high. The second problem is that the points generated by this approach are very close to the feasibility boundary. That usually allows very little movement in the sequence of feasible points, and little reduction in the objective. Hence this alternative is robust but very slow.

The slackless approach has been kept as an alternative, to be used only when the the main algorithm described previously cannot obtain a cheaper feasible point.

Two other approaches were tried and discarded completely. In a first case, when the scheduled injections from the slackless load flow were infeasible, the solver's least-squares solution was touted as the next expansion point. Point C_{1s0} in figure 3.2 is such a solution emanating from S_0.

In the second discarded alternative the states of the subproblem solution were fixed, and the injections were evaluated using the load flow equations. This is by far the fastest way to compute a load flow feasible point. In fig. 3.2 this corresponds to moving from S_0 to C_{0x} in the feasible region. This approach is used in most projected-Lagrangian OPF programs, with the important exact penalty terms added to aid convergence.

In both alternatives the load flow feasible points C_{0x} and C_{1si} do not satisfy the loads; in these approaches the loads can only be satisfied at the optimum. In our tests, even though step size controls were used to aid convergence, these alternatives proved unreliable, because convergence usually occurred for a load slightly different from the prescribed load. In some cases the difference between the load at the computed optimum and the desired load was of the order of a few percent.

This completes the discussion on the illustration of the Rules step.

<u>3.4.4.3 Flow chart of the Rules</u>

A flow chart of the Rules is drawn in figure 3.3. It shows a simple top part which computes a load flow feasible point, and a more complicated lower part which checks for convergence, cost reduction and feasibility. Termination of the Rules is shown on the left side of the figure, with either convergence of the nonlinear program or a better expansion point for the next iteration. The return path for a new iteration of the Rules is shown on the right, after the appropriate step size controls are invoked. The latter path is used when the convergence criteria have not been satisfied. Each element of the flow chart is described briefly below, with the key words underlined.

Upon entry a <u>flag</u> is set indicating that the standard load flow is to be solved by the Newton-Raphson solver.

The <u>initialization</u> step receives information from the previous stages: the subproblem solution, the independent/dependent partition of variables, the present expansion point and its objective value, and an initial step length.

The subproblem solution S_0 is used to compute the E-S_0 line segment of fig. 3.2. It has been observed that S_0 is not a good initial candidate. is because it is usually infeasible to the extent that the cost or feasibility tests of the C_0 it generates fail. Often the required step length in one nonlinear programming iteration, denoted α, is about the same as the one in the previous iteration. Hence the step length from the previous major iteration serves as an initial step length when entering the Rules. The first time through the Rules, however, the initial step length is set to one. This practice has been observed to give good results.

The new candidate (S_i) is computed and stored. It is a weighted sum of the present expansion point (E) and the rejected point on the E-S_0 segment (S_{i-1}):

$$S_i = E + \alpha(S_{i-1} - E) \tag{3.9}$$

Upon entering the Rules, S_{i-1} is the subproblem solution S_0.

A Newton-Raphson solver generates a feasible load flow point C_i. For the standard load flow this almost always converges, but for the slackless load flow the solver often detects infeasibility. When that is the case, the "first" step size is computed to move S_{i+1} closer to the expansion point, and a new Rules iteration is started.

If the point C_i is feasible, it is checked for cost reduction. If the point's cost is not reduced, the "second" step size computation is invoked.

The point is then checked for bounds feasibility. If it is infeasible, the "third" step size is invoked. A final step size is chosen as the minimum of the second and the third step sizes.

If the point C_i was either of increased cost or bounds infeasible, a new iteration of the Rules is required. First though the new stepsize is compared to a small tolerance. Usually the step size is larger than the tolerance, so that the standard Rules are repeated with a new S_{i+1}. If the step size is too small usually the Rules have not generated any improvement, so the initial subproblem solution is reloaded in the initialization step and the slackless load flow strategy is used. In the Rules subroutine of the program, calling the slackless load flow requires only a change in a special flag's status.

If C_i is both cost-reduced and bounds feasible, it is checked for convergence. If it satisfies the convergence requirements, it is declared the optimum solution of the nonlinear programming problem. If not, at least it is retained as the next expansion point. In either case the algorithm exits the Rules.

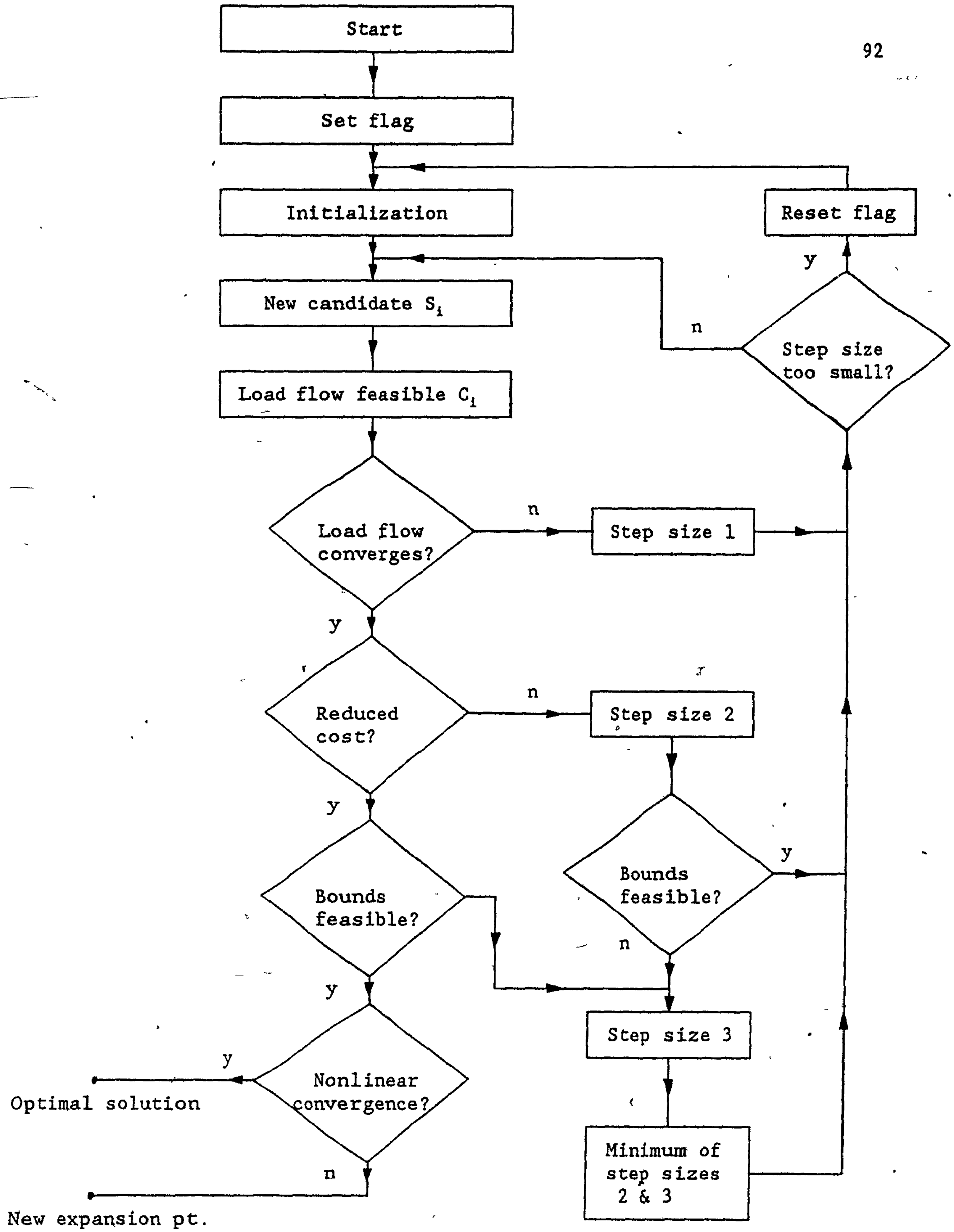


Figure 3.3. Flow chart of the Rules.

We now describe the three step size controls. The first step size correction is applied when the Newton-Raphson solver detects the infeasibility of the point S_i. It is computed as follows:

STEP 1. Compute $a_0 = 1 - \dfrac{\|S_i - C_{lsi}\|_2}{\|S_i - E\|_2}$ (3.10)

where C_{lsi} is the Newton-Raphson solver's least squares solution, and S_i and E have already been defined. All these quantities are computed by the solver.

The quotient is the relative size of the mismatch of the nonlinear solver, compared to the distance to the expansion point. It is always smaller than or equal to one, since S_i is always at least as close to C_{lsi} as to E.

STEP 2. Compute step size $\alpha =$ the submultiple of 2 (i.e, 2^{-n}, n being a natural number) nearest to a_0, but smaller than a_0. An arbitrary minimum of $\alpha = 0.2$ is imposed to avoid very small step sizes.

STEP 3. A cumulative step size α' is also computed, to serve as an initial step length for entry into the Rules, at the next iteration of the nonlinear program. It is computed as:

$$\alpha'_{new} = \alpha'_{old} \times \alpha \qquad (3.11)$$

The idea to reduce the step length by a factor of about a_0 is roughly equivalent to moving along segment E-S from the present S_i by an amount $\|S_i - C_{lsi}\|_2$. By reducing that value of the step length a little, in implementing α, the next value will be a little closer to the feasible region.

A second step size correction is applied when cost(C_i) is higher than cost(E). In this case the step size α and the cumulative step size α' are simply reduced by half.

A third step size computation is implemented when some components in the point C_i are out of bounds. Let S_b be a point on the search segment $E-S_0$ and on the boundary of an dependent constraint. S is the present candidate point on that segment. Then for each violated component j, compute

$$a_j = \frac{\|S_{bj} - E_j\|_2}{\|S_j - E_j\|_2} \qquad (3.12)$$

The smallest of these values is taken as the step length. As was the case for the other step size computations, this would go into a cumulative step size,

The idea behind this computation is that the step size is reduced proportionally to the largest excursion outside the feasible region. The numerator of eq. 3.12 is the distance from the expansion point to the bound and the denominator is the distance from the expansion point to the value of the variable.

Upon entering the Rules, the step length can also be increased, if it has remained stable for a few major iterations. The larger steps would allow for faster decreases in the objective values. Presently when it is increased, it is doubled, with a maximum of one.

There remains one last operation in the Rules, although it is separated from the Rules' main body. It is called after finishing the projection step and after having computed the generalized power balance equation, in the subproblem. It consists in adjusting tight, auxiliary bounds on some variables to avoid a certain kind of instability in optimization, often called <u>zigzagging</u>.

This completes the discussion on the elements of the flow chart.

We proceed now to describe in detail the two particular elements signaled out in the above, the Newton-Raphson solver and the anti-zigzagging step.

3.4.4.4 The Newton-Raphson solver

A flow chart of this solver is presented in figure 3.4. A step size control has been inserted into an otherwise standard Newton-Raphson algorithm. This idea has become common in the numerical mathematics literature [Dennis & Schnabel 1983], where the solution processes for nonlinear equations and unconstrained optimization have been unified. However, this idea seems rarely used in the power literature. Gross [Gross & Luini 1975] has suggested various step size controls for the Newton-Raphson load flow solver. Aoki and Nishikori [Aoki & Nishikori 1984] use optimization techniques, including step size control to avoid bound violations, to solve the constrained load flow problem. Stott and Alsac hint in a commercial prospectus [PCA 1985] that they have implemented such a scheme in their load flow package, but details are lacking.

Basically the Newton-Raphson solver computes a descent direction, to reduce the norm of a mismatch between some specified values of variables and their present values. In the usual procedure, the correction vector is obtained at each iteration by moving along the descent direction with a step size of one. The unit step size is usually accepted without hesitation, even though it might lead to values with greater mismatches. By applying an appropriate step size, smaller than one, a reduction in mismatch will be obtained.

When scheduled injections are feasible and convergence is good, the step size control is never used. Then the mismatch converges to zero. The step size control is usually put to use when the scheduled injections are infeasible. Then this solver converges to a positive mismatch, in effect the least-squares solution of the load flow equations.

The step size control implemented in this work is a heuristic, but it is sure to reach a favorable step size. It is also quite fast. First, at each iteration, zero mismatch convergence of the load flow equations is verified. If the norm of the mismatch reduces to below a small tolerance, then a solution has been found. If not, the present mismatch norm is compared to the previous one. There are two possible responses:

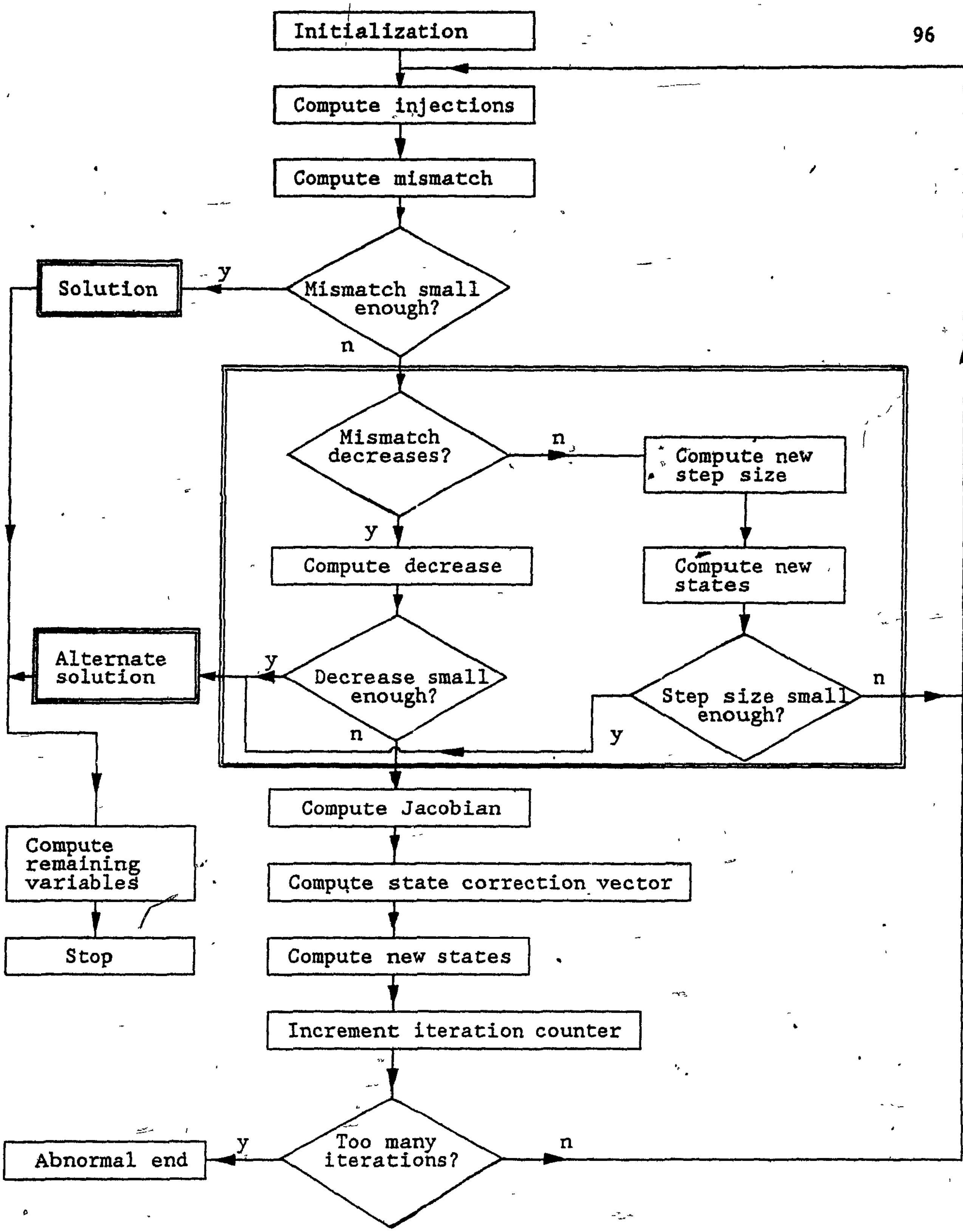


Figure 3.4. Flow chart of the Newton-Raphson solver.

- If the present mismatch is smaller than the previous mismatch, their difference is compared for convergence to some non-zero value. If convergence is achieved, the positive mismatch solution has been found; if not, the algorithm proceeds to the state-correction stage, and no reduction in step size is needed for this iteration.

- If the previous mismatch is smaller than the present mismatch, a step size α is computed. If it is smaller than a certain tolerance, then again the positive mismatch solution has been found; if not, the step size is implemented, a new state vector $x_k = x_k + \alpha \Delta x_k$ is computed and the whole Newton-Raphson iteration is restarted.

In the second case, the process might loop in the upper half of the algorithm more than once before an acceptable step size is found.

The idea behind the step size computation in the second option is simple. Assuming that the mismatch is roughly a linear function of step size, a step size is computed to obtain about the same mismatch as in the previous iteration. It is computed as follows:

STEP 1. Compute $a_0 = \dfrac{\text{norm of previous mismatch}}{\text{norm of present mismatch}}$

STEP 2. Compute the step size α = the submultiple of 2 closest to a_0 but less than a_0.

By reducing the step size a little, in implementing α instead of a_0, the new mismatch about to be computed is more likely to be smaller than the previous mismatch.

The step size can become very small when the load flow Jacobian, used in the computation of the descent direction, approaches singularity. It indicates that the closest feasible point to a solution, for the given scheduled variables, is on the feasibility boundary. When the step size falls below a tolerance, the last iterate is taken as a positive mismatch solution. In fact, this is the usual cause for positive mismatch solutions.

If a positive mismatch solution has not been found in the step size control block, the usual Newton-Raphson procedure resumes. The algorithm terminates normally at a zero-mismatch solution or at a positive mismatch solution, or abnormally due to a high iteration count.

3.4.4.5 The anti-zigzagging device

The difficulty described in this section is inherent to linearizations of nonlinear equations. It is illustrated for an OPF type problem in figure 3.5. In this example, three variables are considered, two real power generations and one transparent variable. The quadratic objective being a function of the real powers only, the cost contours are cylinders, parallel to the transparent variable axis. With this objective, the problem can easily be handled on the real power projection (the bottom face of the region). The projection of the feasible region is shown as the nonlinear shaded region. The segments of the boundary of the projected feasible region correspond, on the nonlinear manifold, either to inequality constraint boundaries or to folds [Fink & Rheinboldt 1986]. The latter occur along a fold line, where hyperplanes tangent to the manifold are parallel to the transparent's axis. The notions of fold and fold line are illustrated in three dimensions, but can be extended to higher dimensions. A subset of the transparent variables can be situated on a fold line. They are important because often in practice, if folds are present on a manifold, components of the optimal solution occur along a fold line.

Solutions can be compared for the nonlinear program and a quadratic subproblem. The expansion point E of the subproblem is chosen close to the solution. The projection of the linear manifold is redrawn in fig. 3.5 at a lower level, to avoid confusion. Even in a quadratic subproblem, the solution P_s^* can have a tendency to move to an inequality boundary, as shown in the figure, even though the true solution P^* is nowhere near that boundary. This will only happen to transparent variables, though.

This difficulty should occur in other quadratic subproblem formulations,

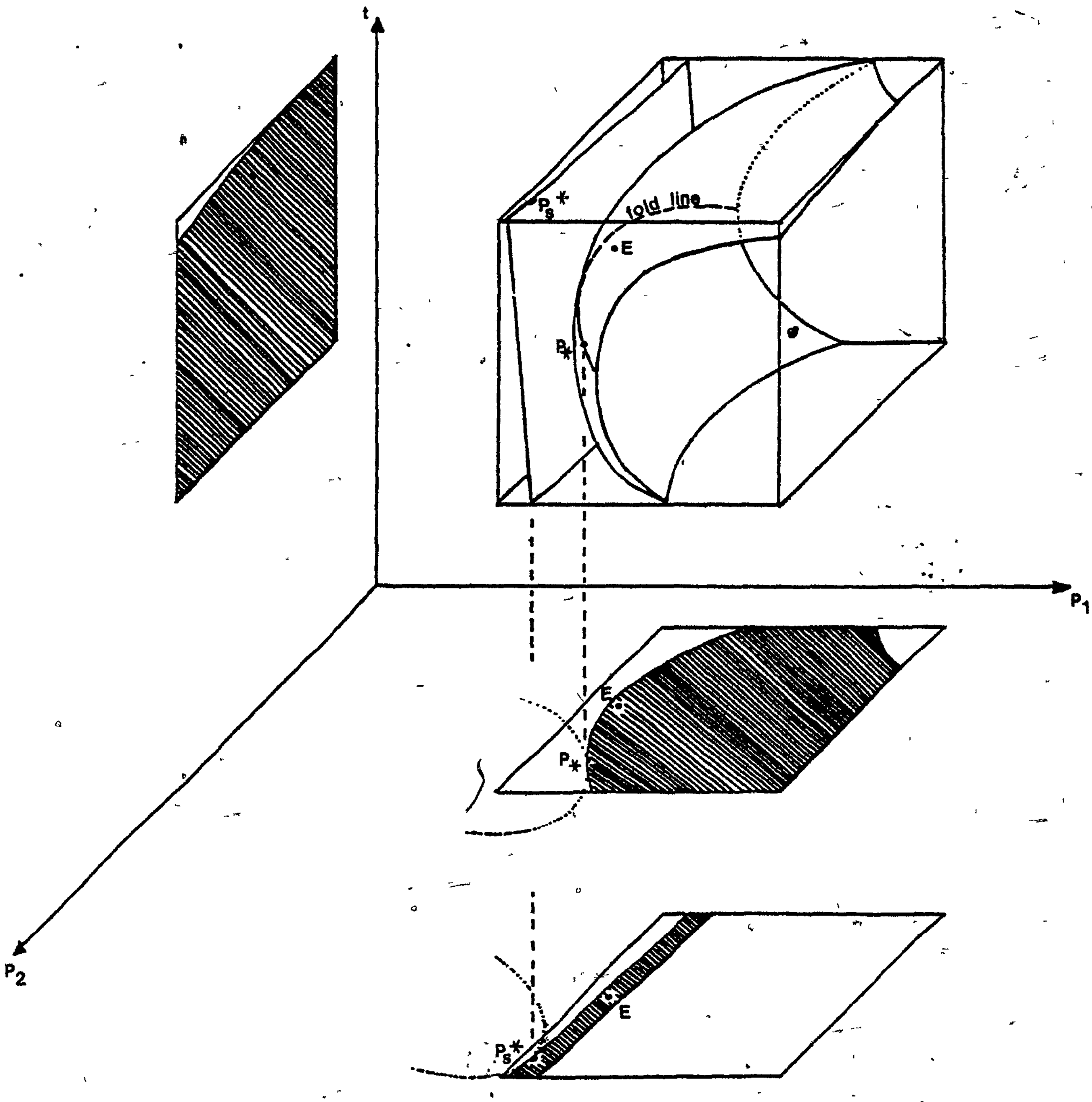


Figure 3.5. Illustration of an optimization with a fold in the nonlinear manifold.

even if the notion of transparents isn't used. Often up to now, it hasn't been recognized, and solution algorithms send variables to their bounds . For example, in an early paper on OPF [Peschon et.al. 1968] the authors state: "All nodes capable of reactive production would be at maximum voltage $V=V^M$ except for those nodes where [...] $Q \leq Q^M$ would be violated." This belief is still widespread in the OPF field. A more recent example comes from the 1987 PICA conference, during the discussion at the panel session on OPF. A complaint from an operations engineer at Florida Power and Light Co. is that their OPF package dispatches all the voltages on the system to their limits.

The behaviour of the transparent variable in a sequence of quadratic subproblems can be erratic, jumping from upper to lower bound as the expansion point moves across the fold line. It causes numerical difficulties, but there is also a theoretical problem. When the subproblem finds the right expansion point on the fold line, the linear manifold is parallel to the cost cylinders. In the subproblem all values within bounds of the transparent variable are valid solutions, although in the nonlinear problem only the expansion point is a solution.

The presence of solutions along the fold line has occurred regularly in our numerical testing of the OPF algorithm. Most often, reactive powers are the variables involved. A possible remedy to this problem, in theory, is to reform the independent/dependent partition of the variables. The effect of this change is illustrated in fig.3.5. The projection of the feasible region in the present space of independent variables (the bottom surface of the three dimensional region) is relatively "narrow". It is compared to the projection of the feasible region in another subspace (on the left face), which fills more of the inequality-feasible region and whose boundaries are mostly due to inequality constraints. Unfortunately, it is virtually impossible to know a priori which variables should be involved in the swap.

Standard remedies for this difficulty in successive linear programming solutions of OPF are the imposition of small step sizes or of a trust region around the expansion point [Ramalyer et.al. 1983]. Various rules exist to update the parameters of these restrictions, but to this author's knowledge none exploits the concept of folds. The problem has not been addressed in successive quadratic programming implementations.

A mechanism is implemented in our algorithm which takes the implicated transparents off their bounds. The presence of the fold is detected and exploited. The proximity of a fold is monitored by the values of the coefficients of the generalized power balance equation. Their values are of the order of one for real powers, and usually smaller for transparents. If a coefficient goes to zero, the corresponding variable is on a fold line. When that happens, steps are taken to peg the variable at the value of the expansion point. Tight auxiliary bounds are placed on that variable, to restrict excursions. In the computation, the coefficient is sufficiently small but not exactly zero. If it is positive the auxiliary upper bound is set to the value of the expansion point; if it is negative, the auxiliary lower bound is placed there. Subsequently the simple problem of the subproblem will maintain the variable at the expansion point. The auxiliary bounds are not too tight, letting the variable move a bit in the continuation process, if necessary. Usually in our tests, a variable going to a fold line stays on the fold line in subsequent subproblems, but if its coefficient departs from zero, the auxiliary bounds are released.

3.4.5 The Load Step Size Control

So far the load step size has been kept very simple in our algorithm. A constant percentage change applied to all loads has been tried. A change of step size would be advantageous if convergence of the nonlinear programs is too slow (decrease step size) or very fast (increase step size). No particular rule has yet been developed to implement the changes in step size. Steps in the range of 1% - 5% have been quite successful in our numerical tests.

3.5 Common and Specialized Modules for Different OPF Tasks

All but two of the modules of this OPF algorithm are common to all the OPF tasks. The two specialized modules are the computation of the objective function and the computation of optimal solution trajectories, STEP 6 of the subproblem tracking algorithm in section 3.4.2.3. In practice, both are implemented in relatively short subroutines. A library of these subroutine pairs could be written to constitute a complete OPF package. This is one of the recommendations for future research.

The objective functions have already been presented in Chapter 2; they are trivial to compute. The optimal solution trajectories have yet to be presented. In the next chapter, these trajectories are worked out for three OPF tasks: economic dispatch, minimum loss, and load shedding.

CHAPTER IV

SOLUTION OF THE OPF SUBPROBLEM USING THE CONTINUATION METHOD

4.1 Introduction

In this chapter, the solution of the OPF subproblem is presented in detail, based on the continuation method. Specifically, subproblems for three tasks are analyzed: economic dispatch, minimum loss, and minimum load shedding. They are chosen because they are basic power system tasks, but also because they demonstrate the applicability of the continuation method to a wide range of mathematical programming formulations.

Economic dispatch and minimum loss dispatch are used in the normal operating state, for dispatching of predominantly thermal or hydro systems. Within the subproblem, economic dispatch is a quadratic program while minimum loss is a linear program. Minimum load shedding is usually called in the emergency state, but we only consider the "steady-state" case where the forecasted demand cannot be met by a feasible dispatch. The solution obtained from this load shedding strategy, which is formulated as a quadratic program, supplies a load which minimizes a norm of the load - demand mismatch and the optimal normal-state dispatch for that load. Barring contingencies, the combination of a normal operating state task and this minimum load shedding makes it possible to form an optimal dispatch policy for any demand, feasible or infeasible.

Two solution techniques are proposed for the normal-state tasks, based on the two different homotopy strategies. These are the varying limits strategy and the varying load strategy, already described in Chapter 3.

For each task, the formulation and the solution of the subproblem is made up of four steps:

i) For economic dispatch and minimum loss, the formulation begun in the previous chapter (subproblem S of section 3.3.2) is completed by specifying the objective function. For minimum load shedding, the entire formulation is worked out from the beginning.

ii) The Kuhn-Tucker first order optimality conditions are derived.

iii) The perturbation function is inserted into the solutions of the optimality conditions, resulting in optimal solution trajectories.

iv) Finally, some initial, simple problems are proposed to start the continuation process.

This presentation begins with the economic dispatch task. Many details, such as the active set strategy, the formulation of the Lagrangian function, and the solution procedure for the optimality conditions are included in this first section. These techniques also went into the analysis of the other two tasks, but they are not repeated in as much detail.

4.2 Solution of the Economic Dispatch Subproblem

4.2.1 Development of the Optimality Conditions

The first order optimality conditions for the quadratic subproblem of economic dispatch will be developed in this section. The mathematical process resulting from the Kuhn-Tucker optimality conditions requires the solution of sets of linear equations and the determination of the correct partition of inactive/active constraints. The group of active variables is referred to as the active set, and the active set at the optimum is called the optimal active set. The optimal active set is easily determined for the simple problem. Then using the continuation process, the optimal active set is always known in the subsequent intermediate problems and in the final problem. Hence the determination of first order optimality conditions consists only in forming a set of linear algebraic equations. A significant advantage of this approach is that the combinatorial search for the optimal active set normally required in most techniques is unnecessary.

a) The subproblem formulation

Subproblem S of section 3.3.2 is the optimization problem to be solved, with one important addition. The objective function is specifically a quadratic function of real power generations, $C(P_g)$. The new subproblem is denoted (ED).

b) The active set strategy

The organization of the calculation of the optimality conditions is based on the active set method [Fletcher 1981]. Since Lagrange multipliers of inactive variables are known to be zero, they and the corresponding constraint functions can be removed from the Lagrangian function. For the mathematical presentation, a set of indices is created to distinguish between upper bound, lower bound, and inactive variables. For the variable i, the index is defined as follows:

$$r_i = \begin{cases} 1 & \text{if variable } i \text{ is at a lower bound,} \\ 0 & \text{if it is inactive,} \\ -1 & \text{if it is at an upper bound;} \end{cases} \tag{4.1}$$

The +1 or -1 for different types of bounds assures that Lagrange multipliers are non-negative. Then form the following diagonal matrices:

$$R_d = \text{diag}(r_i) \tag{4.2.a}$$

for i covering the set of dependent variables, and

$$R_b = \text{diag}(r_j) \tag{4.2.b}$$

for j covering the set of independent variables.

These indices premultiply the inequality constraints, leaving a Lagrangian function without constraint functions for inactive variables.

This notation is advantageous for two reasons: (i) the usual notation for the index sets of constraints (of the type "for all indices j belonging to the group with property J") is replaced by more compact notation, and (ii) it clearly situates the active independent variables in the structure of the optimality conditions.

In the active set formulation, the inactive independent variables, the active dependent variables and the non-zero Lagrange multipliers are present in the optimality conditions, but not inactive dependent variables. The latter are computed from values of the independent variables, once these are found from the solution of the optimality conditions. We say that they are "monitored" rather than "computed".

Since the inactive and active dependent variables appear alternatively in different computations, their notation will be simplified. From here on, the matrix notation for dependent constraints will not be supplemented by superscripts I (inactive) and A (active) unless it is deemed necessary. The distinction between the two groups will usually be clear from the context. The superscript A will always be dropped for active dependent variables, and whenever possible, the superscript I is dropped from the inactive independent variables and their coefficients.

c) The Lagrangian function

With the simplification in notation given above, the Lagrangian for optimization problem (ED) is

$$\mathcal{L} = C(P_g) - \lambda_0 \, [g_0^T b] - \lambda_1^T R_d \, [G_1 b + d_0 - d_g^{lim}] - \mu^T R_b \, [\hat{b}_g - b_g^{lim}] \qquad (4.3)$$

Partitions are created for variables, coefficient matrices and vectors depending on their inactive/active and P_g/transparent status. Notation and dimensions for the above variables and their partitions, and the definition of a new array H needed below are presented in Appendix 4.1.

d) The optimality conditions

The first order optimality conditions are given, starting with their most detailed form, and then with simplifications in notation. The different formulations will be used at various stages of the presentation. Referring to Appendix 4.1 for notation, the optimality conditions are:

$$\left[\begin{array}{cccc|cccc} B^A & & & & -g_{0p}^A & -H_p^{AT} & -R_p & \\ & B & & & -g_{0p} & -H_p^T & & \\ & & & & -g_{0t}^A & -H_t^{AT} & & -R_t \\ & & & & -g_{0t} & -H_t^T & & \\ \hline g_{0p}^{AT} & g_{0p}^T & g_{0t}^{AT} & g_{0t}^T & & & & \\ H_p^A & H_p & H_t^A & H_t & & & & \\ R_p & & & & & & & \\ & & R_t & & & & & \end{array}\right] \begin{bmatrix} P_g^A \\ P_g \\ t_g^A \\ t_g \\ \lambda_0 \\ \lambda_1 \\ \mu_p \\ \mu_t \end{bmatrix} = \begin{bmatrix} -a^A \\ -a \\ 0 \\ 0 \\ k_0^{lim} \\ k_1^{lim} \\ R_p P_g^{lim} \\ R_t t_g^{lim} \end{bmatrix} \qquad (4.4)$$

with

$$k_0^{lim} = g_0^T \, [\, b_1 + b_e \,] \qquad (4.5)$$

$$k_1^{lim} = H \, [\, b_1 + b_e \,] + R_d^A \, [d_g^{lim} - d_0] \qquad (4.6)$$

Together k_0^{lim} and k_1^{lim} form an (ndA + 1) dimensional vector, denoted k^{lim}, where ndA is the number of active dependent constraints.

The solution of eq. 4.4 can be simplified by setting active independent variables to their values and sending them to the right-hand-side. The equations can be split into two parts for convenience.

$$R_p P_g^A = R_p P_g^{lim} \tag{4.7.a}$$

$$R_t t_g^A = R_t t_g^{lim} \tag{4.7.b}$$

$$R_p \mu_p = (B^A P_g^A + a^A) - g_{0p}^A \lambda_0 - H_p^{AT} \lambda_1 \tag{4.7.c}$$

$$R_t \mu_t = - g_{0t}^A \lambda_0 - H_t^{AT} \lambda_1 \tag{4.7.d}$$

and

$$\begin{bmatrix} B & & -g_{0p} & -H_p^T \\ & & -g_{0t} & -H_t^T \\ g_{0p}^T & g_{0t}^T & & \\ H_p & H_t & & \end{bmatrix} \begin{bmatrix} P_g \\ t_g \\ \lambda_0 \\ \lambda_1 \end{bmatrix} = \begin{bmatrix} -a \\ 0 \\ k_0^{lim} - g_0^{AT} b_g^A \\ k_1^{lim} - H^A b_g^A \end{bmatrix} \tag{4.8}$$

The first group (eq.4.7) handles active independent variables and their Lagrange multipliers. The second (eq.4.8) handles inactive independent variables, active dependent variables and their Lagrange multipliers. The two groups are not decoupled, and the Lagrange multipliers λ must be resolved before computing μ.

Before looking for a solution to eq. 4.7 and 4.8, their notation will be further simplified. Regrouping terms in the g_0 vector and the H matrix, the k vector, and the Lagrange multipliers results in the following:

$$R_p P_g^A = R_p P_g^{lim} \tag{4.9.a}$$

$$R_t t_g^A = R_t t_g^{lim} \tag{4.9.b}$$

$$R_p \mu_p = (B^A P_g^A + a^A) - G_p^{AT} \lambda \tag{4.9.c}$$

$$R_t \mu_t = - G_t^{AT} \lambda \tag{4.9.d}$$

and

$$\begin{bmatrix} B & & -G_p^T \\ & & -G_t^T \\ G_p & G_t & \end{bmatrix} \begin{bmatrix} P_g \\ t_g \\ \lambda \end{bmatrix} = \begin{bmatrix} -a \\ 0 \\ k \end{bmatrix} \tag{4.10}$$

In its simplest expression, eq. 4.4 will be reduced to this form:

$$\begin{bmatrix} B' & -A^T \\ A & \end{bmatrix} \begin{bmatrix} b_g \\ \Lambda \end{bmatrix} = \begin{bmatrix} -a' \\ k' \end{bmatrix} \tag{4.11}$$

This notation will be useful later on. The definitions of the new parameters in eq. 4.10 and 4.11 are understood.

4.2.2 Analytical Solution of the Optimality Conditions

Analytical expressions for the optimal solutions of eq. 4.10 are obtained by applying Gaussian elimination to the blocks of its left-hand-side matrix. This is possible as long as the inverse matrices called for in the Gaussian elimination process exist. Causes for singularity of the matrices and possible remedies will be covered further in Appendix 4.2, on degeneracy.

Some structural requirements in our solution technique are that the G matrix of eq. 4.10 be full rank and that inactive generations outnumber the active dependent constraints. If the latter condition is violated, the dependent/independent partition can be reordered.

Taking eq. 4.10 as a three by three block matrix, the second and third block rows and columns are permuted leading to a suitable form for Gaussian elimination. Then the following row operations are performed, in this order:

New row no.2 = old row no.2 - $G_p B^{-1}$ x row no.1 (4.12)

New row no.3 = old row no.3 + G_t^T [$G_p B^{-1} G_p^T$]$^{-1}$ x row no.2 (4.13)

The resulting equation, mathematically equivalent to eq. 4.10, is in block upper triangular form:

$$\begin{bmatrix} B & -G_p^T & \\ & K & -G_t \\ & & L \end{bmatrix} \begin{bmatrix} P_g \\ \lambda \\ t_g \end{bmatrix} = \begin{bmatrix} -a \\ n \\ G_t^T K^{-1} n \end{bmatrix} \qquad (4.14)$$

Analytical expressions for the optimal values of the variables are easily found from eq. 4.14 to be

$$t_g = L^{-1} G_t^T K^{-1} n \qquad (4.15)$$

$$\lambda = K^{-1} [I - MK^{-1}] n \qquad (4.16)$$

$$P_g = B^{-1} \{ G_p^T K^{-1} [I - MK^{-1}] n - a \} \qquad (4.17)$$

where

$$K = G_p B^{-1} G_p^T \qquad (4.18)$$

$$L = G_t^T K^{-1} G_t \qquad (4.19)$$

$$M = G_t L^{-1} G_t^T \qquad (4.20)$$

$$n = G_p B^{-1} a + k \qquad (4.21)$$

The solution for eq. 4.9 can now be rewritten, incorporating the solution for λ:

$$R_p P_g^A = R_p P_g^{lim} \qquad (4.22)$$

$$R_t t_g^A = R_t t_g^{lim} \qquad (4.23)$$

$$\mu_p = R_p \{ (B^A P_g^A + a^A) - G_p^{AT} K^{-1} [I - MK^{-1}] n \} \qquad (4.24)$$

$$\mu_t = R_t \{ \qquad - G_t^{AT} K^{-1} [I - MK^{-1}] n \} \qquad (4.25)$$

To complete the solution, the optimal values of the inactive dependent variables are monitored (next page):

$$d_g^I = d_0 + G(b_g - (b_l + b_e)) \quad (4.26)$$

The optimal b_g vector used to compute eq. 4.26 is made up of the optimal values of P_g and t_g, computed in the preceding equations.

Equations 4.15 to 4.26 constitute the optimal solutions of the quadratic program (ED) for a known active set. The optimal value of the objective function is the value of the objective function evaluated with the optimal real power generations.

Note that in these equations, if all the transparent variables are at their bounds, the terms L and M disappear. Then transparent variables are all sent to the right-hand-side and the structure of the optimality conditions is identical to that of real power dispatch.

Efficient computational techniques are used in the numerical implementation of eq. 4.15 - 4.26 and of the subsequent optimal solution trajectories based on these equations. These techniques reduce the computational effort by taking advantage of quantities already computed, and by avoiding inefficient computing practices, such as computing inverses of matrices. The details of the implementation for the economic dispatch using the varying limits strategy are presented in Chapter 6. The analytic expressions for the segments of the optimal solution trajectories for this problem are now presented.

4.2.3 Solution of the Economic Dispatch Using the Varying Limits Strategy

a) Optimal solution trajectories

The perturbation function of eq. 3.7 is implemented in the right-hand-side vector k of eq. 4.10. The k vector becomes a function of the continuation parameter, θ:

$$k(\theta) = k_0 + \Delta k.\theta \quad (4.27)$$

where

$$k_0 = \begin{bmatrix} g_0^T [b_1 + b_e] - g_0^{AT} b_g^A \\ H [b_1 + b_e] - H^A b_g^A + R_d^A [d_{g0} - d_0 + \Delta d] \end{bmatrix} \quad (4.28)$$

and

$$\Delta k = \begin{bmatrix} 0 \\ \Delta d \end{bmatrix} \quad (4.29)$$

As a result of this, the right-hand-side term n of eq. 4.21 is split into two parts:

$$n(\theta) = n_0 + \Delta k.\theta \quad (4.30.a)$$

with

$$n_0 = G_p B^{-1} a + k_0 \quad (4.30.b)$$

With this choice of perturbation function, optimal solution trajectories can be built, by splitting the terms in eq. 4.15 - 4.26 to form relations of the type of eq. 3.9.

For inactive $b_g(\theta)$:

$$P_g(\theta) = P_{g0} + \Delta P_g \theta \quad (4.31.a)$$

$$P_{g0} = B^{-1} (G_p^T K^{-1} [I - MK^{-1}] n_0 - a) \quad (4.31.b)$$

$$\Delta P_g = B^{-1} G_p^T K^{-1} [I - MK^{-1}] \Delta k \quad (4.31.c)$$

$$t_g(\theta) = t_{g0} + \Delta t_g \theta \quad (4.32.a)$$

$$t_{g0} = L^{-1} G_t^T K^{-1} n_0 \quad (4.32.b)$$

$$\Delta t_g = L^{-1} G_t^T K^{-1} \Delta k \quad (4.32.c)$$

For active b_g:

$$R_p P_g^A = R_p P_g^{lim} \quad (4.33)$$

$$R_t t_g^A = R_t t_g^{lim} \quad (4.34)$$

For Lagrange multipliers:

$$\lambda(\theta) = \lambda^0 + \Delta\lambda.\theta \quad (4.35.a)$$

$$\lambda^0 = K^{-1} [I - MK^{-1}] n_0 \quad (4.35.b)$$

$$\Delta\lambda = K^{-1} [I - MK^{-1}] \Delta k \quad (4.35.c)$$

$$\mu_p(\theta) = \mu_{p0} + \Delta\mu_p\theta \quad (4.36.a)$$

$$\mu_{p0} = R_p ((B^A P_g^A + a^A) - G_p^{AT}K^{-1} [I - MK^{-1}] n_0) \quad (4.36.b)$$

$$\Delta\mu_p = -R_p G_p^{AT}K^{-1} [I - MK^{-1}] \Delta k \quad (4.36.c)$$

$$\mu_t(\theta) = \mu_{t0} + \Delta\mu_t\theta \quad (4.37.a)$$

$$\mu_{t0} = -R_t G_t^{AT}K^{-1} [I - MK^{-1}] n_0 \quad (4.37.b)$$

$$\Delta\mu_t = -R_t G_t^{AT}K^{-1} [I - MK^{-1}] \Delta k \quad (4.37.c)$$

while monitoring inactive dependent variables:

$$d_g^I(\theta) = d_{g0} + \Delta d_g\theta \quad (4.38.a)$$

$$d_{g0} = d_0 + G(b_{g0} - (\bar{b}_1 + b_e)) \quad (4.38.b)$$

$$\Delta d_g = G\Delta b_g \quad (4.38.c)$$

b) The initial, simple problem

In this homotopy strategy, the initial simple problem ignores the functional inequality constraints of program ED. The ensuing problem, although not trivial, is solved quickly. Since this approach has been implemented, many of the fine points can be discussed.

The solutions of the simple problem fall into two categories:

- For very low loads: For loads near minimum generation, the real power generations are set to their minima, and transparent variables are not uniquely defined. In the normal operation of a power system, this case is rare.
- For other loads: For loads greater than a certain "threshold", real power generations are dispatched using standard optimization techniques. The transparent variables with non-zero coefficients in the generalized power balance equation go to their bounds, while those with zero coefficients are dispatched using rules extraneous to the quadratic program.

The solution algorithms for these two categories are introduced in this section, but first a graphical interpretation of the problem should be helpful. This problem is in a form suitable for solution using the equal incremental cost criterion, popular in real power dispatch [Wood & Wollenberg 1984]. The method is based on the result that all inactive dispatchable variables have the same incremental cost at the optimum. This incremental cost is equal to λ_0, the Lagrange multiplier associated with the generalized power balance equation. A variable is sent to its upper bound if its incremental cost cannot be raised to λ_0, or pegged to its lower bound if its incremental cost cannot be lowered to λ_0. In the generalized procedure presented here, the graphs of λ_0 versus all the independent variables are drawn, for all values of the variables between bounds. (Refer to figure 4.1.) For real power generations, these are line segments with positive slopes and λ_0 intercepts. For transparent variables the slopes and λ_0 intercepts are identically zero, because they never cost anything. At the bounds, the incremental cost curves are sent to ± infinity, acting as barriers to avoid infeasible operation. For variables with positive coefficients in the generalized power balance equation, the lower bounds are connected to minus infinity and the upper bounds to plus infinity; the opposite applies to variables with negative coefficients.

The term $g_0^T b_1$ is called the aggregate load. An aggregate load curve is drawn over all permissible values of λ_0, by isolating the load terms of the power balance equation. (See eq. A3.4.3. from Appendix 3.4 for the relationship between the terms making up the power balance equation.) The aggregate load is obtained by adding terms $g_{0i} b_{gi}$ for the corresponding values of b_{gi} (for a given λ_0), read off the incremental cost curves, and subtracting a constant term due to the expansion point, $g_0^T b_e$. Once this graph is built, the optimal dispatch for a given aggregate load can easily be found, by reading values of the variables "horizontally" off the graph. With transparent variables added to the formulation, this procedure is called the generalized equal incremental cost procedure.

In the example of fig. 4.1, the problem considers only four variables, two real power generations and two transparents. Coefficients and bounds are given in the figure or can be read off the graph. For an aggregate load of 15, the optimal values of the independent variables are those which intersect the dotted horizontal line. These values are

$$P_{g1} = 3.5 \qquad t_{g1} = 2$$
$$P_{g2} = 5.5 \qquad t_{g2} = 3$$

The two categories of solutions noted at the beginning of this section (low loads and other loads) can be identified in fig. 4.1. Solutions in the fist category occur on that portion of the aggregate load curve where λ_0 is equal to zero. In this example, this occurs for loads between 9.1 and 11.3. Generations are then set to their minima and transparent variables are adjusted to satisfy the load. The transparent variables lie within their bounds and are non-unique. In the example, the only added restriction on the transparents is that

$$-0.2\ t_{g1} + 0.9\ t_{g2} = \text{Aggregate load} - 9 \tag{4.39}$$

Solutions falling into the second category occupy all the rest of the aggregate load curve.

This completes the graphical interpretation.

The optimality conditions for the simple problem are now analyzed, and solution procedures are introduced. The new optimality conditions are obtained by dropping the H row and the λ_1 column from eq. 4.4. The two categories of solutions are obtained in the analysis as a result of satisfying the new optimality conditions with either a zero or a nonzero value of the Lagrange multiplier λ_0. The two cases are treated separately.

i) For low loads

With $\lambda_0 = 0$, there exist solutions to the simple problem with all real power generations at their lower bounds. The mathematical argument is as follows. The optimality conditions for the simple problem with $\lambda_0 = 0$ tend to send the inactive real power generations to their unconstrained minima, given by the expression

$$P_{gi} = -a_i/B_{ii}. \tag{4.40}$$

Since coefficients a_i and B_{ii} are positive, these values of P_{gi} are all infeasible. However, optimization algorithms based on search direction methods use these values as end points of a search segment. These are the only search directions generated by the algorithms at any iteration. Hence, all P_{gi} are pushed to their lower bounds.

Meanwhile, no values are prescribed for the individual transparent variables. The only restriction on the transparents besides the individual bounds is that

$$g_{0t}{}^T t_g = k_0{}^{lim} - g_{0p}{}^T P_g{}^m \tag{4.41}$$

The low loads which can be satisfied in this case are situated between the minimum aggregate load and the threshold load, defined below. The minimum aggregate load which can be satisfied by a feasible dispatch, denoted MAL, is

$$MAL = \min [\ g_0{}^T b_l\ ] = -g_0{}^T b_e + g_{0p}{}^T P_g{}^m + [\ g_{0t}{}^T t_g\ ]^m \tag{4.42}$$

The maximum load which can be satisfied at minimum generation will be called the threshold load, and denoted TL. It is given by the expression

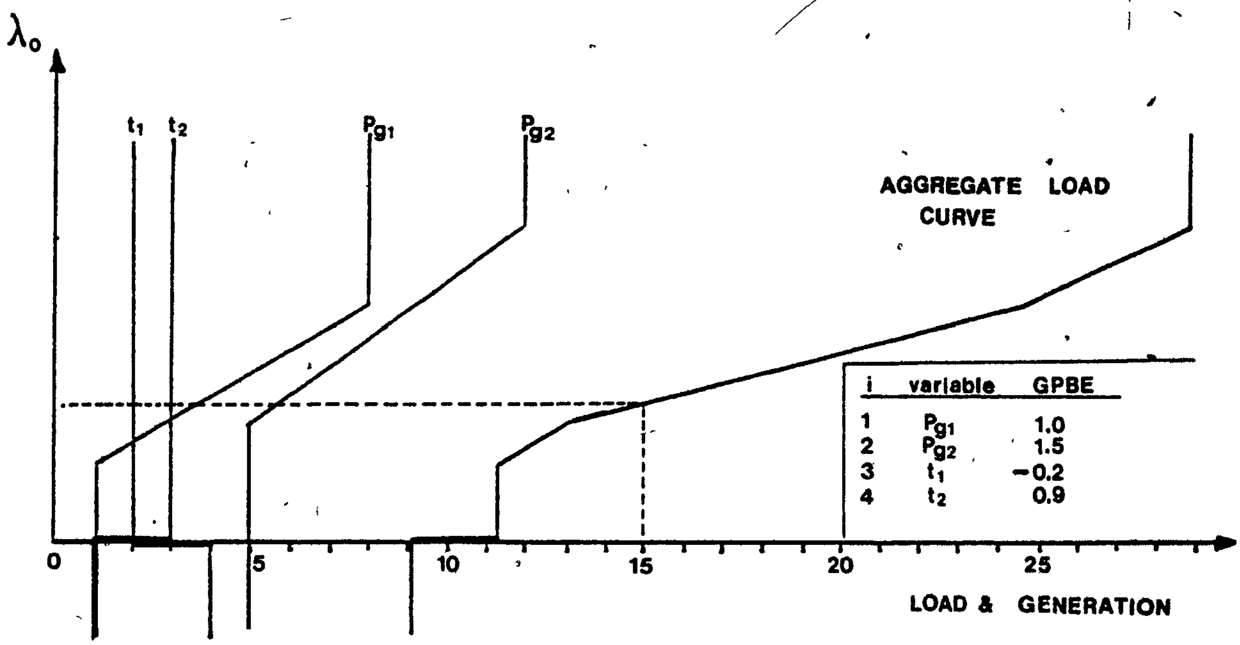


Figure 4.1. The generalized equal incremental cost criterion.

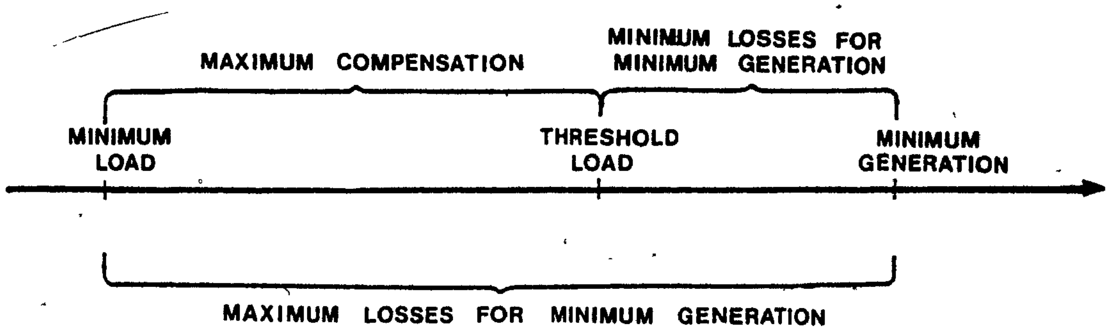


Figure 4.2. Minimum load, threshold load, and compensation.

$$TL = -g_0^T b_a + g_{0p}^T P_g^m + [\, g_{0t}^T t_g \,]^M \qquad (4.43)$$

For aggregate loads between MAL and TL the optimal dispatch is obtained by adjusting $[\, g_{0t}^T t_g \,]$ within bounds. For this, many strategies are possible. The strategy implemented in our algorithm keeps only one transparent variable inactive, since optimal solution equations 4.31 - 4.38 used in the continuation process do not allow for more. The details of the implementation are given in Chapter 6.

ii) For the other loads

For the optimality conditions to be consistent when $\lambda_0 \neq 0$, all the transparent variables must go to their bounds, except for those whose coefficients g_{0t} are zero. The bounds to which the transparents move are such that they reduce as much as possible the right-hand-side term k of the optimality equation of the type eq. 4.10. Alternately, those variables with zero coefficients have no effect on the problem. Either way, eliminating the transparent variables from the optimality equations reduces the structure of the problem to that of a standard real power dispatch. With the active transparents incorporated into the right-hand-side, and again for a given active set, the optimality conditions for this problem are

$$\begin{bmatrix} B & -g_{0p} \\ g_{0p}^T & \end{bmatrix} \begin{bmatrix} P_g \\ \lambda_0 \end{bmatrix} = \begin{bmatrix} -a \\ \hat{k} \end{bmatrix} \qquad (4.44)$$

with solutions

$$\lambda_0 = K^{-1} \, [g_{0p}^T B^{-1} a + \hat{k}] \qquad (4.45)$$

$$P_g = B^{-1} \, [g_{0p} \lambda_0 - a] \qquad (4.46)$$

and where $\hat{k} = k_0^{lim} - [g_{0t}^T t_g]^M$. (4.47)

The solution algorithm for the simple problem must determine not only the optimal values given above, but also the optimal active set. In the hybrid method developed in this thesis, a binary search technique is used in the optimization until the optimal active set is identified, after which the solutions for λ_0 and the inactive real power generations are computed exactly. The details of the implementation are found in Chapter 6.

Finally, the choice of bound for each transparent variable is determined from the Lagrange multipliers μ_t, which are non-negative:

$$\mu_t = - R_t g_{0t} \lambda_0 \tag{4.48}$$

From the non-negativity of B, a, the scalar K^{-1}, g_{0p}, and P_g, the Lagrange multiplier λ_0 of eq. 4.45 is also positive. Hence for the right-hand-side of eq.4.48 to be positive, here are the only possible choices:

i) for $g_{0ti} < 0$, $R_{tii} = 1$, i.e. t_{gi} goes to a lower bound, (4.49)

ii) for $g_{0ti} > 0$, $R_{tii} = -1$, i.e. t_{gi} goes to an upper bound. (4.50)

Recall that for $g_{0ti} = 0$, the corresponding variable is dispatched using rules extraneous to the subproblem.

This completes the analysis of the optimality conditions of the simple problem.

A final observation is that the quantity [$g_{0t}{}^T t_g$] can be thought of as a quantitative measure of compensation. It is the amount by which real power generations can be reduced when compensation devices (in a general sense) are activated. That doesn't mean that real power is generated by transparent variables, to satisfy the loads. In effect, real power losses are reduced by the process, until they can no longer be reduced without violating the bounds. Figure 4.2 illustrates the idea in the region of minimum generation. The range of compensation is the amount by which a load is increased by adjusting compensation without modifying real power generation. Note that the illustration is built around minimum load so that the the previously defined threshold load can be illustraed, but we could slide it into any load region.

4.2.4 Solution of the Economic Dispatch Using the Varying Load Strategy

Note that the presentation in this and subsequent sections parallels that of section 4.2.3, and the same notation is used. That should not cause any confusion, since each section is self-contained.

a) Optimal solution trajectories

Using the perturbation function of eq.3.5, the right-hand-side vector k of eq. 4.10 can be written as a linear function of the load. Hence the load, confined to a trajectory, becomes the continuation parameter. The vector k becomes:

$$k(b_1) = k_0 + \Delta K . b_1 \tag{4.51}$$

where

$$k_0 = \begin{bmatrix} g_0^T b_e - g_0^{AT} b_g^A \\ Hb_e - H^A b_g^A + R_d^A [d_g^{lim} - d_0] \end{bmatrix} \tag{4.52}$$

and

$$\Delta K = \begin{bmatrix} g_0^T \\ H \end{bmatrix} \tag{4.53}$$

and it is understood that $b_1 = b_1(\theta)$.

As in the previous study, the term n of eq. 4.21 is split into two parts:

$$n(\theta) = n_0 + \Delta K . b_1 \tag{4.54.a}$$

with

$$n_0 = G_p B^{-1} a + k_0 \tag{4.54.b}$$

Using the perturbation function of eq. 4.51, the optimal solution trajectories are expressed as functions of the loads. These are quite similar in form to eq. 4.31 - 4.38 of the previous section:

For inactive $b_g(\theta)$:

$$P_g(b_1) = P_{g0} + \Delta P_g b_1 \tag{4.55.a}$$

$$P_{g0} = B^{1-1} (G_p^T K^{-1} [I - MK^{-1}] n_0 - a) \tag{4.55.b}$$

$$\Delta P_g = B^{-1} G_p^T K^{-1} [I - MK^{-1}] \Delta K \tag{4.55.c}$$

$$t_g(b_1) = t_{g0} + \Delta t_g b_1 \tag{4.56.a}$$

$$t_{g0} = L^{-1} G_t^T K^{-1} n_0 \tag{4.56.b}$$

$$\Delta t_g = L^{-1} G_t^T K^{-1} \Delta K \tag{4.56.c}$$

For active b_g:

$$R_p P_g^A = R_p P_g^{lim} \tag{4.57}$$

$$R_t t_g^A = R_t t_g^{lim} \tag{4.58}$$

For Lagrange multipliers:

$$\lambda(b_1) = \lambda^0 + \Delta\lambda . b_1 \tag{4.59.a}$$

$$\lambda^0 = K^{-1} [I - MK^{-1}] n_0 \tag{4.59.b}$$

$$\Delta\lambda = K^{-1} [I - MK^{-1}] \Delta K \tag{4.59.c}$$

$$\mu_p(b_1) = \mu_{p0} + \Delta\mu_p b_1 \tag{4.60.a}$$

$$\mu_{p0} = R_p((B^A P_g^A + a^A) - G_p^{AT} K^{-1} [I - MK^{-1}] n_0) \tag{4.60.b}$$

$$\Delta\mu_p = -R_p G_p^{AT} K^{-1} [I - MK^{-1}] \Delta K \tag{4.60.c}$$

$$\mu_t(b_1) = \mu_{t0} + \Delta\mu_t b_1 \tag{4.61.a}$$

$$\mu_{t0} = -R_t G_t^{AT} K^{-1} [I - MK^{-1}] n_0 \tag{4.61.b}$$

$$\Delta\mu_t = -R_t G_t^{AT} K^{-1} [I - MK^{-1}] \Delta K \tag{4.61.c}$$

while monitoring inactive dependent variables:

$$d_g^I(b_1) = d_{g0} + \Delta d_g b_1 \qquad (4.62.a)$$

$$d_{g0} = d_0 + G\,[b_{g0} - b_e] \qquad (4.62.b)$$

$$\Delta d_g = G\,[\Delta b_g - I] \qquad (4.62.c)$$

b) The initial, simple problem

The solution homotopy must start at a problem with a known optimal solution. That could be obtained by applying the varying limits strategy to a given load, b_{10} say. From a practical point of view, a QP programing code could easily accommodate both homotopy strategies. One subroutine could be used to find an optimal solution for an initial load in a load trajectory, using the solution trajectories of eq. 4.31 - 4.38 of the varying limits strategy. This first subroutine would then be replaced by one which tracks the optimal dispatch for varying loads, using eq. 4.55 - 4.62 of the varying load strategy. The remaining steps of the QP economic dispatch are basically the same for both strategies. This is the preferred approach, since the initial load can take on any value, and its optimal dispatch using the varying limits strategy is quite fast.

Another approach which would stick only to the varying load strategy is to find an initial load for which the solution is easy to find. In real power dispatch, the optimal dispatch for the minimum load is very easy to find, that being the minimum generation. With the full linearized model, either the minimum aggregate load or the threshold load would seem to be good choices. However there is a complication due to the extra degrees of freedom, which is discussed next.

In real power dispatch, the real power line flows are the dependent variables. They are practically never at a bound for minimum load. In the general load flow model considered here, line flows, voltages, and/or reactive powers are the dependent variables. For an arbitrarily chosen independent/dependent partition, with all transparent variables sent to the appropriate bounds, it cannot be guaranteed that the dependent variables are all within bounds. The much sought partition with no violated constraint can be found using a phase-one of linear programming. The procedure is usually quite fast. Basically, it finds a feasible point for a set of linear equalities and inequalities. The constraints would be those of program ED, and also all real power generations set to their lower bounds. However, since the full G matrix has not been computed explicitly, it is preferable to work with the mathematically equivalent constraints of the Jacobian model, $y = Jx$. The latter also has the advantage of being sparse. The solution provides a set of non-basic (independent) variables at their bounds and feasible basic (dependent) variables. This would form the optimal solution for some economic dispatch problem, for a load between minimum aggregate load and threshold load. That would be the initial problem.

There remains a degeneracy to be resolved when the load reaches the threshold load. As was the case for real power dispatch, a rule is needed to free a generation from its lower bound. In so doing its Lagrange multiplier jumps to zero. The rule is the same as before, i.e. the incrementally cheapest generator comes off its bound first.

4.3 Solution of the Minimum Loss Subproblem

4.3.1 Formulation and Optimality Conditions

The objective function in minimum loss is the real power loss in transmission. It can be expressed most simply as the difference between generated and consumed real power,

$$P_{loss} = e^T [P_g - P_l] \tag{4.63}$$

With a linear objective, the subproblem "reduces" to linear programming. The constraints for minimum loss are identical to those in economic dispatch. The minimum loss problem will be denoted (ML).

This section provides the theory for the solution of the minimum loss problem by the continuation method. Although the upcoming ideas have not yet been implemented, it is felt that the varying load strategy can be quite fast, once an initial optimum is provided. As for the varying limits strategy, which solves for a single load, it might not be any faster than the simplex method, particulary since its initial, simple problem is more complicated than for economic dispatch. The merits of the varying limits strategy are best determined by numerical testing.

The parametric techniques developed for quadratic programming in economic dispatch are still valid for linear programming, but there are some noticeable differences between parametric QP and LP. For one, there is no objective term associated with the independent variables in the left-hand-side matrix of the optimality conditions. Hence in an initial simple problem similar to that for economic dispatch, all but one of these variables are sent to their bounds. This is clearly an erroneous dispatching strategy, because almost all real power generations would surely be sent to their bounds. The continuation process would then require many breakpoints to solve the desired problem. Solution techniques for the simple problem will take this difficulty into account. The real power generations are not transparent though, because they are cost-related; the linear objective term is present in the right-hand-side.

Optimality conditions for minimum loss incorporating transparent variables are presented in eq. 4.64 below. Inspection of its structure indicates that to be consistent, the number of active constraints must be equal to the number of independent variables; this agrees with the fundamental theorem of LP. Hence the toughest part in obtaining optimality here is the search for the optimal active set.

The Kuhn-Tucker optimality conditions, in a form similar to eq. 4.4, yield the following:

$$
\left[\begin{array}{cccc|cccc}
 & & & & -g_{0p}^{A} & -H_p^{AT} & -R_p & \\
 & & & & -g_{0p} & -H_p^{T} & & \\
 & & & & -g_{0t}^{A} & -H_t^{AT} & & -R_t \\
 & & & & -g_{0t} & -H_t^{T} & & \\
\hline
g_{0p}^{AT} & g_{0p}^{T} & g_{0t}^{AT} & g_{0t}^{T} & & & & \\
H_p^{A} & H_p & H_t^{A} & H_t & & & & \\
R_p & & & & & & & \\
 & & R_t & & & & &
\end{array}\right]
\begin{bmatrix} p_g^A \\ p_g \\ t_g^A \\ t_g \\ \lambda_0 \\ \lambda_1 \\ \mu_p \\ \mu_t \end{bmatrix}
=
\begin{bmatrix} -e^A \\ -e \\ 0 \\ 0 \\ k_0^{lim} \\ k_1^{lim} \\ R_p p_g^{lim} \\ R_t t_g^{lim} \end{bmatrix}
\tag{4.64}
$$

Manipulations of eq. 4.64 are best handled using the simplified notation:

$$
\begin{bmatrix} 0 & -A^T \\ A & 0 \end{bmatrix}
\begin{bmatrix} b_g \\ \Lambda \end{bmatrix}
=
\begin{bmatrix} -e' \\ k' \end{bmatrix}
\tag{4.65}
$$

Now the dispatchable variables and the Lagrange multipliers are decoupled, leading to the familiar LP equations:

$$A\, b_g = k' \tag{4.66.a}$$

and

$$A^T \Lambda = e' \tag{4.66.b}$$

Equation 4.66.a is the primal problem and eq. 4.66.b is the dual problem. The matrix A is square.

4.3.2 Solution of Minimum Loss Using the Varying Limits Strategy

a) Optimal solution trajectories

The perturbation function of eq. 3.7 is implemented in the vector k′ of eq. 4.66.a, yielding:

$$k'(\theta) = k_0' + \Delta k'.\theta \qquad (4.67)$$

The part of k′ corresponding to the constraints on dependent variables is identical to eq. 4.27, the perturbation function for economic dispatch using the varying limits strategy. The part of k′ corresponding to constraints on independent variables remains unaltered, and independent of θ.

The optimal solution trajectories obtained by inserting k′(θ) into eq. 4.65 are as follows:

For inactive $b_g(\theta)$:

$$b_g(\theta) = b_{g0} + \Delta b_g\theta \qquad (4.68.a)$$

$$b_{g0} = A^{-1}k_0' \qquad (4.68.b)$$

$$\Delta b_g = A^{-1}\Delta k' \qquad (4.68.c)$$

The partition P_g/t_g can be sorted out after computation.

For active b_g:

$$R_pP_g{}^A = R_pP_g{}^{lim} \qquad (4.69)$$

$$R_tt_g{}^A = R_tt_g{}^{lim} \qquad (4.70)$$

For Lagrange multipliers:

$$\Lambda = A^{-T}\ e' \qquad (4.71)$$

The partition λ/μ can be sorted out after computation.

while monitoring inactive dependent variables:

$$d_g^{I}(\theta) = d_{g0} + \Delta d_g \theta \qquad \text{(bis 4.38.a)}$$

$$d_{g0} = d_0 + G(b_{g0} - (b_l + b_e)) \qquad \text{(bis 4.38.b)}$$

$$\Delta d_g = G \Delta b_g \qquad \text{(bis 4.38.c)}$$

An important observation is that Lagrange multipliers are piecewise constant in θ. When an inactive variable moves to its bound, it is added to the active set and the optimality conditions are degenerate, but the primal problem remains consistent at the breakpoint. The dual inherits a degree of freedom, with which the degeneracy is resolved. The process will be explained later in the Appendix on degeneracy.

b) The initial, simple problem

Using this homotopy strategy for economic dispatch, the limits on some functional dependent constraints are initially relaxed, and then returned to their desired positions. However, in this initial, simple problem the functional constraints cannot be neglected. These constraints are needed to "fill out" the formulation of the initial simple problem, as will be explained below.

As pointed out earlier, if the initial, simple problem neglects the functional constraints, the LP structure of the problem pushes all the independent variables (with non-zero g_0 coefficients) except one to their bounds. This can be seen in a graphical interpretation of the solution of this simplified minimum loss, similar to that of figure 4.1 for the initial, simple problem of economic dispatch. Following a simple transformation of variables, $b_i' = b_i/g_{0i}$, this problem becomes "minimize $g_{0p}^T P_g'$ subject to $e^T b_g' = k'$ and bounds on the bg' ". This transformation gives each new variable its own individual incremental cost (dPloss/db_{gi}'), whereas the initial variables all had the same incremental costs. The new variables can now be used in a generalized equal incremental cost procedure.

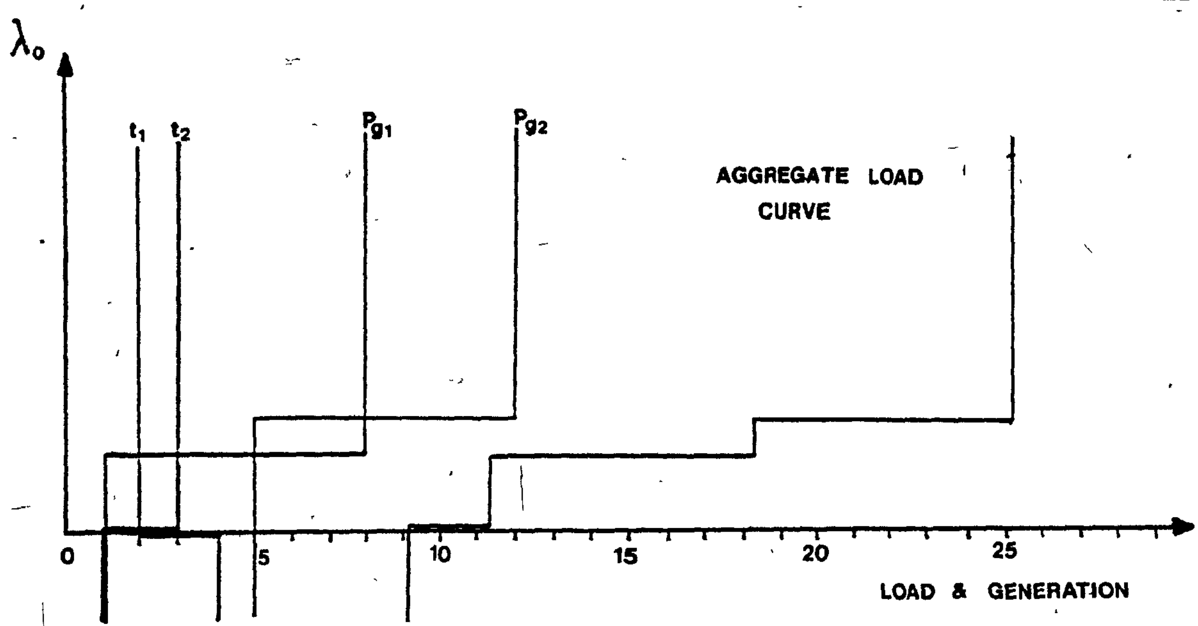


Figure 4.3. The generalized incremental cost procedure applied to the minimum loss problem.

In figure 4.3, the piecewise constant incremental cost curves are drawn for each independent variable and an aggregate load curve is built. This figure shows immediately that the optimal solution of this simpe problem retains only one inactive variable at a time.

If a "bad" choice of independent variables is made in the initial, simple problem proposed immediately above, then the continuation process requires many breakpoints to pass from the optimal active set of the initial, simple problem to that of the desired problem. Ideally, the two optimal active sets should be close.

If a good initial guess of the optimal active set, or even better, of the optimal solution, is available, then it can be incorporated into a better initial, simple problem. This suggests the following improved procedure for the initial simple problem.

Note that in the following discussion, two sets of nomenclature are used in parallel. These are (i) the nomenclature we have been using so far, and (ii) the notions of basic and nonbasic variables from the simplex method of linear programming [Chvatal 1983].

The variables for which there are good initial guesses are set to those values. The active variables in this group are made non-basic, and the inactive ones are basic. Note that the number of fixed values is at most equal to the dimension of the problem, denoted ndim. The remaining dependent variables are computed as a function of the fixed variables. The dependent variables can also be partitioned as basic or non-basic, as long as there is space available in those partitions. For this vector of variables to be an optimum, ndim constraints must be active. To create that many constraints, the limits on some of the non-basic dependent variables could be displaced to coincide with the present values of the variables. The best candidates for this operation would be the dependent variables computed out of bounds and those which are suspected of being at a bound at the optimum. Once the choice of variables for this operation has been made, there only remains to choose if the upper or the lower bound is appropriate. That choice can easily be made upon inspection of the simplex tableau built for this problem. We denote by d_g' the variables affected by the relaxing of their bounds and by d_{gi0} its present value. Then the perturbation function which moves the bounds of dependent constraint d_{gi}' is written as

$$d_{gi}'(\theta) = d_{gi0}' + \Delta d_i . \theta \qquad (4.72.a)$$

with

$$\Delta d_i = d_{gi}^{lim} - d_{gi0}' \qquad (4.72.b)$$

Here the Δd_i are distinct.

This being an optimum for the primal problem, non-negative Lagrange multipliers can be computed to satisfy the dual problem. Then a solution for this initial, simple problem is complete.

A disadvantage of this approach is that possibly many rows of the G matrix must be computed. However, if a good initial optimal active set is chosen, few breakpoints should appear.

4.3.3 Solution of Minimum Loss Using the Varying Load Strategy

a) Optimal solution trajectories

Once again the load trajectory $b_l(\theta)$ is considered the continuation parameter. It is inserted into the vector k′ of eq. 4.66.a, and written as follows:

$$k'(\theta) = k_0' + \Delta K'.b_l \tag{4.73}$$

The part of k′ corresponding to the dependent constraints is identical to eq. 4.51, the perturbation function for economic dispatch using the varying load strategy. The part of k′ corresponding to independent constraints remains unaltered, and independent of θ.

The optimal solution trajectories are obtained by inserting the $k'(\theta)$ vector into eq. 4.66. They are:

For inactive $b_g(\theta)$:

$$b_g(\theta) = b_{g0} + \Delta B_g b_l \tag{4.74.a}$$

$$b_{g0} = A^{-1}k_0' \tag{4.74.b}$$

$$\Delta b_g = A^{-1}\Delta K' \tag{4.74.c}$$

The partition P_g/t_g can be sorted out after computation.

For active b_g:

$$R_pP_g^A = R_pP_g^{lim} \tag{4.75}$$

$$R_tt_g^A = R_tt_g^{lim} \tag{4.76}$$

For Lagrange multipliers:

$$\Lambda = A^{-T} e' \qquad (4.77)$$

The partition λ/μ can be sorted out after computation.

while monitoring inactive dependent variables:

$$d_g^I(\theta) = d_{g0} + \Delta D . b_1 \qquad \text{(bis 4.62.a)}$$

$$d_{g0} = d_0 + G [b_{g0} - b_e] \qquad \text{(bis 4.62.b)}$$

$$\Delta D = G [\Delta b_g - I] \qquad \text{(bis 4.62.c)}$$

Here again the Lagrange multipliers are piecewise constant in θ. The comments presented for the solution homotopies in the varying limits strategy also apply here.

b) The initial, simple problem

The remarks made for economic dispatch using the varying load strategy also apply for minimum loss.

4.4 Solution of the Minimum Load Shedding Subproblem

4.4.1 Development of the Problem Formulation

A fairly general objective function is proposed. It consists of a quadratic function of the difference between the customer demand b_D and the feasible load b_L supplied by the utility.

It is written

$$S(b_1) = [\ b_D - b_1\ ]^T\ S_0\ [\ b_D - b_1\ ] \tag{4.78.a}$$

$$= b_D{}^T b_D - 2[\ b_D{}^T S_0\ ]b_1 + b_1{}^T S_0 b_1 \tag{4.78.b}$$

$$= s_0 - s_1{}^T b_1 + \tfrac{1}{2}\, b_1{}^T S_2 b_1 \tag{4.78.c}$$

with S_0 symmetric.

The unknown variables are the supplied feasible loads b_1, for a known demand b_D. The value of the objective function is zero in the normal state, when b_1 and b_D are identical. It takes on positive values in the load shedding state.

There really is no definite choice of objective function for the load shedding problem, so some discussion is warranted. The above function is a weighted norm of the demand-load mismatch. The weights could be attributed according to priority lists, or revenues,etc.. Off-diagonal weights might also be considered, placed symmetrically to retain the usual advantageous quadratic form. If S_0 is the identity matrix, the objective is simply the Euclidean norm of the mismatch. This thesis does not propose to study the merits of the different objectives, but rather to furnish a mathematical framework in which they can be incorporated.

As in many load shedding subproblem formulations, the load shedding proposed in this section uses the full linearized load flow model. Hence voltages, reactive power sources and passive network controls can participate in the control action.

Two formulations are regrouped, for load shedding based on: (i) economic dispatch, or (ii) minimum loss. The dispatchable variables can be computed, along with the optimal feasible load, to satisfy either one of the normal-state dispatches.

The formulations contain the objective function above, plus the constraints described in this paragraph. The loads are bounded between some predetermined lower bound and the varying demand.[1] In addition, the optimality equations, eq. 4.4 for economic dispatch or eq. 4.64 for minimum loss, are added as equality constraints, to impose the optimal dispatch for the given (as yet unknown) load. These constraints contain the usual linearized load flow and the bounds on the dispatchable variables. The two formulations are denoted (LS,ED) for load shedding based on economic dispatch and (LS,ML) for load shedding based on minimum loss.

An important role of the added equality constraints in this load shedding formulation is to define a unique set of dispatchable variables for a given load. As pointed out by [Chan & Yip 1979], in a load shedding problem with the usual load flow constraints and where the loads alone make up the objective function, the optimal loads can be computed, but the corresponding dispatchable variables are non-unique. The added equality constraints given by the optimality equations define unique and optimal values of the dispatchable variables for a given load. This addition is also necessary to uniquely define the dependent variables in our formulation, since they are expressed as functions of the loads and the dispatchable independent variables.

The formulation of the load shedding problem is now written. The compact notation of eq. 4.11 is used to express the equality constraints for either one of the two variants, with the dependence of the k' term on the load b_1 written explicitly. Due to its excessive length, the formulation, denoted (LS), is presented on the next page.

[1] We note that the choice of very high lower bounds on the loads (to satisfy priority loads, for example) could lead to infeasibility in the load shedding problem.

$$\min_{b_l, b_g} \quad s_0 - s_{1T} b_l + \tfrac{1}{2} b_l^T . S_2 . b_l$$

s.t.

$$b_l^m \leq b_l \leq b_D$$

$$\begin{bmatrix} B' & -A^T \\ A & \end{bmatrix} \begin{bmatrix} b_g \\ \Lambda \end{bmatrix} = \begin{bmatrix} -a' \\ k_0 + A_l . b_l \end{bmatrix} \qquad \text{(LS)}$$

for A containing all the active constraints on the dispatchable variables, while monitoring the inactive dispatchable variables:

$$d_g^{Im} < d_{g0}^{\cdot I} + G^I \; [b_g - b_l - b_e] < d_g^{IM}$$

$$b_g^m < b_g < b_g^M$$

4.4.2 Development of the Optimality Conditions

The Lagrangian function for the minimum load shedding problem (LS) is a function of the loads and of the unknowns for the normal-state dispatch. It is written as

$$\mathcal{L} = S(b_l) - \lambda_1^T \; [Uz - v] - \mu_1^T \; [b_l - b_l^{lim}] \qquad (4.79)$$

where the new notation is defined as follows:

U, z, and v — are simplifying notation, used temporarily to express the left-hand-side matrix, the vector of unknown dispatchable variables and normal-state Lagrange multipliers, and the right-hand side vector of the equality constraints of problem (LS),

λ_1 is the vector of Lagrange multipliers associated with the equality constraints of load shedding,

μ_1 is the vector of Lagrange multipliers associated with the inequality constraints on the loads,

and

R_1 is the matrix of constraint status indices for loads.

The λ_1 vector is broken down into its two main components, λ_{1-g} and λ_{1-A}, corresponding to the two main partitions in the equality constraints of problem (LS).

Having introduced this notation, the first order optimality conditions for problem (LS) can now be given. They are:

$$\left[\begin{array}{ccccc|c|c}
S_2^A & & & & & A_1^{AT} & -R_1^{AT} \\
\cline{7-7}
 & S_2 & & & & A_1^T & \\
 & & & & -B' & -A^T & \\
 & & & & A & & \\
\cline{1-6}
 & & B' & -A^T & & & \\
A_1^A & A_1 & A & & & & \\
\cline{1-4}
R_1^A & & & & & &
\end{array}\right]
\begin{bmatrix} b_1^A \\ b_1 \\ b_g \\ \Lambda \\ \lambda_{1-g} \\ \lambda_{1-A} \\ \mu_1 \end{bmatrix}
=
\begin{bmatrix} s_1^A \\ s_1 \\ 0 \\ 0 \\ -a' \\ k_0 \\ R_1 b_1^{lim} \end{bmatrix}
\qquad (4.80)$$

These optimality conditions can be split into two groups, as for the previous tasks, separating active loads and active dispatchable independent variables from the remaining variables. These two groups of equations are given below. In the second group, the partitions of the variables b_g, the Lagrange multipliers λ_1, and of the corresponding coefficient arrays are given in greater detail than in eq. 4.81 above. These equations are:

$$R_1{}^A b_1 = R_1 b_1{}^{lim} \tag{4.81.a}$$

$$R_p P_g{}^A = R_p P_g{}^{lim} \tag{4.81.b}$$

$$R_t t_g{}^A = R_t t_g{}^{lim} \tag{4.81.c}$$

$$R_1{}^T \mu_1 = (S_{2A} b_1{}^A - s_1{}^A) + A_1{}^T \lambda_{1-A} \tag{4.81.d}$$

$$R_p{}^T \mu_p = (B^A P_g{}^A + a^A) - A_p{}^{AT} \Lambda \tag{4.81.e}$$

$$R_t{}^T \mu_t = \qquad - A_t{}^{AT} \Lambda \tag{4.81.f}$$

and

$$\begin{bmatrix} S_2 & & & & & & A_1{}^T \\ & & & & -B & & -G_p{}^T \\ & & & & & & -G_t{}^T \\ & & & & G_p & G_t & \\ & B & & -G_p{}^T & & & \\ & & & -G_t{}^T & & & \\ A_1 & G_p & G_t & & & & \end{bmatrix} \begin{bmatrix} b_1 \\ P_g \\ t_g \\ \lambda \\ \lambda_{1-Pg} \\ \lambda_{1-tg} \\ \lambda_{1-A} \end{bmatrix} = \begin{bmatrix} s_1 \\ 0 \\ 0 \\ -G^{AT} b_g{}^A \\ -a \\ 0 \\ k_0 + A_1{}^{AT} b_1{}^A - G^{AT} b_g{}^A \end{bmatrix} \tag{4.82}$$

where λ_{1-Pg} and λ_{1-tg} are the partitions of λ_{1-g} corresponding to the inactive real power generations and transparent variables respectively. The terms of the λ_1 vector corresponding to active dispatchable variables need not be computed, because no restriction has been imposed on their values and because they are not needed in the computation of other unknowns.

4.4.3 Solution of Minimum Load Shedding by the Varying Demand Strategy

a) Optimal solution trajectories

The perturbation function of eq. 3.8 is implemented. It affects two terms of the right-hand-side vector of eq. 4.81. The linear objective term becomes

$$s_1(\theta) = 2S_0{}^T [b_{D0} + \Delta b_D \theta] \tag{4.83.a}$$

$$= s_{10} + \Delta s_1 \theta \tag{4.83.b}$$

In the limits vector, the upper bounds in the load term become

$$R_1 b_1^M(\theta) = R_1[b_{D0} + \Delta b_D \theta] \tag{4.84}$$

The remaining terms k_0, $G^{A^T} b_g^A$ and a of the right-hand-side vector are independent of θ.

Equations of the optimal solution trajectories for load shedding cannot be broken down into groups for the individual variables, as was done for the previous tasks. This is because the block in the optimality conditions corresponding to the normal-task dispatch is rank deficient. Hence terms like K, L and M of eq. 4.18 - 4.20 which result from the partitioning of the left-hand-side matrix of the optimality conditions cannot be formed here. The addition of the A_1 row and column for load shedding restores full row rank to the constraint block of the optimality conditions. In order to retain full rank of the left- hand-side matrix of the optimality conditions, the number of inactive loads (where load shedding takes place) is at least equal to the rank deficiency of the normal-task dispatch block.

The solution trajectories are computed from the optimality equations in a more general form. The two components of the solution trajectories $[s(\theta)] = [s]_0 + [\Delta s].\theta$, for eq.4.83, are obtained from the solutions of those sets of linear equations with the right-hand side $[rhs] = [rhs]_0 + [\Delta rhs].\theta$. Then the solution trajectories for the remaining unknowns in eq. 4.82 are easily computed.

An important observation is that the optimality equations of eq. 4.81 form a general set of optimality conditions, also valid for the normal-state tasks. For the normal-state tasks, the Lagrange multipliers λ_1 are identically zero, because the normal-state block which multiplies λ_1 is full rank, and the corresponding right-hand-side term is zero. With λ_1 identically zero, no load can be inactive because the values of the unconstrained minima $b_1 = S_2^{-1} s_1$ are identically equal to the demand b_D. Then all the loads are placed in the right-hand-side of the optimality conditions, and eq. 4.81 reduces to either eq.4.11 for economic dispatch or eq. 4.65 for minimum loss. Two important conclusions are that

1) Equation 4.81 is a general optimality equation for all three tasks studied in this chapter, with the normal state tasks being particular cases.
2) Load shedding is used only as a last resort in this dispatching strategy, when the demand trajectory leaves the loadability region, and normal-state dispatch becomes infeasible.

b) The initial, simple problem

Load shedding is invoked when the dispatch task used in the normal state reaches a feasibility limit. The optimum on the boundary is a "final" optimal solution for the normal state task, but an "initial" optimal solution for load shedding. The load shedding objective starts at a value of zero and increases with a changing demand. The optimal active set and values of the variables are transferred from the normal-state task to the load shedding.

A difficulty occurs when changing tasks, because of the degeneracy in the optimality conditions of the normal-state task. The choice of load which will leave its bound to resolve the degeneracy is decided by techniques discussed in Appendix 4.2 on degeneracy. It is felt that usually loads will be released one at a time, because the feasibility limit in the normal-state task is usually reached when a single, final constraint is added to the optimality conditions. If that is the case, the load to be freed is found quickly.

Once the degeneracy is resolved, this algorithm for tracking the optimal loads and dispatch as a function of the varying demand is straightforward. The return to the normal-state dispatch, when conditions for load shedding have subsided and the value of the objective function returns to zero, should occur in the reverse order of the description above.

CHAPTER V

OTHER APPLICATIONS OF THE CONTINUATION METHOD IN OPF

5.1 Introduction

This short chapter presents four other applications of the continuation method in OPF. Three new applications investigate certain changes or additions to the set of constraints. They are: the addition of dynamic ramp constraints to the economic dispatch formulation, the perturbation of constraint functions by continuation methods to redispatch following contingencies, and the substitution of the load flow Jacobian model by the DC load flow model. For those cases, the OPF subproblems are described and solution procedures are only sketched. In a fourth application, following an economic dispatch solved by the varying load strategy, expressions are derived for bus and system incremental costs.

5.2 The Ramp Constrained Economic Dispatch Problem

So far, only static OPF problems have been studied. In this section, optimality conditions are given for the dynamic ramp-constrained problem, based on the subproblem model of OPF. Some ideas for their solution are sketched, using the continuation method, but a full solution procedure will not be developed. The added time dimension makes the problem much larger, although well-structured.

In keeping with previous solution methodologies, nonlinear programming is used to solve the dynamic problem. A sequence of optimal dispatches is sought over a period of time, for a sequence of different loads obtained by load forecasting. In the absence of dynamic constraints, the varying load strategy can be used in the solution of the static problem. When dynamic constraints become active, some new technique must be used to allow optimality conditions to look ahead over a certain time span. This time span would cover only the period when dynamic constraints are active. It would be suggested a priori, based on rapid variations in the load forecast, although its exact knowledge is not necessary beforehand.

In this approach, time is discretized into NT segments over span T. All load flow variables and their Lagrange multipliers are sought at instants $t = n\Delta t$, $n = 1, \ldots, NT$. An example of notation for variable v is

$$v_i^{(n)}$$

where subscript i identifies the variable itself,

superscript n identifies the time instant.

The number of variables increases to NT times the number of variables in the static problem. The usual static load flow constraints hold at each instant. Dynamic ramp constraints are added to the formulation; they place limits on the variations of real power generations over a single time interval Δt. They are written

$$| P_{gi}^{(n)} - P_{gi}^{(n-1)} | \leq r_i \tag{5.1}$$

$$i = 1, \ldots, ng$$

Regrouped in vector form and split along the active/inactive partition, the set of ramp constraints at period n is written

$$R_r^{(n)} [P_g^{(n)} - P_g^{(n-1)} - r] = 0 \tag{5.2}$$

where R_r is the active/inactive index matrix for the ramp constraints.

These are the simplest dynamic constraints for OPF, since no new variable is introduced, and each constraint introduces only two non-zero terms ,±1, in the constraint matrix. However, the upcoming methodology can easily be expanded to more complex dynamic constraints.

To simplify the presentation, only the case of dynamic economic dispatch is given. The network and cost parameters are considered to be time-invariant, and bus loads are varied over time. Formulations for other OPF tasks or with time-varying parameters can be built in similar fashion.

The objective of the dynamic problem is the fuel cost over the entire time span. That is the summation of the fuel costs at each time period. It is written

$$C = \sum_{n=1}^{NT} C^{(n)}(P_g) = NTc_0 + \sum_{n=1}^{NT} a^T P_g^{(n)} + 0.5 \sum_{n=1}^{NT} P_g^{(n)T} B P_g^{(n)} \qquad (5.3)$$

All the elements of the dynamic optimization having been stated, the formulation of the ramp constrained economic dispatch, denoted (ED,R), can be written:

$$\begin{array}{lrcl}
\min\limits_{b_g^{(n)}} & C & = & \sum_{n=1}^{NT} C(P_g^{(n)}) \\
\text{s.t.} & g_0^T b^{(n)} & = & 0 \\
& d_g^m \le d_0 + G_1 b^{(n)} & \le & d_g^M \\
& b_g^m \le b_g^{(n)} & \le & b_g^M \\
& | P_g^{(n)} - P_g^{(n-1)} | & \le & r \\
& \text{for all } n = 1,\ldots,NT & &
\end{array} \qquad \text{(ED,R)}$$

Lagrange multipliers associated with ramp constraints are denoted ρ. The active set is assumed known, as in previous analyses, so matrices $R_d^{(n)}$, $R_b^{(n)}$, and $R_r^{(n)}$ are known. The Lagrangian for (ED,R) is given by the following equation:

$$\mathcal{L} = \sum_{n=1}^{NT} \Big(C(P_g^{(n)}) - \lambda^T R_d^{(n)} [G b_g^{(n)} - k^{lim}] - \mu^T R_b^{(n)} [b_g^{(n)} - b^{lim}] - \rho^T R_r^{(n)} [P_g^{(n)} - P_g^{(n-1)} - r] \Big) \qquad (5.4)$$

The static constraints can be regrouped, as before, using the notation A and k'. That regroups the second and third terms of the Lagrangian. The constraints are split along the real power/transparent partition. The derivatives of the Lagrangian with respect to all the independent variables and all the Lagrange multipliers for each time period form the optimality conditions. They are given by eq. 5.5. The basic blocks in these equations are the static optimality conditions, placed side by side along the block diagonal. The ramp constraints can be thought to link the individual static problems. Constraint submatrices R_r (or their transposes) usually appear four times at each period, as shown in eq. 5.5, linking two adjacent problems. For the first and last periods though, they appear only twice.

$$
\left[\begin{array}{cccc|cccc|c|cccc}
B & & -A_p^{(1)T} & -R_r^{(1)T} & & & & R_r^{(2)T} & & & & & \\
 & & -A_t^{(1)T} & & & & & & & & & & \\
A_p^{(1)} & A_t^{(1)} & & & & & & & & & & & \\
R_r^{(1)} & & & & & & & & & & & & \\
\hline
 & & & & B & & -A_p^{(2)T} & -R_r^{(2)T} & & & & & \\
 & & & & & & -A_t^{(2)T} & & & & & & \\
 & & & & A_p^{(2)} & A_t^{(2)} & & & & & & & \\
-R_r^{(2)} & & & & R_r^{(2)} & & & & & & & & \\
\hline
 & & & & & & & & \ddots & & & & R_r^{(NT)T} \\
\hline
 & & & & & & & & & B & & -A_p^{(NT)T} & -R_r^{(NT)T} \\
 & & 0 & & & & & & & & & -A_t^{(NT)T} & \\
 & & & & & & & & & A_p^{(NT)} & A_t^{(NT)} & & \\
 & & & & & & & & -R_r^{(NT)} & R_r^{(NT)} & & &
\end{array}\right]
\left[\begin{array}{c}
P_s^{(1)} \\ t_s^{(1)} \\ \Lambda^{(1)} \\ \rho^{(1)} \\ \hdashline
P_s^{(2)} \\ t_s^{(2)} \\ \Lambda^{(2)} \\ \rho^{(2)} \\ \hdashline
* \\ * \\ * \\ \hdashline
P_s^{(NT)} \\ t_s^{(NT)} \\ \Lambda^{(NT)} \\ \rho^{(NT)}
\end{array}\right]
=
\left[\begin{array}{c}
-a \\ 0 \\ k'^{(1)} \\ R_r^{(1)} r \\ \hdashline
-a \\ 0 \\ k'^{(2)} \\ R_r^{(2)} r \\ \hdashline
* \\ * \\ * \\ \hdashline
-a \\ 0 \\ k'^{(NT)} \\ R_r^{(NT)} r
\end{array}\right]
\qquad (5.5)
$$

A chain of static problems is formed when some ramp constraints are active over a group of consecutive periods. The effect of each ramp constraint is felt by all the variables in the chain. That is why each ramp constraint must be considered over a long time span.

Ideally, the first block of the equation is not part of a previous chain. If that is the case, the term $R_r^{(1)}$ can be dropped from eq. 5.5. Also, the last block should end a chain; if not, the time window over which the optimization is carried out can be shifted or expanded.

The major difficulty in solving the optimality conditions is in identifying the active ramp constraints, given a load trajectory. A successful technique for identifying them was developed for hydrothermal scheduling, and is based on the continuation method [Calderon 1985]. In this method, all the $k'^{(n)}$ subvectors in the right-hand-side of eq. 5.5, which contain the load terms, are made identical initially. Then a perturbation function resets them to their original values. The perturbation function is

$$k'(\theta) = \begin{bmatrix} k'^{(1)} \\ k'^{(1)} \\ \cdots \\ k'^{(1)} \end{bmatrix} + \begin{bmatrix} 0 \\ k'^{(2)} - k'^{(1)} \\ \cdots \\ k'^{(NT)} - k'^{(1)} \end{bmatrix} \theta \qquad (5.6)$$

At $\theta=0$, the NT blocks yield identical solutions, so that discrepancies in the generations at different instants are nil. The solution of a single block suffices to start the problem. That would be handled using techniques described in Chapter 4. Then as θ is increased, differences between the blocks appear and ramp constraints might be activated. As long as the blocks are decoupled, the change in k' can be carried out using the static varying load method. Maybe some similar technique could be applied to chains of problems. When θ reaches 1, the optimal solution is found.

Numerically, such a solution technique is quite taxing, mostly because of the large amount of memory required to carry the NT blocks. In terms of size of computation, this should not be much larger than the computation of NT static problems, but the time of computation could be much longer on small computers due to the paging of information in and out of memory.

It is suggested that numerical techniques be sought which take advantage of the decoupling and the weak coupling of ramp constraints. The matter will not be investigated any further in this study. A final suggestion is that if one block of the optimality conditions is seen to be infeasible, it could be replaced by a load shedding block. The detection of infeasibility at a particular period and the changeover to load shedding are not developed in this study.

5.3 Redispatching Following a Contingency

A contingency is a sudden, unexpected loss of some element of the system, i.e. generation, load, or transmission. Immediately following a contingency, emergency procedures are implemented. First, controls serve to protect any vulnerable part of the system, and fend off instability. These are very fast controls, usually within less than a second after the inception of the contingency, and are usually performed automatically. Then if some load flow quantities have moved beyond their bounds, a human operator tries to restore quickly the secure operation. After some trials, the system is brought back to a secure state, and then a new optimal dispatch is sought.

One step can be cut from the procedure if the operator is furnished an optimal post-contingency dispatch right away. In this section, numerical techniques based on the continuation method are suggested and briefly sketched for computing the post-contingency optimal dispatch. It is thought that with the proper adjustments, these methods are likely to be quite fast.

The dispatch results are target values towards which the operator should move the system. The operator still has to guide the system, since the optimal solution trajectories produced by these calculations bear no meaning in the post-contingency dynamics.

The idea behind the use of the continuation method in post-contingency redispatch is the following. The contingency removes some element from the power system. Instantly the corresponding parameter jumps to a new value. Computing the post-contingency optimum from scratch, or even using pre-contingency information as an initial guess, seems to be lengthy in most cases, especially if the active set is significantly modified. Instead, in this new approach a perturbation function is introduced to vary continuously the value of the affected parameter, from the pre-contingency value to the new value:

$$p(\theta) = p_i + (p_f - p_i)\theta \qquad (5.7)$$

θ in the interval $[0,1]$

where p_i and p_f are initial and final values of parameter p.

The pre-contingency optimal dispatch is known, so it serves as the initial simple problem. As the parameter is varied, optimal solution trajectories are produced, ending at the post-contingency optimum. The intermediate values along the trajectory might be useful in operations planning, since they express the optimal dispatch versus system parameters. They are of no use in redispatching.

a) Right-hand-side perturbations

Optimal solution trajectories can be furnished explicitly when the varying parameter affects only the right-hand-side. Then, as for all the previous cases, the optimal solution trajectories are piecewise linear. Two contingency cases involve only right-hand-side perturbations of the optimality conditions: any loss of bus load or the complete loss of a generation. Their treatment is basically similar to that in real power dispatch. For the loss of plant i, both upper and lower bounds of generation i are sent to zero, using a perturbation function

$$P_{gi}^{lim}(\theta) = P_{gi0}^{lim}(1-\theta) \qquad (5.8)$$

where P_{gi0}^{lim} is the true value of the limit.

That has the effect of shutting off plant i. When adding a generator, the procedure is implemented in reverse. The extra generation limits $P_{gN+1}^{m} = P_{gN+1}^{M} = 0$ and the associated Lagrange multipliers are added to the formulation of the optimality conditions. With that particular choice of limits, the addition does not cause a jump in the optimal dispatch. The new Lagrange multipliers are computed as a function of existing ones and the cost data. Then allow the generation limits to take on their true values:

$$P_{gi}^{lim}(\theta) = \theta P_{gi0}^{lim} \qquad (5.9)$$

A loss of bus load, whether partial or complete, could be handled as a special case of the varying load strategy, in which a single load would be varied.

b) Left-hand-side perturbations

In the real power redispatch problem using a DC load flow model, some piecewise nonlinear solution trajectories were reported [Huneault et.al.1985] for perturbations affecting the left-hand side of the optimality conditions. For partial loss of generation at a bus, the solution trajectories are of the form

$$\lambda(\theta) = \lambda_0 + \theta \frac{N_\lambda(\theta)}{D_\lambda(\theta)} \qquad (5.10.a)$$

$$P_g(\theta) = P_{g0} + \theta \frac{N_p(\theta)}{D_p(\theta)} \qquad (5.10.b)$$

where

λ_0, P_{g0}	are pre-contingency values.
$N_\lambda(\theta), D_\lambda(\theta)$	are second degree vector and scalar polynomials in θ, respectively.
$N_p(\theta), D_p(\theta)$	are third degree vector and scalar polynomials in θ, respectively.

For the loss of a transmission line, it was reported that the polynomials are of much higher degree. In fact, they are of degrees 19 and 20 in θ. Because the terms of the dispatch are much more complicated with the full linearized load flow model, the evaluation of nonlinear solution trajectories would lead to some unreasonably difficult expressions. Hence no attempt has been made to work them out.

For the contingency problems involving left-hand-side perturbations, a numerical technique is proposed. Instead of varying $p(\theta)$ continuously, it can be varied discretely, i.e.

$$p(\theta) = p_i + (p_f - p_i)n\Delta\theta \qquad (5.11)$$
$$n = 1,\ldots,N \quad \text{and} \quad N\Delta\theta = 1$$

If variations in $p(\theta)$ from problem (n) to problem (n+1) are small, then so will the variations in their optimal solutions. To solve one problem, for a given n, the varying limits strategy could be applied. The optimal active set from the previous problem could be retained to form the basis of the new initial simple problem. Often, the previous active set would be carried over intact to the new problem, or little change would occur. That would avoid lengthy searches of the active set, or lengthy applications of the continuation procedure. The discrete solution trajectory could advance quickly if it consists mostly in solving a sequence of simple problems.

The number of steps N in which to divide the perturbation interval would be a practical matter, to be determined experimentally. Also, a stepsize control could be explored, to allow larger steps when adjacent problems have very close solutions.

One possible difficulty with left-hand-side perturbations is that their solution trajectories, whether analytical or discrete, can be volatile. Without a constraint box to limit its excursions, an optimal solution can break down when for some $\theta=\theta'$ the left-hand-side matrix becomes singular. One case of degeneracy of this type was illustrated in Appendix 4.2. The presence of the constraint box avoids the associated problem of sending dispatch variables to infinity, because when the problem occurs the dependent constraints which are forcing the problem get replaced by simple bounds on the independent variables. These are usually isolated numerical problems; all in all, a single perturbation of a large system should not be expected to result in such drastic redispatching.

5.4 Extensions to the DC Loadflow Model

Many examples of real power economic dispatch using a DC load flow model served as starting points for studies with the full linearized load flow model. Now some applications to the DC load flow, inspired by the larger model, can be suggested to replace the load flow Jacobian.

The DC load flow model expresses real power generations and real power line flows as a function of voltage phase angles:

$$P_g = Y\delta \quad (5.12)$$
$$P_t = T\delta \quad (5.13)$$

Y is the system susceptance matrix, and rows of the T matrix contain terms y_{ij} and $-y_{ij}$ in positions i and j respectively, for line flow P_{ij}. The model comes as a result of fixing all voltage magnitudes to one per unit, replacing trigonometric functions $\sin\delta$ by δ and $\cos\delta$ by 1, and by neglecting resistance and shunt admittance in the load flow equations.

All generators except one can be retained as independent variables. The remaining slack generation (the manifold variable) is expressed as a function of the independent generations through the usual power balance equation. One voltage phase angle is set to zero, as a reference, and in the optimization the others can be monitored outside the optimization. The line flows are dependent injections, expressed, conceptually at least, in terms of independent generations.

A natural extension to the DC load flow model would see the addition of phase shifters. These devices help control the real power flows on transmission lines. The system equations become

$$P_g = Y\delta + Y_1\phi \quad (5.14)$$
$$P_t = T\delta + T_1\phi \quad (5.15)$$

where

ϕ is the nf dimensional vector of phase shifter angles.

Y_1 is an (ng x nf) matrix. Its columns contain elements y_{ij} and $-y_{ij}$ in positions i and j respectively, for phase shifter ϕ_{ij}. Position i is the near bus.

T_1 is an (nj x nf) matrix. Let ϕ_{ij} be the m^{th} phase shifter in the list. For line flow P_{ij}, the row of T_1 contains element y_{ij} in position m if the phase shifter is connected at bus i, or $-y_{ij}$ if it is connected to bus j. The row is zero if no phase shifter is connected to line ij.

The phase shifters would be independent states, and also transparent variables. The methods expounded in Chapter 4 could then be applied to the problem, with eq. 5.14 and 5.15 replacing the full linearized load flow model.

Based on analysis developed in section 4.2.3, for the varying limits strategy, optimality conditions will likely push phase shifters to their appropriate bounds, starting at low loads. Some would come off their bounds as a control measure only when some line flow reaches a limit.

The addition of varying voltages or taps to the DC load flow model could create transparent states, which could be handled like the phase shifters. However the additions result in a nonlinear model. Rather than linearizing the new equations, it would be preferable to work with the full linearized load flow model. Hence these two additions will not be considered useful.

Minimum loss dispatch and load shedding with the DC load flow model are straightforward. Methods developed earlier can be applied, replacing the full linearized load flow model by the usual DC load flow of eq. 5.12 - 5.13, or by the extended model of eq. 5.14 - 5.15. The former would be slightly different from previous models, in that no transparent variable is present. That would be, in fact, a simpler case. These two applications will not be pursued any further here.

5.5 Bus and System Incremental Costs for Economic Dispatch

Expressions are derived for bus and system incremental costs in economic dispatch, based on solutions of the subproblem. They provide exact numerical values in a nonlinear OPF algorithm, because differential information provided by the subproblem and the nonlinear problem are identical at the optimum.

a) Bus incremental costs

From the optimal dispatch trajectories for economic dispatch, it is a simple matter to develop expressions for bus incremental costs. They are the derivatives of the optimal cost with respect to the bus loads. Using the chain rule in differentiation, they are seen to be formed as the product of two easily obtainable quantities. They are denoted, taken together, as the BIC vector:

$$BIC = \frac{\partial C}{\partial b_1^T} = \frac{\partial C}{\partial P_g^T} \cdot \frac{\partial P_g^T}{\partial b_1} \tag{5.16}$$

From eq. 3.3, the first term of the above product is

$$\frac{\partial C}{\partial P_g^T} = a^T + P_g^T B \tag{5.17}$$

Inserting the value of P_g^T from eq. 4.17 for free generations, and simplifying, this becomes

$$\frac{\partial C}{\partial P_g^T} = [G_p B^{-1} a + k]^T [I - MK^{-1}]^T K^{-1} G_p \tag{5.18.a}$$

$$= \lambda^T G_p \tag{5.18.b}$$

with all notation referring to free generations only.

To simplify the presentation, the dependence of λ on the load will only be shown at the end.

From eq. 4.59, the second term of the product of eq.5.16 is

$$\frac{\partial P_g}{\partial b_1} = B^{-1} G_p^T K^{-1} [I - MK^{-1}] \Delta K \qquad (5.19.a)$$

for inactive generations,

and $$\frac{\partial P_g}{\partial b_1} = 0 \qquad (5.19.b)$$

for active generations.

Recall that ΔK is the coefficient matrix in the right-hand-side vector of the optimality conditions associated with the load, as in eq. 4.55. The product of the two terms of eq. 5.16 yields this expression for the BIC:

$$BIC = \lambda^T (G_p B^{-1} G_p^T) K^{-1} [I - MK^{-1}] \Delta K \qquad (5.20)$$

Terms due to bounded generations in the first term of the product are multiplied by zero in the second term of the product, so they disappear. Recalling the definition of the term K in eq. 4.18, two middle terms in eq. 5.20 cancel out. This simple formulation of bus incremental costs results:

$$BIC = \lambda^T [I - MK^{-1}] \Delta K \qquad (5.21)$$

Note that this expression is valid only when the load is greater than the threshold load. For smaller loads, K and M are undefined. From previous arguments, it is known that for loads between minimum load and threshold load, no free generation is being used.

The incremental bus cost for any particular load i is easily obtained:

$$BIC_i = \frac{\partial C}{\partial b_{1i}} = BIC . \frac{\partial b_1}{\partial b_{1i}} \qquad (5.22)$$

The second term of the product of eq. 5.22 is simply $e_i = [0,\dots,1,0,\dots,0]$ with the 1 in position i. Then the individual bus incremental costs are

$$BIC_i = \lambda^T [I - MK^{-1}] \Delta K_i \qquad (5.23)$$

The term ΔK_i is the i^{th} column of matrix ΔK. This is the simplest expression for bus incremental costs.

Finally, the dependence of λ on the loads is highlighted. The bus incremental costs are written

$$BIC(b_l) = [\lambda_0 + \Delta\lambda . b_l][I - MK^{-1}]\Delta K \quad (5.24.a)$$

$$= BIC_0 + \Delta BIC . b_l \quad (5.24.b)$$

b) The system incremental cost

In lossless real power dispatch, the system incremental cost (SIC) is the derivative of the optimal cost with respect to the total system real load. With a lossy load flow model, the concept remains valid only if a load trajectory is specified.

Before undertaking the analysis, real power loads are separated from reactive power loads. If the two are naturally partitioned in the b_l vector, expressions for the system real power load P_D and the reactive power load Q_D are given by

$$P_D = e_p^T b_l \quad (5.25.a)$$

$$Q_D = e_q^T b_l \quad (5.25.b)$$

with

$$e_p^T = [e_{dim(Pl)} \quad 0_{dim(Ql)}] \quad (5.25.c)$$

$$e_q^T = [0_{dim(Pl)} \quad e_{dim(Ql)}] \quad (5.25.d)$$

and e is the unit vector.

System loads could be expressed for other partitions of real and reactive loads by applying the appropriate e_p and e_q.

A load trajectory of the type eq. 3.6 is proposed. Then the system loads become scalar functions of the scalar θ, as shown below:

$$P_D = e_p^T[b_{10} + \Delta b_1\theta] \quad (5.26.a)$$

$$= P_{D0} + \Delta P_D\theta \quad (5.26.b)$$

and

$$Q_D = e_q^T[b_{10} + \Delta b_1\theta] \quad (5.27.a)$$

$$= Q_{D0} + \Delta Q_D\theta \quad (5.27.b)$$

The real power SIC, denoted SIC(P), is the derivative of the optimal cost with respect to the system real power load:

$$SIC(P) = \frac{\partial C}{\partial P_D} = \frac{\partial C}{\partial b_1^T} \cdot \frac{\partial b_1}{\partial P_D} \quad (5.28)$$

The second term on the right-hand-side is a vector of participation factors for the next increment of load. This is where the specified load trajectory is required. The values of the elements of P_1 are expressed as a function of the system load; the participations for real power loads in eq. 5.28 are positive and for reactive power loads are nil. The product of the two vectors in eq. 5.28 only affect real power terms. The formulation simplifies to the following:

$$SIC(P) = \frac{\partial C}{\partial P_D} = \frac{\partial C}{\partial P_1^T} \cdot \frac{\partial P_1}{\partial P_D} \quad (5.29.a)$$

$$= BIC_p . PF_p \quad (5.29.b)$$

The subscript p pertains to terms in real power load, and PF is a vector of participation factors, the second term of the product of eq. 5.28. For loads below the threshold load, the real power system incremental cost, as seen earlier, is identically zero.

A similar expression can be formed for the reactive power system incremental cost, from reactive load derivatives and participation factors:

$$SIC(Q) = BIC_q . PF_q \tag{5.30}$$

We stress the importance of knowing the load trajectory to compute the SIC. If no load trajectory is specified, P_D and Q_D are scalar functions of the vector b_1, and derivatives

$$\frac{\partial b_1}{\partial P_D} \quad \text{and} \quad \frac{\partial b_1}{\partial Q_D}$$

are undefined. That makes sense physically, since for a load flow model with lossy transmission, the addition of an increment of load solicits different responses, depending on the added load's location.

If the power factor of a load can be modeled as being constant, the reactive power load can be removed from the formulation, as a parameter. If that is the case for all loads, then the formulation reduces to one in real power loads only. That would reduce the dimension of the problem, but it would make it less sparse.

c) Discussion

A prime use of incremental bus costs is in determining the economic benefits in supplying load to a particular bus. They can be compared to the bus incremental revenues. Whereas the incremental costs are linked to the system load, the incremental revenues are linked by contract to the bus loading. Trajectories of incremental bus costs and incremental revenues can be compared, for a given load forecast, as in figure 5.1. From a utility's point of view, it is advantageous to supply load as long as the incremental cost of supplying is smaller or equal to the incremental revenue gained through the sale. The region to the left of point A on the graph is advantageous to the utility. The amount of profit per increment of load is the difference between the two curves. Regulations force the utility to supply load to a bus beyond point A if required. In that region, the utility is losing money on each increment of load supplied to that bus. That constitutes, willingly or not, a subsidy to the consumer. The amount of the subsidy per increment of load is again the difference between the two curves.

Recently much interest has been shown in adjusting the customer rates to the bus incremental cost, possibly in real time [Ponrajah & Galiana 1985, Allan & Peddle 1986, Ghoudjehbaklou 1986, Lescoeur & Galland 1986, Luo et.al.1986, Oyama 1986]. A major difficulty with such a scheme is that bus incremental costs depend as much on system loading as on bus loading. The customers at a bus are not entirely responsible for their bus incremental cost, so it could be unfair to change their rates based on bus incremental costs alone. The assessment of whether a bus rate should be changed would have to be made over a larger set of information.

The recognition of significant disparities between bus incremental costs and revenues over a wide range of loads would lead to a more equitable collection of revenues by the utilities. It is a complex economic issue, influenced by many conditions, including the one presented in the previous paragraph, the costs of alternate energy, decisions of regulatory agencies representing consumers, the need for some subsidies, and the consumer's willingness to pay more, if need be. Such a study is not undertaken in this thesis; it simply suggests the use of optimal bus incremental cost trajectories as a new tool. It could be useful in justifying a review of rates, or to find alternate revenues to compensate for subsidies. In operations planning, it could suggest load management strategies, or in transmission planning, the best locations for additional equipment.

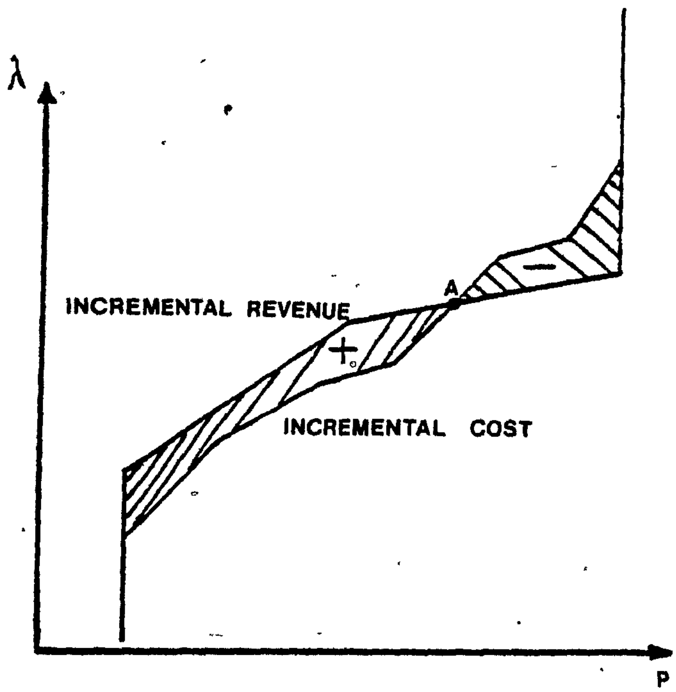


Figure 5.1. Bus incremental costs and revenues.

CHAPTER VI

DETAILS OF THE NUMERICAL IMPLEMENTATION OF AN ECONOMIC DISPATCH - OPF ALGORITHM

6.1 Introduction

An optimal power flow algorithm solving the economic dispatch task has been implemented on the computer, using the algorithm described in Chapter 3 and one of the subproblem solutions of Chapter 4. This chapter provides the details for many of the numerical solution procedures used in the program.

, Information of a general nature, presented in a first section, describes data structures and solvers used throughout the program. Then a detailed account of the computation for the economic dispatch subproblem is given. It is based on the continuation method using the varying limits strategy.

The real power dispatch algorithm presented in this chapter is of particular interest. It is a hybrid of the two most used techniques, one iterative, the other direct. Advantages from both techniques are incorporated into our algorithm, along with some new ideas for identifying active variables. Although it plays a pivotal role in the nonlinear OPF algorithm, this real power dispatch could be used on its own, in simpler dispatching schemes using a lossy linear network model.

Other solution procedures presented here, such as the computation of solution trajectories or the updates of optimality conditions, are specific to our OPF algorithm. Although important, the remaining sections are highly technical, and the reader might wish to skip them. These details are meant to document the program for future research.

The computations involving nonlinear equations will not be covered in this chapter. It is felt that previous descriptions of the Rules step and of the nonlinear solver in Chapter 3 are sufficient.

6.2 Some General Considerations in the Program Implementation

A few considerations, important in the general organization of the program, are presented in this section. Specifically, three topics are covered. The data structure for the program is described first. There follows a discussion on the linear equation solvers considered for various uses in the program. A short note on matrix - vector computations completes this section.

6.2.1 Synopsis of the Data Structures Used in the Program

In large problems, a well-planned scheme for storing data is essential for quick and easy retrieval. The data structure, as it is called, simplifies the retrieval of data by adding arrays of auxiliary information to the arrays of variables and parameters.

Basically two types of structures are used in the program: ordered lists and linked lists. The ordered list is a collection of values placed in a sequence; it is the most common form in general. In linked lists [Aho et.al. 1983], additional information vectors called pointers indicate where to locate in the list the next element having some given property. Some advantages of linked lists are that they avoid the reordering of lists with dynamic partitions or new entries, they allow easy access to elements sharing a common property, they avoid repeating common information, and they reduce the storage requirements of sparse arrays. For these reasons, much of the program's information is stored in linked lists.

A disadvantage of linked lists is the added memory requirements for the pointers, but that is considered a reasonable price to pay for the ease of handling variables. Usually pointers can be stored in the smallest available memory cells by assigning the proper variable type (for example, INTEGER*2 in FORTRAN). That makes the linking of data much more economical than repeating data, even zeros.

Three types of data are present in the program: system parameters, variables, and coefficients created by the computation.

The system parameters are stored in both ordered and linked lists. Bus data and data for passive network controls are placed in ordered lists. The line data is kept in linked lists, mostly to keep track of the adjacency structure of the network. It also allows to access line data from either end bus, without having to repeat the data.

The variables are stored in various linked lists, for different applications. A master linked list holds values of the variables, the Lagrange multipliers, the solution trajectories, and some status information in the subproblem. It communicates with two secondary linked lists with expansion point information, one on the states, the other on the injections and the load flow Jacobian. The nonzeros of the Jacobian are stored in compact form in a vector along with added information for easy retrieval. The two secondary lists are structured for easy use in the Newton-Raphson solver, and in particular in the linear equation solver. Ordered lists for each state variable are formed when needed, from the expansion point information, for use in computing the load flow equations and the Jacobian.

The third group of data consists of coefficients which go into the make-up of the optimality equations of the subproblem, eq. 4.9 - 4.10. Rows of sensitivity coefficients G_i are stored as columns for easy access in FORTRAN, in a dense matrix. Pointers in the master list link the independent variables to their column positions and the active dependent variables to their row positions. Only the pointers are rearranged, thereby avoiding the shuffling of rows or columns of values following the removal of constraints from the active set. This also allows to keep the sensitivity coefficients for the deactivated dependent variables. They are set aside simply by changing the status of their pointers, and can just as easily be reactivated. Right-hand-side coefficients are stored in another ordered list. All the moving limits created by the continuation process are placed in an ordered list and accessed from the master list when needed.

6.2.2 Linear Equation Solvers Considered and Used in the Program

The linear equation solvers are the basic building blocks of the program. Both sparse and dense matrix solvers have been implemented for use in different computations:

- Computations involving the load flow Jacobian require the sparse matrix solver. It is found in the Newton-Raphson solver, and in the computation of sensitivity coefficients and dependent variables, before and during the continuation process.

- The optimality equations of the subproblem are sparse, but because of their structure, their solution is organized to use the dense matrix solver.

Options available for sparse matrix computations fall into two categories. The indirect (or iterative) methods such as conjugate gradient [Evans 1985] or Lanczos [Golub & van Loan 1983] methods are most useful for very large, very sparse systems of equations. Theoretically, they require N iterations to solve an N-dimensional system of equations, but ill-conditioned systems can take many more iterations. Direct methods [Duff 1984] reorder the equations and variables to reduce the extra computational burden due to fill-in, followed by the LU decomposition of the reordered matrix, and the solution by forward and backward substitution. Direct methods are more useful in applications with a single matrix and many right-hand-side vectors, because the reordering and the factorization remain unchanged from one problem to another. The remaining operations in the solution process represent a small portion of the computation.

Ordering schemes for direct methods also fall into two broad categories. Sparse-matrix methods, popular since the 1950's, act only on the matrix structure, neglecting the structure of the right-hand-side vector in the ordering strategy. Some of the more important ordering schemes in this group can be found in the following references [Markowitz 1957, Tinney & Walker 1967, Hellerman & Rarick 1971, Duff 1977, George & Liu 1981, Pissanetsky 1984]. Recently developed sparse-vector methods [Tinney et.al. 1985, Gomez &

Franquelo 1988] show that great advantage can be taken of very sparse right-hand-side vectors.

Five sparse matrix solvers were available to us. Because of the general nature of our matrices however, three of them had to be rejected. These were the solvers SPARSPAK [George and Liu 1981] and the Yale University package, which use direct methods, and the conjugate gradient package PCGSOL. The first two, designed to exploit symmetry, cannot handle zero-valued diagonal elements. The third only handles symmetric positive-definite matrices. The other two solvers were tried in our program. The Lanczos-type iterative solver LSQR [Paige & Saunders 1982] was tried first. It was very reliable and user-friendly, but for the problems we tested it turned out to be very slow. Our choice for the sparse matrix solver settled on the well-known Harwell library subroutine package MA28 [Hopper 1977]. It was much faster than LSQR, even as much as an order of magnitude faster in the larger tests. Furthermore, its use of a direct method made the OPF algorithm faster overall, because of the repeated use of the fixed Jacobian matrix in the linear equation solutions of the subproblem.

Sparse-vector solvers are not readily available, but if one could be found or written, it is the best suited for computing sensitivity coefficients. In those computations, a sparse row of the Jacobian is used as the right-hand-side of the required linear equations. In particular, this technique simplifies the computation of the coefficients of the generalized power balance equation. To achieve the greatest simplification in this case, the manifold variable should be chosen as the real power generation at the most isolated bus. It has the least number of elements in its row of the Jacobian.

Dense matrix solvers are used in computing optimal solutions in the subproblem, eq. 4.27-4.37 for economic dispatch. The optimality equation from which they were derived however, eq. 4.6, is large and sparse. It has a single bordered block structure, with a diagonal main submatrix B; that already makes it optimally ordered. Its reduction to a set of small dense matrix equations, as in our solution, is a standard procedure in this case [ch.6 of George & Liu 1981]. As seen in eq. 4.12 - 4.14, this procedure is equivalent to sparse Gaussian elimination without partial pivoting.

Options available for dense matrix computations are numerous. The reader is referred to [Stewart 1973, Golub & Van Loan 1984] for descriptions of the upcoming matrix properties and solution techniques. A Gaussian elimination algorithm is used in one instance for the solution of a general matrix equation. However, most applications in the program involve symmetric positive definite (SPD) matrices. In particular, the reduced Hessian terms K and L of eq. 4.18-4.19 are SPD and are often involved. Algorithms solving for this type of matrix need only store the diagonal elements and half of the nondiagonal elements, avoiding their repetition. The Cholesky factorization, used on SPD matrices, produces symmetric factors, so that again, about half the computation and half the storage space are saved. Following changes in the active set, techniques are available to update some factorizations, rather than to restart them. The updates are preferred because they are much faster. The Cholesky factorization and its updates are used for most of the dense linear equation solutions.

The reduced Hessian terms K and L are inherently prone to ill-conditioning, occasionally rendering the Cholesky factorization useless. A back-up factorization, more robust than the Cholesky factorization but requiring more computation, is available for those cases. It is based on the QR decomposition, and is described later in this chapter.

An added precaution against ill-conditioning was added in the form of an iterative refinement algorithm. It is called after a solution of an ill-conditioned system, when the residual of the system of equations is larger than a certain tolerance. In practice it should rarely be called.

The Gaussian elimination, the Cholesky factorization, and the QR decomposition are taken from LINPACK, a general package of linear equation solvers [Dongara et.al. 1979]. Some of the updates are based on LINPACK subroutines, while others are built from scratch. Details of the updates of the optimality conditions and some of the other computations described in this section are presented in section 6.3.

6.2.3 Matrix-Vector Products

Chains of matrices and vectors are multiplied in various parts of the computation. The operations implicating products of a matrix by a vector are always computed first. The result being a vector, that greatly reduces the dimensions and the computational burden of subsequent products.

6.3 Implementation of the Solution Algorithm for the Economic Dispatch Subproblem using the Varying Limits Strategy

The remainder of the chapter presents the specific solution procedures used in the program. The algorithm of section 3.4.2.3 can serve as a general guide to the procedure. First the initial, simple problem is solved here in three steps: an initialization step, a real power dispatch, and the computation of dependent variables. Then the continuation process, steps 3.a to 10 of the forementioned algorithm, is explained in detail.

6.3.1. Starting the subproblem: computing sensitivity coefficients, checking for feasibility, and setting transparent variables

Data required for the initial simple problem are the cost data and the coefficients of the generalized power balance equation. The latter are computed using equations eq. A3.4.3 and A3.4.4 b. and e., given in Appendix 3.3. Referring to that Appendix for nomenclature, the computational procedure is as follows:

Solve for α: $J_{bd}{}^{T}\alpha = J_{md}{}^{T}$ (6.1)

Compute β: $\beta = J_{mb}{}^{T} - \alpha^{T}J_{bb}$ (6.2)

Combine: $g_0{}^{T} = [\alpha^{T}\ \beta^{T}]$ (6.3)

Here, and in subsequent computations, α and β are auxiliary vectors, used as workspace arrays.

The injections are placed before the states in the vector of independent variables, so that g_0 need not be reordered. All sensitivity coefficients are computed in this manner, with the appropriate row of J_d replacing J_m.

The first step of the optimization is to check for the feasibility of the initial simple problem over the set of independent variables b_g. To do this, compute $[g_0^T b]^m$ and $[g_0^T b]^M$ as follows:

$g_{0i}b_i^m$ results from setting b_{gi} to its lower bound if $g_{0i} > 0$, or results from setting b_{gi} to its upper bound if $g_{0i} < 0$.

$g_{0i}b_i^M$ results from setting b_{gi} to its upper bound if $g_{0i} > 0$, or results from setting b_{gi} to its lower bound if $g_{0i} < 0$.

$$\text{Then } [g_0^T b]^m = \sum_i g_{0i} b_i^m \tag{6.4}$$

$$\text{and } [g_0^T b]^M = \sum_i g_{0i} b_i^M \tag{6.5}$$

If the former is non-positive and the latter is non-negative, then the generalized power balance equation is sure to contain at least one feasible point inside the constraint box.

If the feasibility check fails, a new subproblem must be submitted. This procedure is necessary if the subproblem is to be used on its own, or for a first pass in the nonlinear OPF problem. With Rules assuring that subsequent expansion points are feasible in the nonlinear problem, this procedure need not be repeated in subsequent iterations.

Having passed the feasibility check, the next step is to send transparent variables to the appropriate values. Assume for now that the aggregate load is greater than the threshold load. Then the transparent variables are sent to the appropriate bounds, according to conditions 4.53 and 4.54 of Chapter 4. At the same time, the right-hand-side coefficient $\hat{k}$ is computed from eq. 4.48 (and 4.5). Transparent variables having zero sensitivity coefficients are left at the values of the expansion point and auxiliary bounds are set (see section 3.4.4.5) to ensure convergence of the nonlinear problem.

The next step is to compare the aggregate load to the threshold load. If the aggrgate load is larger than the threshold load, the algorithm can proceed to the computation of the real power dispatch. If the aggregate load is smaller than the threshold load, all real power generations are set to their minima and the following simple procedure is applied to find one optimal solution (amongst many):

- Free one transparent variable at a time. If an adjustment of that variable can satisfy the generalized power balance equation, then an optimal solution is found. Stop.

- If not the variable goes to the opposite bound. With this change in the active set the right-hand term $\hat{k}$ and the active/inactive partition are adjusted.

- Repeat with the next transparent variable. Continue until the generalized power balance equation is satisfied or until all transparent variables have switched bounds.

If the power balance equation can be satisfied, the algorithm proceeds to the continuation process. If not, the aggregate load is too small to be satisfied by a feasible operating point. Then a control or an error message would be called from outside the simple problem.

The next step, the real power dispatch, is more complex and will be treated on its own in the upcoming section.

6.3.2. Algorithm for the real power dispatch

An algorithm is presented for the solution of the real power economic dispatch problem (ed,P), first described in Chapter 4. It comes about after the imposition of values on the transparent variables in the initial, simple problem. It is in the form of a standard real power dispatch.

The solutions of the optimality conditions for this problem are given by eq. 4.50 for the Lagrange multiplier λ_0, and eq. 4.51 for real power generations, but they are only optimal once the proper active set is found. Finding the optimal active set is the most time-consuming part of the optimization. Two solution procedures are commonly used:

- In the direct approach, it is assumed at the outset that all generations are free. Both eq. 4.50 - 4.51 are solved. If in the solution some generations violate their bounds, usually the first to be violated is set to its bound and its Lagrange multiplier is activated. The process is repeated with the new active set. This continues until solutions for free generations from eq. 4.51 are completely feasible. That is the optimal solution.

- In the iterative approach, often called lambda dispatch [Wood & Wollenberg 1984], upper and lower bounds are computed for the Lagrange multiplier λ_0. A feasible value of λ_0 is proposed, instead of being computed in eq. 4.50, and inserted into eq. 4.51. If some of the resulting generations violate their bounds, they are set temporarily to their bounds. Then the desired system load is compared to the computed system generation:

  i) If the load is larger than the generation, then λ_0 must be increased.
  ii) If the load is smaller than the generation, then λ_0 must be decreased.
  iii) If the two are equal (within a tolerance), then the computed generations are optimal.

  Changes in λ_0 are made by binary search within its feasible region. The process is repeated until it converges in option iii).

The algorithm proposed here uses the iterative approach only until the proper active set is identified. Since each iteration is relatively simple, it is felt that the iterative approach arrives at the active set more quickly than the direct approach. Then with the proper active set, a single application of the direct method yields an optimal solution. That avoids having to home in on the solution.

The proper active set is easily identified in applications with monotonically increasing cost functions. If in two consecutive iterations A and B, the search direction for the Lagrange multiplier λ_0 changes but the proposed active set remains the same, then the active set remains constant everywhere within the interval $[\lambda_{0A}, \lambda_{0B}]$. In particular, it is the active set for the optimal λ_0, which must be situated in that interval.

The notions of system load and system generation are respectively replaced in this application by aggregate load, $g_0^T b_1$, and aggregate generation, $g_0^T(b_g + b_e)$. The only variables left to determine, then, are the real power generations.

Another refinement implemented in the iterative part of the algorithm, and not found in standard lambda dispatch algorithms, is the identification of active generations as the algorithm progresses. This information reduces the dimension of the search. Generations on their lower bounds when λ_0 decreases are sure to remain there, as is the case for upper bound generations when λ_0 increases.

Assuming that the transparent variables have been determined, and that the aggregate load is greater than the threshold load, here then is the algorithm for the solution of the real power economic dispatch problem (ed,P):

STEP 1. Place all generations in the free partition.

STEP 2. Compute the right-hand-side term $\hat{k}$ of eq. 4.48.

$$\hat{k} = g_0^T(b_1 + b_e) - [g_{0t}^T t_g]^M \qquad \text{(bis 4.48)}$$

STEP 3. Determine lower and upper bounds on λ_0 from cost data.

i) For all generations, compute

$$\lambda_{0i}^m = a_i + B_{ii} P_{gi}^m \qquad (6.6)$$

and

$$\lambda_{0i}^M = a_i + B_{ii} P_{gi}^M \qquad (6.7)$$

ii) Then pick out the extreme values as bounds on λ_0.

$$\lambda_0^m = \min_i \{\lambda_{0i}^m\} \qquad (6.8)$$

and

$$\lambda_0^M = \max_i \{\lambda_{0i}^M\} \qquad (6.9)$$

STEP 4. Set the initial λ_0 to the median value in the interval $[\lambda_0^m, \lambda_0^M]$.

STEP 5. If the active set has not changed over two consecutive iterations and the search direction for λ_0 has changed, then the optimal active set has been found. Go to STEP 11.

STEP 6. Compute generations for the given λ_0.

From eq. 4.51,

$$P_{gi} = (g_{0i}\lambda_0 - a_i)/B_{ii} \quad \text{for free generations only.} \qquad (6.10)$$

If P_{gi} violates a bound, set it to that bound and place the index i in a corresponding list of temporary upper or lower bound generations.

STEP 7. Compute the weighted sum of generations.

$$S = g_{0p}^T P_g \qquad (6.11)$$

STEP 8. Compare S to $\hat{k}$.

i) If S is less than $\hat{k}$, then go to STEP 9.
ii) If S is greater than $\hat{k}$, then go to STEP 10.
iii) If S equals $\hat{k}$ (within a tolerance), then go to STEP 12.

STEP 9. Update λ_0, $\hat{k}$ and active/inactive partitions on generations.

i) Update lower bound on λ_0: $\lambda_0^m = \lambda_0$;
update $\lambda_0 = (\lambda_0^m + \lambda_0^M)/2$.

ii) Update the active/inactive partition on generations by including indices from the temporary upper bound list in the permanent upper bound list.

iii) Update $\hat{k}$: $\hat{k} = \hat{k} - g_{0i}P_{gi}$ for all newly activated P_{gi}.

iv) If control comes from STEP 12, go to STEP 10 ii). If not, go to STEP 5.

STEP 10. Update λ_0, $\hat{k}$ and active/inactive partitions on generations.

i) Update upper bound on λ_0: $\lambda_0^M = \lambda_0$; update $\lambda_0 = (\lambda_0^m + \lambda_0^M)/2$.

ii) Update the active/inactive partition on generations by including indices from the temporary lower bound list in the permanent lower bound list.

iii) Update $\hat{k}$: $\hat{k} = \hat{k} - g_{0i}P_{gi}$ for all newly activated P_{gi}.

iv) If control comes from STEP 12, STOP. If not, go to STEP 5.

STEP 11. The optimal active set being known, compute solutions from eq. 4.50 -4.51. STOP.

STEP 12. The optimum has been reached iteratively. Update the active set and the right-hand-term one last time. Go to STEP 9 ii).

The algorithm terminates in STEP 11 or converges after STEP 12.

A possible improvement would be the implementation of a secant search technique [Dahlquist & Björck 1974] instead of a binary search, in updating λ_0, as described in [Wood & Wollenberg 1984]. This is suggested for future implementation.

6.3.3 Computing the dependent variables

The free dependent variables are computed from the newly determined values of the independent variables. That is done using eq. A3.4.3 and A3.4.4 c. and f., from Appendix 3.4. The dependent variables are computed at various stages: before the continuation procedure with all dependent variables considered inactive, and in the continuation procedure in computing coefficients of the solution trajectories. Depending on the particular classification of each variable, there are different computation procedures. This can be seen in the computational procedure presented a little further.

The computation for the components of the solution trajectories, of the form $s(\theta) = s_0 + \Delta s.\theta$ (eq. 4.38), is performed in two parts. The procedure to compute s_0 serves also to compute the values of the dependent variables preceding the continuation process. The two parts of the computation are given below, side by side. Those steps which are common to both components are written only once.

Operation	Δs component	s_0 component	
Form α:			
- If x_b is inactive	$\alpha = \Delta x_{bg}$	$\alpha = x_{bg0} - x_{be}$	(6.12)
- If x_b is on a fixed bound, $x_{bg}{}^b$	$\alpha = 0$	$\alpha = x_{bg}{}^b - x_{be}$	(6.13)
- If x_b is on a moving bound $d_0 + \Delta d$	$\alpha = \Delta d$	$\alpha = d_0 - x_{be}$	(6.14)
Compute β:	$\beta = J_{db}\alpha$		(6.15)
Form α:	$\alpha = y_{bg} - (y_{bl} + y_{be}) - \beta$		(6.16)
Solve for Δx_d:	$J_{bd}\Delta x_d = \alpha$		(6.17)
For inactive states only, compute:	$x_{di} = \Delta x_{di}$	$x_{di} = x_{dei} + \Delta x_{di}$	(6.18)
(For active states	$x_{di} = 0$	$x_{di} = x_{di}{}^b$)	
For each inactive dependent injection y_{di}:			
Compute α_i:	$\alpha_i = J_{dd(i)}x_d$		(6.19)
Compute y_{di}:	$y_{d(i)} = \alpha_i + J_{db(i)}x_b$	$y_{d(i)} = \alpha_i + J_{db(i)}x_b + x_{de}$	(6.20)

This format for presenting the computation is adopted for all the upcoming computations involving two components.

6.3.4. The continuation procedure

The details of the computation of the continuation procedure are presented in this section. The process is made up of five steps: shifting violated constraints, updating the optimality conditions, computing solution trajectories, determining the next breakpoint, and computing the solution at the breakpoint. The updates to the optimality conditions are considered an important feature of the algorithm, so they will be presented in detail. Another feature in this implementation is a test for resolving certain forms of degeneracy; it is also presented, in a final subsection.

a) Computing violations on dependent variables

The values of dependent variables computed from the optimal solution of the initial, simple problem are compared to their bounds. If violations are found, the continuation process is invoked.

A violation counter is initialized to zero. Then, for each dependent variable d_i:

- If $d_i > d_i^M$, then place index i in the set of upper violated constraints. Compute the violation $(d_i - d_i^M)$.
- If $d_i < d_i^m$, then place index i in the set of lower violated constraints. Compute the violation $(d_i^m - d_i)$.
- For either violation, increment the violation counter. If the violation is the largest seen so far, record the index i and the value of the violation.

Once all the dependent variables are processed, if the violation counter indicates zero, then the solution to the initial, simple problem is also the optimal solution to the economic dispatch subproblem. Then the continuation process and the subproblem are complete. If the violation counter indicates a positive value, then create perturbation functions for all violated constraints, by shifting all the violated constraint bounds by the amount of the largest violation Δd:

- For upper bound violations, $d_i^M(\theta) = (d_i^M + \Delta d) - \Delta d.\theta$ (6.21)
- For lower bound violations, $d_i^m(\theta) = (d_i^m - \Delta d) + \Delta d.\theta$ (6.22)

All the dependent variables are then inactive, except one which is "just active". Its Lagrange multiplier is activated, and its value is zero.

Before proceeding with the updating step, the Cholesky factorization of the K matrix of eq. 4.18 is computed. That is very easy because at this point K is a scalar, whose factorization is simply its square root. The notation U_k is used to denote the upper triangular factor of matrix K:

$$U_k = [g_0^T B^{-1} g_0]^{1/2} \qquad (6.23)$$

The index of the newly activated constraint is retained and update no.1 is invoked.

b) The six updates

These updates reform the components of the calculation - Cholesky factorizations, right-hand side vectors, partitions of vectors and their dimensions - following the updating of the optimality conditions at a breakpoint.

An advantage of the computational scheme in general, and the updating scheme in particular, is that at the beginning of the continuation process left-hand-side matrices K and L are actually scalars. Then at each update of the optimality conditions, the dimensions of K or L either remain unchanged or are modified by one. Quick updates of the Cholesky factorizations of K are available for all update conditions. Quick updates for the Cholesky factorization of L are available only in cases involving transparent variables. Normally the dimensions of these matrices should remain small, making the computation relatively fast.

Here then are the detailed computational procedures for the six update conditions and their factorizations.

i) Update no.1

This updates the optimality conditions when a functional constraint d_i becomes active. An extra row and column of sensitivity coefficients are added to the optimality equation, eq. 4.4 . The computational procedure is as follows:

- Update the active/inactive partition on dependent variables.
- Update dimensions of these partitions.
- Check for structural degeneracy (active dependent constraints outnumber inactive real power generations). This will be explained further.
- Compute (or retrieve, if available) the sensitivity coefficient vector g_i for the active constraint as described in section 6.3.1.
- Compute the corresponding right-hand-side term:

$$k_i = g_i^T[b_l + b_e] - g_i^{bT} b_g^b \tag{6.24}$$

The second term of the right-hand-side is computed using the present active/inactive partition of independent variables.

- Check for numerical degeneracy.
- Update Cholesky factorizations U_K and U_L of matrices K and L respectively (eq. 4.18 and 4.19).

For U_K, a new column u and then a new row are concatenated along the right/bottom edges of the existing U_K. The new bottom row has a single nonzero element v in the last column, so that the new U_K remains upper triangular. The procedure for computing u and v is as follows:

- For all inactive generations
  j, compute α_j,: $\alpha_j = g_{ij}/B_{jj}$ (6.25)

- Compute β: $\beta = G_p\alpha$ (6.26)
  - Solve for u: $U_K^T u = \beta$ (6.27)
  - Compute v: $v = g_{pi}^T\alpha - u^T u$ (6.27)
    If v > 0, then $v = \sqrt{v}$ (6.28)
    < 0, then resort to the back-up decomposition.
  - Update the G matrix to include the g_i vector.

The matrix U_L is recomputed from scratch.

ii) Update no.2

This updates the optimality conditions when a functional constraint d_i becomes inactive. An existing row and column of the optimality equation is deleted. The computational procedure is as follows:

- Update the active/inactive partition for dependent variables.
- Find the position of the sensitivity coefficient vector g_i to be removed. Remove its position index from the ordered list of sensitivity vector positions (but keep g_i and information to access it).
- Remove the corresponding right-hand-side element.
- Update the Cholesky factorizations of U_K and U_L:

For U_K, use LINPACK subroutine DCHEX. It updates the Cholesky factorization of matrix K with the row and column corresponding to the deactivated constraint permuted to the bottom/right edge of the matrix. Then update the dimensions of the active/inactive partition on dependent variables. The last row and column of U_K are then discarded.

The matrix U_L is recomputed from scratch.

iii) Update no.3

This updates the optimality conditions when a transparent variable t_j becomes active. A row and a column of the optimality equation are transferred from the left-hand-side to the right-hand-side. The computational procedure is as follows:

- Update the active/inactive partition for transparent variables.
- Extract the column vector g_j from G_t corresponding to transparent variable t_j.
- Update the right-hand-side term:

$$k = k - g_j t_{gj}^{lim} \qquad (6.29)$$

- Update the Cholesky factorizations of U_K and U_L:

U_K remains unchanged.

For U_L, if there remain free transparent variables, the LINPACK subroutine DCHEX is used as above, to remove the row/column corresponding to t_j from matrix L. The dimensions of the active/inactive partitions for transparent variables are then updated, and again the last row/column of U_L are discarded.

iv) Update no.4

This updates the optimality conditions when a transparent variable t_j becomes inactive. A row and a column of the optimality equation are transferred from the right-hand-side to the left-hand-side. The computational procedure is as follows:

- Update the active/inactive partition of the transparent variables.
- Update the dimensions of the partitions.
- Extract the column vector g_j from G_t corresponding to transparent variable t_j.

- Update the right-hand-side term:

$$k = k + g_j t_{gj}^{lim} \qquad (6.30)$$

- Update the Cholesky factorizations of U_K and U_L:

U_K remains unchanged.

For U_L, the update procedure is similar to that in Update no.1, in that a new row and a new column are added to the existing U_L.

- Compute v:
  - Solve for α: $K\alpha = g_j$ (6.31)
  - Compute v: $v = g_j^T \alpha$ (6.32)
- If the number of free transparents is greater than 1,
  - Compute β: $\beta = G_t^T \alpha$ (6.33)
  - Solve for u: $U_L^T u = \beta$ (6.34)
  - Compute v: $v = v - u^T u$ (6.35)
- Check v:
  - If $v > 0$, then $v = \sqrt{v}$ (6.36)
  - < 0, then resort to the back-up decomposition.

v) Update no.5

This updates the optimality conditions when a real power generation P_{gj} becomes active. A row and a column of the optimality equation are transferred from the right-hand-side to the left-hand-side. The computational procedure is as follows:

- Update active/inactive partitions for real power generations.
- Update the dimensions of the partitions.
- Check for structural degeneracy.

- Extract the column vector g_j from G_p corresponding to real power generation P_{gj}.
- Update the right-hand-side term:

$$k = k - g_j P_{gj}^{lim} \tag{6.37}$$

- Update the Cholesky factorizations of U_K and U_L:

For U_K, first the down-dated row/column $\alpha = B^{-1/2} g_j$ is computed:

For all elements of column g_j,
Compute α_i: $\alpha_i = g_{ij}/\sqrt{B_{jj}}$ (6.38)

Then LINPACK subroutine DCHDD down-dates the existing factorization to obtain the new one:

$$U_K^{new} = \text{factorization of } [U_K - \alpha\alpha^T]. \tag{6.39}$$

The matrix U_L is recomputed from scratch.

vi) Update no.6

This updates the optimality conditions when a real power generation P_{gj} becomes inactive. A row and a column of the optimality equation are transferred from the left-hand-side to the right-hand-side. The procedure is as follows:

- Update active/inactive partitions for real power generations.
- Update the dimensions of the partitions.
- Extract the column vector g_j from G_p corresponding to real power generation P_{gj}.
- Update the right-hand-side term:

$$k = k + g_j P_{gj}^{lim} \tag{6.40}$$

- Update the Cholesky factorizations of U_K and U_L:

For U_K, first create the updated row/column α as in Update no.5. Then LINPACK subroutine DCHUD updates the existing factorization to obtain the new one:

$$U_K^{new} = \text{factorization of } [U_K + \alpha\alpha^T]. \qquad (6.41)$$

- The matrix U_L is recomputed from scratch.

vii) Factorization of L from scratch

Updates of U_L are only possible when the change in the optimality conditions leaves K unchanged, since L requires the inverse of matrix K. Most often it is necessary to recompute the factorization U_L of the matrix L from scratch. Once the matrix L has been recomputed, its factorization is performed using the LINPACK subroutine DCHDC.

viii) The back-up factorization

The usual Cholesky factorization can break down due to ill-conditioning. It manifests itself in a negative value of the square of the new diagonal element v, when a row and a column are added to matrix K or L. Either K or L can be written in the form of the normal equations i.e., in the form A^TA:

$$K = [G_p B^{-1/2}]\ [B^{-1/2} G_p^T] = A_K^T A_K \qquad (6.42)$$

$$L = [G_t U_K^{-1}]\ [U_K^{-1} G_t^T] = A_L^T A_L \qquad (6.43)$$

The QR decomposition applied to matrix A expresses A as a product of matrices: A = QR. It can be shown [George & Ng 1986] that the upper triangular portion of the R matrix is the Cholesky factorization of A^TA. Hence in our problem the Cholesky factorizations of K and L can be built from the matrices A_k and A_l, without forming the normal equations. The LINPACK subroutine DQRDC is used to produce this R matrix.

The QR technique is more robust than the usual Cholesky factorization algorithm. It is best left as a backup, though, since it requires twice as much computation as the usual Cholesky technique and since the Cholesky technique should rarely bog down.

c) Computation of solution trajectories

The computation of solution trajectories for economic dispatch is presented. The computation differs from the analytical expressions, eq. 4.31 - 4.38, in that values already computed for some variables serve to compute values for the next variables. That saves much computational effort. The two components of the solution trajectories are given, in the format first used in presenting the computation of dependent constraints. The computational procedure is as follows.

Eq. 4.32. If $\dim(t^I) = ntf > 0$, then compute free transparent variables.

Operation	Δs component	s_0 component	
Solve for β:	$U_K^T\beta = \Delta n$	$U_K^T\beta = n_0$	(6.44)
Solve for α:	$U_K\alpha = \beta$		(6.45)
Solve for β:	$U_L^T\beta = \alpha$		(6.46)
Solve for t_g:	$U_L t_g = \beta$		(6.47)

Eq. 4.35. Compute Lagrange multipliers of active functional constraints.

Operation	Δs component	s_0 component	
Set α:	$\alpha = \Delta n$	$\alpha = n_0$	(6.48)
If ntf > 0, compute α:	$\alpha = \alpha + G_t t_g$		(6.49)
Solve for β:	$U_K^T\beta = \alpha$		(6.50)
Solve for λ:	$U_K\lambda = \beta$		(6.51)

Eq. 4.31. Compute real power generations.

Operation	Δs component	s_0 component	
Compute α:	$\alpha = G_p{}^T\lambda$	$\alpha = G_p{}^T\lambda - a$	(6.52)
For all inactive real power generations, compute P_{gi}:	$P_{gi} = \alpha_i/B_{ii}$		(6.53)

Eq. 4.36 - 4.37. Compute Lagrange multipliers for active real power generations and active transparent variables.

Operation	Δs component	s_0 component	
Compute μ_p:	$\mu_p = -G_p{}^{bT}\lambda$	$\mu_p = B^bP_g{}^b + a^b - G_p{}^{bT}\lambda$	(6.54)
Compute μ_t:	$\mu_t = -G_t{}^{bT}\lambda$		(6.55)

Eq. 4.38 for dependent variables has already been treated in section 6.3.3. The remaining eq. 4.33 - 4.34 describe variables fixed at their bounds, which need not be computed.

The s_0 components for the upcoming segment of the solution trajectory are computed explicitly, using the procedure above, only when starting the continuation process ($\theta = 0$) or in restarting it after resolving a degeneracy. In most cases, it can be computed very simply from the newly computed Δs vector and the vector of known values of the variables, s_I, at the latest breakpoint θ_I.

Since $s_0 + \Delta s.\theta_I = s_I$ (6.56)

at the initial breakpoint of the next segment of the solution trajectory,

then $s_0 = s_I - \Delta s.\theta_I$. (6.57)

Hence a simple vector subtraction suffices to obtain s_0.

The precision of this quick computation of s_0 was compared to that of the drawn-out computation. It was feared that the quick computation might drift, due to the accumulation of numerical errors, as in some simple schemes for the computation of numerical integration [Dahlquist & Björk 1974]. However, tests indicate that the precision of the quick computation is very good.

d) Determining the next breakpoint

The optimality conditions remain valid until the next breakpoint, where either an inactive variable becomes active or Lagrange multiplier reduces to zero. The values of θ at which these things occur for each variable are computed as follows:

- For an inactive variable s_i going to a fixed bound:

  If $\Delta s > 0$, compute θ_i needed to reach the upper bound s_i^M,

  $$\theta_i = (s_i^M - s_{i0})/\Delta s. \quad (6.58)$$

  If $\Delta s < 0$, compute θ_i needed to reach the lower bound s_i^m,

  $$\theta_i = (s_{im} - s_{i0})/\Delta s. \quad (6.59)$$

  If $\Delta s = 0$, set θ_i greater than 1.

- For an inactive variable s_i going to a moving bound $d^{lim}(\theta) = d_0 + \Delta d\theta$ (possible for dependent and sometimes transparent variables):

  If $\Delta s > 0$, compute θ_i needed to reach the moving upper bound,

  $$\theta_i = (d_0 - s_{i0})/(\Delta s - \Delta d). \quad (6.60)$$

  If $\Delta s < 0$, compute θ_i needed to reach the moving lower bound,

  $$\theta_i = (d_0 - s_{i0})/(\Delta s - \Delta d). \quad (6.61)$$

  If $\Delta s = 0$, set θ_i greater than 1.

- For Lagrange multipliers $\Lambda(\theta) = \Lambda_0 + \Delta\Lambda.\theta$:

  If $\Delta\Lambda < 0$, compute the θ_i needed to reach zero,

  $$\theta_i = -\Lambda_0/\Delta\Lambda. \quad (6.62)$$

  If $\Delta\Lambda \nless 0$, set θ_i greater than 1.

Compare all θ_i, and pick the smallest as the next breakpoint θ_I. If that value is larger than one, then set it to one. If it is less than one, record the condition which causes θ_I and invoke its update condition. Then go on to compute the values of the variables $s(\theta_I)$.

e) Computing the solution at the breakpoint

The values of the inactive variables and Lagrange multipliers for active variables at the newly determined breakpoint θ_I are computed using eq. 6.56. If in the previous step, the breakpoint was set to one, then the values just computed are the solutions of the subproblem. If not, the continuation process returns to the updates.

f) Resolving degeneracy for three cases

Provisions have been implemented in the program to avoid certain cases of degeneracy. A first case is that of structural degeneracy, when in the continuation process, the active functional constraints come to outnumber the inactive generations. When that happens, the reduced Hessian term K of eq. 4.18 is a singular matrix, and subsequent calculations bog down. The problem is numerical, in that the forming of K imposes a certain block ordering of the left-hand-side matrix of the optimality equation, as shown in eq. 4.14. That matrix is nonsingular, but the imposed ordering makes the proposed solution equations, eq. 4.15 - 4.26, impossible to compute. This problem has been observed in numerical testing, but it rarely occurs once the appropriate basis is chosen.

A remedy to this problem is to reform the basis. It requires the application of the projection step of section 3.4.1, to reduce the number of active dependent variables, and the reordering of the Jacobian. Coefficients for a new generalized power balance equation and for the few remaining active dependent variables must be computed. This is just a restructuring of the computation, and does not invalidate the portions of the solution trajectories already found. The computation with the new basis picks up where the previous one left off.

The other two cases implemented in the program were described in Appendix 4.2. When the constraints outnumber the inactive variables, the reordering described above is insufficient. Constrained variables must be found to come off their bounds, while satisfying the optimality conditions. If no such variable can be found, a feasibility boundary has been reached. The simplest test is to check if there remain real power generations at their lower bounds. If so, the incrementally cheapest generation is taken off its bound and its Lagrange multiplier is set to zero. Then this solution is verified for optimality.

The other case resolves the degeneracy due to one constraint too many. It was described at length in Appendix 4.2, so it will not be repeated here.

CHAPTER VII

DESCRIPTION AND ANALYSIS OF THE NUMERICAL SIMULATIONS

7.1 Introduction

An optimal power flow program has been written implementing the ideas of the previous chapters for the economic dispatch task, and has been tested on systems of 6, 10, 30 and 118 buses. This chapter documents and analyzes the results, taken not only from the output, but also from the various important stages of the calculation.

The results are presented for each test system separately. The formats and the contents of the various tables and graphs will be discussed in detail only for the first test, on the 6 bus system. By then the reader should be well-acquainted with the format. Hence the results for the three subsequent tests are presented using the same format, but only the points deemed important are highlighted in the discussion.

The format for presenting the results in this chapter is as follows. First the global performance of the algorithm in solving for the initial load is given in detail. Then, in the first three simulations, the OPF is solved for a sequence of loads in a load-tracking scheme, and solution results for the subsequent loads are provided. Although less detailed than for the initial solution, the latter results provide for a good comparison of the computational effort and the computation time of the algorithm for the sequence of loads. Following this global description, the details of the system variables and costs through the various stages of the computation are presented graphically.

For the six bus system only, two other topics are added to this study. These verify a couple of fine points in the theory of the previous chapters. First we analyze the numerical stability of the subproblem solutions for the transparent variables, and their relation to the sensitivity coefficients in the power balance equation. Secondly, we look at the case where the computation of the first load continues until no breakpoint occurs in a long sequence of subproblems, and where the system is solved to extremely tight tolerances. The computational effort for the solution of the subsequent loads is then monitored to see if the initial effort was worthwhile.

Following the descriptive sections mentioned above, some general results and observations are regrouped. Here we summarize our numerical experience with our OPF program, and discuss its general behaviour. Those readers who wish to skip the details can move ahead to this section 7.6.

Results for the 30 bus system are then compared to those documented in a recent publication [Ponrajah 1987] and to those in a well-known paper in the OPF literature [Alsac & Stott 1974]. This comparison confirms some of the conclusions from our results, especially those concerning the roles of the different types of variables in the optimization and the relative difficulty in computing them.

The chapter closes with a discussion on the numerical difficulties observed in the computation, particularly as the test systems increase in size.

7.2 Simulations on a 6 Bus System

The six bus system is taken from the book by Dhar [Dhar 1982]. No cost data was given in that reference, so some values were created more or less arbitrarily. Also, to increase the size and complexity of the problem, generations were placed at every bus. The data and the schematic diagram for this system can be found in Appendix 7.1.

The number of variables in this system is as follows:

Number of buses:	6
Number of generations:	6
Number of loads:	4
Number of transmission lines:	7
Number of variable transformer taps:	2
Total number of load flow variables:	32

The program solved this system for an initial load, and then in three different runs for sets of 10 subsequent loads, increased in intervals of 1, 2 and 4 percent respectively.

7.2.1 Global Characteristics of the Solutions

The solution process for the six bus system is summarized in four tables: Table 7.1 for the solution to the initial load, and tables 7.2 to 7.4 for the three cases of the load-tracking solutions.

7.2.1.1 The optimal solution for the initial load

Table 7.1 gives a detailed account of the computational effort required for the solution of the initial load, for all the major steps of the algorithm. Horizontally, the columns tabulate the values for each major iteration until convergence is achieved. Vertically, the information is broken up into two main categories: information from the subproblem and from the load flow feasibility search.

For the subproblem, the table indicates first how many of the neglected dependent constraints are violated in the simple problem solution. It then shows how many breakpoints are required to obtain a completely feasible subproblem optimum using the continuation method.

In the search for the load flow feasible point, the table provides three important groups of information: the number of Newton-Raphson iterations needed in the load flow solver, the variables used to gauge convergence, and the step size described in section 3.4.4 to curtail unduly large movements from an expansion point.

The three convergence criteria considered here are (1) the relative reduction in the load flow feasible point's fuel costs from one iteration to the next, (2) the relative gap between the load flow feasible point's cost and that of the subproblem's optimal cost, and (3) the relative distance between the load flow feasible point and the subproblem solution. The term relative is used because the three difference terms which make up these convergence criteria were divided by the corresponding variables, to give a better idea of their relative size. Fairly tight tolerances have been placed on these convergence criteria in the program, allowing to study the rate of convergence over a larger number of iterations.

The table ends with the cost of the initial guess, the cost of the optimal solution, and the computation time. The latter is taken from an AT compatible personal computer equipped with a coprocessor. These timings are about two to three orders of magnitude slower than those which could be expected using fast mainframe computers [Dongarra 1987], but they are useful in comparing our results amongst themselves.

Our description of the results of Table 7.1 will start with a detailed look at the first column, followed by the general progression across the table for each entry. In all, 9 major iterations were required to solve the first load of this system to the chosen tolerances.

At the top of the first column, we have the results of the first subproblem of the optimization. The solution to the simple problem of the subproblem violated 4 of the 19 dependent constraints, the largest violation being 1.695 p.u. on the reactive power generation Q6. In the continuation process, 12 breakpoints were encountered before reaching the optimum of the subproblem. This process is documented in Table 7.6 and in figures 7.2, and will be described later.

TABLE 7.1 - TEST ON THE 6 BUS SYSTEM

SUMMARY OF THE ALGORITHM'S PERFORMANCE

SOLUTION FOR THE FIRST LOAD

Major iteration no.	1	2	3	4	5	6	7	8	9
SUBPROBLEM									
Violations in the subproblem at the simple optimum	4	3	5	5	4	3	1	4	4
Breakpoints in the continuation process	12	6	20	14	10	4	2	6	8
SEARCH FOR LOAD FLOW FEASIBLE POINT									
Newton-Raphson iterations in finding load flow feasible pt.	3	4	2	2	4	2	4	3	1
Relative reduction in cost of load flow feasible pt.	-	.0138	782e-2	.139e-2	.144e-2	628e-3	.373e-3	.528e-3	.173e-4
Relative gap between costs of load flow feasible pt. and subproblem solution	.0652	.0323	.118e-1	.104e-2	.462e-2	.677e-2	.606e-2	235e-2	.191e-2
Relative norm of the difference between the variables at the load flow feasible point and the subproblem solution	.1159	.1084	.0758	.0592	.0362	.0228	.0186	.0161	.0155
Step size in searching feasible load flow pt	9	.45	.45	45	.45	.45	.225	.1125	.1

Initial cost. 1.6431 Optimal cost 1.6009 Computation time: 19.72 sec.

Upon completion of the first subproblem, the three convergence criteria are computed. Convergence cannot be declared after just one iteration, but this information is still useful in showing how much progress is achieved towards reaching the optimum. It can be noted, in comparison with the subsequent columns of the table, that the first iteration results in the largest changes in the candidate solution.

The computation of the load flow feasible point in the first major iteration required 3 Newton-Raphson iterations. The first load flow feasible point computed by this part of the program was also bounds-feasible and of lower cost than the previous expansion point, so it was kept as the expansion point for the next major iteration and the step size remained unchanged from its previous value.

The sequence of subproblem solutions over the 9 major iterations, as will be seen with the other simulations, is somewhat typical. The first few subproblems required a relatively large numbers of breakpoints, in resolving cases with relatively large violations. For example, the first three subproblems required 12, 6 and 20 breakpoints, while the last three required only 2, 6 and 8 respectively. The largest violations in the first three subproblems were 1.70, 0.48 and 1.58 p.u., while in the last three they were 0.01, 0.55 and 0.19 p.u. It is worth noting however that even for the longest subproblem, in major iteration no.3, the number of breakpoints compares favorably with the typical number of linear or quadratic programming iterations. Recent works place an empirical upper limit on the usual number of LP iterations around 1.5 times the number of constraints [Chvatal 1983]. For this system, that number would be about 80.

The sequence of Newton-Raphson load flow solvers required relatively little computation, averaging less than 3 iterations in each major iteration. For the 6 bus system, as well as for the other 3 systems used for these tests, load flow convergence was rarely a problem. In three cases, major iterations nos. 2, 7 and 8, the Newton-Raphson solver was used twice, since the first computed candidate was unacceptable. That explains the reduction in step size. However the most iterations in any major iteration was only four, in iterations 2 and 7.

The two cost-related convergence criteria started off with relatively small values, thanks to the good initial guess, and then reduced apparently with a linear rate of convergence. This rate could be expected, because of the use of the step size in the procedure [Dennis & Schnabel 1983]. The change in cost, from its initial value to its optimal value, is only 2.5%, from 1.6431 to 1.6009 units. The change in the real power generations is correspondingly small.

Figure 7.1 shows the progression of the costs of the load flow feasible points and of the subproblem solutions versus the number of major iterations. It illustrates that the subproblem solution can effectively serve as a lower bound on the optimal solution. It also shows that the upper curve, that of the feasible load flow points, reaches its bottom rather quickly. Hence, the idea of using the gap between the two curves as a measure of convergence is sound, but it might unduly prolong the computation. In our case, the tolerance on the gap between the two curves was chosen 5 times larger than the tolerance on the cost reduction to avoid prolonged computation, but even then the cost reduction bottomed out before gap value reached its tolerance.

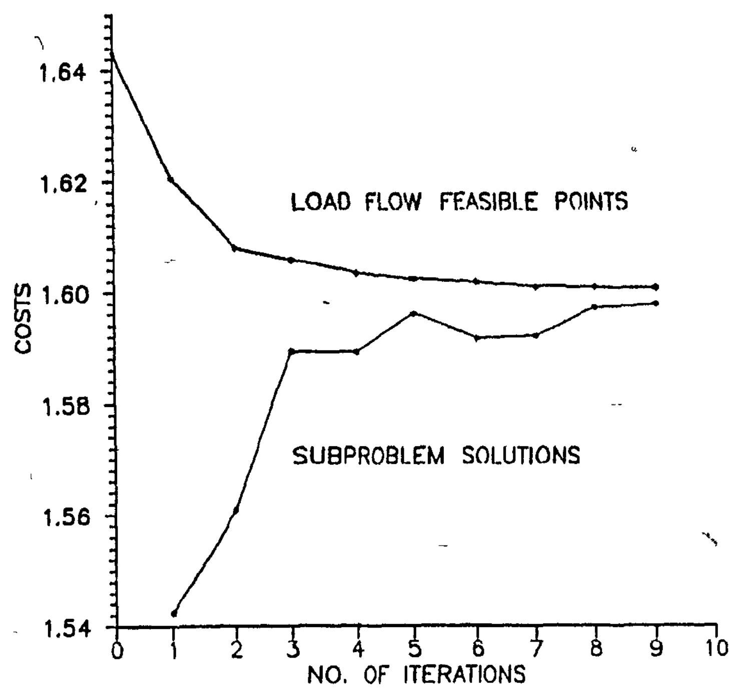


Figure 7.1. Costs of the load flow feasible points and the subproblem solutions at each major iteration, solution to the 6 bus system.

The third convergence criterion, involving the load flow variables, decreased from a fairly large initial value of 11.6% to a much smaller 1.5%. Most of this change is in the reactive power generations, and to a lesser extent in the voltages and the variable transformer tap positions. This improvement is possibly the most significant in the optimization process, in that it assigns optimal values to the many variables which have no direct cost.

The step size used in the search for the load flow feasible point reduced automatically as the optimization proceeded. This, as we shall see, is the typical behaviour. Recalling the argument of section 3.4.4.5 for the solution of systems of nonlinear equations with fold lines, this behaviour seems reasonable. As the expansion point approaches the optimum, the correction step suggested by the subproblem will not reduce to zero; hence the step size must do so. The final value of the step size in this case is 0.1. In a case where this test was allowed to run for 30 major iterations, the step size reduced to 0.025; that seems to confirm the theory given above.

The computation time to solve for the first load of this system was 19.72 sec with computations in double precision. This will be compared to the solution times of the subsequent loads.

7.2.1.2. The optimal solutions of the load-tracking step

The next three tables provide an account of the solutions for the subsequent loads in the load-tracking mode. The loads are increased by 1 percent after each solution in the first case, and by 2 and 4 percent in the second and third cases. These cases are documented in Tables 7.2, 7.3 and 7.4 respectively.

In each case the load-tracking is tested on sequences of 10 loads. The tight convergence tolerances for the initial load solution are maintained on every fifth load, but are relaxed a bit for the other loads. In comparing costs from the different load tracking tests, the accuracy obtained with the relaxed tolerances seems quite good.

The information contained in these tables is as basically the same as that in Table 7.1, except that it is not broken down into major iterations. For each load, the corresponding column provides the number of major iterations required in the solution, the total number of breakpoints in the subproblems, the total number of Newton-Raphson iterations, the optimal cost, and the computation time.

i) The loads are increased by 1 percent

In Table 7.2, results for 1 percent changes in load are seen to be very encouraging. Six of the ten loads required only one major iteration to reach the optimal solution; the other four required only a second major iteration. For most loads the number of subproblem breakpoints is very small. In fact, for four of the loads, the continuation process of the subproblem was not required, resulting in quick subproblem solutions. Only for loads nos. 2 and 3, the total number of subproblem breakpoints seemed large (22 and 18), although these are relatively few compared to the 82 breakpoints in the solution for the initial load. The solutions for the fifth and the tenth loads, with their tighter convergence tolerances, were just as fast as the other solutions with the looser tolerances.

TABLE 7 2 - TEST ON THE 6 BUS SYSTEM

SUMMARY OF THE ALGORITHM'S PERFORMANCE

SOLUTIONS FOR THE SUBSEQUENT LOADS - 1% VARIATIONS IN LOAD

Load no.	1	2	3	4	5	6	7	8	9	10
No. of major iterations	1	2	2	1	1	1	2	1	1	2
Total number of breakpoints in the subproblems	2	22	18	2	0	8	12	0	0	0
Total number of Newton-Raphson iterations	3	4	6	3	4	3	5	3	6	3
Optimal cost	1.6223	1.6440	1 6661	1.6886	1.7114	1.7347	1 7583	1.7823	1.8068	1.8316
Computation time (sec.)	1.64	4 67	4.89	1 59	1.93	1 81	3.13	1.15	2.15	3.46

The total number of Newton-Raphson iterations for the 10 loads was 40. This includes the iterations of a load flow solution immediately following the load increase, before the optimization. The average is then 4 iterations for each load, with two usually coming in the initial step mentioned above.

Costs are seen to increase by a little more than one percent from one load to another. These values will be compared to the costs of Tables 7.3 and 7.4 a little further.

Solution times for the loads in Table 7.2 range from 1.15 sec. to 4.89 sec. with an average of 2.64 sec.. This is only 13% of the 19.72 sec. required for the initial solution.

ii) The loads are increased by 2 percent

Solutions in Table 7.3, with the varying loads incremented by 2 percent, are also very encouraging. With larger changes in the load, the initial guess, taken from the previous optimal solution, is farther from the optimum of the new problem. That increases the difficulty in solving the new problem only slightly however, because the load variations are still quite small.

All but two of the load-tracking solutions in this case required a single major iteration; the remaining solutions required two major iterations. The total number of subproblem breakpoints increased from 64 in the previous case to 96 in this case, and the number of Newton-Raphson iterations remained the same as in the previous case. It is interesting to note that iterations 2 and 3 in this case require more breakpoints than average, as for the previous case. This is because the sets of independent variables upon entering these subproblems are similar for the two cases. That demonstrates that the complexity of the subproblem solution is related to the chosen set of independent variables. The solution times for this case are somewhat similar to those of the previous case. Hence, despite the added computation, the 2 percent change in loads performs very well in load-tracking.

TABLE 7.3 - TEST ON THE 6 BUS SYSTEM SUMMARY OF THE ALGORITHM'S PERFORMANCE SOLUTIONS FOR THE SUBSEQUENT LOADS - 2% VARIATIONS IN LOAD										
Load no.	1	2	3	4	5	6	7	8	9	10
No. of major iterations	1	2	2	1	1	1	1	1	1	1
Total number of breakpoints in the subproblems	0	14	18	12	4	10	8	2	12	16
Total number of Newton-Raphson iterations	3	4	5	3	3	5	3	10	2	2
Optimal cost	1.6439	1.6882	1.7341	1.7814	1.8305	1.8781	1.9336	1 9878	2.0438	2.1019
Computation time (sec.)	1.64	4.67	4.89	1.59	1.87	1.87	3.13	1.10	2.14	3.41

iii) The loads are increased by 4 percent

The results of Table 7.4, with the varying loads incremented by 4 percent, show that slightly more computation is required for each solution than in the previous cases. The total counts for subproblem breakpoints and Newton-Raphson iterations over the 10 loads in this case are 112 and 44, respectively. All the loads except two require a single major iteration for their solution; one requires two major iterations and one requires five. The average computation time per solution was 3.26 sec. (17% of the time for the initial solution), with individual timings ranging from 1.48 sec. to 8.13 sec.. Load tracking in this case with 4% load changes is clearly faster than computing with 1% load changes 4 times or with 2% load changes twice. Hence the 4% load changes performed very well in load tracking.

TABLE 7.4 - TEST ON THE 6 BUS SYSTEM

SUMMARY OF THE ALGORITHM'S PERFORMANCE

SOLUTIONS FOR THE SUBSEQUENT LOADS - 4% VARIATIONS IN LOAD

Load no.	1	2	3	4	5	6	7	8	9	10
No. of major iterations	1	2	1	1	1	1	1	1	1	5
Total number of breakpoints in the subproblems	4	20	18	10	8	10	10	6	2	26
Total number of Newton-Raphson iterations	4	5	4	3	5	3	4	6	3	7
Optimal cost	1.6874	1.7797	1.8783	1.9837	2.0964	2.2169	2.3460	2.4842	2.6322	2.7894
Computation time (sec.)	2.20	4.61	3.13	2.20	2.86	2.19	2.75	3.02	1.48	8.13

iv) Comparison of optimal costs from the three load-tracking runs

It is interesting to compare optimal costs computed in the different load tracking tests. These are regrouped to form Table 7.5 below. The load increases are compounded differently (i.e. Load(k+1) = (1+e).Load(k) with different values of e), but their differences are very small; the largest discrepancy in loads is by a factor of 0.0019, on the final load of the table. Hence an accurate comparison can be made. These costs compare very well, with discrepancies never occurring before the third significant digit and usually not before the fourth. The largest discrepancy, very small, occurs for the final load, with a difference in costs of 0.26 percent of their average cost. The average discrepancy in optimal costs over all the loads is a very small 0.089 percent, which is about the same as the average discrepancy in loads. The remarkable concordance in the costs indicates that the computed optimal cost trajectories stray very little from the exact optimal cost trajectories.

TABLE 7.5 - TEST ON THE 6 BUS SYSTEM SUMMARY OF THE ALGORITHM'S PERFORMANCE COMPARISON OF OPTIMAL COSTS OBTAINED FROM THE LOAD TRACKING MODES (TAKEN FROM TABLES 7.2 - 7.4)								
% change in load	2	4	6	8	10	12	16	20
Optimal Costs for load tracking with								
1% load changes	1.6440	1.6886	1.7347	1.7823	1.8316			
2% load changes	1.6439	1.6882	1.7341	1.7814	1.8305	1.8781	1.9878	2.1019
4% load changes		1.6874		1.7797		1.8783	1.9837	2.0964

This completes our description of the overall solution of the 6 bus system.

7.2.2 A study of the subproblem solutions

This section describes in detail the solution of the first subproblem in the solution of the initial load of the 6 bus system, and includes some results from two other subproblems which can serve for comparison.

7.2.2.1 The first subproblem

The solution to the simple problem of the first subproblem violates four constraints. These are:

- Q6 violated its lower bound by 1.695 p.u.
- Q4 violated its upper bound by 1.032 p.u.
- V2 violated its upper bound by 0.296 p.u.
- V4 violated its upper bound by 0.113 p.u.

It will be interesting to follow the solution trajectories of these constraints in the continuation process.

The continuation process starts by adding the most violated constraint Q6 to the active set, after having shifted the bounds of the violated constraints by 1.695 p.u.. From there the continuation process encounters 11 more breakpoints as the shifted bounds reintegrate their original positions. These breakpoints are chronicled in Table 7.6.

TABLE 7.6 - TEST ON THE 6 BUS SYSTEM

A SUMMARY OF BREAKPOINTS ENCOUNTERED IN THE FIRST SUBPROBLEM

θ	Variable name and type		Cause of breakpoint
0.0000	Q6	dependent	most violated dependent constraint set to its moving lower bound.
.1390 e-5	Q5	transparent	released from its upper bound.
0.2097	Q5	transparent	set to its lower bound.
0.2097	Q1	transparent	released from its lower bound.
0.5007	Q1	transparent	set to its upper bound.
0.5013	V6	transparent	released from its lower bound.
0.8270	V2	dependent	set to its moving upper bound.
0.8270	V5	transparent	released from its upper bound.
0.9139	V5	transparent	set to its lower bound.
0.9139	V3	transparent	released from its upper bound.
0.9619	V3	transparent	set to its lower bound.
0.9619	Q2	transparent	released from its upper bound.

When the process starts, the transparent variables are for the most part on their bounds, and the dependent Q6 is active on its moving lower bound. Then the continuation parameter θ is increased. The typical scenario for the continuation process, which can be seen in Table 7.6 and in subsequent tests, is as follows. At some point a variable reaches a bound. Almost immediately, a transparent variable is forced off its bound as a form of compensation, when the trajectory of its Lagrange multiplier plummets to zero. The continuation process moves ahead until again another variable reaches a bound and the process is repeated. Hence, breakpoints occur in pairs. For example, in Table 7.6, we see that the response to the addition of the dependent Q6 to the active set at θ=0 is the removal of the transparent Q5 from the active set at 0.1390 e-5. The next pairs of breakpoints are fairly well spaced, around θ = 0.21, 0.50, 0.83, 0.91 and 0.96. The largest difference in θ between the two members of the pair is $\Delta\theta$ = 0.0006 for θ around 0.50, and in the other cases the differences are smaller than 0.0001.

Inspection of Table 7.6 shows that the words "set to" and "released from" a bound always alternate. This will always be the case in our tests. The variable which goes to a bound can be dependent, as seen here twice, to a fixed or to a moving bound. It can also be transparent, as seen here four times, when this variable is released from one bound but moves to the other.

The optimal solution trajectories for all the variables of the subproblem are furnished in fig. 7.2, a. to f.. The values on the left edge of the graphs are the simple problem solutions. The solution trajectories provide the optimal solutions to the intermediate problems where the relaxed constraints are reintegrating their bounds, and lead to the subproblem solutions on the right edge of the graphs.

The graphs indicate, in brief, that the simple solution allows for large imbalances in reactive power generation and unreasonably high bus voltages. The continuation process redistributes reactive power through the system so as to avoid a very high Q4 and a very low Q6, and reduces bus voltage magnitudes V2 and V4 within their bounds. The intermediate steps resemble something of a balancing act, with variables reacting to each other, sometimes in tandem and other times offsetting each other. It can be noted also that for such a small system, all the variables interact fairly closely to each other.

Figure 7.2.a shows the trajectories of the real power generations. We see that the reallocation of real powers, to accommodate the tighter bounds on Q and V in the continuation process, is very small. From common dispatching practices, that result was expected.

Figure 7.2.b. illustrates the trajectories of the reactive power generations. The outer trajectories Q4 and Q6 are typical of the dependent variables with relaxed bounds. The envelope which they form on the graph

narrows from left to right, as the constraints are tightened. In this example, Q6 follows its moving lower bound from the outset to its final, original position. Q4, originally less violated, is reduced to acceptable values in the continuation process but remains inactive throughout. The other reactive powers are transparent variables in this case. Q5 is released from its upper bound in response to Q6 being set to its lower bound, and Q1 is released from its lower bound after Q5 reaches its lower bound. Both Q5 and Q1 are forced to change bounds in the process. Q2 is released from its upper bound towards the end of the process, after V3 is set to its lower bound. The remaining Q3 remains at its lower bound throughout.

Figure 7.2.c shows the trajectories of the bus voltage magnitudes. V2 and V4 are dependent variables having violated their upper bound of 1.10 p.u.. The former meets up with its moving upper bound at θ=0.827, and from there follows it to 1.1 p.u.. The latter remains inactive but succeeds in reintegrating its feasible region. The other voltages are transparent variables. V3 and V5 switch bounds rather quickly; we observe in this and other tests that the slopes of transparent voltages can be quite steep. V6 leaves its lower bound, and V1 remains on its upper bound of 1.05 p.u.

Figure 7.2.d shows the trajectories of the bus voltage phase angles. These are unbounded, so they are of less interest. We see however the progression of the values towards the subproblem optimum. In this example, the only outstanding feature is d2 increasing sharply after V2 hits its moving upper bound.

Figure 7.2.e shows the trajectories of the variable transformer tap positions. Both are dependent variables in this case. They remain within bounds but move over a wide range. T2 reacts strongly when the voltage V5 at an adjacent bus is released from its upper bound. It then strongly reacts in the opposite direction when V5 reaches its lower bound. T1 also reacts to V5 reaching its bound, although in the opposite direction. This reaction is "reasonable", since T1 is acting to maintain reactive powers in another part of the system.

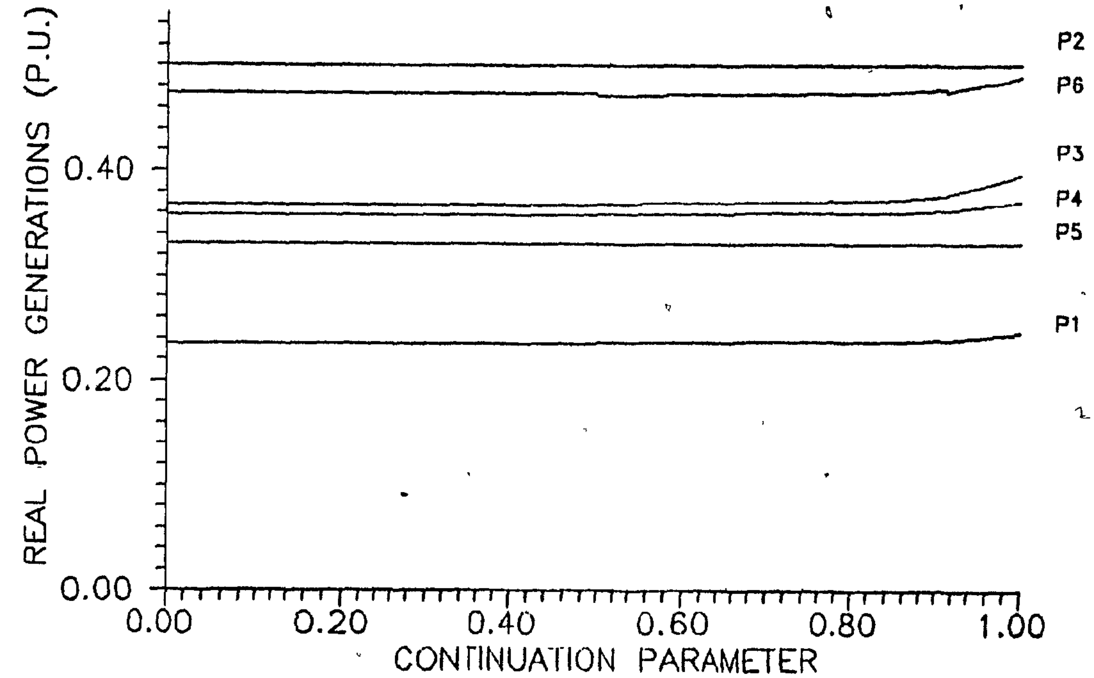


Figure 7.2.a. Real power generations vs. the continuation parameter, first subproblem in the solution of the 6 bus system.

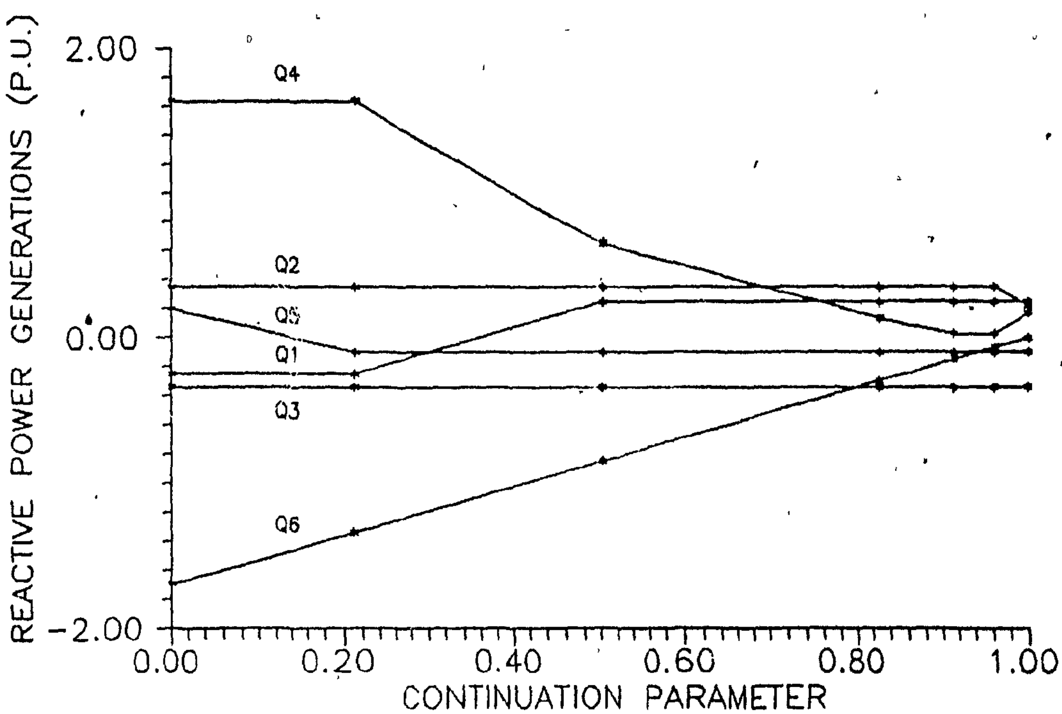


Figure 7.2.b. Reactive power generations vs. the continuation parameter, first subproblem in the solution of the 6 bus system.

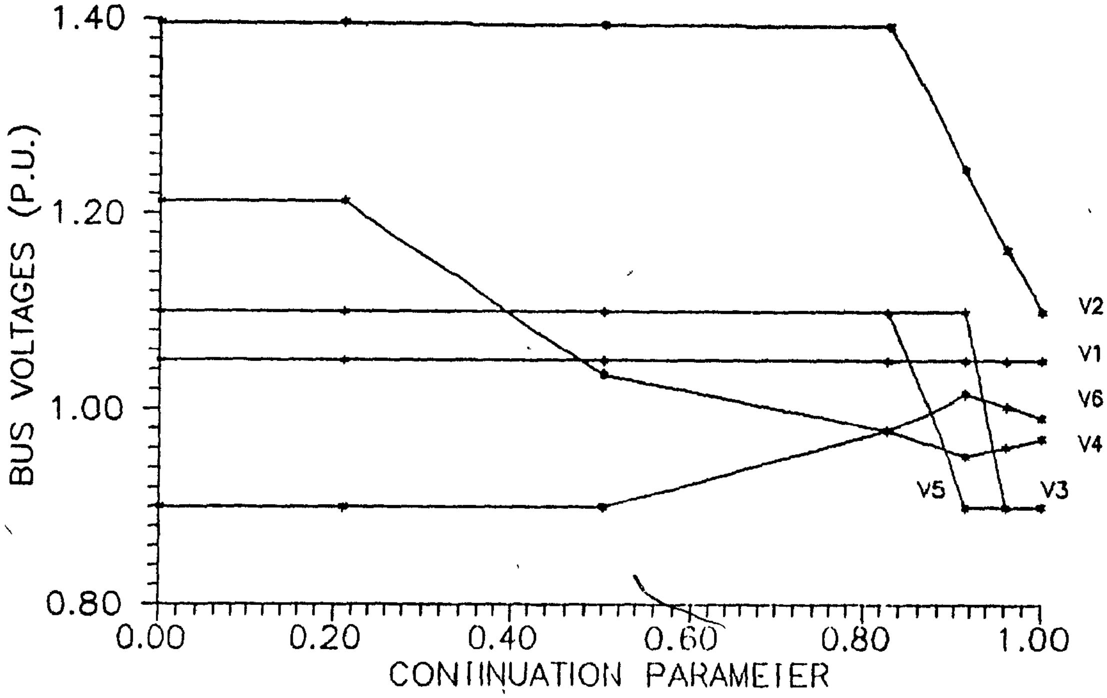


Figure 7.2.c. Bus voltage magnitudes vs. the continuation parameter, first subproblem in the solution of the 6 bus system.

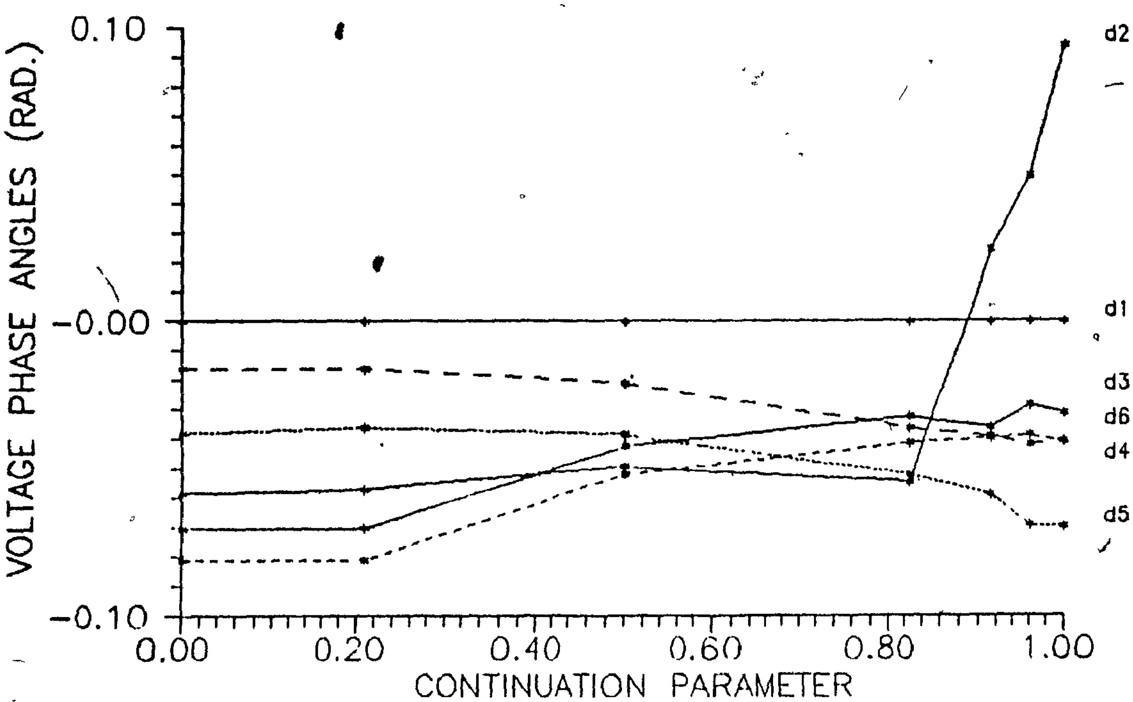


Figure 7.2.d. Bus voltage phase angles vs. the continuation parameter, first subproblem in the solution of the 6 bus system.

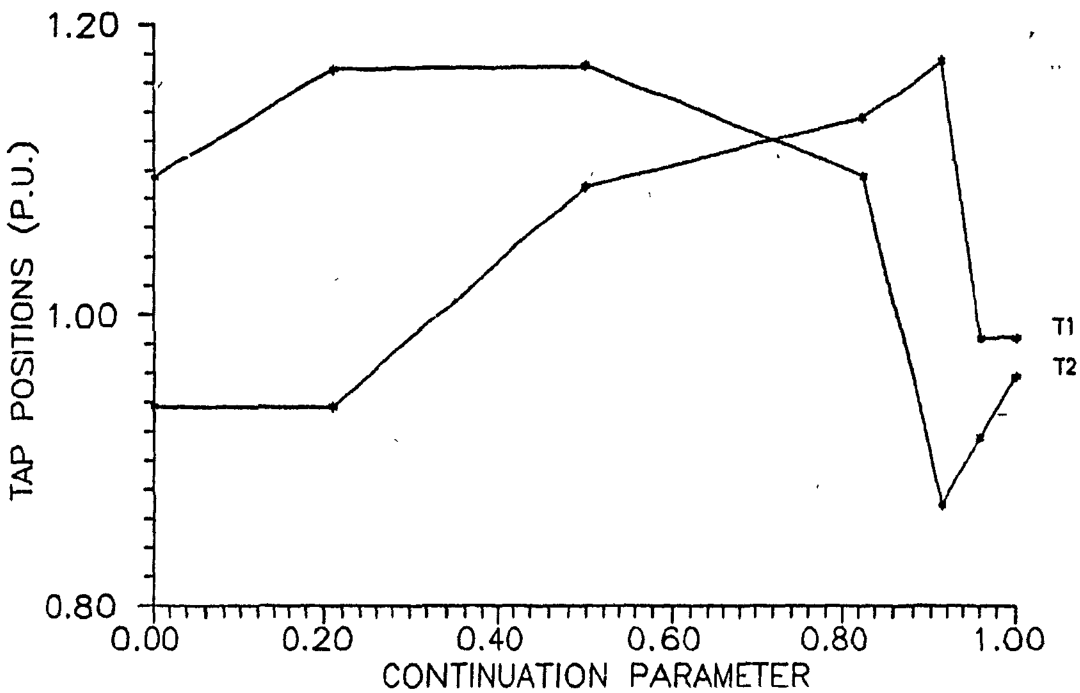


Figure 7.2.e. Variable transformer tap positions vs. the continuation parameter, first subproblem in the solution of the 6 bus system.

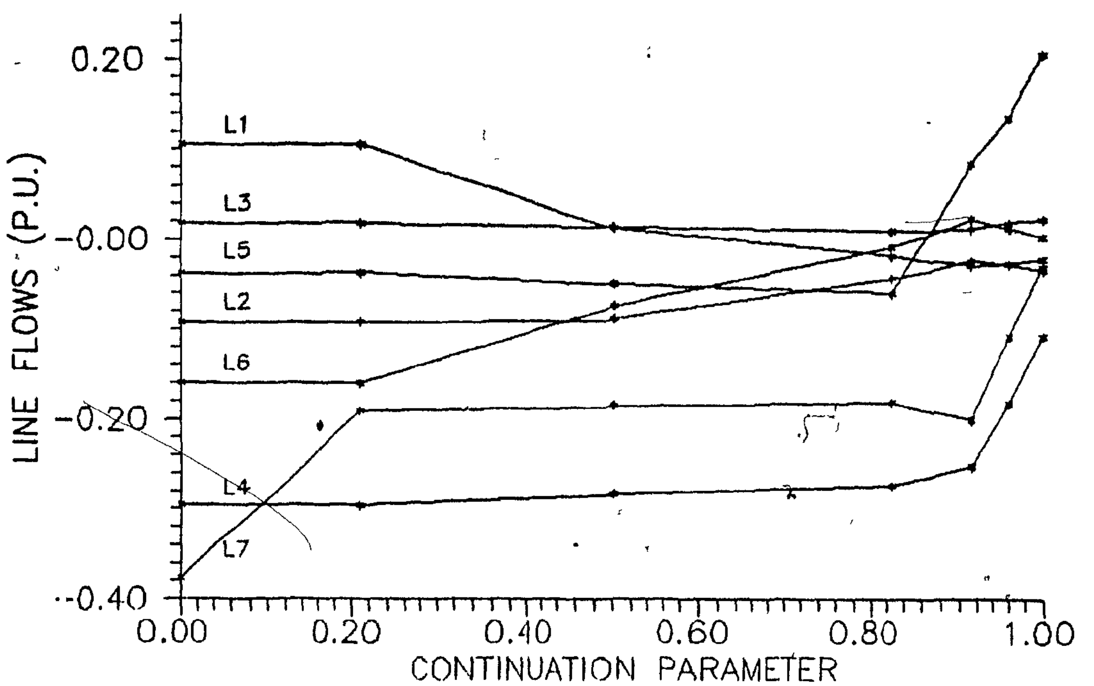


Figure 7.2.f. Line flows vs. the continuation parameter, first subproblem in the solution of the 6 bus system.

Finally, figure 7.2.f shows the line flows (or more precisely, the linear approximations of the line currents squared). Negative values are possible in the subproblem, since the linear equations for these variables do not impose non-negativity. Their values are all relatively small compared to their bounds. We note near the end of the process, as V5 is released from its bound, that the line flow L4 (between buses 5 & 2) increases sharply, and as V3 is released the line flows L5 (buses 4 & 3) and L7 (buses 3 & 2) increase sharply.

7.2.2.2 Some Results from Two Other Subproblems

The more interesting graphs of some variables from two other subproblems are presented, to give a better idea of the possible behaviour of the continuation algorithm and to compare with the first subproblem. The chosen subproblems are the third, with 20 breakpoints, and the seventh, with only 2 breakpoints.

For the third subproblem, fig. 7.3 a. to d. present the reactive power generations, the bus voltage magnitudes, the variable transformer tap positions and the line flows. The set of independent variables has been changed slightly from the first subproblem. We notice in fig. 7.3.a the same reactive power violations as in the first subproblem, and in fig. 7.3.b, an additional voltage violation on V1. Most noticeable in the trajectories are the increased number of breakpoints (marked by asterisks on the curves) and in some instances the sharper variations and the reversals in the values of some variables. For example in fig. 7.3.a, reactive powers Q6 and Q3 vary sharply and in alternating directions towards the end of the process. In fig. 7.3.b the bus voltage magnitudes all dip rapidly in that same final interval of θ between 0.86 and 1.0. The two taps, illustrated in fig. 7.3.c, reacted differently: T1 left its lower bound briefly but then returned to it, while T2 decreased until reaching θ=0.86 and then rapidly increased. The effect of these rapid changes on line flow L5 was to quickly send it to its lower bound and then just as quickly return it to values in its initial range. These quirks in the intermediate solutions might be due to the high sensitivities of

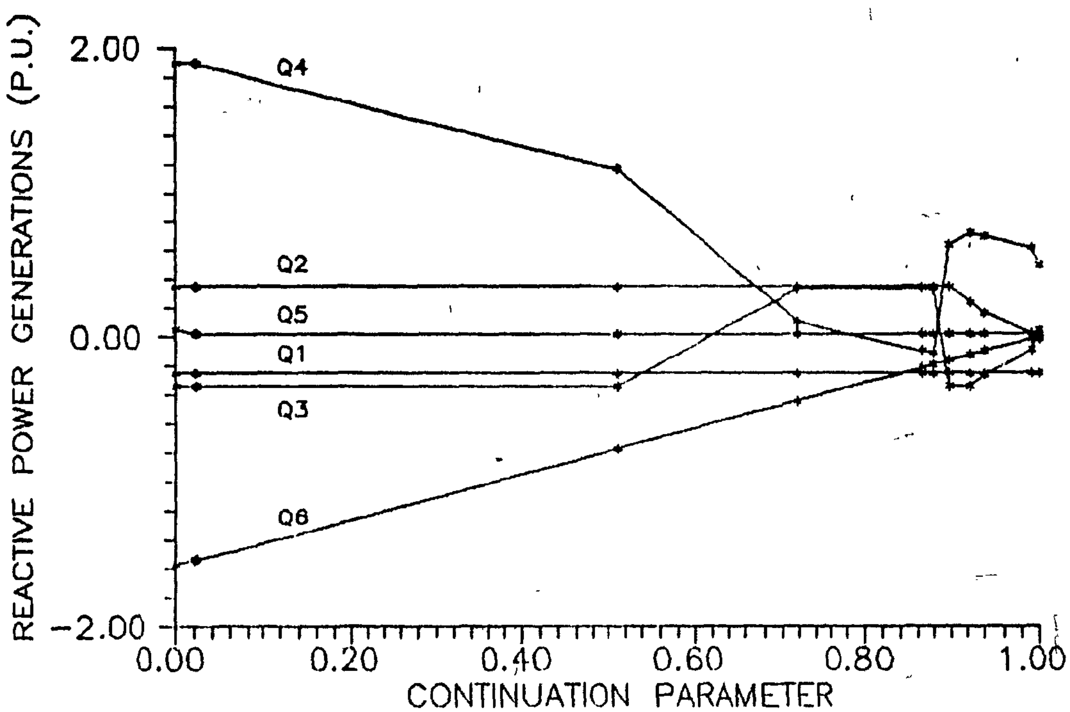


Figure 7.3.a. Reactive power generations vs. the continuation parameter, third subproblem in the solution of the 6 bus system.

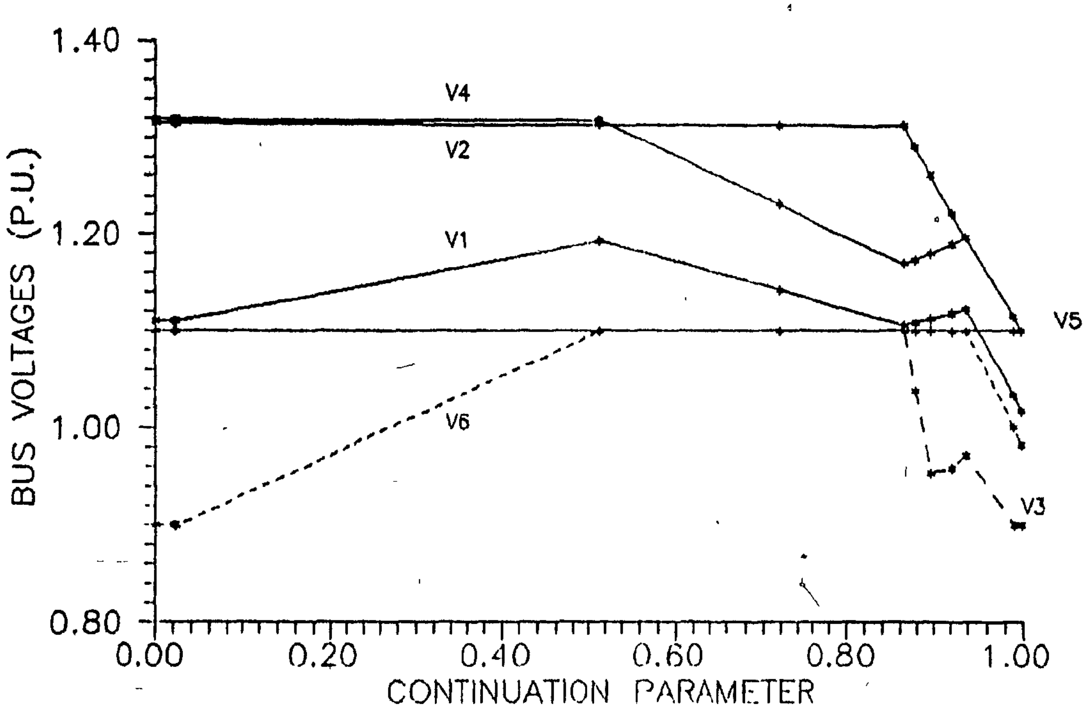


Figure 7.3.b. Bus voltage magnitudes vs. the continuation parameter, third subproblem in the solution of the 6 bus system.

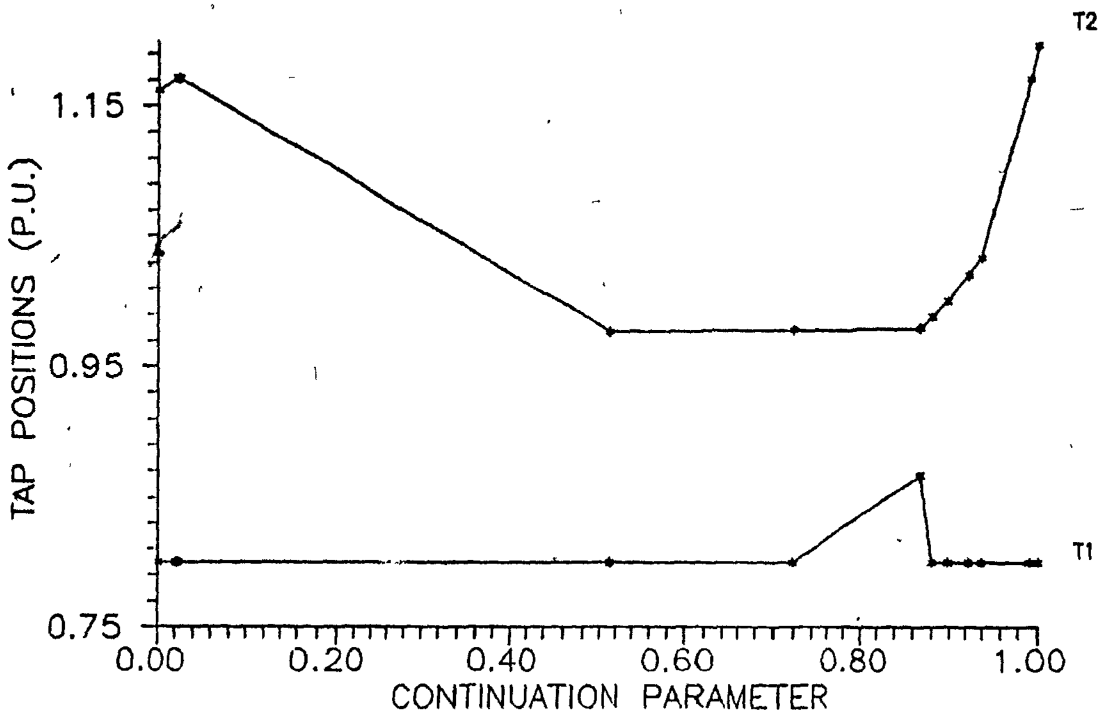


Figure 7.3.c. Variable transformer tap positions vs. the continuation parameter, first subproblem in the solution of the 6 bus system.

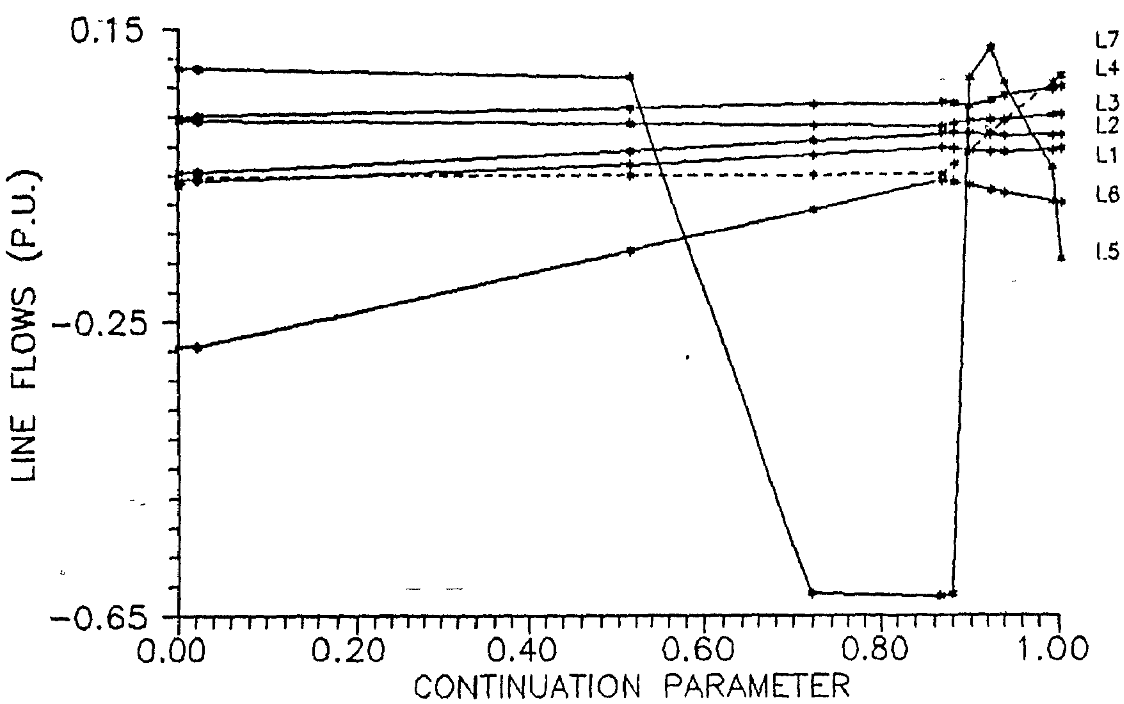


Figure 7.3.d. Line flows vs. the continuation parameter, third subproblem in the solution of the 6 bus system.

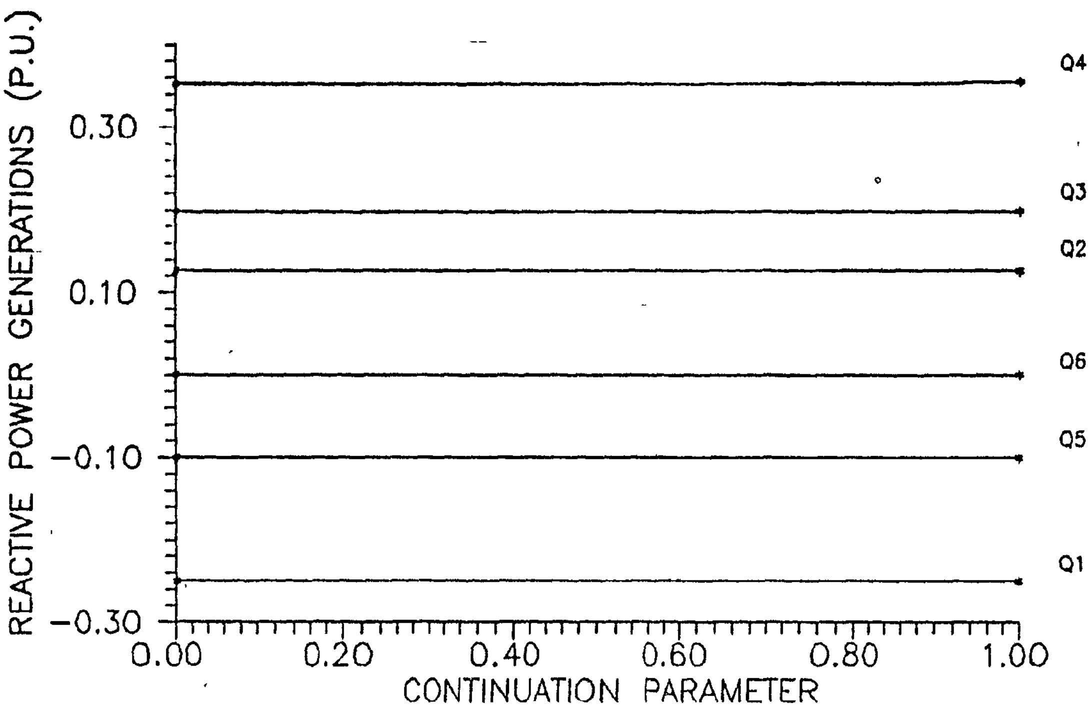


Figure 7.4.a. Reactive power generations vs. the continuation parameter, seventh subproblem in the solution of the 6 bus system.

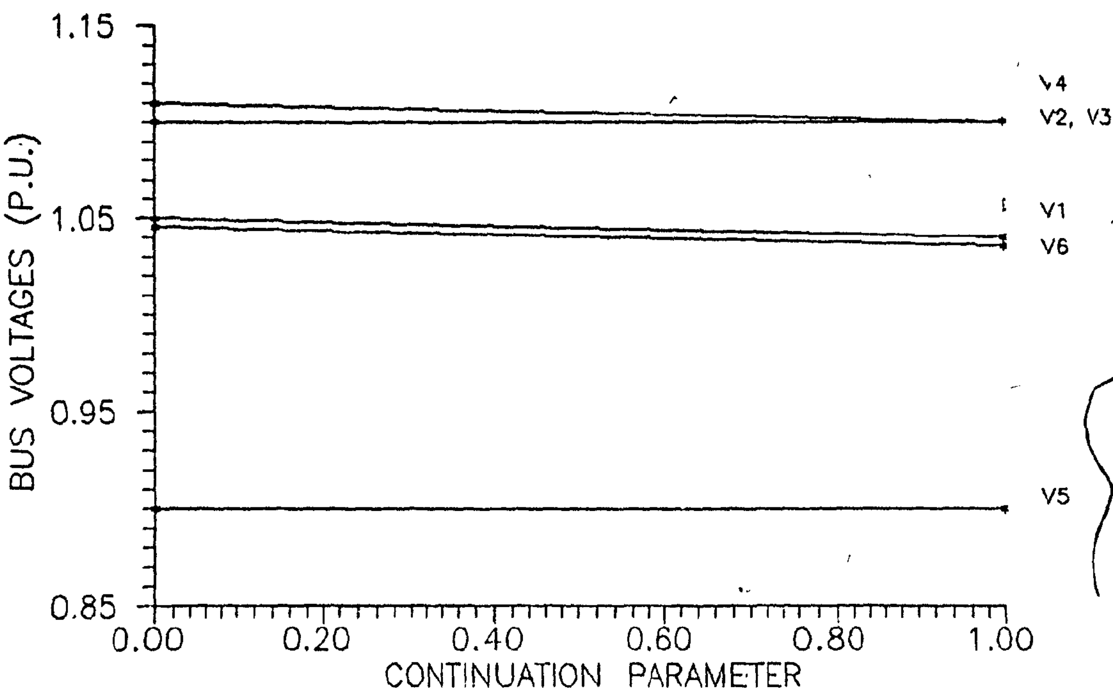


Figure 7.4.b. Bus voltage magnitudes vs. the continuation parameter, seventh subproblem in the solution of the 6 bus system.

the dependent variables in the subproblem. These variables would not be so sensitive over such a wide range in the full nonlinear problem.

The seventh subproblem is the most easily solved, so in a sense it is the most successful. Upon entering this subproblem, the set of independent variables contains all the active constraints except one. The single violated constraint in the simple problem is V4. After having added it to the active set and having released V1, no other breakpoint is encountered. Fig. 7.4 a. and b. illustrate the almost stationary reactive power generations and bus voltage magnitudes. Only the initially violated voltage V4 shows any noticeable change during the continuation process.

7.2.3 Description of the System Variables in the Nonlinear Optimization Process

This section describes the progression of the more important load flow variables through the major iterations of the nonlinear optimization process. More specifically, the sequences of solutions at two points in the algorithm are studied: the subproblem solutions and the load flow feasible points. Basically the results show how the quantities vary with the change in the expansion point, as it moves towards the optimum.

7.2.3.1 The sequence of subproblem solutions

Figures 7.5. a. to d. show the selected load flow variables at the end of the subproblems versus the major iteration number. They illustrate the varied behaviours of the different types of variables in the optimization.

Figure 7.5.a shows the slight variation in the allocation of real power generations. Referring back to Fig. 7.1, we saw that the subproblems produced optimistic costs which eventually merged with the nonlinear load flow costs. The rise in subproblem costs vs. the major iterations is reflected here also, as the real power generations increase slightly from left to right on the

graph. These increases are in the range of 2 to 4 percent, and take different values for each P.

The next three figures in this group show that the cost is highly insensitive to the other variables, which are not directly cost-related. They can oscillate over a wide range, in some cases jumping from one bound to another, from one major iteration to the next. Figure 7.5.b. shows 3 reactive powers and fig.7.5.c. shows 3 bus voltage magnitudes, exhibiting oscillations of various magnitudes. The remaining Q's and V's follow the same patterns, but were removed from the graphs to avoid clutter. The same erratic behaviour is apparent in fig. 7.5.d. for the variable transformer taps.

The reasoning behind the large swings in these variables was sketched briefly in Chapter III, and had to do with the sensitivities of the variables in the subproblem. This aspect is verified on its own in section 7.2.5 a little further.

7.2.3.2 The sequence of load flow feasible points

The subproblem solution serves as an end point of the search segment in finding the next load flow feasible point. Despite some large oscillations in the subproblem solutions, the sequence of the load flow feasible points settles down after a few major iterations. The oscillations are effectively damped out in this part of the computation by applying smaller step sizes in the search. This is illustrated in fig. 7.6. a. to d.

Figure 7.6.a. shows the progression of the real power generations in the optimization. The initial guess was quite good for all the P's except P1. As the optimization proceeded P1 reduced noticeably, from 0.334 to 0.264. As for the others, P2 was a fixed quantity and the remaining P's increased slightly.

Figures 7.6. b. to d., for the reactive power generations, the bus voltage magnitudes and the taps, show that these variables eventually converge, after a few iterations of oscillatory behaviour.

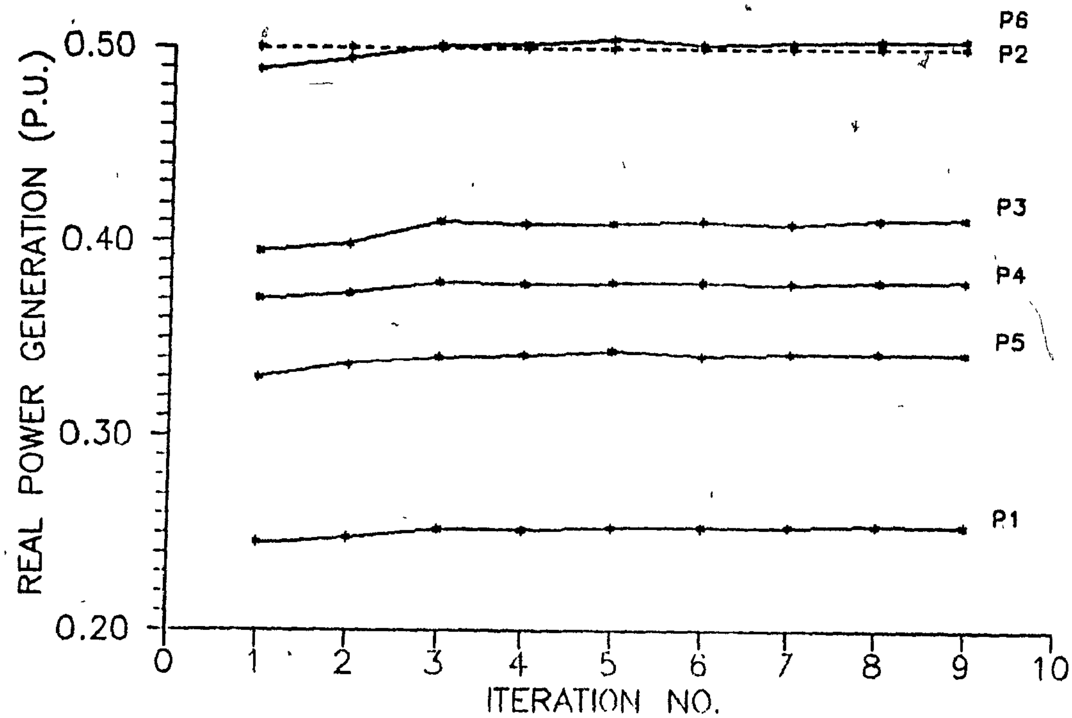


Figure 7.5.a. Subproblem real power generations vs. the major iteration number, solution of the first load of the 6 bus system.

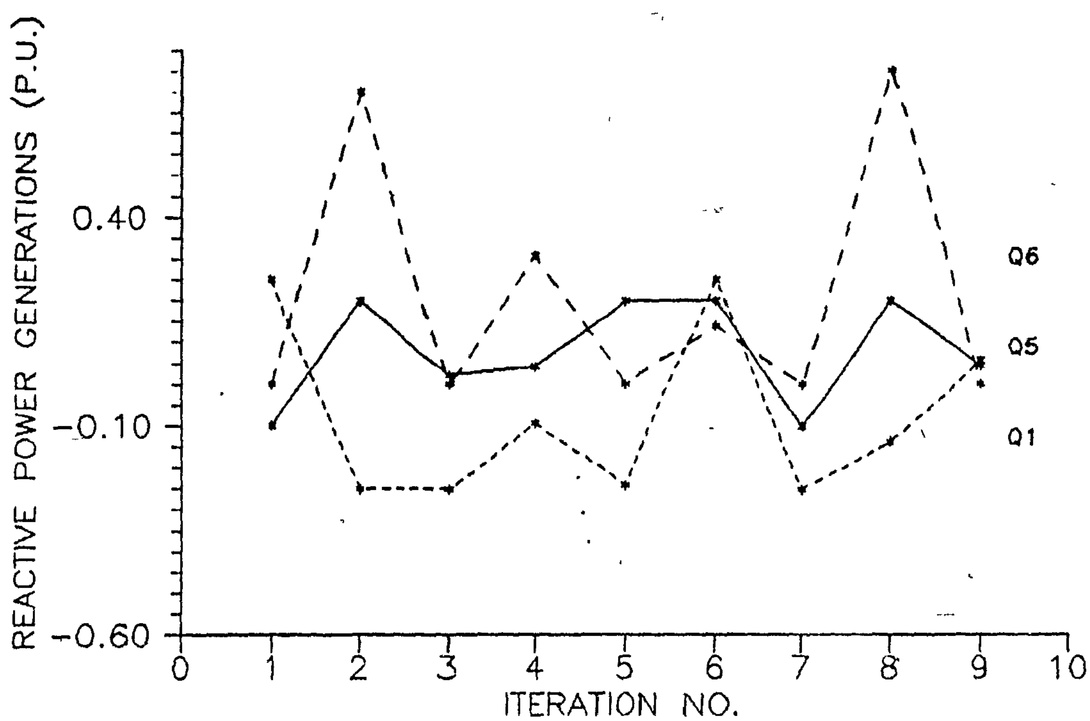


Figure 7.5.b. Subproblem reactive power generations vs. the major iteration number, solution of the first load of the 6 bus system.

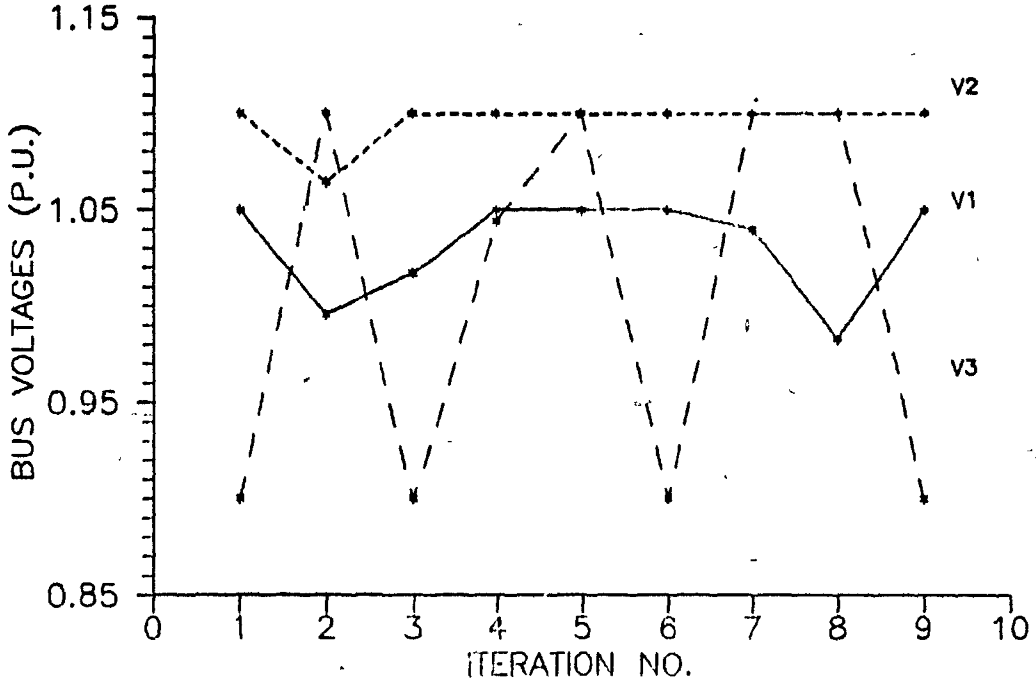


Figure 7.5.c. Subproblem bus voltage magnitudes vs. the major iteration number, solution of the first load of the 6 bus system.

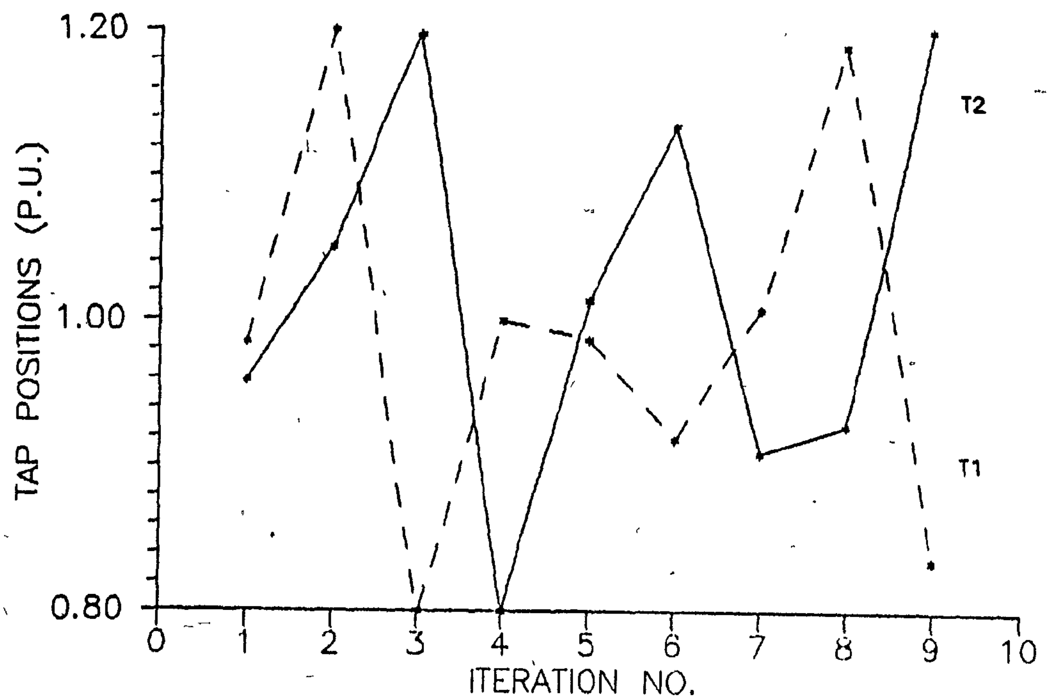


Figure 7.5.d. Subproblem variable tap positions vs. the major iteration number, solution of the first load of the 6 bus system.

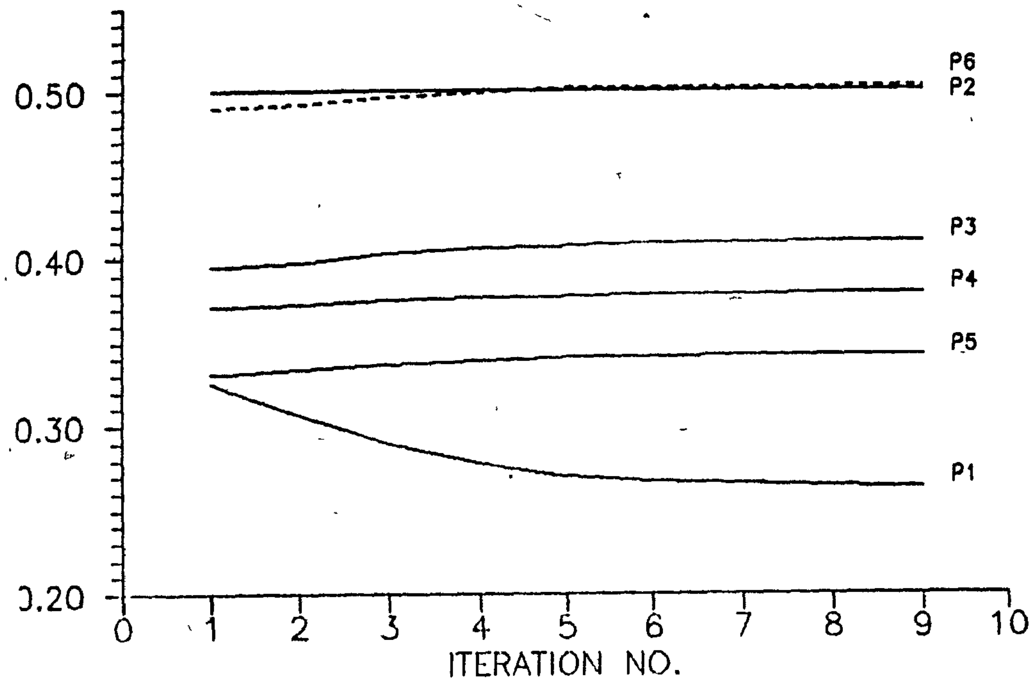


Figure 7.6.a. Feasible real power generations vs. the major iteration number, solution of the first load of the 6 bus system.

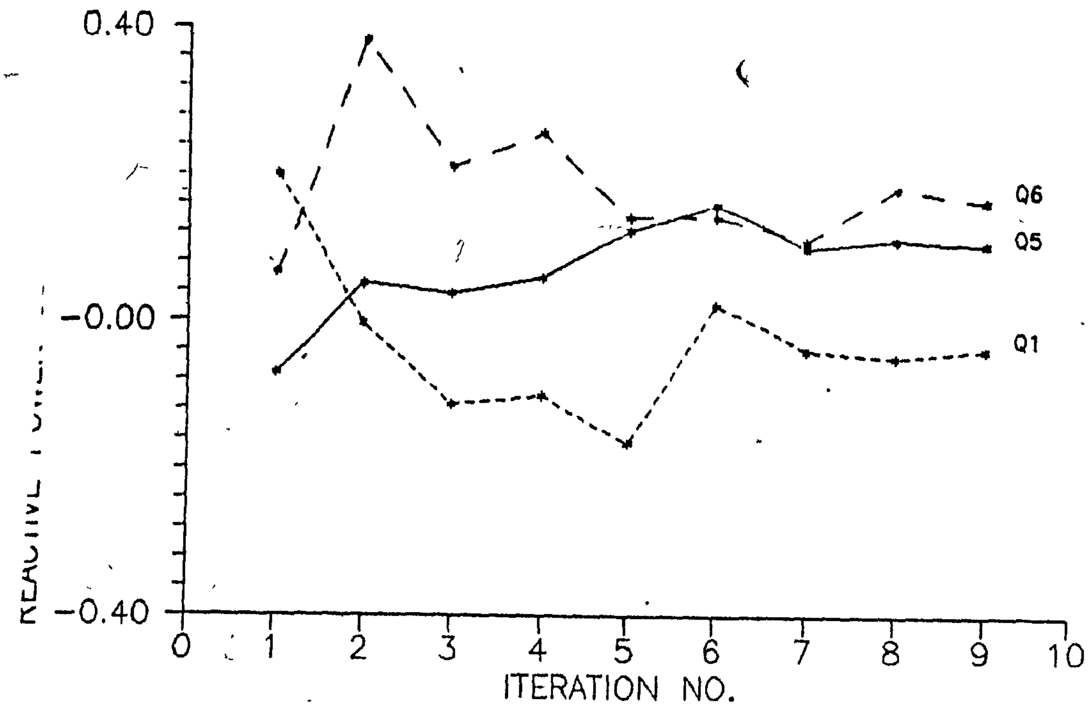


Figure 7.6.b. Feasible reactive power generations vs. the major iteration number, solution of the first load of the 6 bus system.

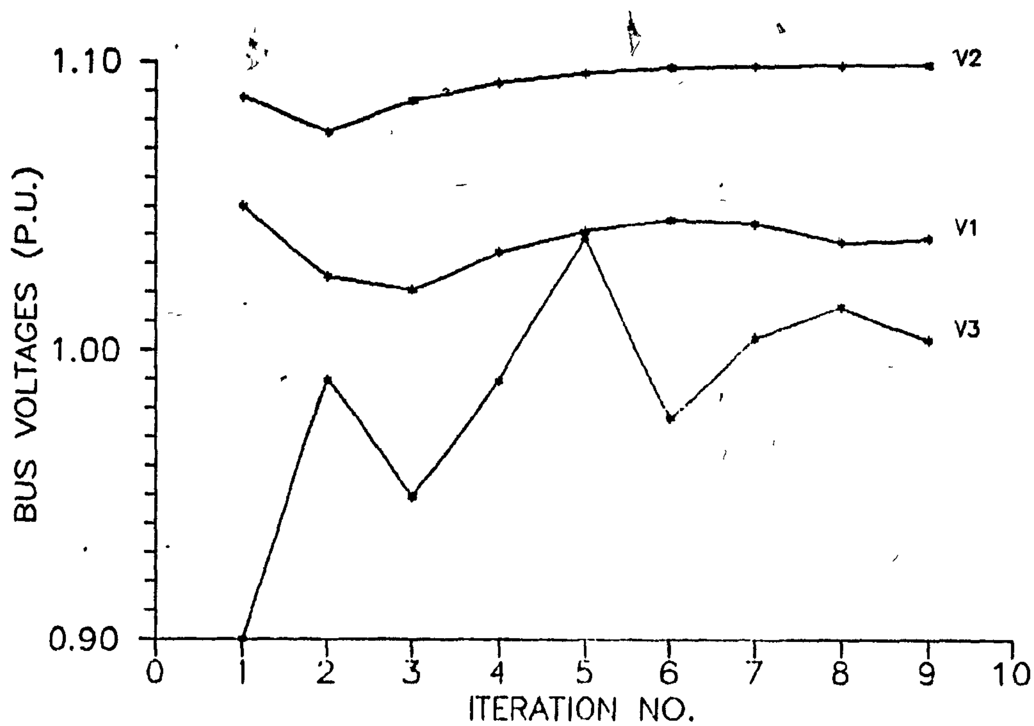


Figure 7.6.c. Feasible bus voltage magnitudes vs. the major iteration number, solution of the first load of the 6 bus system.

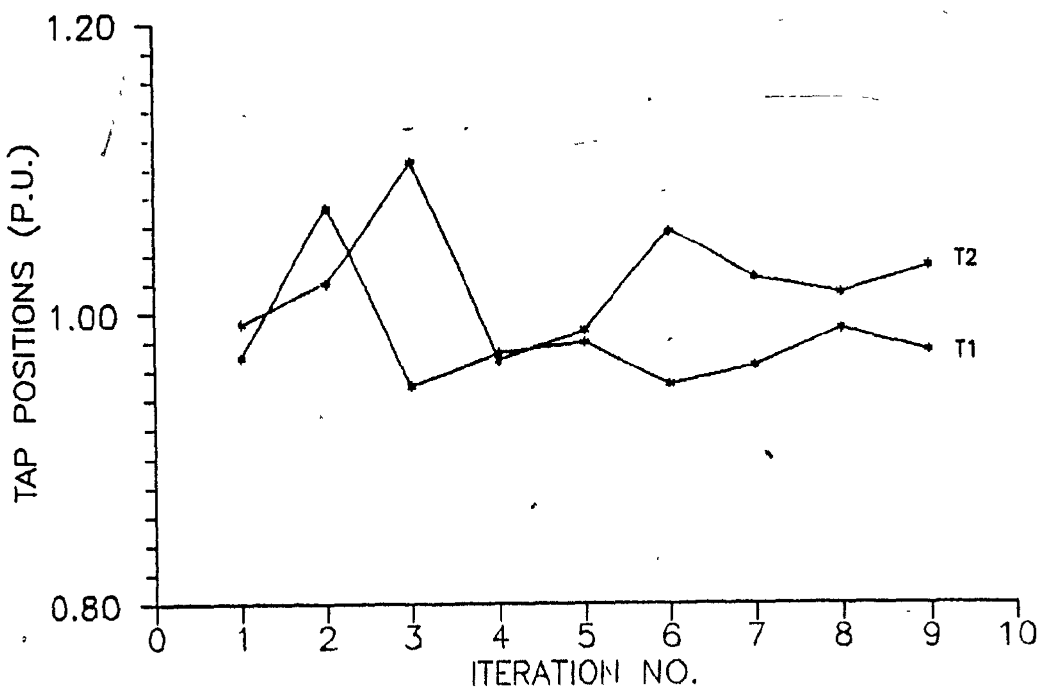


Figure 7.6.d. Feasible variable tap positions vs. the major iteration number, solution of the first load of the 6 bus system.

7.2.4 Description of the System Variables in the Load Tracking Mode

Figures 7.7 a. to e. follow the changes in the computed optimal values of the system variables in the load tracking mode, with 4% percent changes in the loads. It is important in load tracking that the computed trajectories be relatively smooth, to avoid abrupt swings in the dispatch. The results in these figures are quite satisfactory in that respect.

The real power generations in fig 7.7.a form a very smooth dispatch schedule. The corresponding optimal cost trajectory is drawn in fig. 7.7.b. The variations in the computed optima of the remaining variables are shown in fig. 7.7 c. to e. They are not as smooth as for the real power generations, but for the most part they are smooth enough. No general trend emerges for groups of variables as the load increases, since they remain more or less constant. Although the changes are slight, the direction of change usually appears quite clearly for each variable, without random swings. For example, in fig. 7.7.c. the trends for Q3, Q4 and for Q6 are clearly increasing. In fig. 7,7.d, three of the bus voltage magnitudes, V3, V4 and V5 exhibit a slightly oscillatory behaviour but their general trends are apparent. Figure 7.7.e shows that the tap positions move little throughout the load-tracking process. The taps are probably of little use until some quantity at an adjacent bus hits a bound.

The oscillations in these curves are small, but if need be, the curves of the independent variables could be smoothed before being used for dispatching.

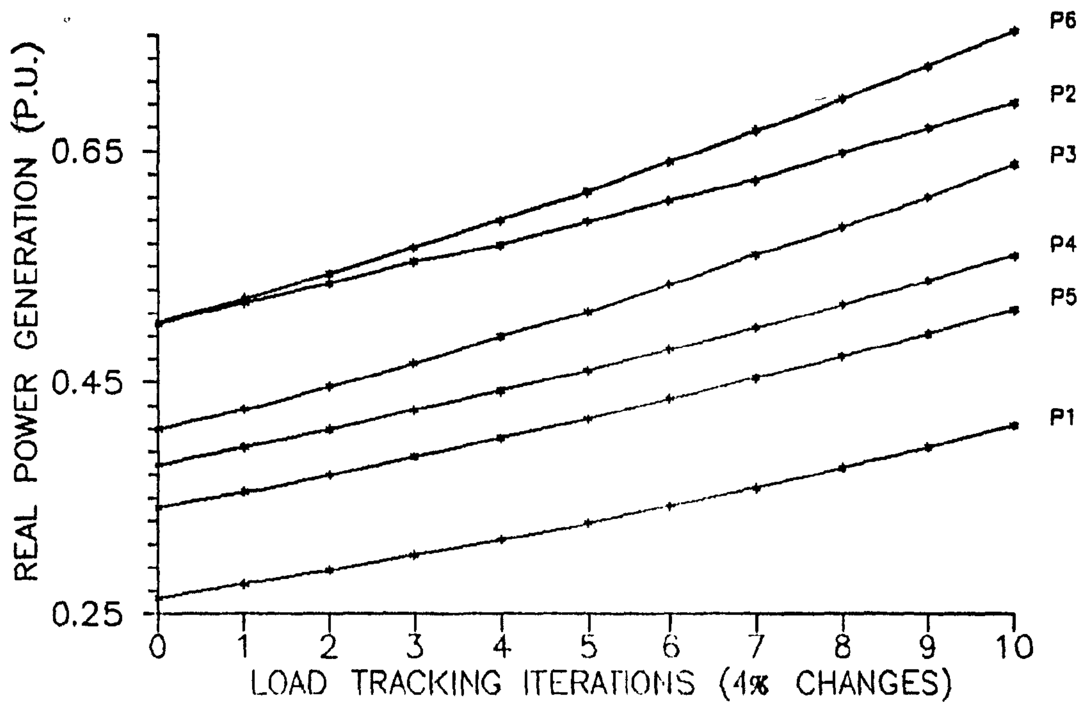


Figure 7.7.a. Real power generations vs.load, load tracking on the 6 bus system.

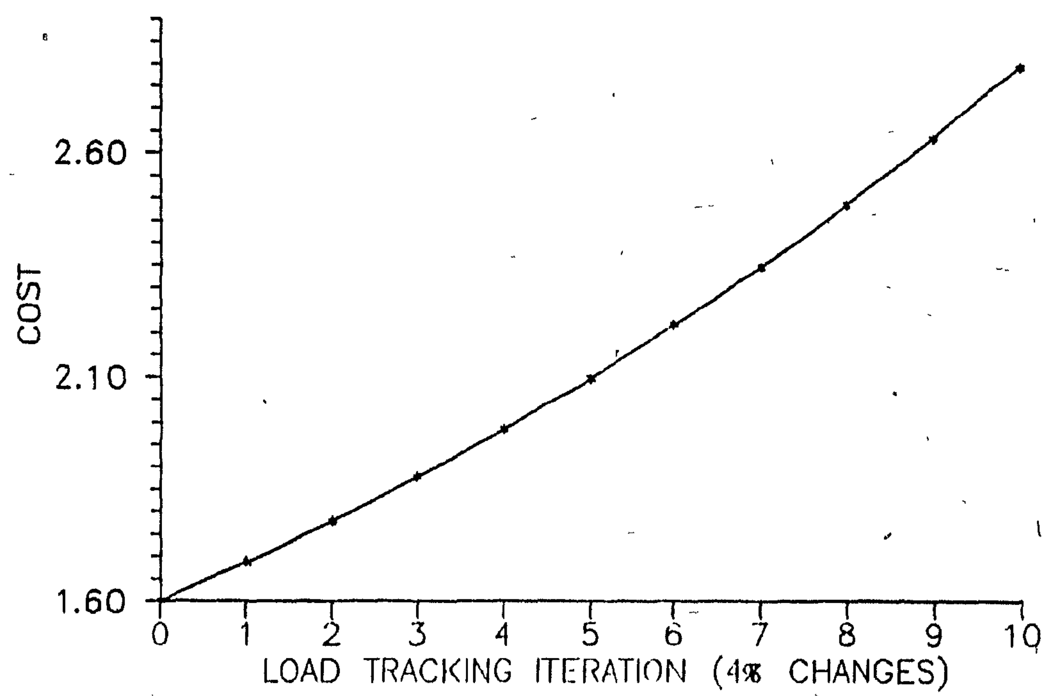


Figure 7.7.b. Optimal fuel costs vs.load, load tracking on the 6 bus system.

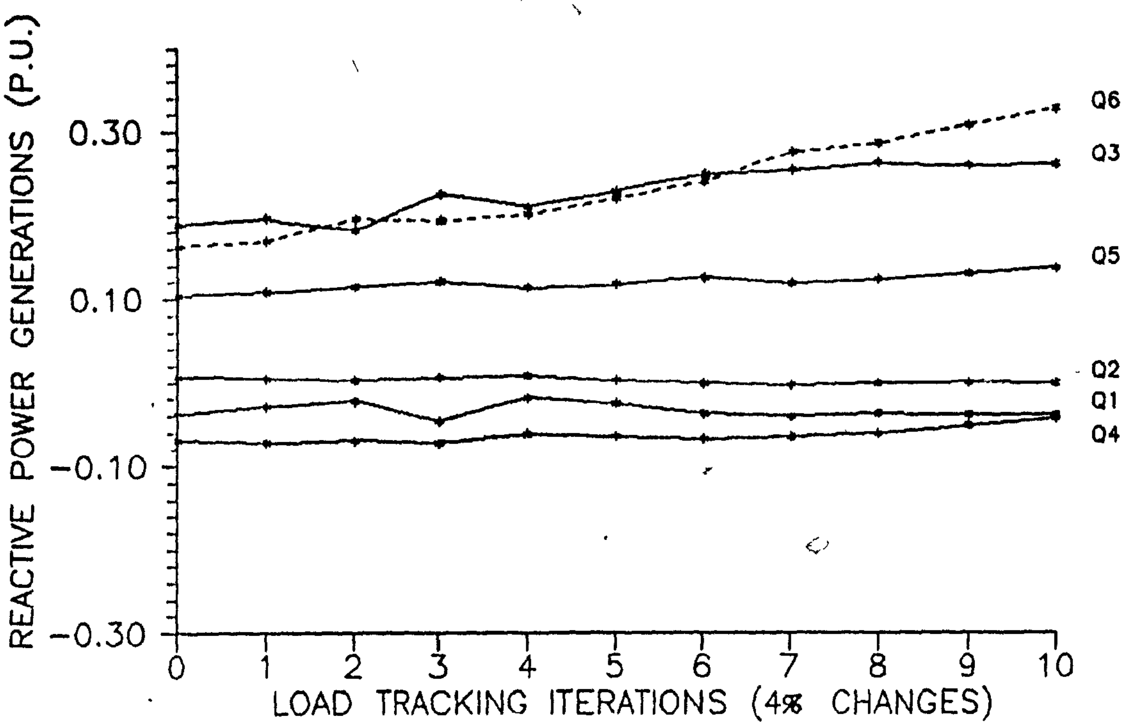


Figure 7.7.c. Reactive power generations vs.load, load tracking on the 6 bus system.

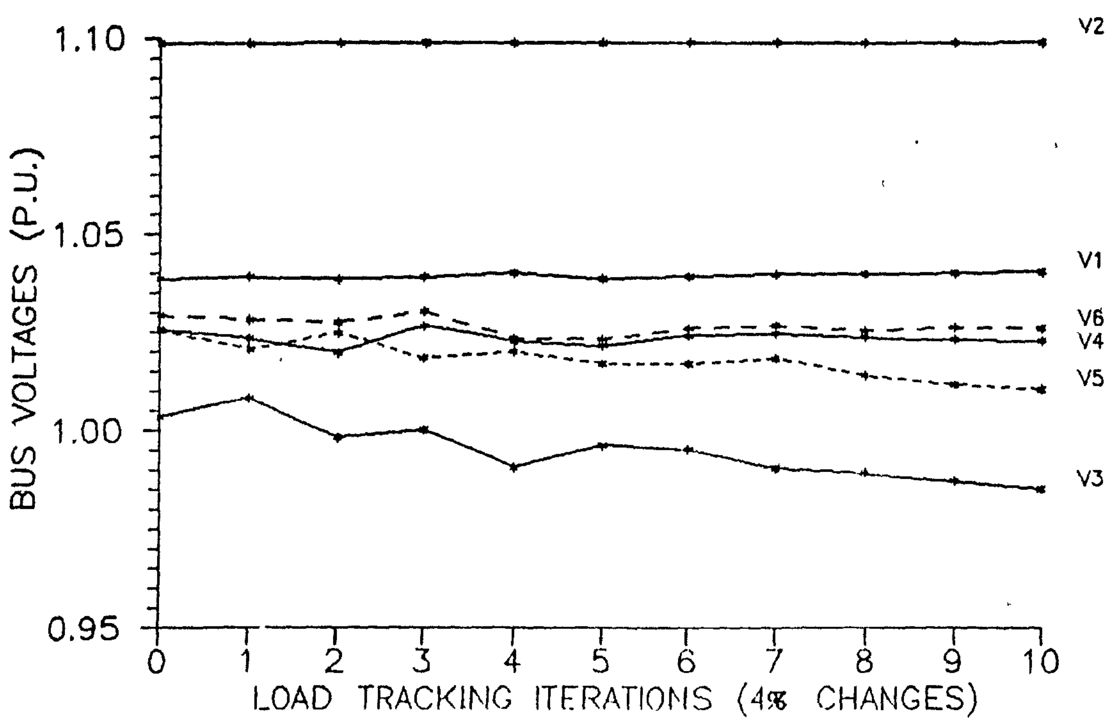


Figure 7.7.d. Bus voltage magnitudes vs.load, load tracking on the 6 bus system.

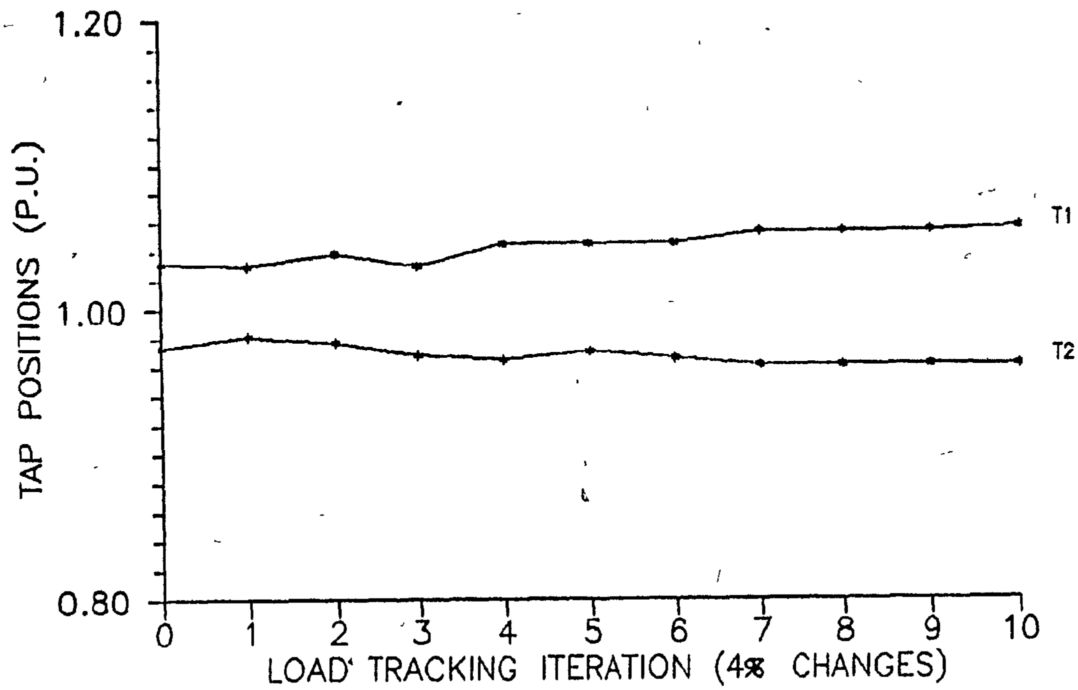


Figure 7.7.e. **Variable tap positions vs.load, load tracking on the 6 bus system.**

7.2.5. Analysis of the Sensitivity Coefficients

It was stated in section 3.4.4.5 that when an optimum is situated on a fold line, some optimal transparent variable settings, although unique, are very difficult to locate. One indication of the presence of the fold line is the very small values of some of the sensitivity coefficients in the power balance equation. We surmised that numerical problems would occur if from one major iteration to the next a very small coefficient changes its sign; then in the simple problem solutions of the subproblem, the corresponding transparent variable can oscillate between its upper and lower bounds. To avoid this problem in our program, the transparent variables were set to their expansion point values when the magnitude of its sensitivity coefficient was below 10^{-3}.

Even though other factors enter into play in the computation of the subproblem optimum, we will try to detect the influence of these sensitivity coefficients on the sequence of subproblem solutions.

Table 7.7 provides the sensitivity coefficients of the variables portrayed in fig. 7.5.b. to d. when they were transparent variables. By construction the real power generation P1 has a sensitivity of one, and the sensitivities for the other inactive generations range from 1.015 and 1.035. The sensitivities of the transparent variables listed below can be seen to be much smaller, ranging from 0.889e-4 to 0.192, with many of them in the 10^{-3} to 10^{-2} range.

TABLE 7 7 - TEST ON THE 6 BUS SYSTEM

SENSITIVITY COEFFICIENTS OF THE VARIABLES IN FIG. 7.5

	Major iteration no.								
	1	2	3	4	5	6	7	8	9
Variable	Sensitivity coefficients								
Q1	- 154e-2	-.202e-1	-.137e-2	162e-1	-	-	- 672e-2	.399e-2	-
Q5	342e-4	722e-2	889e-4	.180e-1	-	.336e-2	- 263e-2	695e-2	- 581e-2
Q6	-	641e-2	-	.158e-1	-	.560e-3	- 259e-2	.637e-2	-.560e-2
V1	.113e 0	139e-1	-	-	.536e-1	.650e-1	608e-2	-	-
V2	-	-.150e 0	-	.103e 0	.185e-1	107e 0	.711e-1	746e-1	628e-1
V3	.192e 0	107e 0	.107e 0	.445e-1	-	- 334e-1	316e-1	441e-2	- 468e-2
T1	-	-	- 795e-1	-.134e-3	.111e-1	-	-	-	-
T2	-	-	-	-	-.519e-3	-	-	-	-

From the information in this table, the behaviour of most of the variables confirms the ideas of the previous paragraphs. The coefficients for Q5 are so small in the first few major iterations that Q5 is maintained close to its expansion point; in the last few iterations the coefficients are a bit larger and their signs oscillate. In fig. 7.5.b, Q5 exhibits the prescribed behaviour. The same can be said for the oscillatory behaviour in the last four iterations of Q6 and of V3. Little information is given in the table on the variable tap settings, but the very small coefficient for T1 in the fourth major iteration can explain the large jump from a bound to a value in the middle of its range. The stable solutions to V1 and V2 in fig. 7.5.b are also reflected in a sequence of large positive coefficients in Table 7.7. Only Q1 does not fill the prescribed behaviour.

This illustrates, at least qualitatively, the desired relationship between the behaviour of the sensitivity coefficients and the numerical stability of the subproblem solutions. As for improvements to our OPF algorithm to reduce the occurrence of numerical instability, more numerical studies are needed, but it might now be worthwhile and justifiable to try increasing the threshold level for which variables are set to the expansion points.

7.2.6 Solving the Initial Load to Very Tight Tolerances

To illustrate the importance of an accurate first solution for load tracking, the optimal solution to the initial load for the 6 bus system was solved with very tight tolerances. More importantly, this allowed the program to solve for a long sequence of subproblems without any violation in the simple problem.

Upon solution for the first load, the program had run for 80 major iterations, the last 48 of which had not seen a violation in the subproblem. In the final iteration the reduction in cost was 0.279e-7, the gap between the costs of the load flow feasible point and the subproblem was 0.242e-6, and the distance between the load flow feasible solution and the subproblem solution was 0.254e-4. The optimal cost reduced only marginally from the value of 1.6009 given in Table 7.1 to 1.60065.

It is interesting to note that the computation of the solution to the first load was finally terminated when the linear equation solver declared the load flow Jacobian singular, in the Newton-Raphson solver. This strengthens the arguments made in this thesis about the OPF optimum lying on fold lines of the load flow manifold.

The improvement in the load-tracking solutions for a sequence of ten loads following this lengthened first solution is spectacular. All the solutions except one required a single major iteration, with the remaining one requiring two. The improvement came in the fact that for the first nine loads, no breakpoint was encountered in the subproblems. Only in the tenth load, three violations requiring 14 breakpoints were required to solve the subproblem.

This illustrates the value of an accurate initial solution for load tracking. In this case the expense in solving the first load was exaggerated; about 30 major iterations would have sufficed to settle on an active set, the last 15 requiring almost no breakpoints. Hence we can state that solving an initial solution to fairly tight bounds is recommendable for load tracking.

7.3 Simulations on a 10 Bus System

The ten bus system is taken from [Adielson 1972]. In that paper several generations could be accounted for at a single bus. This feature is not available in our program however; instead, at each generation bus a single incremental cost segment was built to cover the same power and incremental cost ranges as the combined generation. The data and the schematic diagram for this system can be found in Appendix 7.2.

The number of variables in this system is as follows:

Number of buses:	10
Number of generations:	7
Number of loads:	7
Number of transmission lines:	13
Number of variable transformer taps:	5

Total number of load flow quantities: 57
Total number of load flow variables: 51

The program solved this system for an initial load, and then in two different runs, for 8 loads increased by 1 percent and 5 loads increased by 2 percent.

7.3.1 Highlights of the Solution to the 10 Bus System

In this section we present the main observations concerning the solutions of the 10 bus system. The complete set of results is presented in condensed form in section 7.3.2.

i) The overall solution

All in all, the 10 bus system showed slower convergence than the 6 bus system, although no major numerical difficulty was encountered. The results from the solution to the first load are presented in Table 7.8. In brief, here are the major points:

- 8 major iterations were required to converge to the prescribed tolerances.
- Many breakpoints were required to solve for the first four subproblems. After that, the subproblem's active set settled down, and the next three subproblems produced no breakpoint. The final subproblem produces only a few breakpoints. This is a "nice" behaviour.
- The values of the convergence criteria started quite small, and their reduction was slow. That is because of the application of a very small step size, as of the third major iteration.
- The large step size reduction warrants some explanation. Even though the step size was repeatedly reduced in the third major iteration, the costs of the feasible points produced by the Newton-Raphson solver were always higher than that of the previous expansion point. Then the alternate, slackless load flow was used to find a sequence of cheaper but usually infeasible points. The slackless load flow finally found a cheaper

feasible point with a step size of 0.176e-2. All this indicates that the optimum was not far away from the suggested expansion points.

- Many Newton-Raphson iterations (32) were computed in the search for the appropriate step size in the third major iteration. Other than that, very few iterations were required.
- The cost of the initial guess was very close to the optimal cost. This can be seen in fig. 7.8, along with the curve of the subproblem costs. In this case the gap between the two curves hardly decreased.
- The computation time of 55.59 sec. is almost 2.5 times the time required by the 6 bus system. The four long subproblems and especially the many Newton-Raphson iterations in the third major iteration contribute the most to this computation time.

The load-tracking solutions for the 10 bus system also required considerably more computational effort than for the 6 bus system. This is illustrated in Table 7.9 for 1% load variations and in Table 7.10 for 2% load variations.

- Most loads were solved in 2 or 3 major iterations (on average one more iteration than for the 6 bus case), but in both load tracking runs the second load required many iterations (8 and 9).
- The number of breakpoints is quite large for most loads.
- Many Newton-Raphson iterations were computed for some loads, usually when step sizes had to be reduced.
- Despite the relatively large computational effort for the load-tracking solutions, the computation times were all better than for the initial load. The timings range from 3.57 sec. to 43.45 sec., with an average of 16.70 sec. This average is 30% of the time required to solve for the first load. The average solution times per load for the two load-tracking runs are almost similar, at 16.15 and 17.61 sec.
- The optimal solutions from the 2 load-tracking runs compare very well. Of the four cost values which can be compared, the largest discrepancy was 0.1% and the average discrepancy was 0.04%.

ii) The first subproblem solution

The 22 breakpoints required to solve the first subproblem are given in Table 7.11. The pattern of breakpoints occuring in pairs is evident here. Also noticeable is the clustering of most of the breakpoints towards the end of the continuation process. This is also the case for other tests with many breakpoints.

Optimal solution trajectories for the variables are presented in fig. 7.9 a. to f.. The behaviour of the different types of variables is as described in the results of the 6 bus system. Most noticeable, as was the case for the third subproblem of the 6 bus system, are the many breakpoints towards the end of the continuation process and the sharp changes in certain transparent voltages and taps, as seen in fig. 7.8.c. and e.. The sharp changes might be due to the magnitude of the largest violation in this continuation process, which was 24.7 p.u.. The variable with the largest violation, Q4, can be seen in fig. 7.8.b. to be following its moving lower bound. Some line flows in this test, particularly L11 and L10, varied substantially.

iii) The sequences of subproblem and load flow feasible solutions

Large changes occur in the set of independent variables over the first four subproblems. As a result, other than for real power generations, there are many changes in the subproblem solutions from one iteration to the next. This is illustrated in fig. 7.10 a. to d. The clutter on the left sides of the graphs of fig. 7.10 b. to d., for the Q's, V's and T's, was left intentionally, to show the large swings in all the variables. The next three subproblems, where no breakpoint was encountered, resulted in very small changes. Finally, the last subproblem with a few breakpoints resulted in some fairly large changes. This indicates that even though the problem is considered solved for this load, the subproblems have not settled on an active set. That is verified in Tables 7.9 and 7.10, with the subsequent loads picking up many breakpoints.

The sequence of load flow feasible points mimicked the subproblem solutions until the step size dropped to 0.176e-2. After that, as expected the load flow feasible values moved very little. This is illustrated in fig. 7.11. a. to d..

vi) The sequences of optimal solutions in the load-tracking mode

Figures 7.12 a. to e. follow the changes in the computed optimal values of the system variables in the load tracking mode, with 1% percent changes in the loads. Despite the small load variations, trends in the dispatching are clear. For example, in fig. 7.12.a. the real power generation P5 picks up a larger proportion of the load as it increases, while most of the other P's vary little. Once again, the real power dispatch curves vs. load are very smooth. The resulting optimal cost vs. load curve is shown in fig. 7.12.b. The other variables seem to follow a general trend more closely in this test. In fig. 7.12.c., except for the single jump in Q4 after the second load, the reactive power generations are just about constant or show a slight lowering trend. In fig. 7.12.d, the bus voltages show a general increasing trend. Again, in fig. 7.12.e., the variable transformer taps vary little.

7.3.2 Tables and Graphs of the Results of the 10 Bus System

Tables 7.8 to 7.11 and Figures 7.8 to 7.12 on the following pages contain the results of the 10 bus system.

TABLE 7.8 - TEST ON THE 10 BUS SYSTEM

SUMMARY OF THE ALGORITHM'S PERFORMANCE

SOLUTION FOR THE FIRST LOAD

Major iteration no.	1	2	3	4	5	6	7	8
SUBPROBLEM								
Violations in the subproblem at the easy optimum	7	9	11	9	0	0	0	2
Breakpoints in the continuation process	22	28	26	22	0	0	0	4
SEARCH FOR LOAD FLOW FEASIBLE POINT								
Newton-Raphson iterations in finding load flow feasible pt.	4	2	32	0	1	1	1	1
Relative reduction in cost of load flow feasible pt.	-	.541e-3	.194e-3	.367e-7	.102e-4	.102e-4	.201e-3	.194e-3
Relative gap between costs of load flow feasible pt and subproblem solution	.764e-1	.733e-1	.213e-2	578e-2	.577e-2	.570e-2	562e-2	.553e-2
Relative norm of the difference between the variables at the load flow feasible point and the subproblem solution	.494	.267	.197	.199	.193	193	.192	.194
Step size in searching feasible load flow pt	.45	.45	.176e-2	.176e-2	176e-2	176e-2	176e-2	.176e-2

Initial cost 13 085   Optimal cost 13 074   Computation time: 55.59 sec.

TABLE 7.9 - TEST ON THE 10 BUS SYSTEM

SUMMARY OF THE ALGORITHM'S PERFORMANCE

SOLUTIONS FOR THE SUBSEQUENT LOADS - 1% VARIATIONS IN LOAD

Load no.	1	2	3	4	5	6	7	8
No. of major iterations	2	8	2	2	2	2	2	2
Total number of breakpoints in the subproblems	4	50	42	36	40	58	49	12
Total number of Newton-Raphson iterations	4	10	4	4	4	4	63	4
Optimal cost	13.215	13.346	13.491	13.640	13.790	13.943	14.100	14.259
Computation time (sec.)	3.84	22.47	13.07	10.99	11.20	17.85	43.45	6.26

TABLE 7.10 - TEST ON THE 10 BUS SYSTEM

SUMMARY OF THE ALGORITHM'S PERFORMANCE

SOLUTIONS FOR THE SUBSEQUENT LOADS - 2% VARIATIONS IN LOAD

Load no.	1	2	3	4	5
No. of major iterations	2	9	3	3	2
Total number of breakpoints in the subproblems	4	90	52	38	50
Total number of Newton-Raphson iterations	4	12	5	38	4
Optimal cost	13.360	13.639	13.940	14.253	14.581
Computation time (sec.)	3.57	30.49	15 76	26.91	11.32

TABLE 7.11 - TEST ON THE 10 BUS SYSTEM

A SUMMARY OF BREAKPOINTS ENCOUNTERED IN THE FIRST SUBPROBLEM

θ	Variable name and type		Cause of breakpoint
0.0000	Q4	dependent	most violated dependent constraint set to its moving lower bound.
.8810 e-4	T4	transparent	released from its lower bound.
0.0999	T4	transparent	set to its upper bound.
0.1000	V4	transparent	released from its lower bound.
0.3697	V3	dependent	set to its upper bound.
0.3697	T3	transparent	released from its lower bound.
0.5324	V2	dependent	set to its upper bound.
0.5324	T2	transparent	released from its lower bound.
0.8562	Q5	dependent	set to its lower bound.
0.8563	T4	transparent	released from its upper bound.
0.9313	Q9	dependent	set to its moving lower bound.
0.9313	V6	transparent	released from its upper bound.
0.9458	T4	transparent	set to its upper bound.
0.9458	Q5	dependent	released from its lower bound.
0.9584	Q6	dependent	set to its lower bound.
0.9585	V3	dependent	released from its upper bound.
0.9619	Q5	dependent	set to its lower bound.
0.9619	T1	transparent	released from its upper bound.
0.9720	V3	dependent	set to its upper bound.
0.9720	V2	dependent	released from its upper bound.
0.9850	T2	transparent	set to its upper bound.
0.9851	T4	transparent	released from its upper bound.

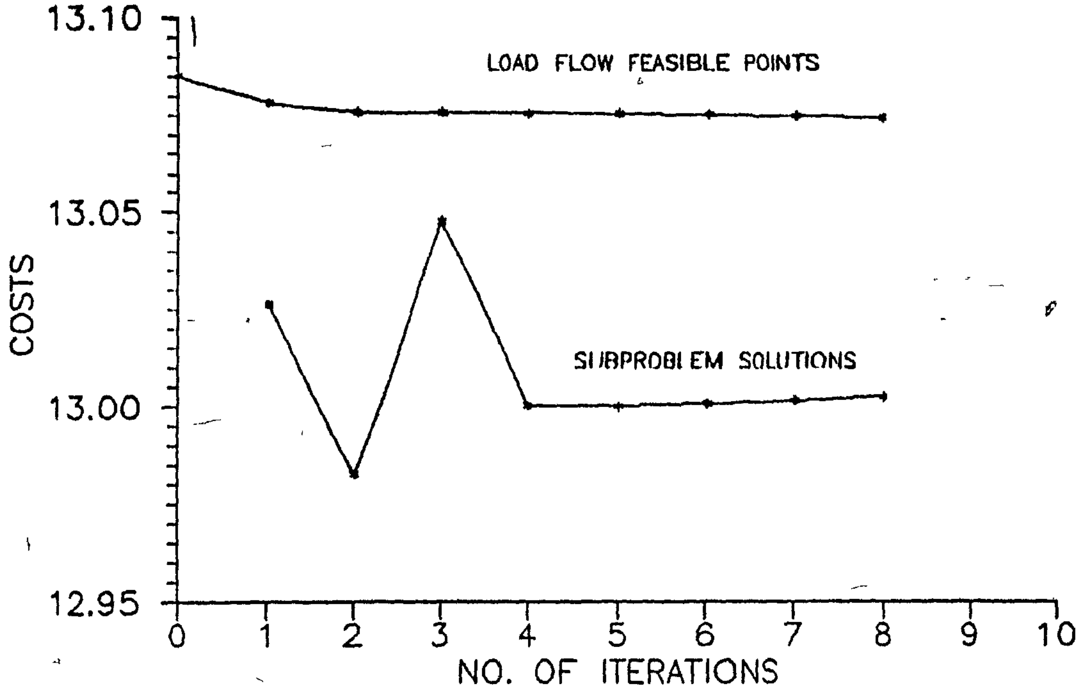


Figure 7.8. Costs of the load flow feasible points and the subproblem solutions at each major iteration, solution to the 10 bus system.

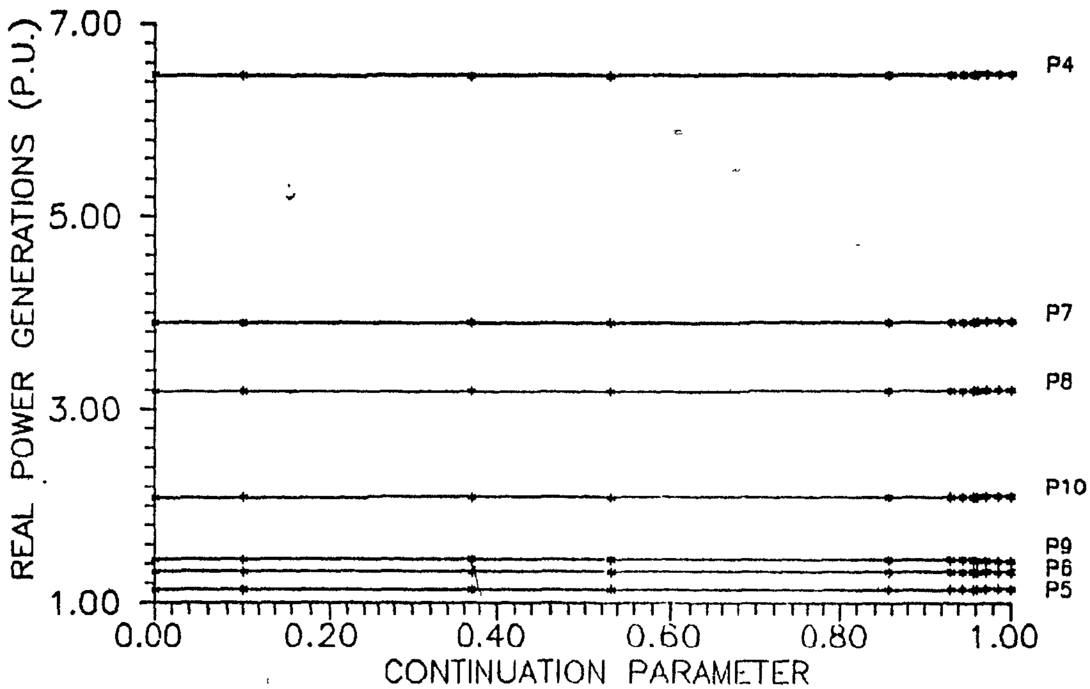


Figure 7.9.a. Real power generations vs. the continuation parameter, first subproblem in the solution of the 10 bus system.

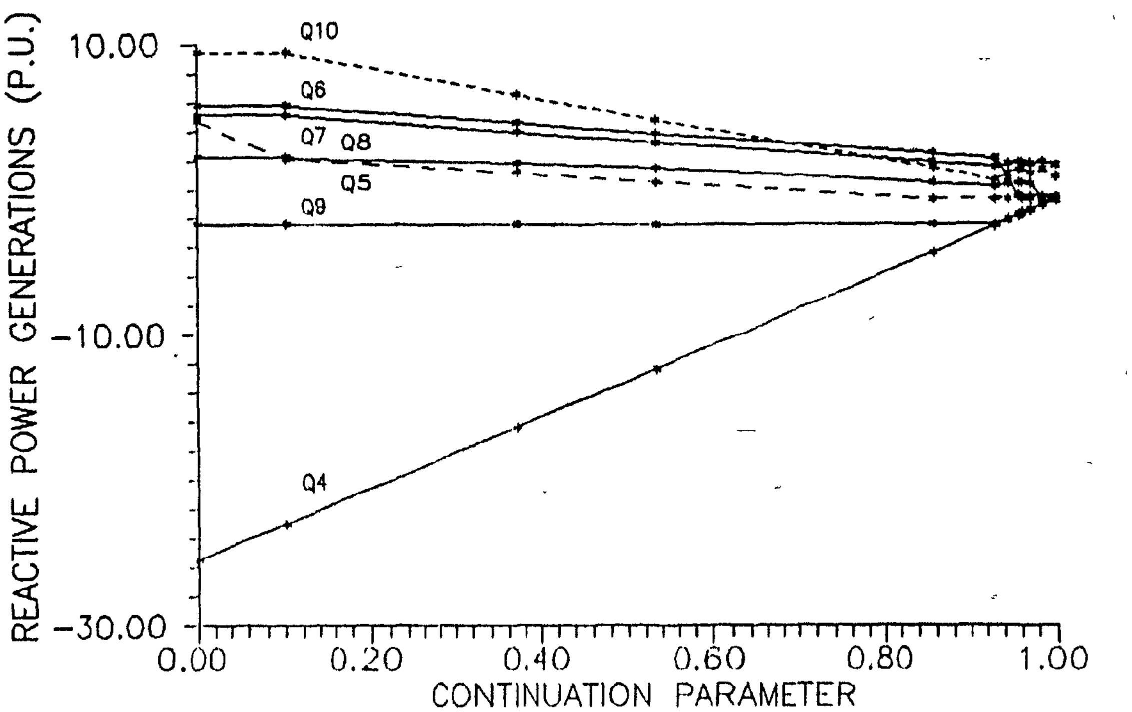


Figure 7.9.b. Reactive power generations vs. the continuation parameter, first subproblem in the solution of the 10 bus system.

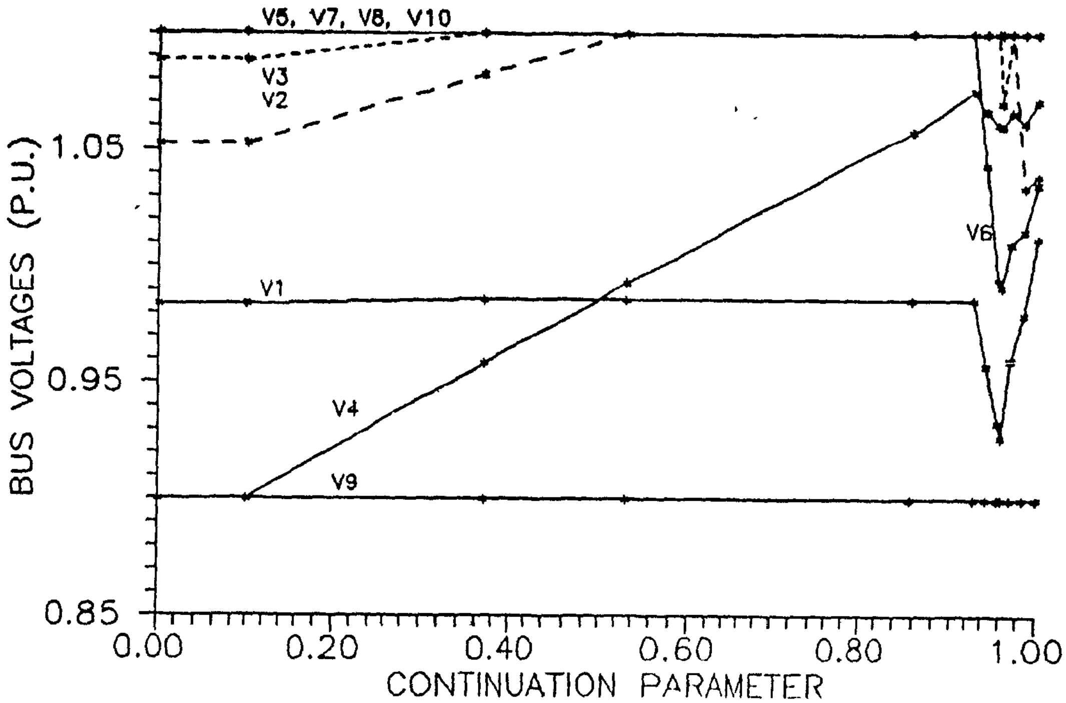


Figure 7.9.c. Bus voltage magnitudes vs. the continuation parameter, first subproblem in the solution of the 10 bus system.

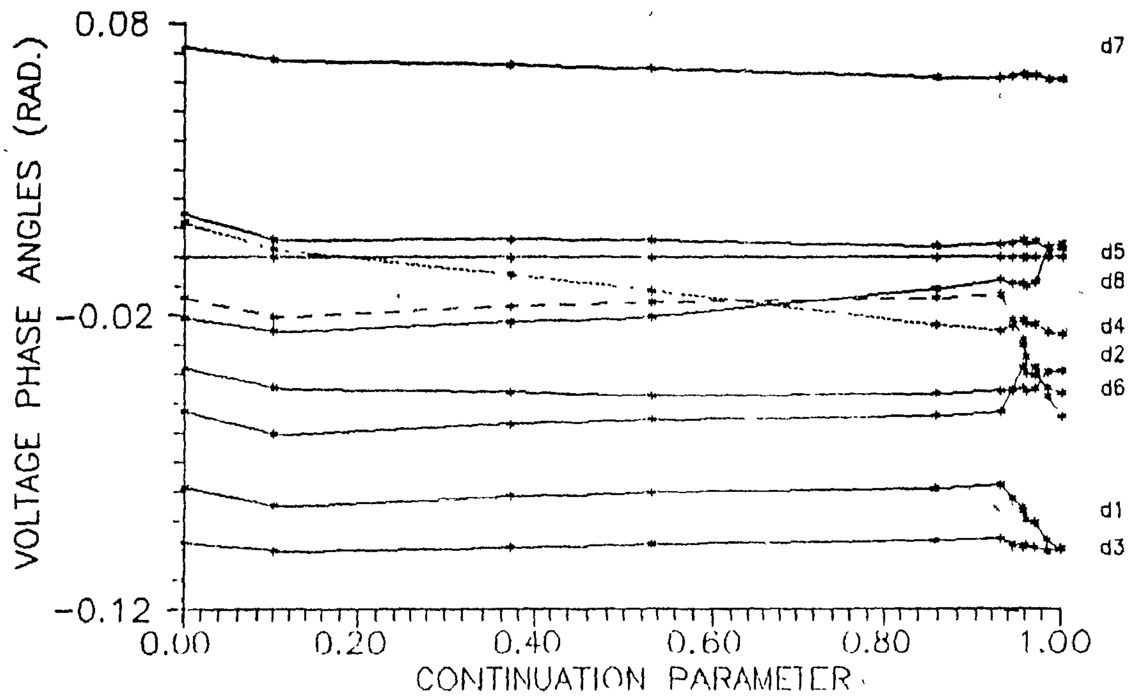


Figure 7.9.d. Bus voltage phase angles vs. the continuation parameter, first subproblem in the solution of the 10 bus system.

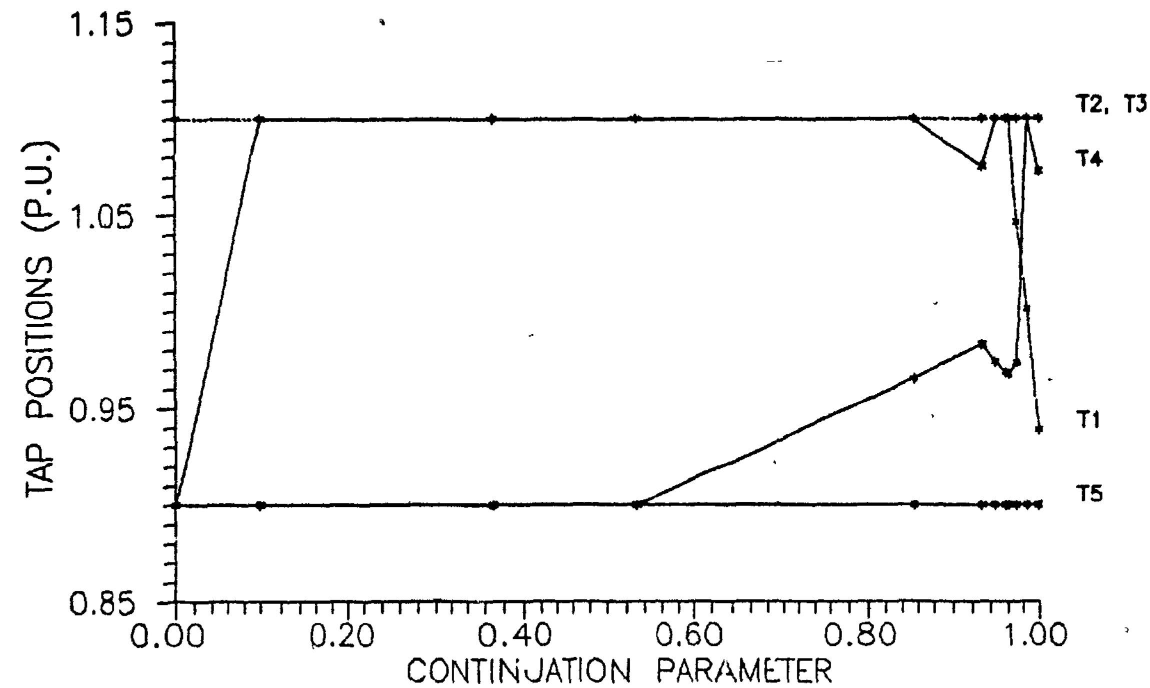


Figure 7.9.e. Variable transformer tap positions vs. the continuation parameter, first subproblem in the solution of the 10 bus system.

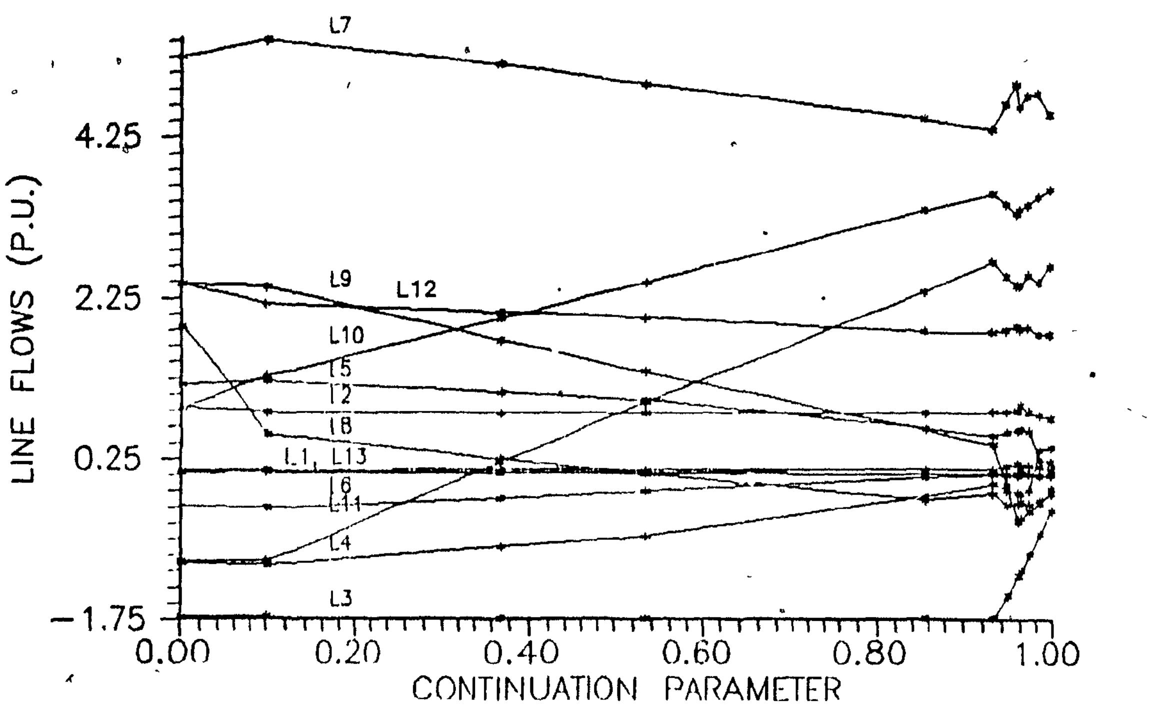


Figure 7.9.f. Line flows vs. the continuation parameter, first subproblem in the solution of the 10 bus system.

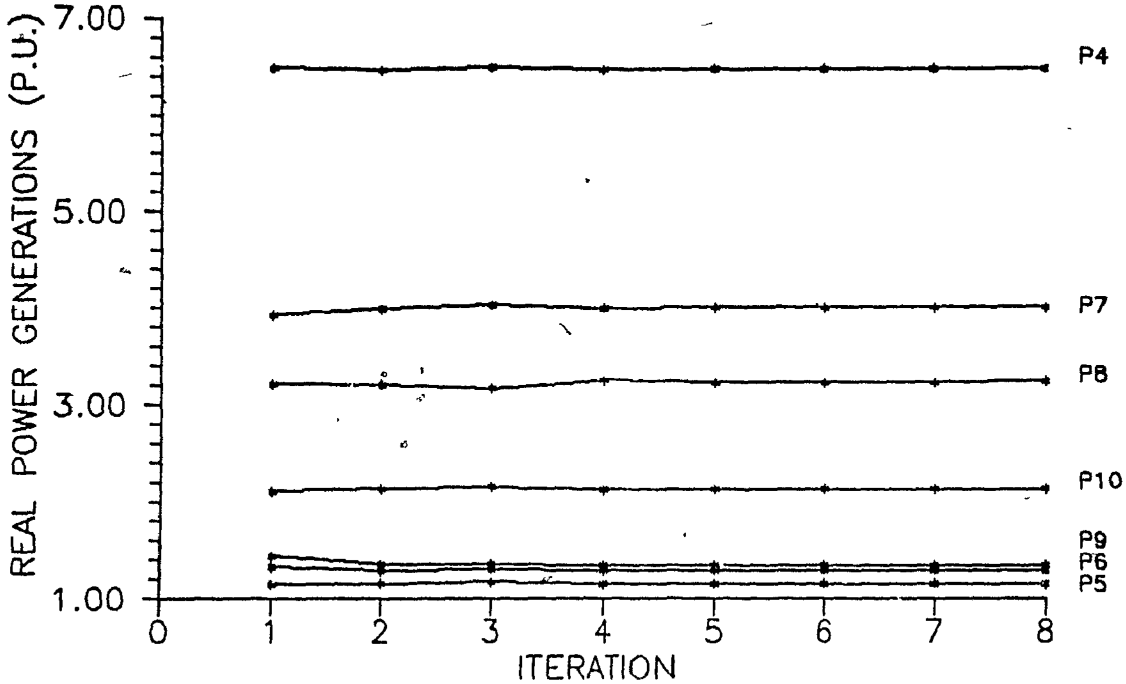


Figure 7.10.a. Subproblem real power generations vs. the major iteration number, solution of the first load of the 10 bus system.

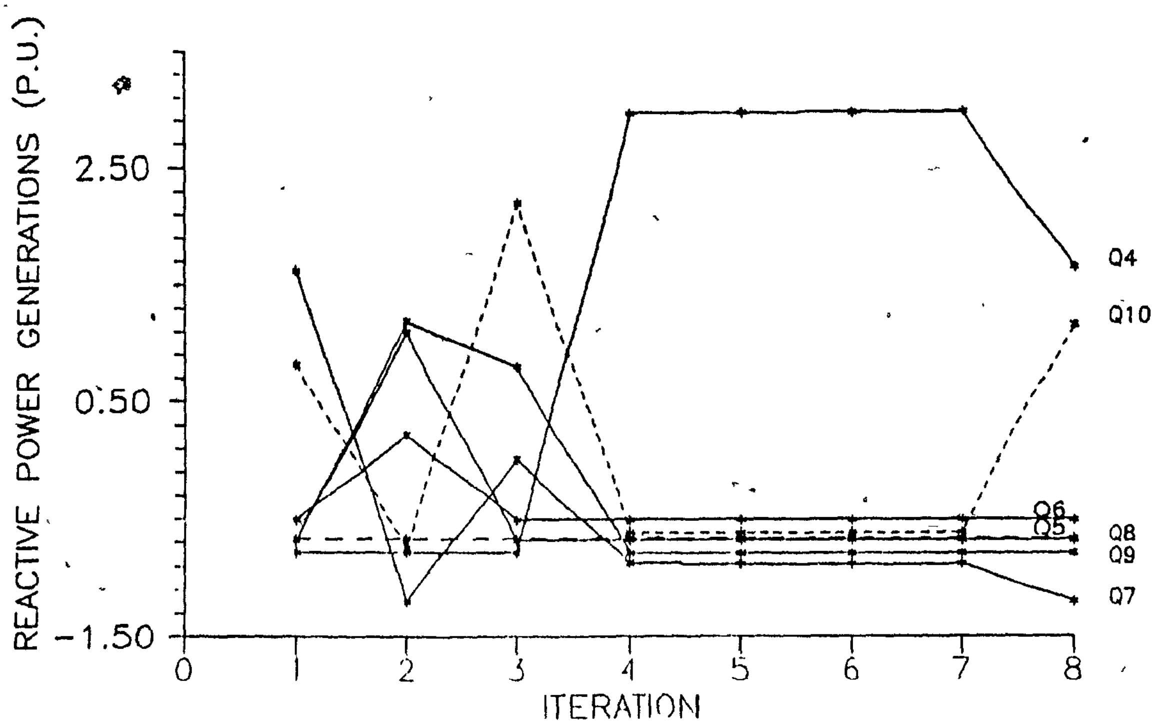


Figure 7.10.b. Subproblem reactive power generations vs. the major iteration number, solution of the first load of the 10 bus system.

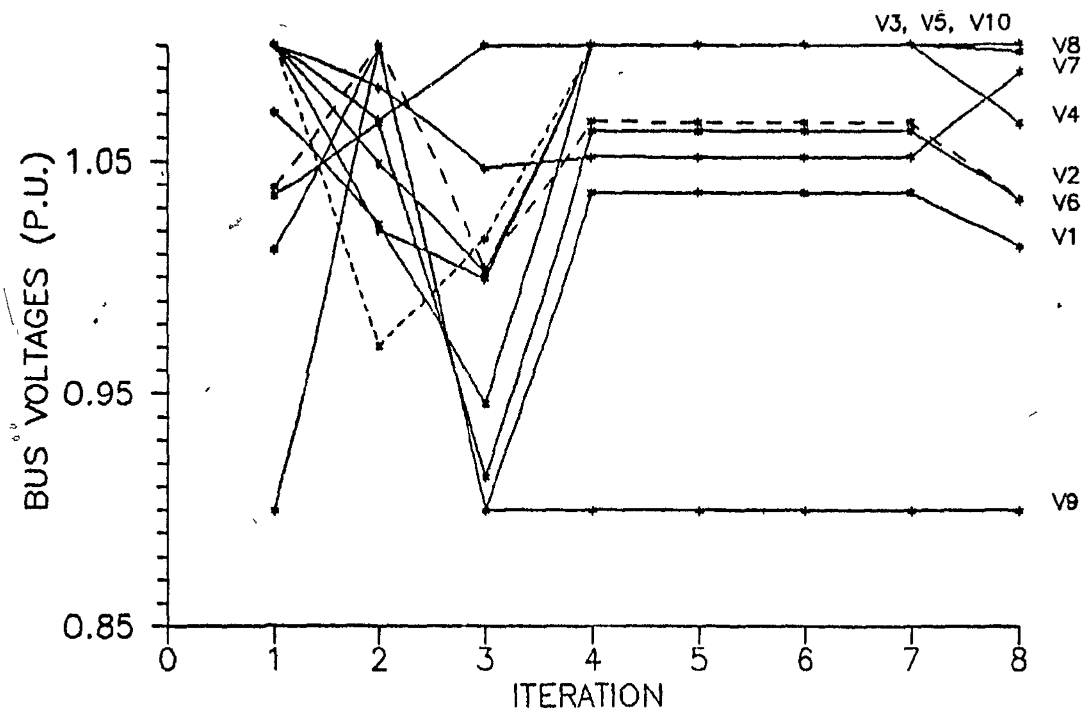


Figure 7.10.c. Subproblem bus voltage magnitudes vs. the major iteration number, solution of the first load of the 10 bus system.

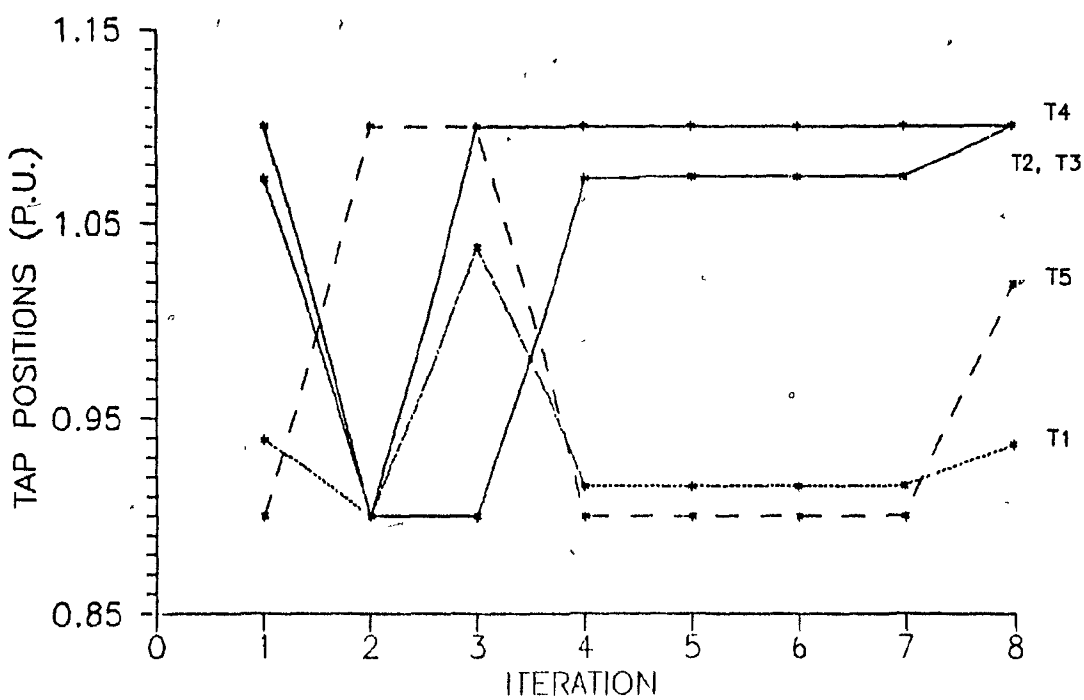


Figure 7.10.d. Subproblem variable tap positions vs. the major iteration number, solution of the first load of the 10 bus system.

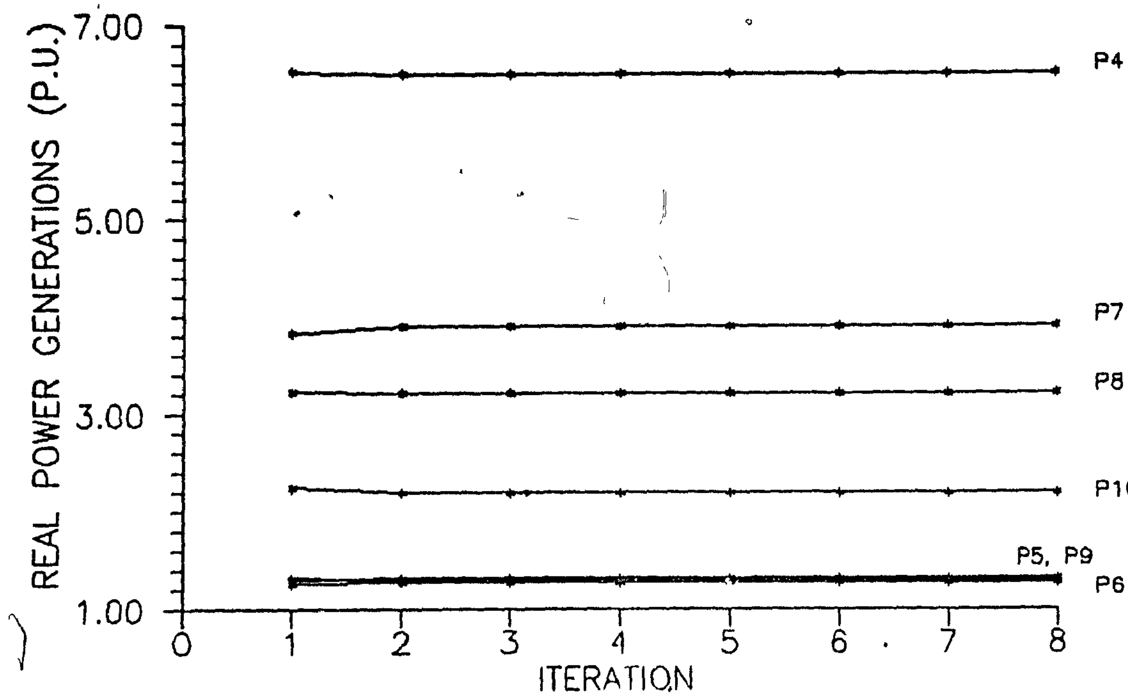


Figure 7.11.a. Feasible real power generations vs. the major iteration number, solution of the first load of the 10 bus system.

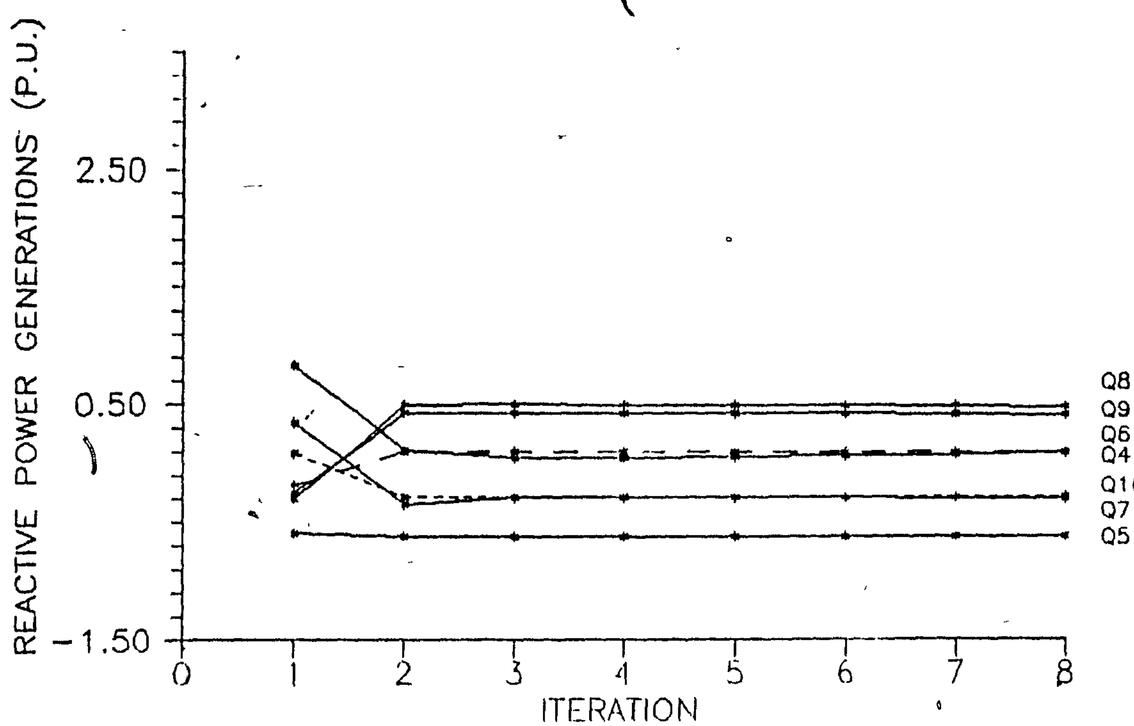


Figure 7.11.b. Feasible reactive power generations vs. the major iteration number, solution of the first load of the 10 bus system.

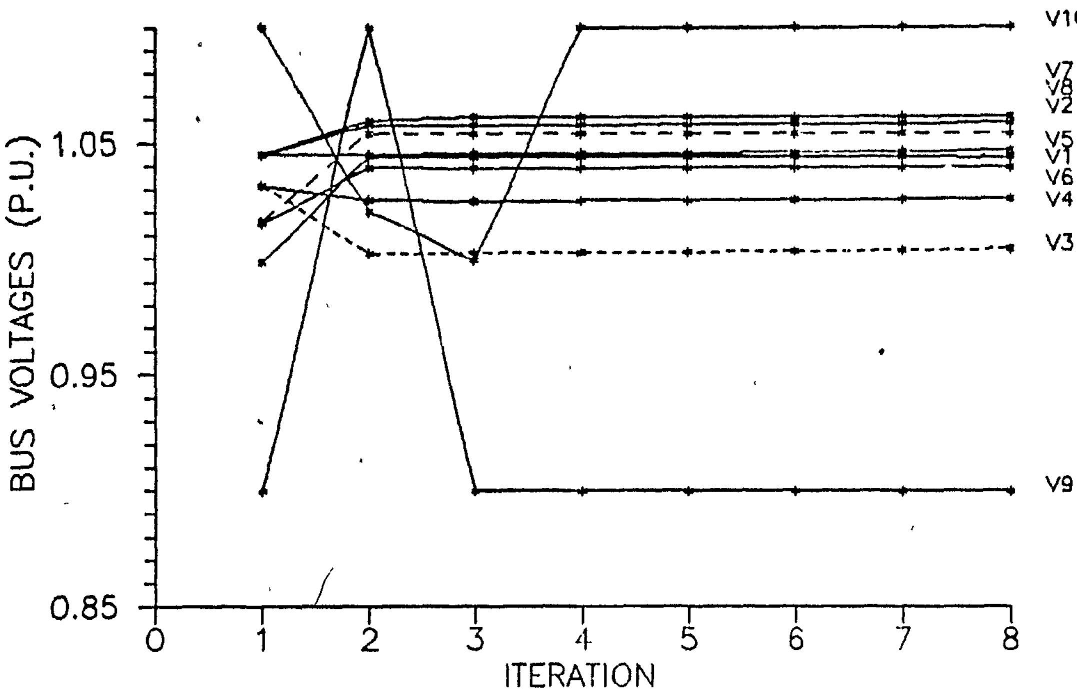


Figure 7.11.c. Feasible bus voltage magnitudes vs. the major iteration number, solution of the first load of the 10 bus system.

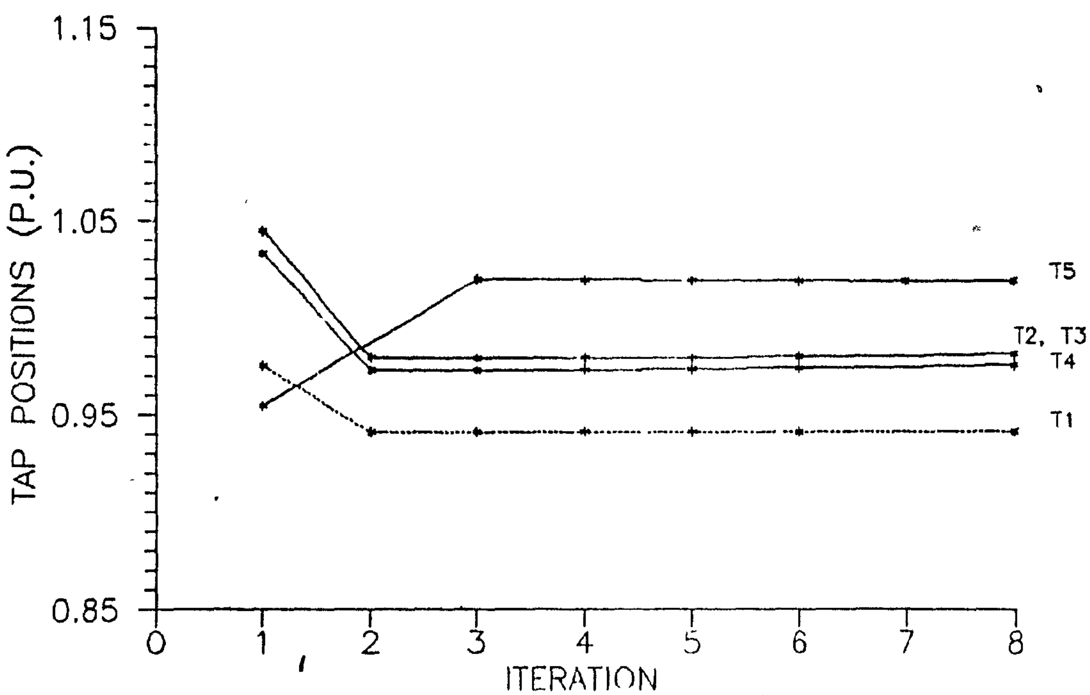


Figure 7.11.d. Feasible variable tap positions vs. the major iteration number, solution of the first load of the 10 bus system.

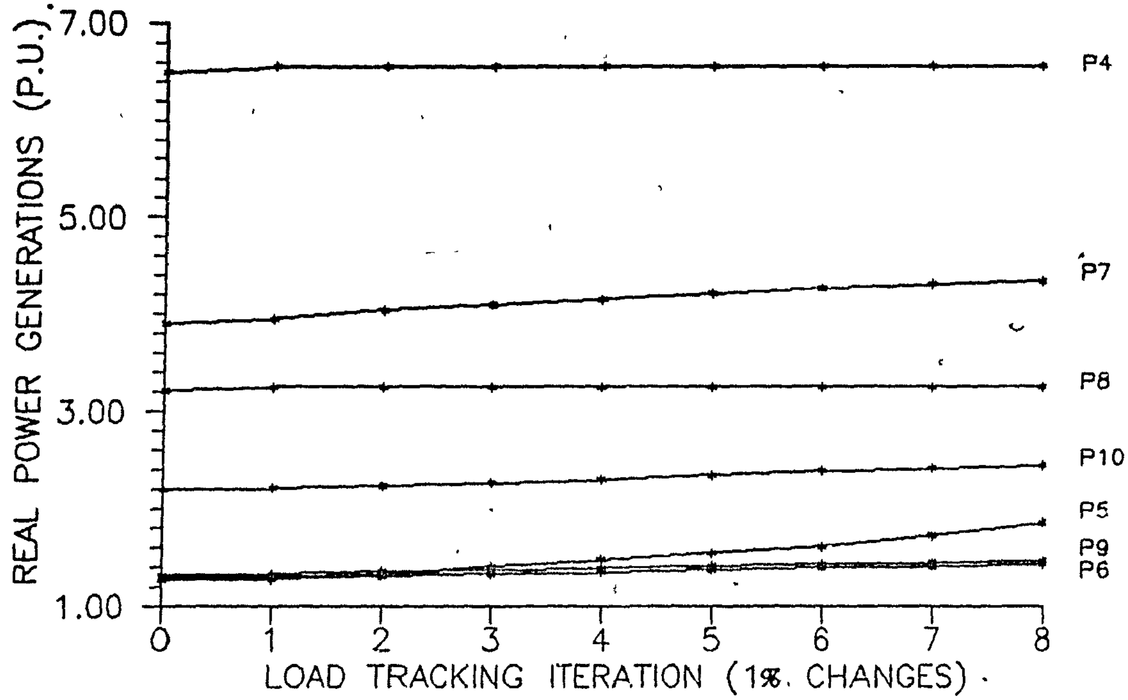


Figure 7.12.a. Real power generations vs. load, load tracking on the 10 bus system.

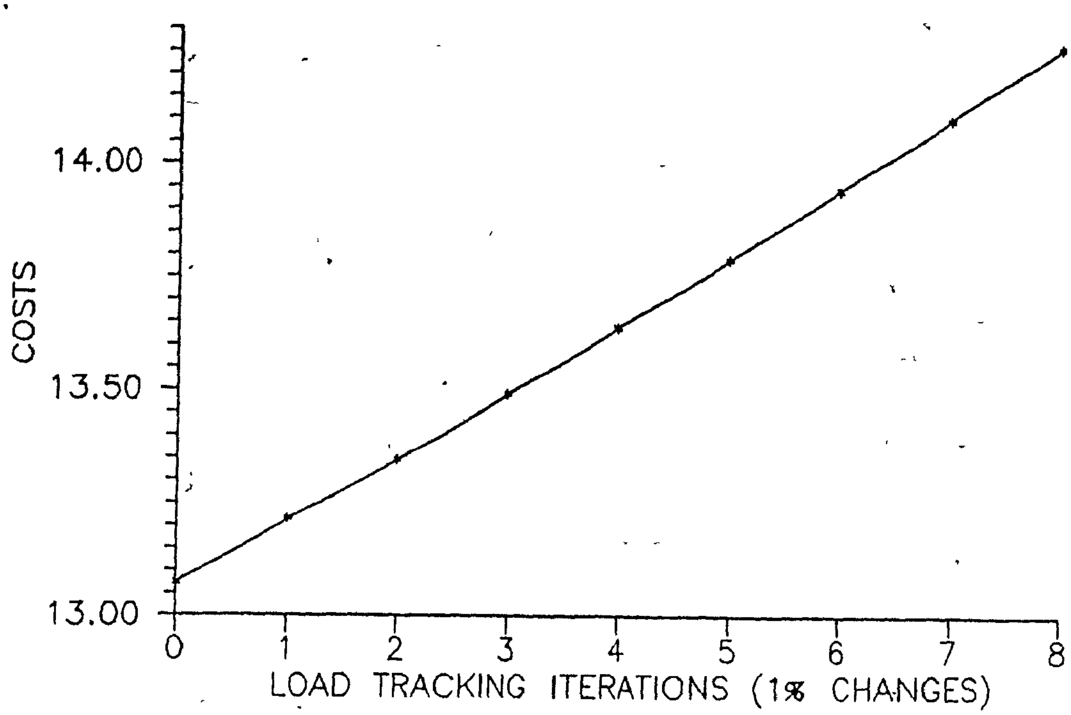


Figure 7.12.b. Optimal fuel costs vs. load, load tracking on the 10 bus system.

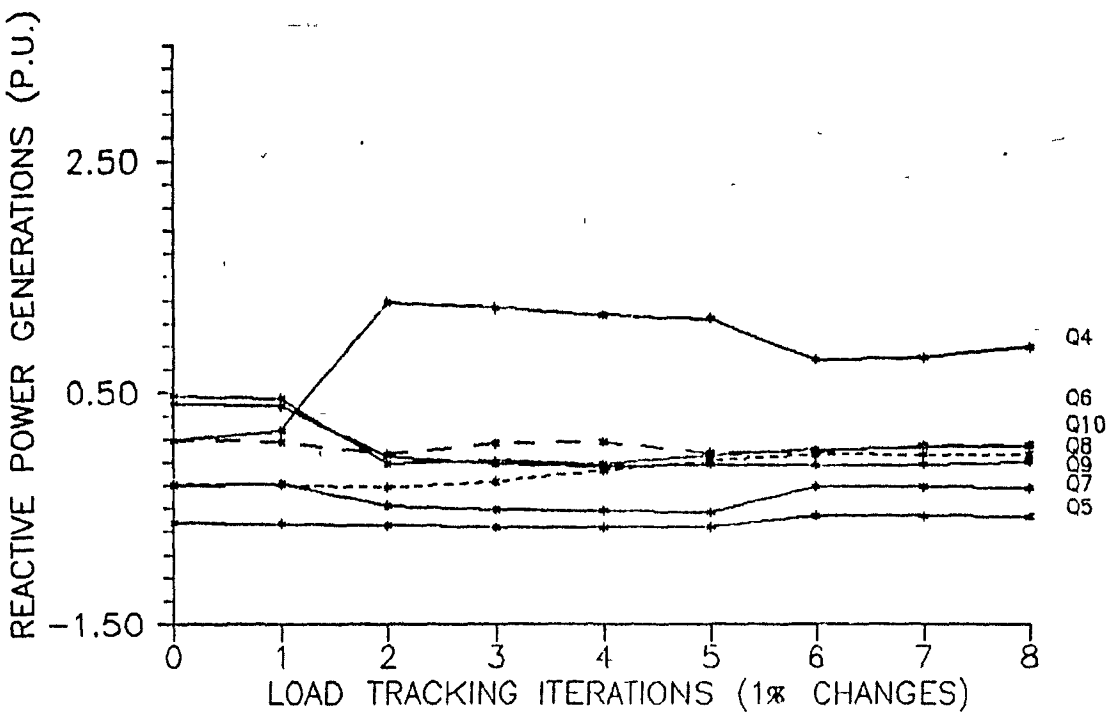


Figure 7.12.c. Reactive power generations vs.load, load tracking on the 10 bus system

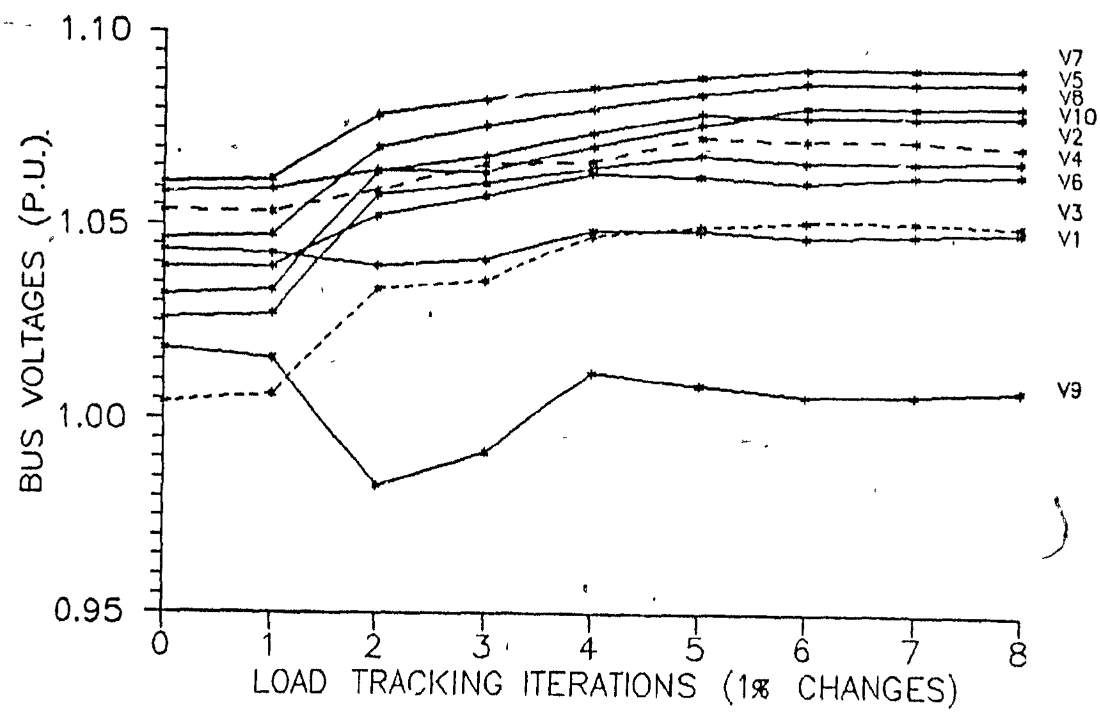


Figure 7.12.d. Bus voltage magnitudes vs.load, load tracking on the 10 bus system.

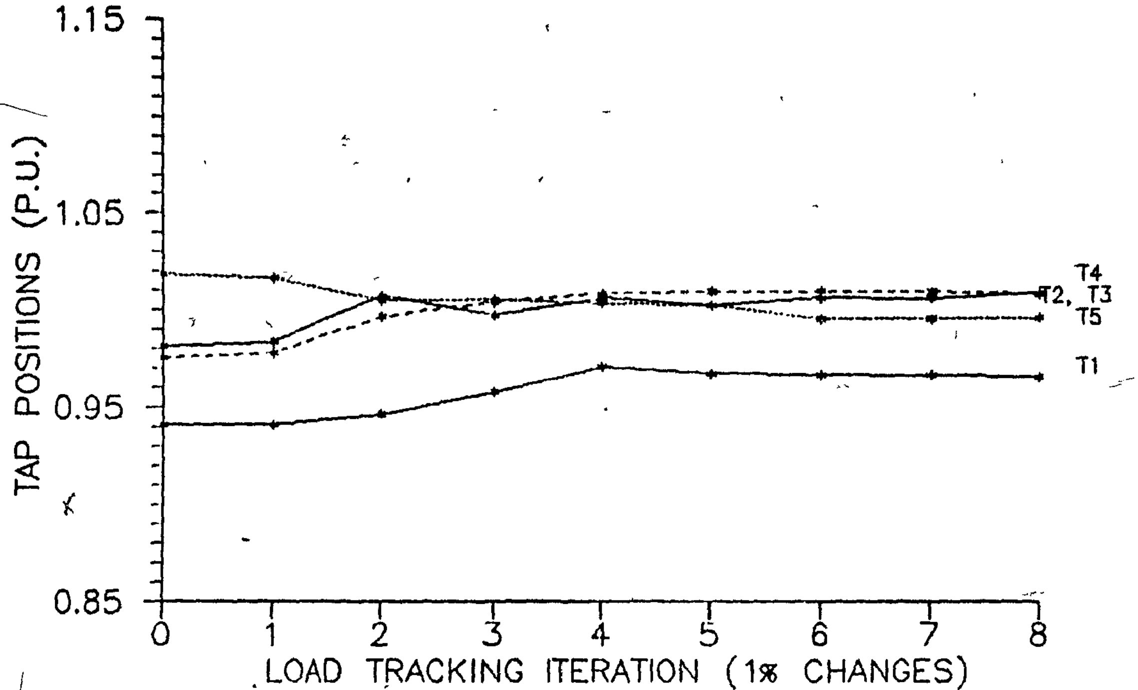


Figure 7.12.e. Variable tap positions vs.load, load tracking on the 10 bus system.

7.4 Simulation on a 30 Bus System

The 30 bus system is an adaptation of the IEEE 30 bus standard load flow test system taken from [Alsac & Stott 1974]. In addition to the data found in that paper, 9 controllable shunt admittances were added to the network. The data and the schematic diagram for this system can be found in Appendix 7.3.

The number of variables in this system is as follows:

Number of buses:	30
Number of generations:	6
Number of loads:	27
Number of transmission lines:	41
Number of controllable shunt elements:	9
Number of variable transformer taps:	4
Total number of load flow quantities:	173
Total number of load flow variables:	125

The program solved this system for an initial load, and then in three different runs, for 10 loads increased by 1 percent, 6 loads increased by 2 percent, and 3 loads increased by 4 percent.

7.4.1 Highlights of the Solution to the 30 Bus System

Again we present our observations, followed by the complete set of results for the 30 bus system.

i) The overall solution

Of the four systems tested in this thesis, this one presented the best nonlinear convergence characteristics. The rate of convergence of the solution still seems to be linear, but with quicker reductions in the convergence criteria. This can be seen in Table 7.12, which illustrates the detailed solution for the first load. The convergence criteria and the step size in the search for the load flow feasible point all reduced noticeably from one iteration to the next.

Some other important observations follow.

- Only 6 major iterations were required to converge to the prescribed tolerances.
- Many breakpoints were requirei to solve for all but the last subproblem. We note however that even though this test system is substantially larger than the previous two, the typical number of breakpoints remains the same. This indicates once again that the number of breakpoints depends more on the closeness to the final active set in a combinatorial sense.
- The values of the convergence criteria and the step size started quite large, and their reduction was fast.
- Relatively few Newton-Raphson iterations were required.
- The optimization produced a sizeable improvement in the cost. This can be seen in fig. 7.13, along with the curve of the subproblem costs.
- The computation time of 96.50 sec. is about 5 times the amount required by the 6 bus system and less than twice the time required by the 10 bus system.

Concerning the steady number of breakpoints in the subproblem solutions, there were probably too few major iterations in this solution to feed the right active set to the subproblem. More iterations probably would would have allowed the subproblems to settle on the right active set and avoid breakpoints.

The load-tracking solutions for the 30 bus system also presented some very attractive characteristics. This is illustrated in Table 7.13 for 1% load variations, in Table 7.14 for 2% load variations, and Table 7.15 for 4% load variations.

- Most loads were solved in 1 or 2 major iterations in the first two runs, with a maximum of 6 major iterations. The third run averaged 3 major iterations per load.

- The number of breakpoints is quite small for all loads except one in the first run. The average number of breakpoints increased in the second run. In the third run, one very long solution found its way between two quick solutions.
- Few Newton-Raphson iterations were computed.
- The computation times for the load-tracking solutions were very good, ranging from 12.03 sec. to 53.82 sec., with an average of 21.54 sec. This average is 22% of the time required to solve for the first load. The average solution times per load for the three load-tracking runs are 18.70, 24.45 and 25.11 sec.
- The optimal solutions from the 3 load-tracking runs compare very well. Six costs can be compared between the first two runs, 2 costs between the three runs, and one cost between the last two runs. The largest discrepancy between the first two runs was 0.05% and the average discrepancy was 0.03%. The third run with 4% load variations provided solutions with slightly lower costs, but with a maximum discrepancy of only 0.15% compared to the optima of the other two runs.

ii) The first subproblem solution

The first subproblem in this test required 18 breakpoints. These are given in Table 7.16. Again, there is a clustering of most of the breakpoints towards the end of the continuation process. In this test, most of the variables except those on their moving bounds showed little variation.

This system being larger and sparser than the previous test systems, the pairs of breakpoints now feature more closely related variables. For example, when Q27 hit its bound V27 was released at Θ=0.8484, and when Q28 met its bound V28 was released at Θ=0.9650.

Optimal solution trajectories for the variables are presented in fig. 7.14 a. to h.. To facilitate the presentation, the bus voltage magnitudes were separated into two groups, the transparent V's and a selection of dependent V's. The behaviour of the different types of variables is once again as described in the results of the 6 bus system. For the most part, the voltage magnitudes and the tap positions in this test do not exhibit the sharp variations seen in the 10 bus test, possibly because the largest violation in the continuation process was only 2.45 p.u. In fig. 7.14.b. we see the variable with the largest violation, Q25, follow its moving lower bound back to the feasible region. One of the faster moving variables, V1 in fig. 7.14.d., is following its moving upper bound at the very end of the continuation process. Among the transparent variables, only two of the shunt controllers move sharply from one bound to another.

iii) The sequences of subproblem and load flow feasible solutions

Despite the good nonlinear convergence characteristics of this test system, many of the variables which are not directly cost-related oscillate between their bounds at the output of the subproblem. This is illustrated in fig.7.15 a. to e. As before, the reduction in step size limited the excursions of the load flow feasible variables, so that convergence of most variables is quite good over the last few major iterations. This is illustrated in fig. 7.16 a. to e.

iv) The sequences of optimal solutions in the load-tracking mode

The results of the load tracking with 1% load variations are portrayed in fig. 7.17 a. to f. These are similar to previous results, with clear trends in the directions of movement of the variables. Most noticeable are the voltages and especially the shunt admittances, which increase very smoothly.

7.4.2 Tables and Graphs of the Results of the 30 Bus System

Tables 7.12 to 7.16 and Figures 7.13 to 7.17 on the following pages contain the results of the 30 bus system.

TABLE 7.12 - TEST ON THE 30 BUS SYSTEM

SUMMARY OF THE ALGORITHM'S PERFORMANCE

SOLUTION FOR THE FIRST LOAD

Major iteration no.	1	2	3	4	5	6
SUBPROBLEM						
Violations in the subproblem at the easy optimum	7	4	27	14	21	6
Breakpoints in the continuation process	18	24	20	12	26	4
SEARCH FOR LOAD FLOW FEASIBLE POINT						
Newton-Raphson iterations in finding load flow feasible pt.	2	5	2	4	1	1
Relative reduction in cost of load flow feasible pt	.1132	.0970	411e-2	.419e-3	.271e-3	.470e-4
Relative gap between costs of load flow feasible pt. and subproblem solution	.1108	.0201	.309e-2	.348e-2	.108e-2	.483e-3
Relative norm of the difference between the variables at the load flow feasible point and the subproblem solution	.6559	.0355	.0198	.0105	.797e-2	.749e-2
Step size in searching feasible load flow pt.	.9	.45	.45	225	.225	.1

Initial cost 884 440   Optimal cost: 802.310   Computation time: 96 50 sec.

TABLE 7.13 - TEST ON THE 30 BUS SYSTEM

SUMMARY OF THE ALGORITHM'S PERFORMANCE

SOLUTIONS FOR THE SUBSEQUENT LOADS - 1% VARIATIONS IN LOAD

Load no.	1	2	3	4	5	6	7	8	9	10
No. of major iterations	1	1	1	1	1	1	1	2	2	6
Total number of breakpoints in the subproblems	8	4	12	4	6	8	6	12	12	42
Total number of Newton-Raphson iterations	3	3	3	3	3	3	3	4	4	6
Optimal cost	812.67	823.19	833.86	844.69	855.66	866.79	878.08	889 48	901.06	912.69
Computation time (sec)	13.57	12.08	14.34	12 03	13.63	13.13	12.63	20 75	20 99	53 82

TABLE 7 14 - TEST ON THE 30 BUS SYSTEM

SUMMARY OF THE ALGORITHM'S PERFORMANCE

SOLUTIONS FOR THE SUBSEQUENT LOADS - 2% VARIATIONS IN LOAD

Load no.	1	2	3	4	5	6
No of major iterations	1	2	2	2	2	3
Total number of breakpoints in the subproblems	10	4	34	24	18	36
Total number of Newton-Raphson iterations	3	4	4	4	4	5
Optimal cost	823.11	844 497	866.495	889.079	912 300	936.141
Computation time (sec.)	14.06	21 04	28.67	27 08	21 58	34.44

TABLE 7.15 - TEST ON THE 30 BUS SYSTEM SUMMARY OF THE ALGORITHM'S PERFORMANCE SOLUTIONS FOR THE SUBSEQUENT LOADS - 4% VARIATIONS IN LOAD			
Load no.	1	2	3
No. of major iterations	2	3	4
Total number of breakpoints in the subproblems	4	42	8
Total number of Newton-Raphson iterations	4	5	6
Optimal cost	844.090	888.232	934.784
Computation time (sec)	18.08	35.84	21.42

TABLE 7.16 - TEST ON THE 30 BUS SYSTEM A SUMMARY OF BREAKPOINTS ENCOUNTERED IN THE FIRST SUBPROBLEM		
θ	Variable name and type	Cause of breakpoint
0.0000	Q25 dependent	most violated dependent constraint set to its moving lower bound.
.9479 e-4	V25 transparent	released from its lower bound.
0.5883	V4 dependent	set to its upper bound.
0.5883	B9 transparent	released from its upper bound
0.8482	Q27 dependent	set to its moving lower bound.
0.8484	V27 transparent	released from its lower bound.
0.8801	V25 transparent	set to its upper bound.
0.8804	V26 transparent	released from its upper bound.
0.9694	Q28 dependent	set to its moving upper bound.
0.9650	V28 transparent	released from its upper bound.
0.9825	B9 transparent	set to its upper bound.
0.9826	V4 dependent	released from its upper bound.
0.9850	V1 dependent	set to its moving upper bound.
0.9850	B9 transparent	released from its upper bound.
0.9562	B9 transparent	set to its lower bound.
0.9862	B6 transparent	released from its upper bound.
0.9874	B6 transparent	set to its lower bound.
0.9874	T1 transparent	released from its upper bound.

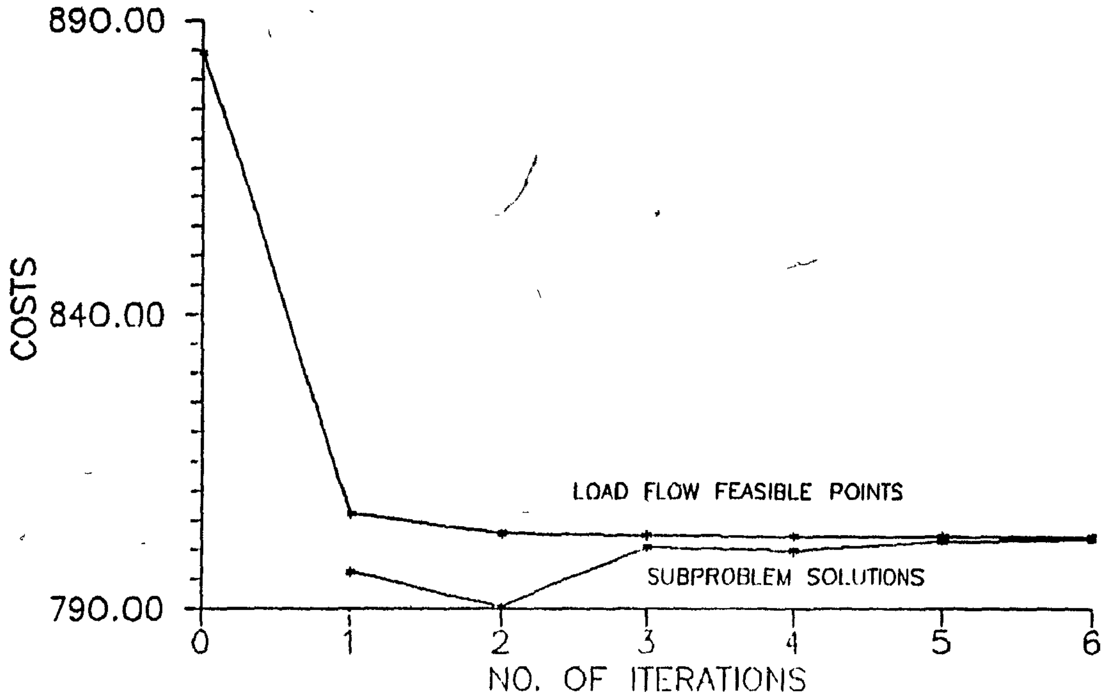


Figure 7.13. Costs of the load flow feasible points and the subproblem solutions at each major iteration, solution to the 10 bus system.

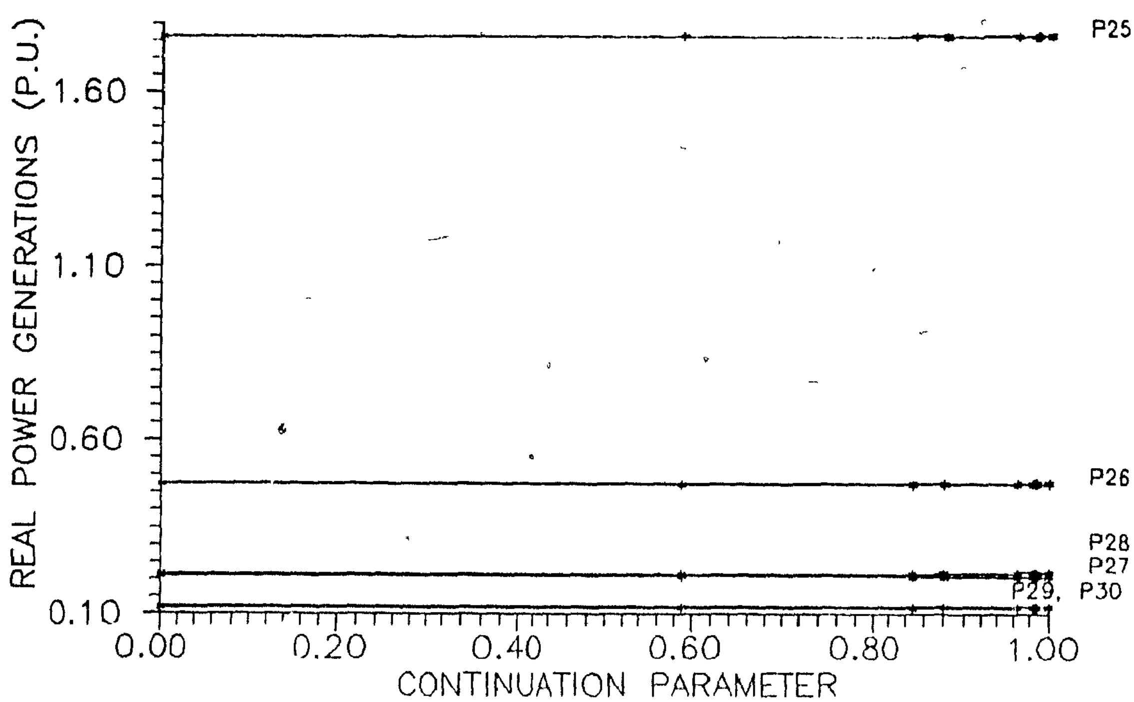


Figure 7.14.a. Real power generations vs. the continuation parameter, first subproblem in the solution of the 30 bus system.

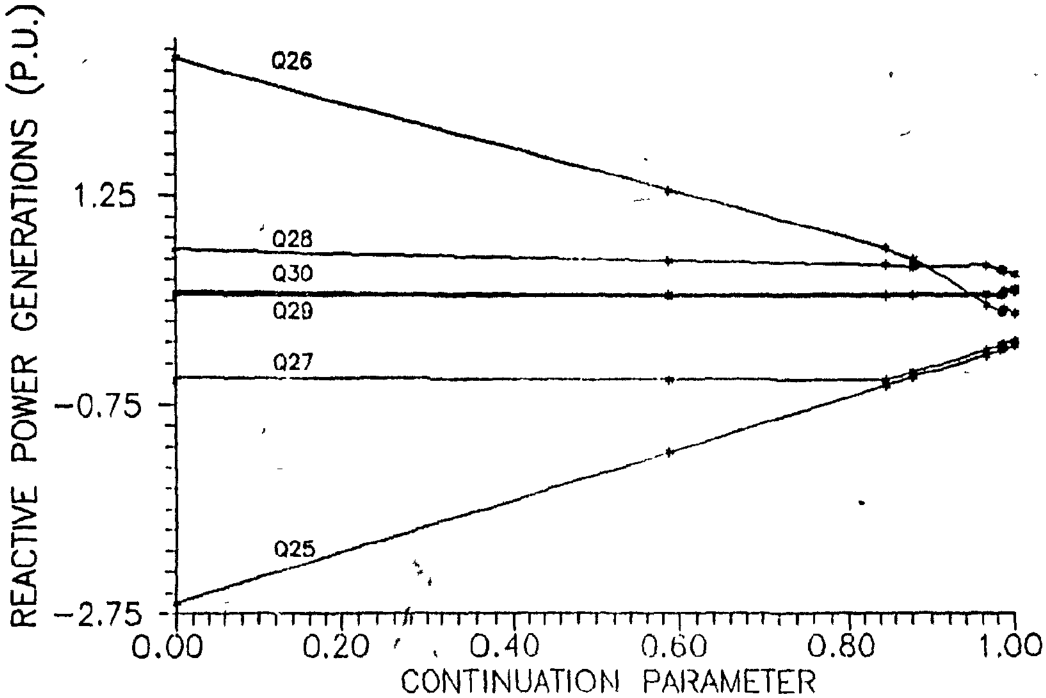


Figure 7.14.b. Reactive power generations vs. the continuation parameter, first subproblem in the solution of the 30 bus system.

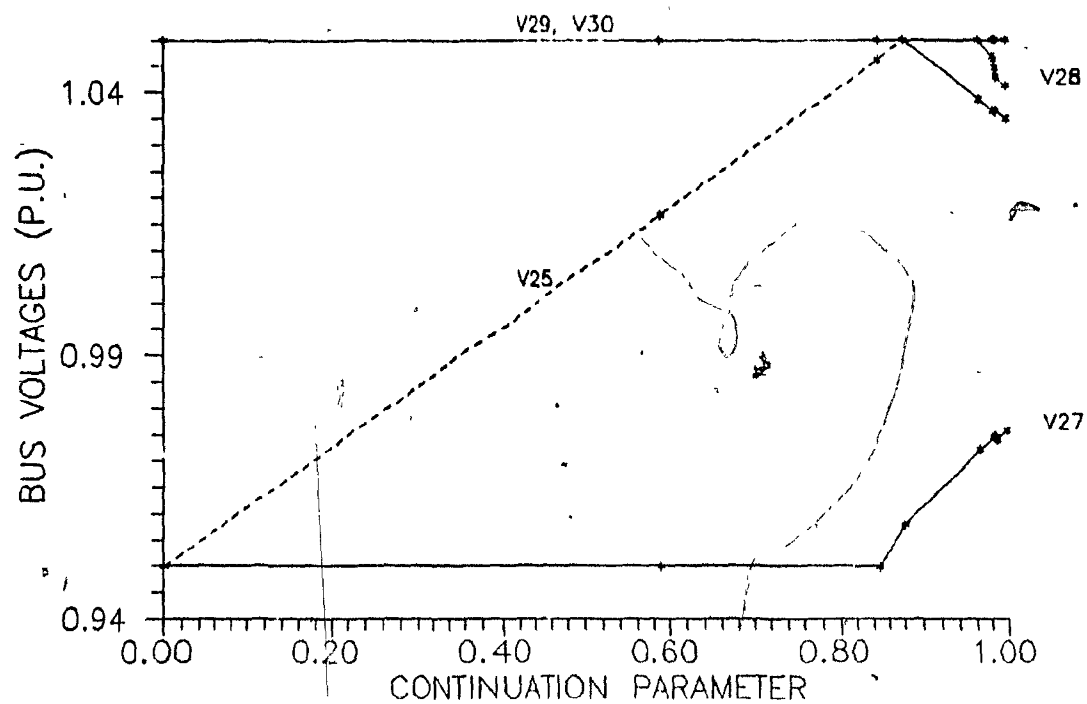


Figure 7.14.c. Transparent bus voltage magnitudes vs. the continuation parameter, first subproblem in the solution of the 30 bus system.

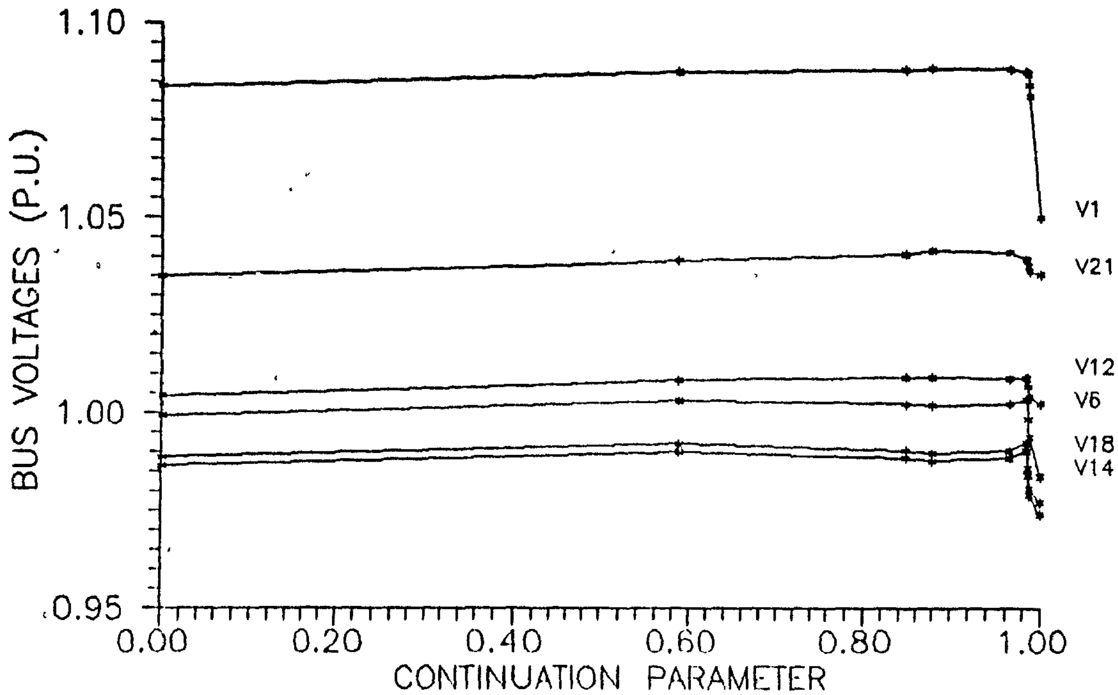


Figure 7.14.d. Selected dependent bus voltage magnitudes vs. the continuation parameter, first subproblem in the solution of the 30 bus system.

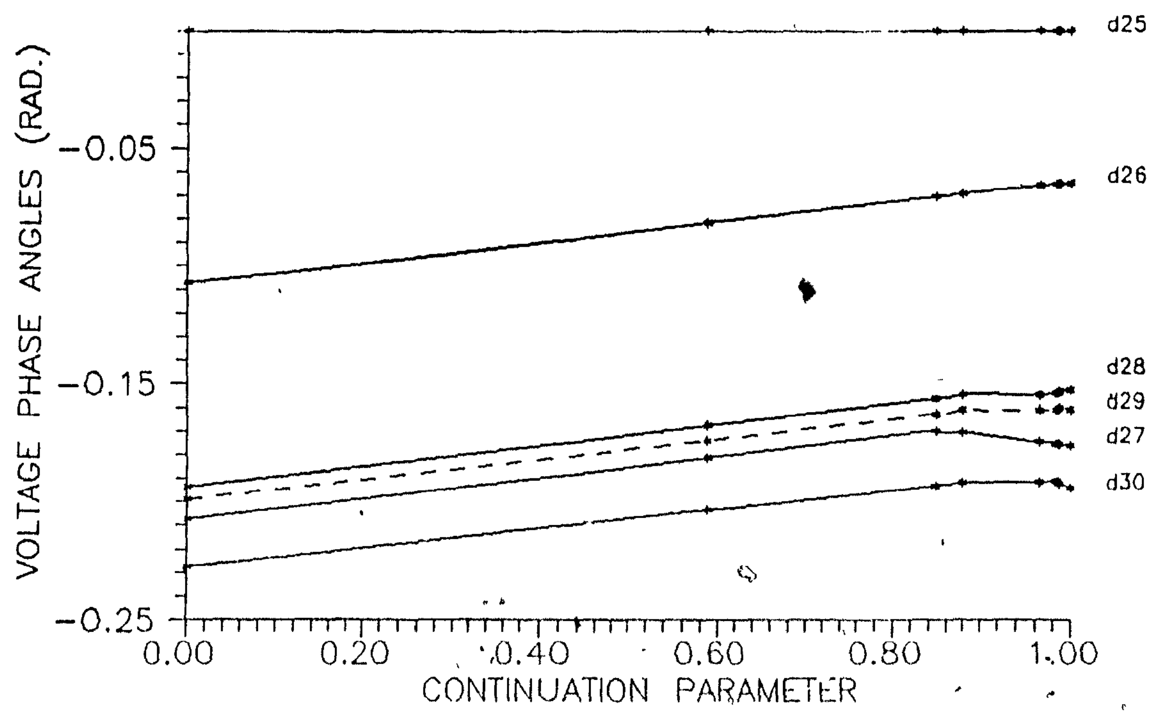


Figure 7.14.e. Bus voltage phase angles vs. the continuation parameter, first subproblem in the solution of the 30 bus system.

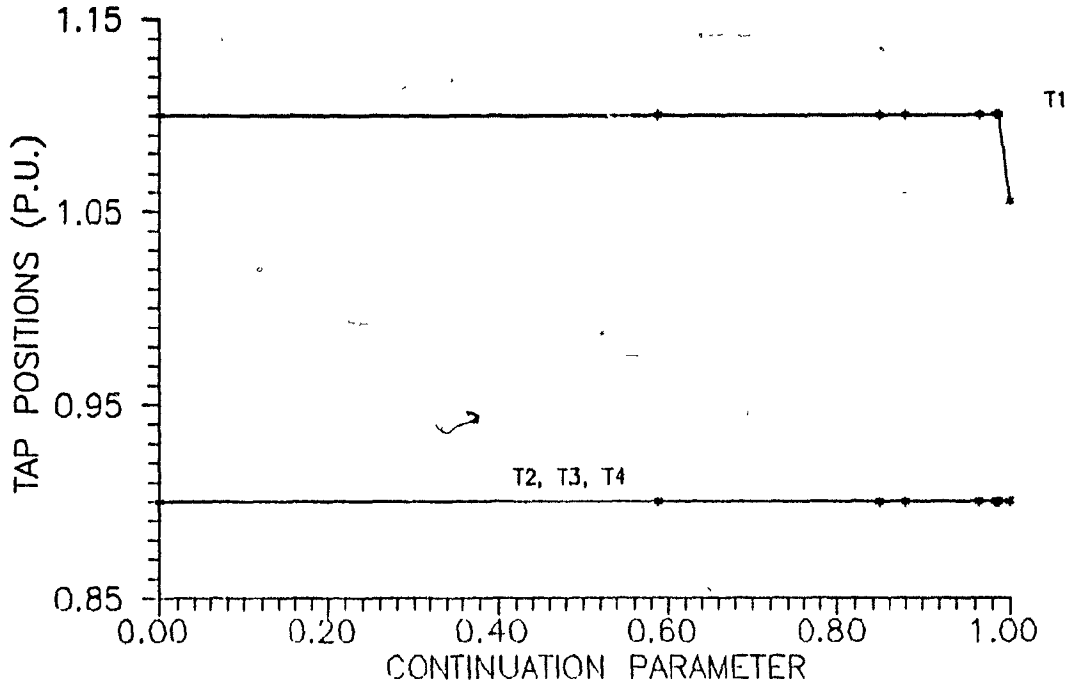


Figure 7.14.f. Variable transformer tap positions vs. the continuation parameter, first subproblem in the solution of the 30 bus system.

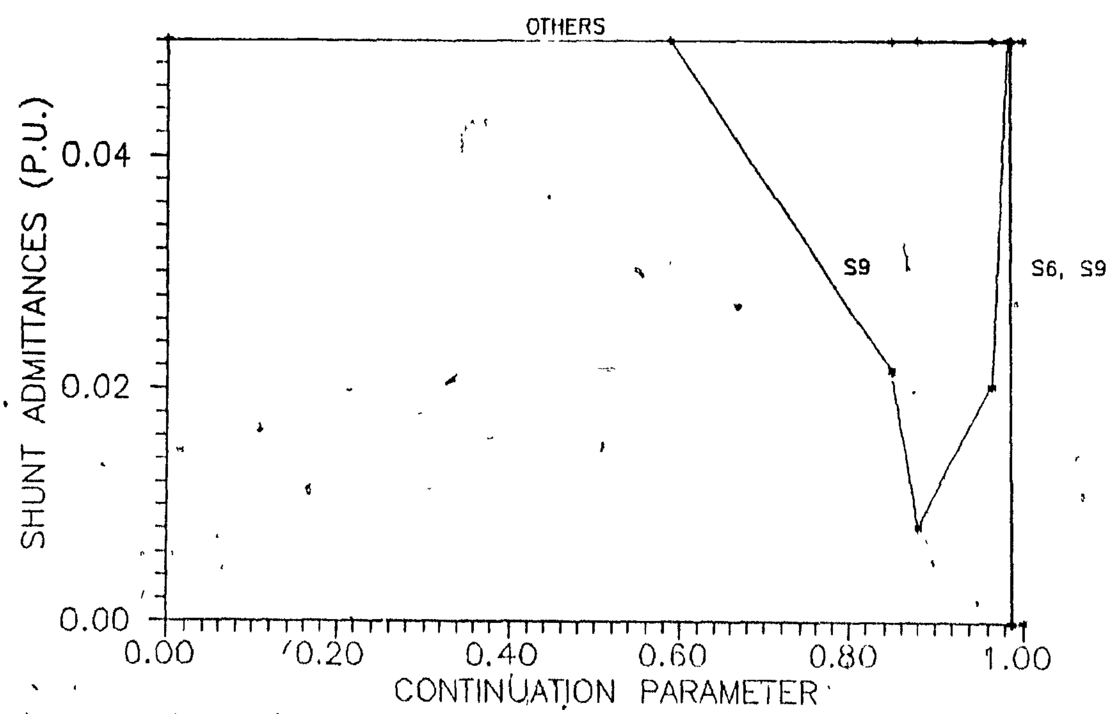


Figure 7.14.g Shunt controller admittances, first subproblem in the solution of the 30 bus system.

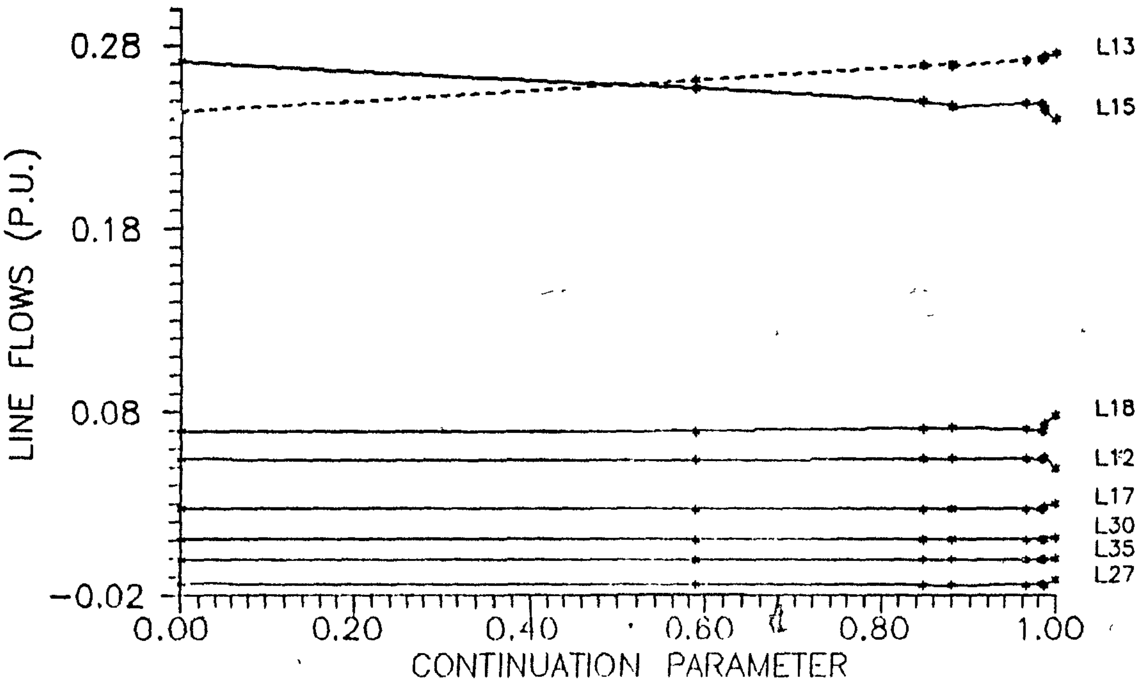


Figure 7.14.h. Line flows vs. the continuation parameter, first subproblem in the solution of the 30 bus system.

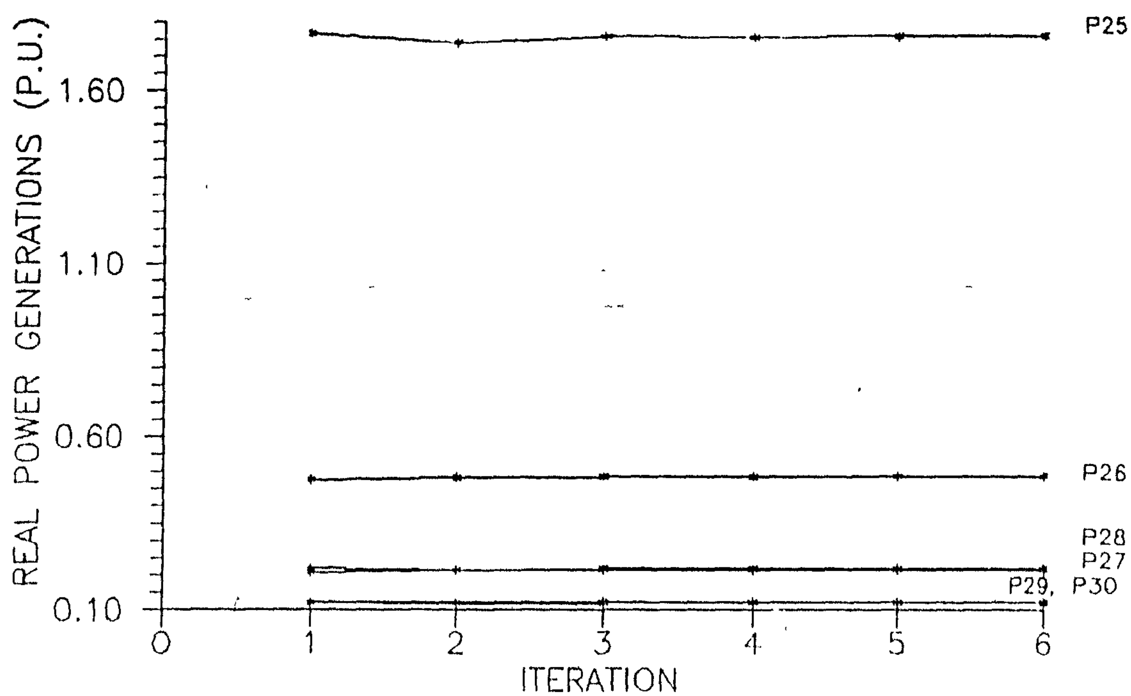


Figure 7.15.a. Subproblem real power generations vs. the major iteration number, solution of the first load of the 30 bus system.

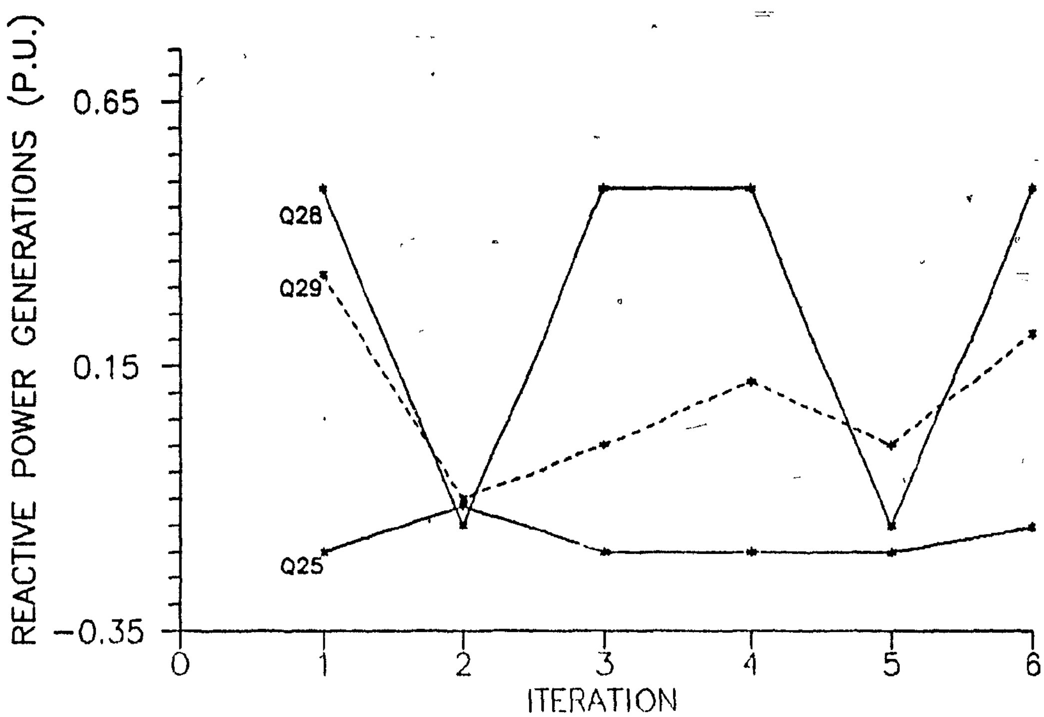


Figure 7.15.b. Subproblem reactive power generations vs. the major iteration number, solution of the first load of the 30 bus system.

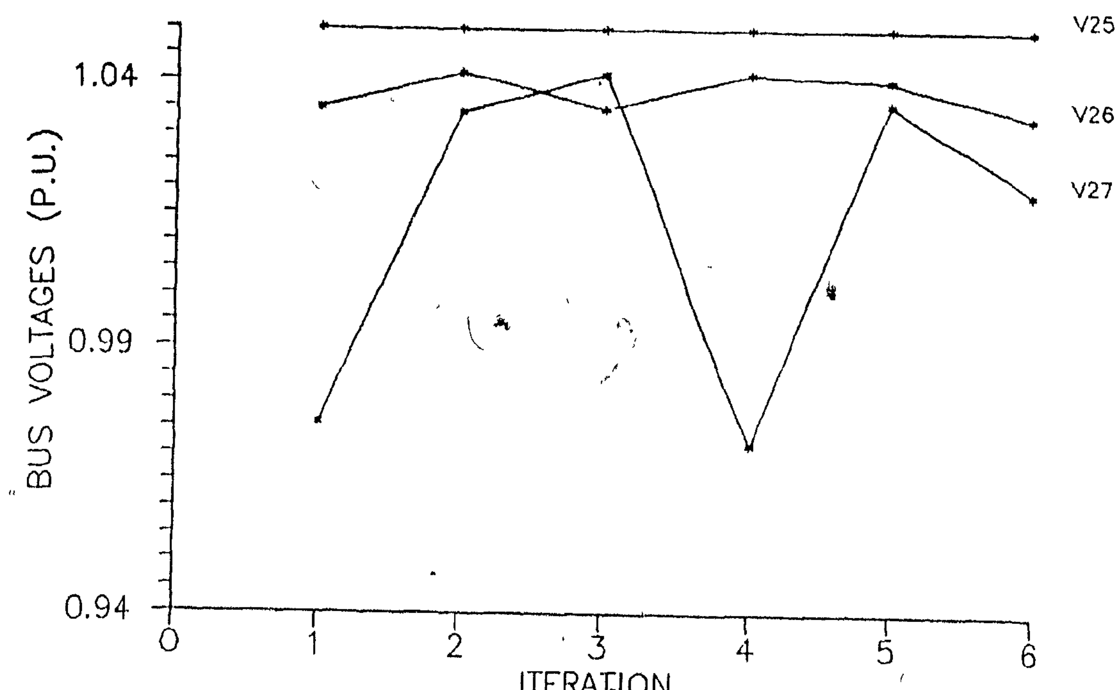


Figure 7.15.c. Subproblem bus voltage magnitudes vs. the major iteration number, solution of the first load of the 30 bus system.

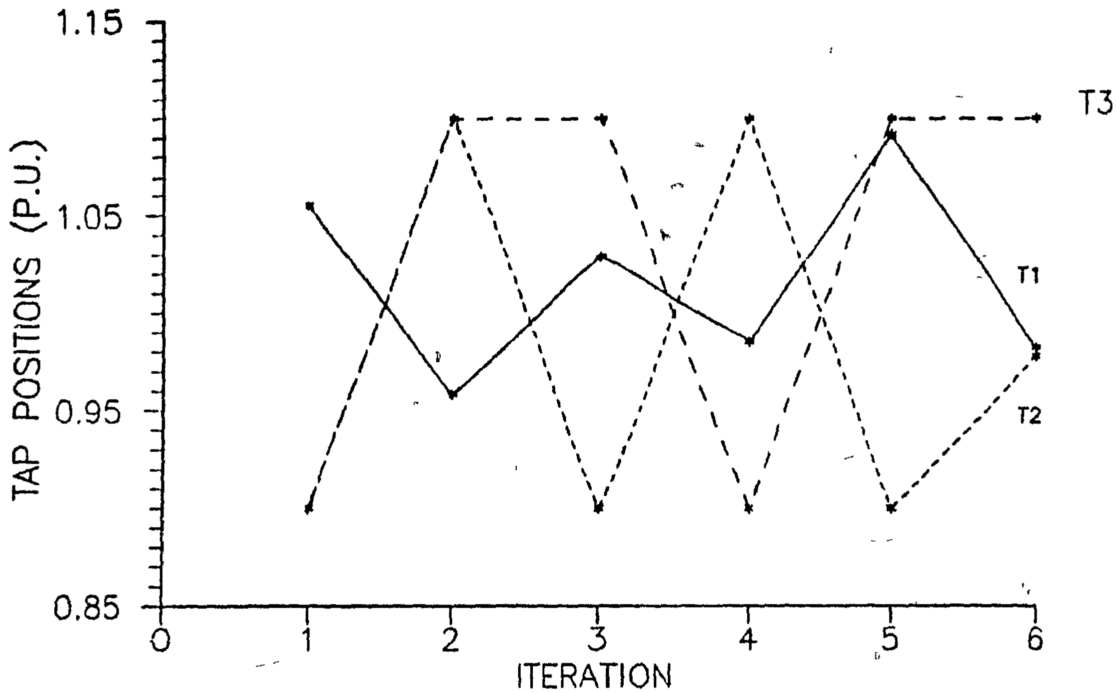


Figure 7.15.d. Subproblem variable tap positions vs. the major iteration number, solution of the first load of the 30 bus system.

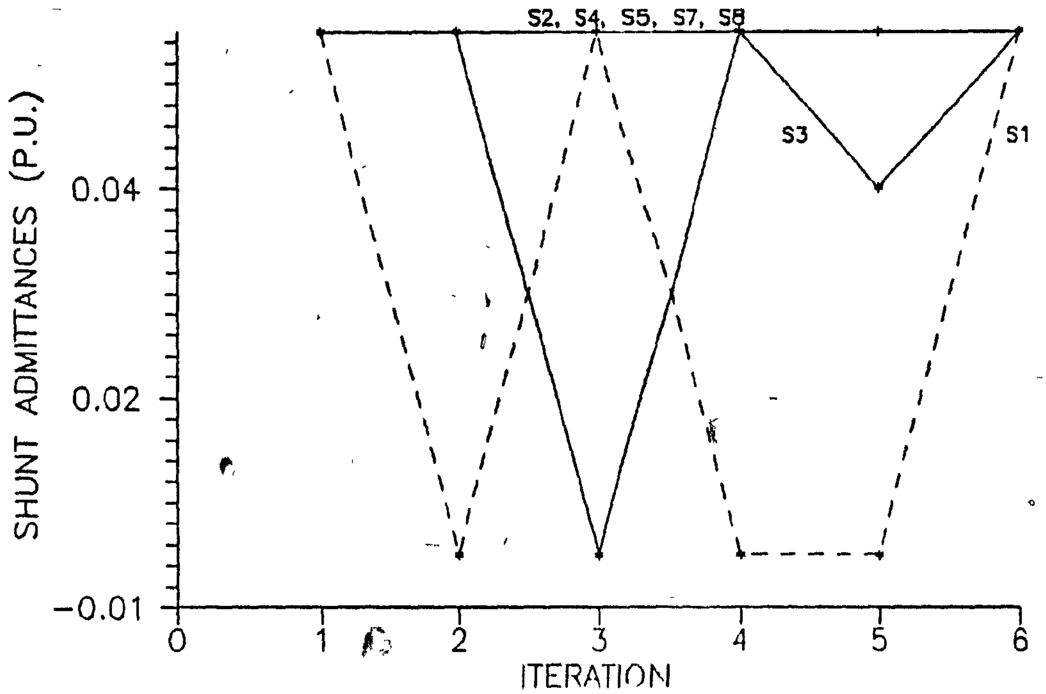


Figure 7.15.e **Subproblem shunt controller admittances vs. the major iteration number, solution of the first load of the 30 bus system.**

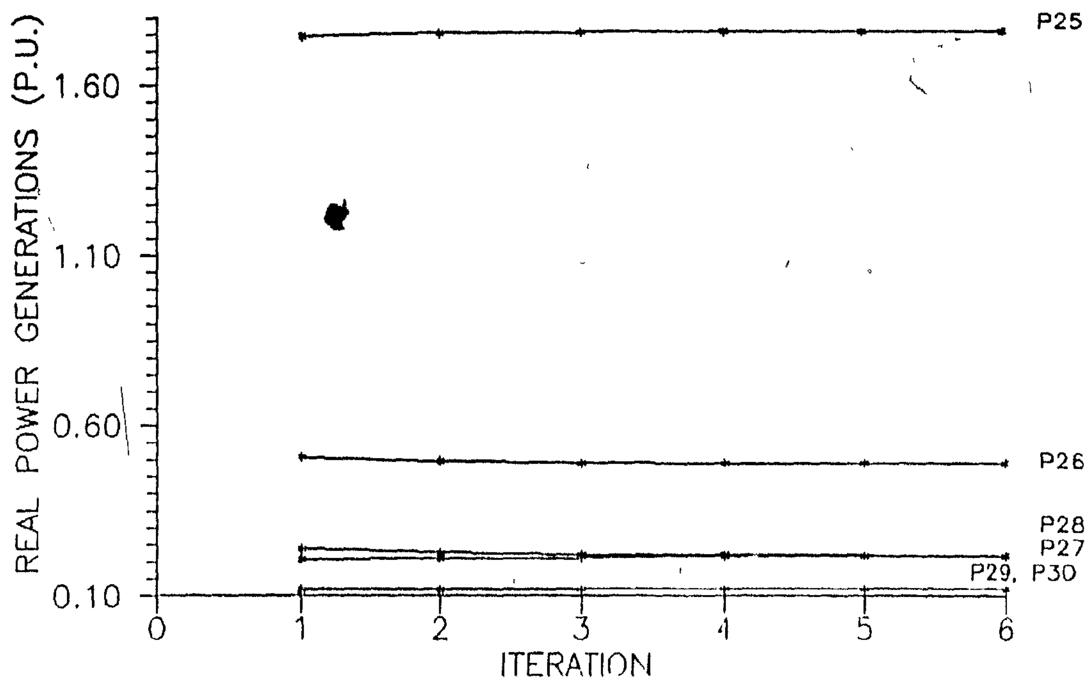


Figure 7.16.a. Feasible real power generations vs. the major iteration number, solution of the first load of the 30 bus system.

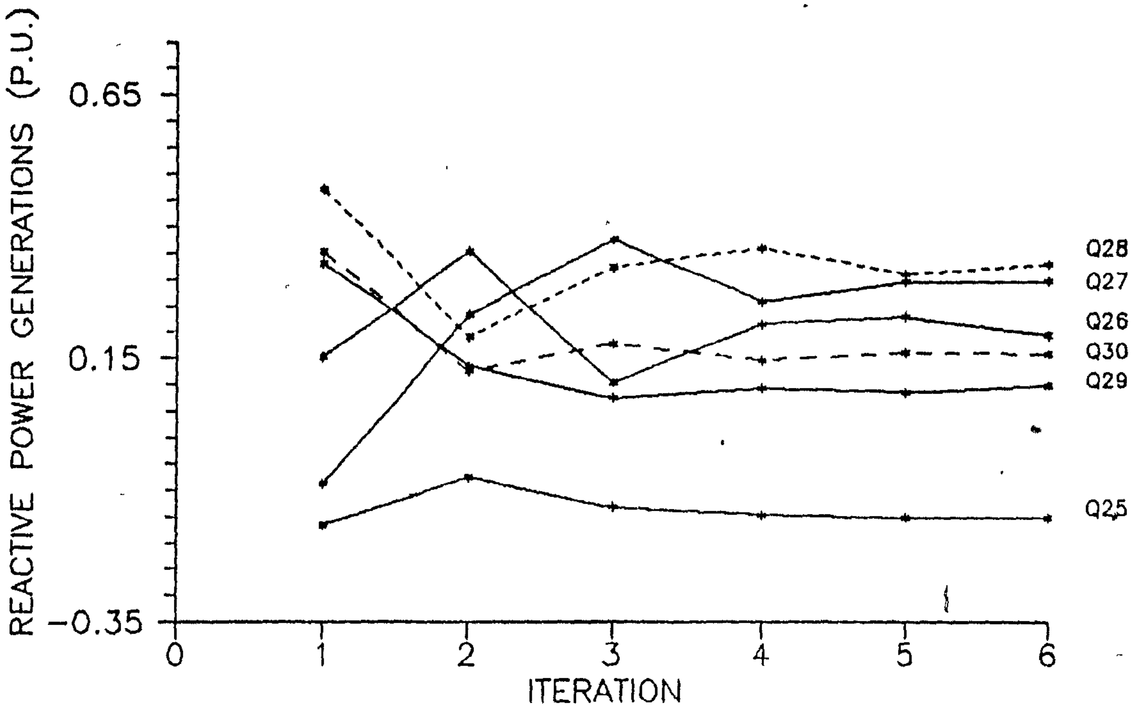


Figure 7.16.b. Feasible reactive power generations vs. the major iteration number, solution of the first load of the 30 bus system.

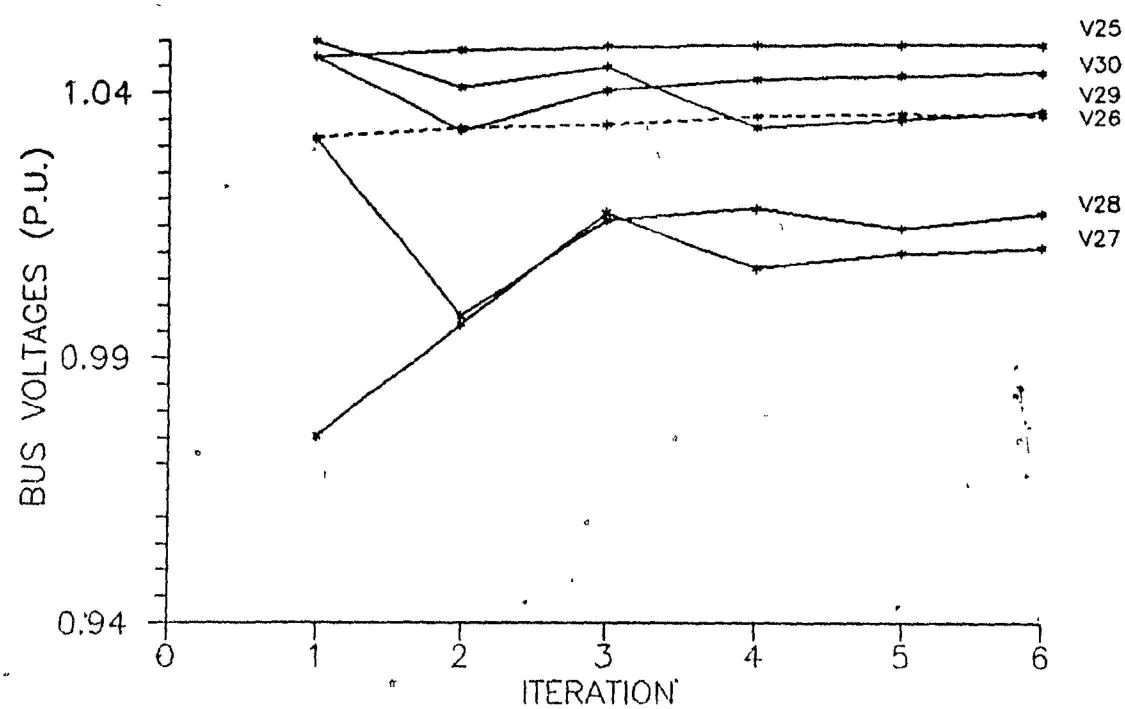


Figure 7.16.c. Feasible bus voltage magnitudes vs. the major iteration number, solution of the first load of the 30 bus system.

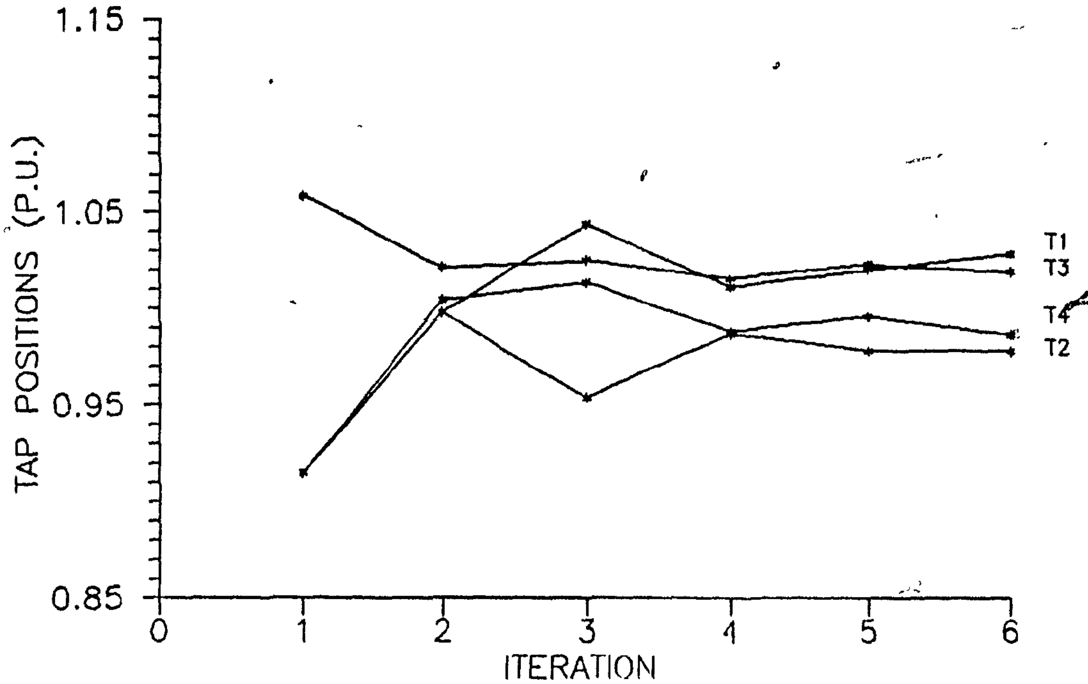


Figure 7.16.d. Feasible variable tap positions vs. the major iteration number, solution of the first load of the 30 bus system.

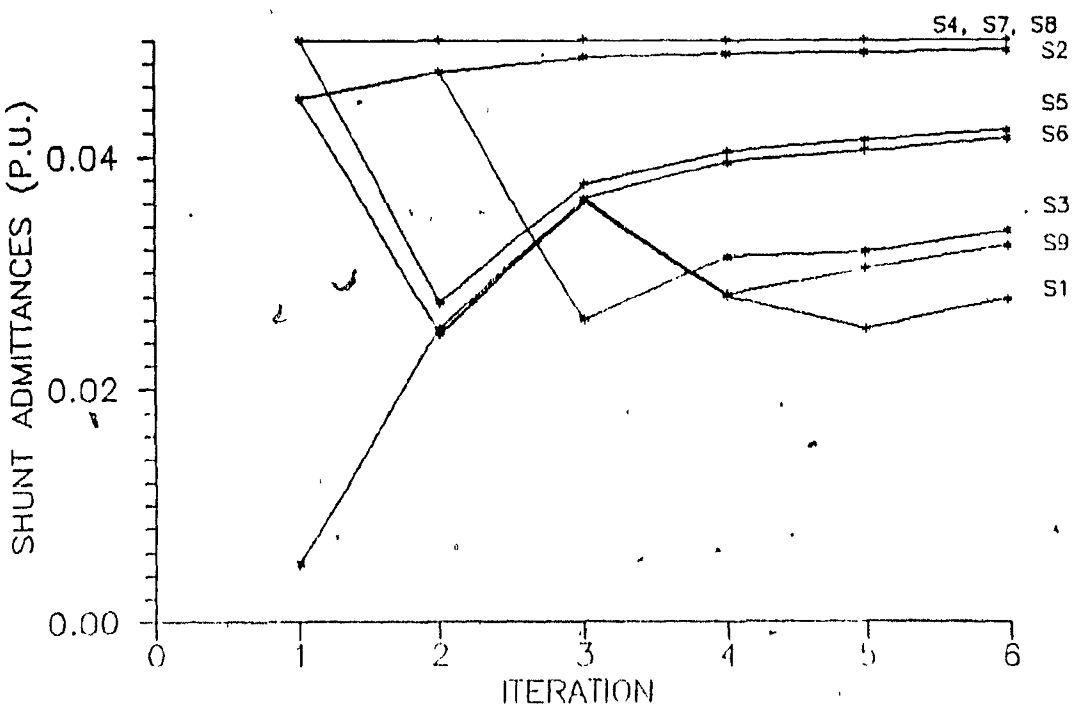


Figure 7.16.e. Feasible shunt controller admittances vs. the major iteration number, solution of the first load of the 30 bus system.

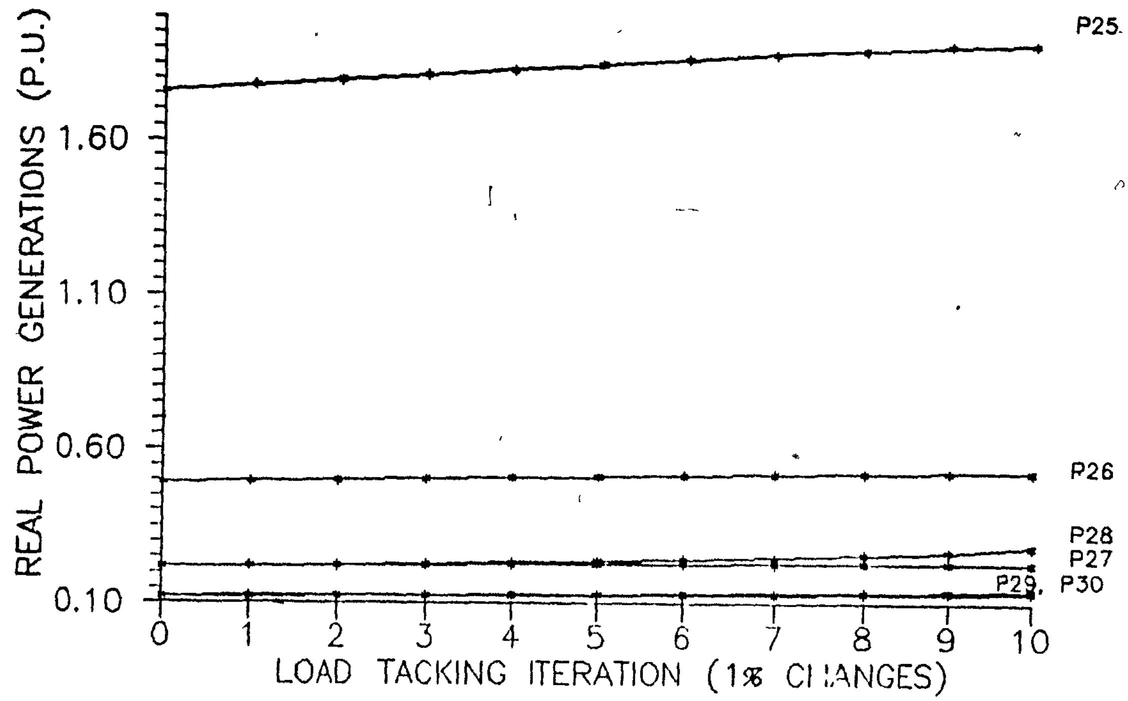


Figure 7.17.a. **Real power generations vs.load, load tracking on the 30 bus system.**

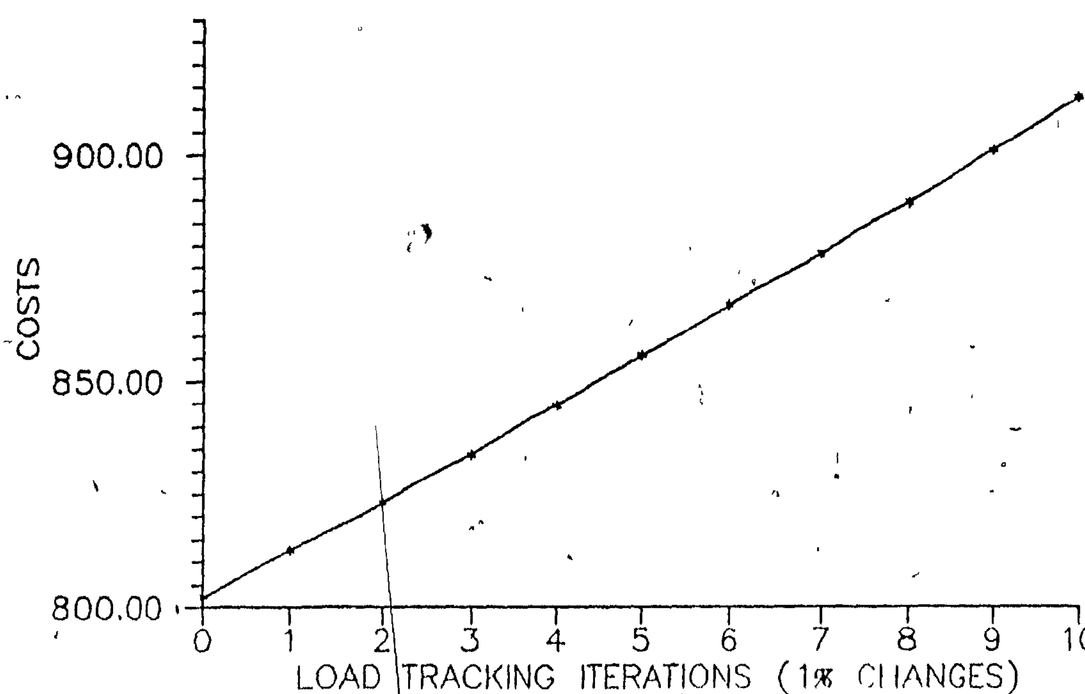


Figure 7.17.b. Optimal fuel costs vs.load, load tracking on the 30 bus system.

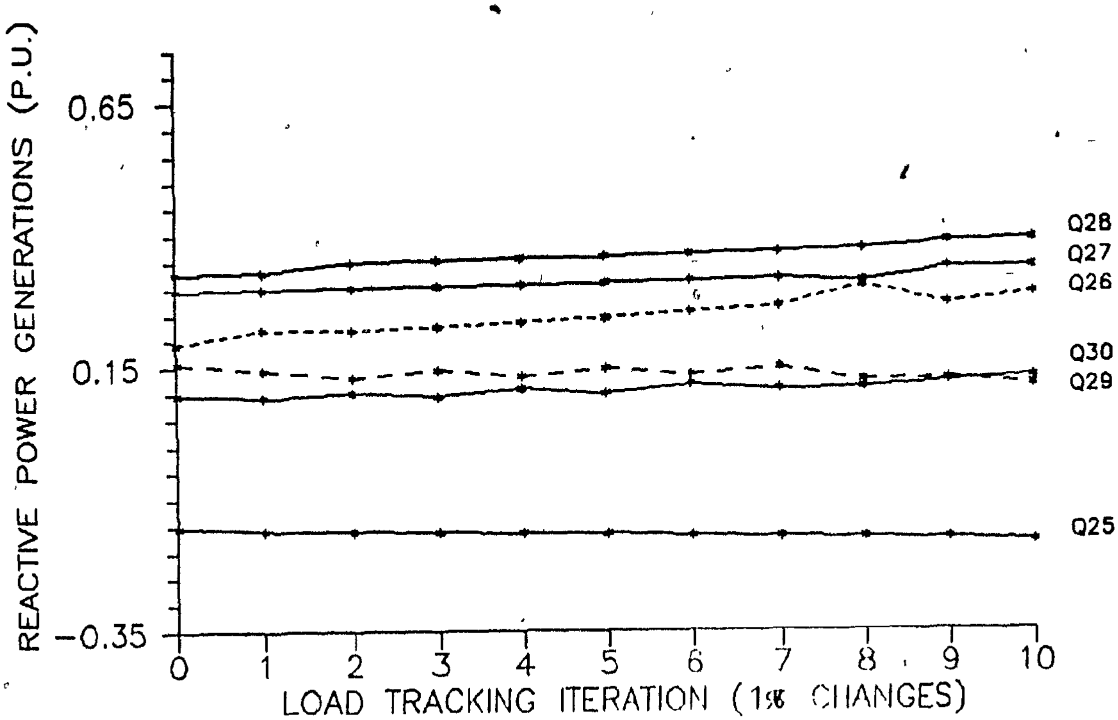


Figure 7.17.c. Reactive power generations vs. load, load tracking on the 30 bus system.

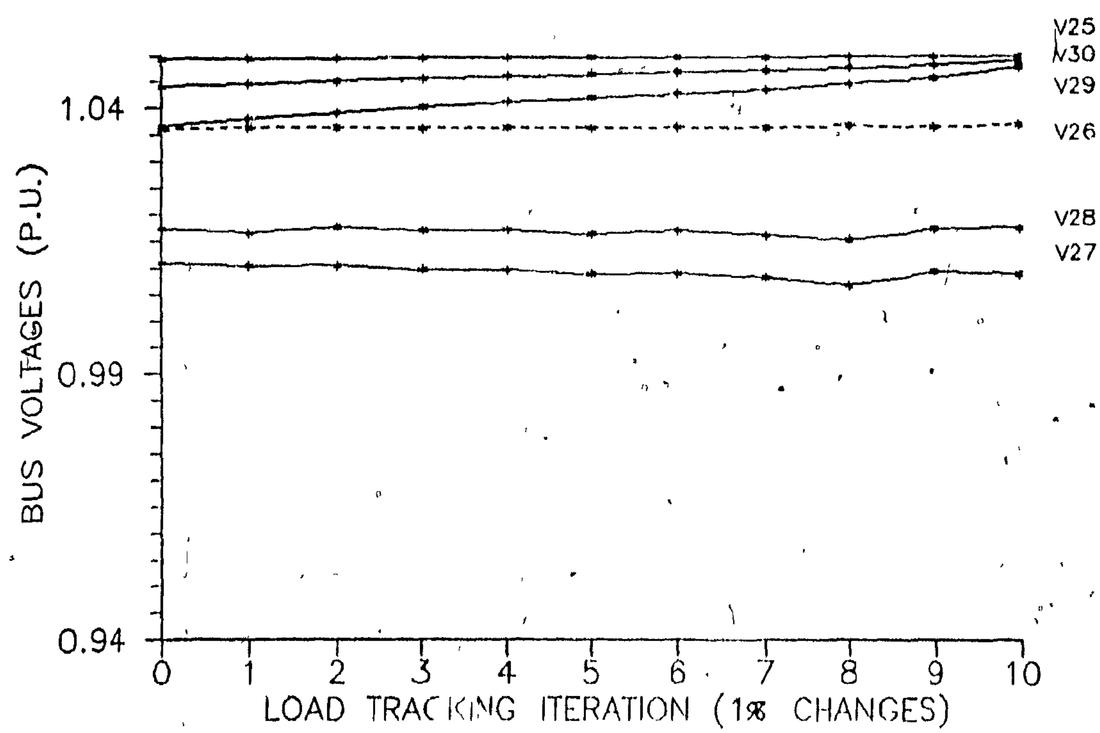


Figure 7.17.d. Bus voltage magnitudes vs. load, load tracking on the 30 bus system.

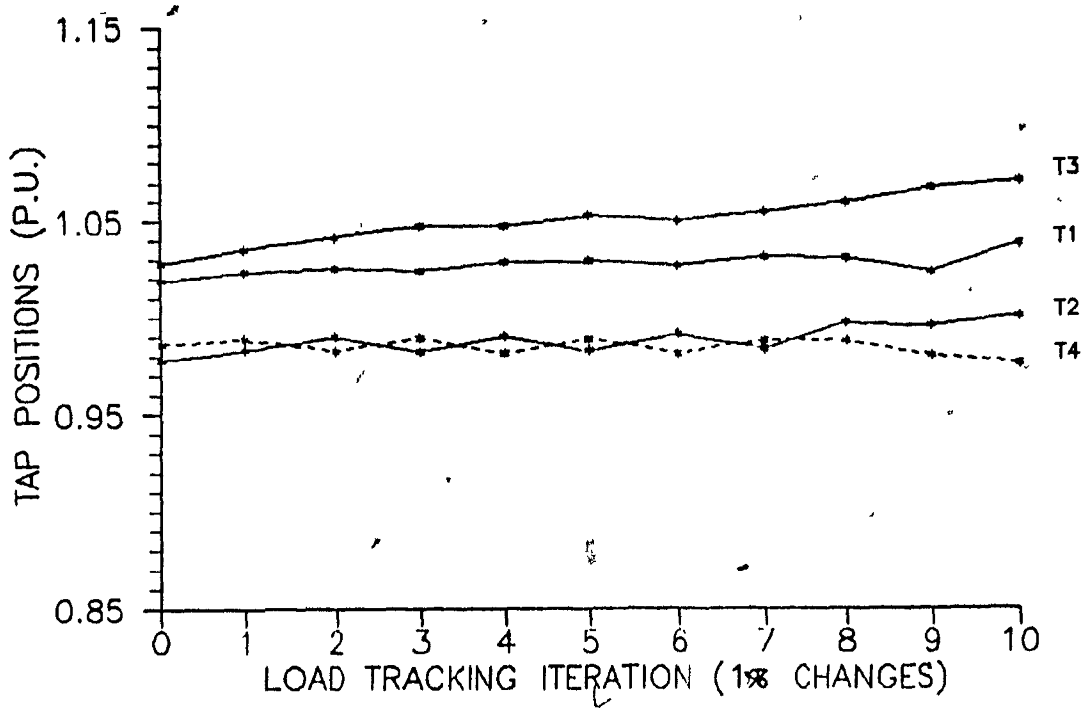


Figure 7.17.e. Variable tap positions vs.load, load tracking on the 30 bus system.

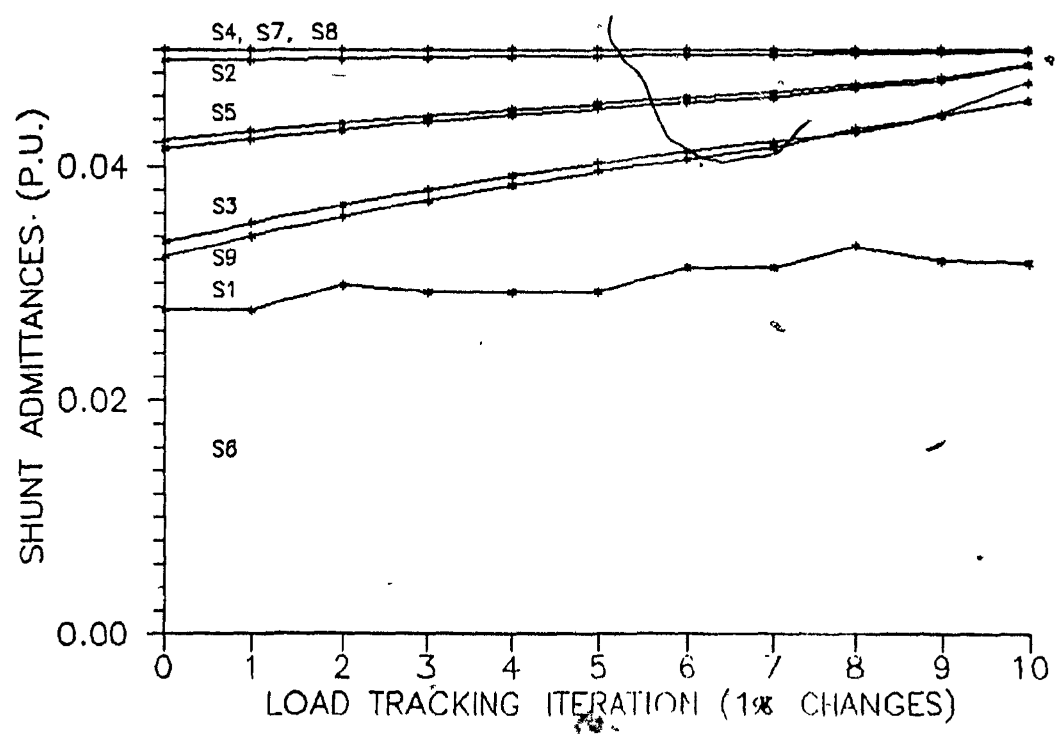


Figure 7.17.f. Shunt controller admittances vs. load, load tracking on the 30 bus system.

7.5 Simulation on a 118 Bus System

The 118 bus system is an adaptation of the IEEE 118 bus standard load flow test system, taken from a report of the Engineering Foundation Conference [Podmore et.al. 1977], and with additional material taken from [Ponrajah 1987]. The data and the schematic diagram for this system can be found in Appendix 7.3.

The number of variables in this system is as follows:

Number of buses:	118
Number of generations:	49
Number of loads:	98
Number of transmission lines:	173
Number of controllable shunt elements:	9
Number of variable transformer taps:	8
Number of phase shifting transformers:	2
Total number of load flow quantities:	669
Total number of load flow variables:	531

Due to the large size of this system, a constraint-feasible and load flow-feasible initial guess could not be arrived at by trial and error, as was the case for the previous tests. Hence an initial guess was manufactured from the optimal solution in [Ponrajah 1987].

The program could solve this system for an initial load only, due to problems with the program's present load changing algorithm. This and other problems are discussed in the upcoming section 7.8.

Another difficulty encountered with this large test system was due to numerical problems in modelling the subproblem's dependent reactive power generations; this also will be discussed later. Hence, in a first test, the limits on these variables were removed. The results of the OPF were quite good, with performances similar to those in the previous tests. Then in a second test as many reactive powers as possible were returned to the formulation, and a subproblem was solved. This test required much more computation, and the results exhibited a more erratic behaviour. Results in this section are taken mostly from the simpler first test, but the graphs and the discussion for the solution trajectories of the subproblem solution are taken from the more interesting second test.

7.5.1 Highlights of the Solution to the 118 Bus System

Again we present our observations, followed by the complete set of results for the 118 bus system.

i) The overall solution

The overall solution results for the tests on the 10 bus and 118 bus systems offer somewhat similar performances. As was the case for the test on the 10 bus system, the initial guess for the 118 bus test was taken very close to the actual optimal solution. It is felt that because of this, the algorithm performs sluggishly, with improvements being slow from one major iteration to the next. The overall solution is presented in Table 7.17.

Some important observations on these results follow.

- Eight major iterations were required, although very little progress was achieved in the process.
- Three of the first four subproblems required of the order of 20 breakpoints. Again, the maximum number of breakpoints does not seem to be related to system size per se.
- The next four subproblems required few breakpoints.
- Very few Newton-Raphson iterations were needed, because most of the subproblem solutions were almost load flow-feasible (to within tolerances).
- The cost and the other convergence criteria improved very little. The costs vs. major iteration curves are drawn in fig. 7.18 a. and b.

- The computation time of 253.81 sec. is very good - this is only 2.65 times more than for the computation of the 30 bus system. Aside from the few Newton-Rahson iterations, one reason is that part of the computation for this test was converted to single precision, to avoid memory size problems. Even without this simplification, the increase in the timing would probably still be proportional to the increase in system size.

ii) The first subproblem solution

Table 7.18 gives the list of 22 breakpoints encountered in the first subproblem of the solution reported above. This pattern of breakpoints is similar to the those observed in the previous tests, with all breakpoints occuring in pairs and most occuring near the end of the continuation process. The pairs of breakpoints usually contain variables which are close to each other in the network topology, although not necessarily at the same bus.

Table 7.19 gives the list of 129 breakpoints encountered in the subproblem solution where many of the limits on the reactive power generations were maintained. Even though this is a much larger number of breakpoints than what was encountered in previous tests, it still remains comparable to the number of iterations which could be expected in LP or QP.

More than a dozen variables, among them reactive powers Q46, Q49, Q54, Q55, Q56 and voltage magnitudes V40, V41, V55, V56 and V64, have an erratic behaviour, moving onto and off of their given limits regularly. Their actual movements are very small, but the algorithm forces them in and out of the active set at great computational expense. The problem, to be discussed later, is due to ill-conditioning in the optimality conditions.

We note in Table 7.19 that most control actions are performed by a variable very close the newly activated constraint. Furthermore, long sequences of breakpoints often involve neighboring variables, all interacting to each other but having little affect on other parts of the network. This behaviour seems to be characteristic of large systems, but mainly because of their sparseness.

Optimal solution trajectories for the more important variables of the subproblem described in Table 7.19 are presented in fig. 7.18 a. to d. The different types of variables behave as in the previous tests, but with sharp variations occuring on more of the transparent variables. Figure 7.18.b shows a large set of trajectories for the dependent Q's. There are basically three types of variables portrayed here: (1) some dependent Q's with large violations follow their moving bounds much of the way; (2) other Q's start on their bounds but then move freely, on irregular trajectories; and (3) Q's which remain on their bounds throughout the process. Many transparent V's in fig. 7.18.c. move very sharply off their bounds. Some wander irregularly as seen mostly between $\Theta=0.5$ and $\Theta=0.95$ while others jump from one bound to the other near $\Theta=1$. The same behaviour is seen in fig. 7.18.d with the variable transformer tap positions.

iii) The sequences of subproblem and load flow feasible solutions

Figure 7.20 a. to c. give the real power generations, the bus voltage magnitudes and the variable transformer tap positions at the end of each subproblem. As usual, the real power generations show little change from one subproblem to the next. The other variables oscillate for the first 4 major iterations, but move little after that as the active set has settled down.

Figure 7.21. a. to c. shows the same variables at the end of each search for a feasible load flow point. The step size reduced substantially in the first major iteration. Consequently, little change occured in most of the variables.

7.4.2 Tables and Graphs of the Results of the 118 Bus System

Tables 7.17 to 7.19 and Figures 7.18 to 7.21 on the following pages contain the results of the 118 bus system.

TABLE 7.17 - TEST ON THE 118 BUS SYSTEM SUMMARY OF THE ALGORITHM'S PERFORMANCE								
Major iteration no.	1	2	3	4	5	6	7	8
SUBPROBLEM								
Violations in the subproblem at the easy optimum	15	12	4	18	4	1	1	1
Breakpoints in the continuation process	22	22	4	20	6	2	2	2
SEARCH FOR LOAD FLOW FEASIBLE POINT								
Newton-Raphson iterations in finding load flow feasible pt	1	0	1	0	0	0	0	1
Relative reduction in cost of load flow feasible pt.	.185e-3	627e-4	.672e-4	.575e-4	.121e-3	.119e-3	236e-3	231e-3
Relative gap between costs of load flow feasible pt. and subproblem solution	.0317	.0294	.0301	0304	.0302	.0301	.0298	.0296
Relative norm of the difference between the variables at the load flow feasible point and the subproblem solution	0.2644	0 2624	0.2624	0 2633	0.2625	0.2619	0.2606	0 2592
Step size in searching feasible load flow pt	.00125	.00125	.00125	.00125	.00125	.00125	.00125	01

Initial cost 5409.10 Optimal cost 5402 02 Computation time 253 81 sec

TABLE 7.18 - TEST ON THE 118 BUS SYSTEM

A SUMMARY OF BREAKPOINTS ENCOUNTERED IN THE FIRST SUBPROBLEM

CASE WITH NO REACTIVE GENERATION

θ	Variable name and type		Cause of breakpoint
0.0000	J59	dependent	most violated dependent constraint set to its moving lower bound.
.9516 e-3	T2	transparent	released from its lower bound.
0.2995	J61	dependent	set to its moving lower bound.
0.3042	V77	transparent	released from its upper bound.
0.4934	V75	dependent	set to its moving upper bound.
0.4935	V34	transparent	released from its lower bound.
0.9429	V63	dependent	set to its moving upper bound.
0.9429	V59	transparent	released from its upper bound.
0.9685	V64	dependent	set to its moving upper bound.
0.9685	V63	dependent	released from its moving upper bound.
0.9708	V30	dependent	set to its moving upper bound.
0.9709	V8	transparent	released from its upper bound.
0.9877	V112	dependent	set to its moving lower bound.
0.9877	V105	transparent	released from its lower bound.
0.9932	V86	dependent	set to its moving lower bound.
0.9932	T1	transparent	released from its upper bound.
0.9969	J172	dependent	set to its upper bound.
0.9969	V18	transparent	released from its upper bound.
0.9979	V63	dependent	set to its lower bound.
0.9979	T4	transparent	released from its lower bound.
0.9980	V23	transparent	set to its moving upper bound.
0.9980	V32	transparent	released from its upper bound.

TABLE 7.19 - TEST ON THE 118 BUS SYSTEM

A SUMMARY OF BREAKPOINTS ENCOUNTERED IN THE FIRST SUBPROBLEM

CASE WITH REACTIVE GENERATIONS CONSIDERED

θ	Variable name and type		Cause of breakpoint
0.0000	Q56	dependent	most violated dependent constraint set to its moving upper bound.
.6424 e-6	V56	transparent	released from its upper bound.
0.4236	Q65	dependent	set to its moving lower bound.
0.4236	T3	transparent	released from its lower bound.
0.5064	Q8	dependent	set to its moving lower bound.
0.5064	T8	transparent	released from its lower bound.
0.5103	T3	transparent	set to its lower bound.
0.5103	V65	transparent	released from its lower bound.
0.5400	Q34	dependent	set to its moving lower bound.
0.5400	V34	transparent	released from its lower bound.
0.6266	Q77	dependent	set to its moving upper bound.
0.6266	V76	transparent	released from its lower bound.
0.6719	Q76	dependent	set to its upper bound.
0.6719	V77	transparent	released from its lower bound.
0.6882	V38	dependent	set to its upper bound.
0.6883	T2	transparent	released from its lower bound.
0.7625	V81	dependent	set to its lower bound.
0.7628	V40	transparent	released from its upper bound.
0.8203	V41	dependent	set to its lower bound.
0.8204	V36	transparent	released from its upper bound.
0.8614	Q61	dependent	set to its moving upper bound.
0.8614	V62	transparent	released from its upper bound.
0.8855	Q26	dependent	set to its moving lower bound.
0.8855	T7	transparent	released from its lower bound.
0.8863	V60	dependent	set to its upper bound.
0.8863	V61	transparent	released from its upper bound.
0.8871	Q59	dependent	set to its moving upper bound.
0.8871	V55	transparent	released from its lower bound.
0.8963	V67	dependent	set to its upper bound.
0.8963	V66	transparent	released from its upper bound.
0.9024	Q55	dependent	set to its upper bound.
0.9024	V54	transparent	released from its lower bound.
0.9030	Q62	dependent	set to its upper bound.
0.9030	V60	transparent	released from its upper bound.
0.9109	V66	transparent	set to its upper bound.
0.9109	V81	dependent	released from its lower bound.
0.9214	Q74	dependent	set to its moving upper bound.
0.9214	V74	transparent	released from its upper bound.
0.9496	Q89	dependent	set to its moving lower bound.
0.9496	V92	transparent	released from its upper bound.

TABLE 7.19 (cont.) - TEST ON THE 118 BUS SYSTEM

A SUMMARY OF BREAKPOINTS ENCOUNTERED IN THE FIRST SUBPROBLEM

CASE WITH REACTIVE GENERATIONS CONSIDERED

θ	Variable name and type		Cause of breakpoint
0.9520	J59	dependent	set to its upper bound.
0.9520	V67	dependent	released from its upper bound.
0.9565	V79	dependent	set to its lower bound.
0.9565	V80	transparent	released from its lower bound.
0.9567	Q85	dependent	set to its moving upper bound.
0.9567	T1	transparent	released from its upper bound.
0.9698	Q105	dependent	set to its moving lower bound.
0.9698	V105	transparent	released from its lower bound.
0.9745	Q12	dependent	set to its moving upper bound.
0.9745	V12	transparent	released from its lower bound.
0.9749	V55	transparent	set to its upper bound.
0.9749	Q55	dependent	released from its upper bound.
0.9784	V56	transparent	set to its upper bound.
0.9784	V59	transparent	released from its upper bound.
0.9800	V62	transparent	set to its upper bound.
0.9800	T3	transparent	released from its lower bound.
0.9851	V36	transparent	set to its lower bound.
0.9852	Q56	dependent	released from its moving upper bound.
0.9854	V54	transparent	set to its upper bound.
0.9854	T4	transparent	released from its lower bound.
0.9880	Q72	dependent	set to its moving lower bound.
0.9880	V72	transparent	released from its lower bound.
0.9890	Q54	dependent	set to its upper bound.
0.9890	V54	transparent	released from its upper bound.
0.9915	V63	transparent	set to its moving upper bound.
0.9915	Q54	dependent	released from its upper bound.
0.9915	Q56	dependent	set to its moving upper bound.
0.9915	V36	transparent	released from its lower bound.
0.9920	T3	transparent	set to its lower bound.
0.9920	T5	transparent	released from its lower bound.
0.9933	V59	transparent	set to its upper bound.
0.9933	V56	transparent	released from its upper bound.
0.9939	Q55	dependent	set to its upper bound.
0.9939	J59	dependent	released from its upper bound.
0.9951	Q107	dependent	set to its lower bound.
0.9951	V107	transparent	released from its lower bound.
0.9958	V30	dependent	set to its moving upper bound.
0.9958	V18	transparent	released from its upper bound.
0.9958	Q18	transparent	set to its lower bound.
0.9958	V26	transparent	released from its upper bound.
0.9962	Q99	dependent	set to its moving upper bound.
0.9962	V99	transparent	released from its upper bound.

TABLE 7.19 (cont.) - TEST ON THE 118 BUS SYSTEM

A SUMMARY OF BREAKPOINTS ENCOUNTERED IN THE FIRST SUBPROBLEM

CASE WITH REACTIVE GENERATIONS CONSIDERED

θ	Variable name and type		Cause of breakpoint
0.9970	V26	dependent	set to its lower bound.
0.9970	V19	transparent	released from its upper bound.
0.9971	Q19	dependent	set to its lower bound.
0.9971	V8	transparent	released from its upper bound.
0.9973	V43	dependent	set to its upper bound.
0.9973	V55	transparent	released from its upper bound.
0.9974	J143	dependent	set to its lower bound
0.9974	V32	transparent	released from its upper bound.
0.9974	Q31	dependent	set to its upper bound.
0.9974	V15	transparent	released from its upper bound.
0.9974	V32	trasnsparent	set to its upper bound.
0.9974	V31	transparent	released from its upper bound.
0.9974	Q49	dependent	set to its moving upper bound.
0.9974	V49	transparent	released from its upper bound.
0.9979	Q15	dependent	set to its lower bound.
0.9979	T6	transparent	released from its lower bound.
0.9982	V64	dependent	set to its moving upper bound.
0.9982	Q49	dependent	released from its moving upper bound.
0.9983	Q46	dependent	set to its upper bound.
0.9983	V46	transparent	released from its upper bound.
0.9983	V36	dependent	set to its upper bound.
0.9983	V41	transparent	released from its lower bound.
0.9987	V112	dependent	set to its lower bound.
0.9987	Q105	transparent	released from its moving lower bound.
0.9988	Q49	dependent	set to its lower bound.
0.9988	Q46	dependent	released from its upper bound.
0.9990	V40	dependent	set to its upper bound.
0.9990	V38	transparent	released from its upper bound.
0.9994	Q105	transparent	set to its upper bound.
0.9994	Q107	transparent	released from its lower bound.
0.9995	V8	transparent	set to its lower bound.
0.9995	V40	dependent	released from its upper bound.
0.9996	Q46	dependent	set to its upper bound.
0.9996	V43	dependent	released from its upper bound.
0.9998	V41	transparent	set to its lower bound.
0.9998	V64	dependent	released from its moving upper bound.
0.9998	V81	dependent	set to its lower bound.
0.9998	V36	dependent	released from its upper bound.
0.9998	V23	dependent	set to its moving upper bound.
0.9998	V32	transparent	released from its upper bound.
0.9999	V64	dependent	set to its moving upper bound.
0.9999	V59	transparent	released from its upper bound.

TABLE 7.19 (cont.) - TEST ON THE 118 BUS SYSTEM

A SUMMARY OF BREAKPOINTS ENCOUNTERED IN THE FIRST SUBPROBLEM

CASE WITH REACTIVE GENERATIONS CONSIDERED

θ	Variable name and type	Cause of breakpoint
0.9999	J159 dependent	set to its upper bound.
0.9999	V24 transparent	released from its upper bound.
0.9999	V107 dependent	set to its upper bound.
0.9999	V104 transparent	released from its lower bound.

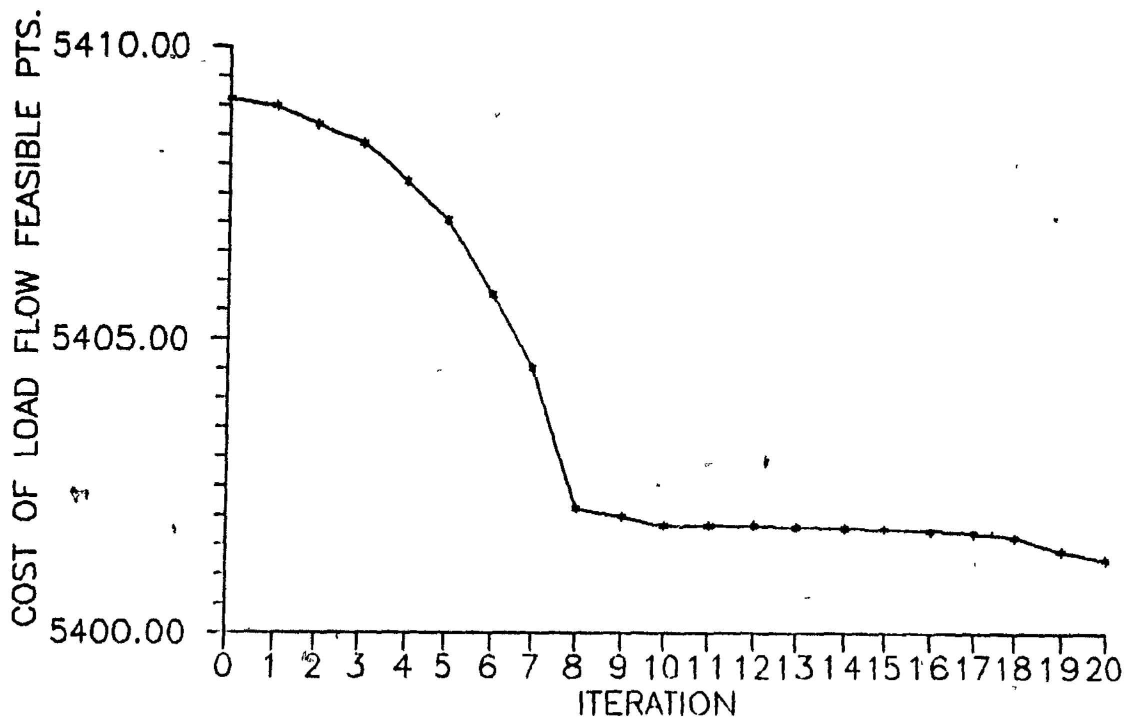


Figure 7.18.a. Costs of the load flow feasible points at each major iteration, solution to the 118 bus system.

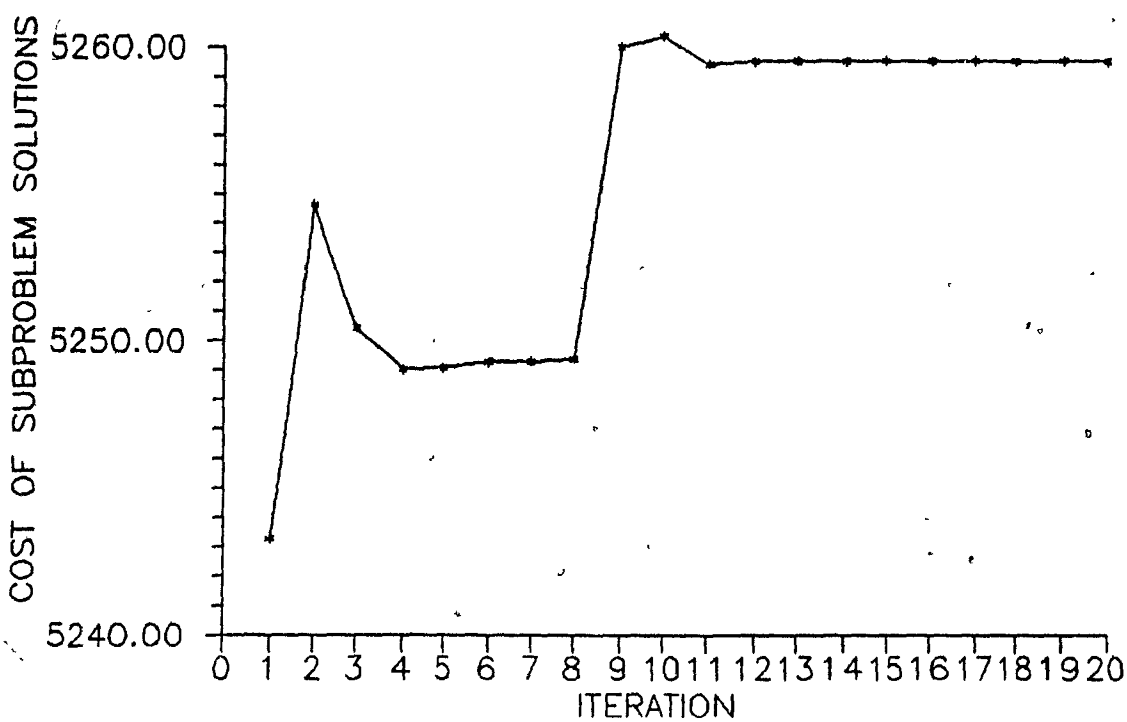


Figure 7.18.b. Costs of the subproblem solutions at each major iteration, solution to the 118 bus system.

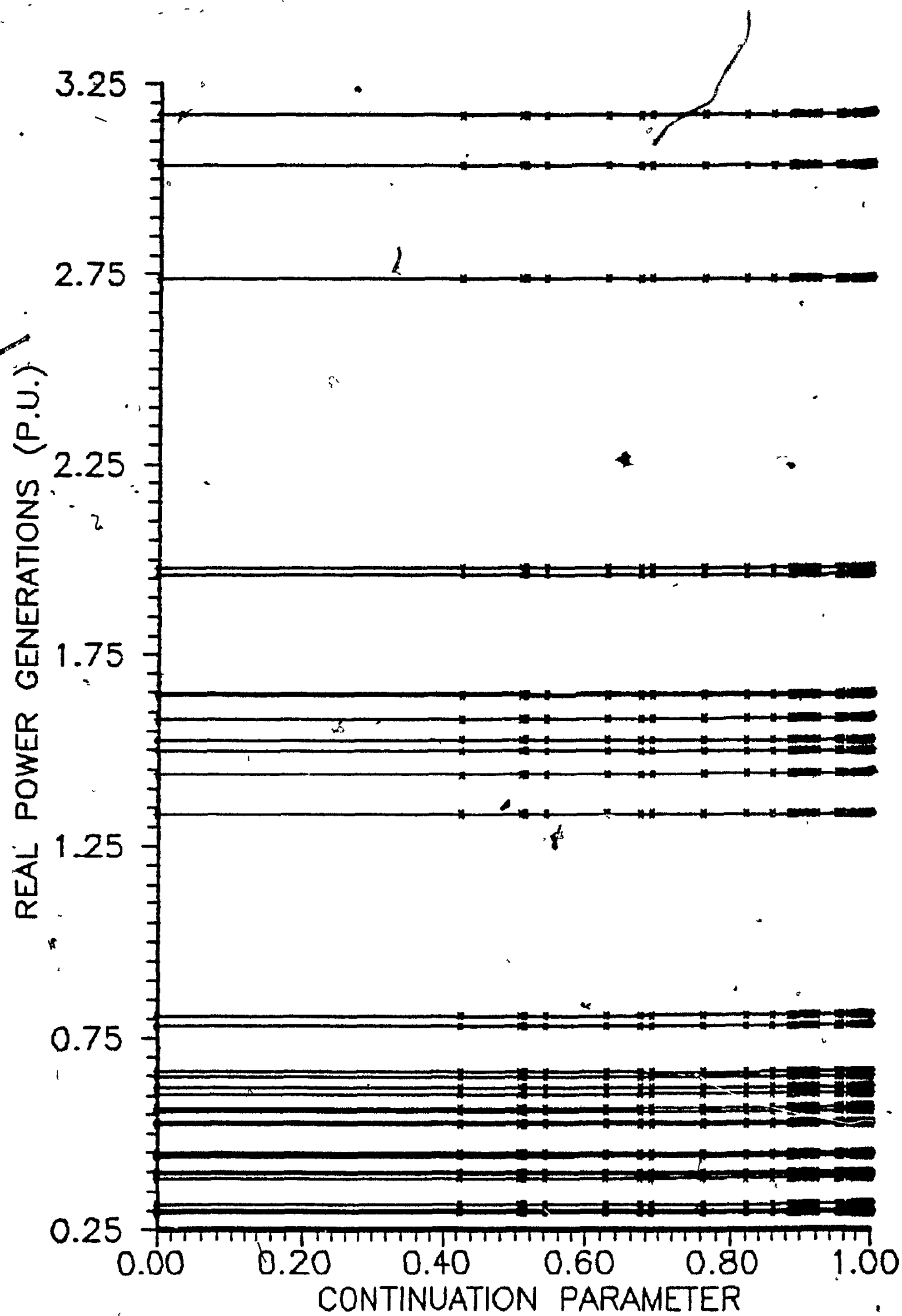


Figure 7.19.a. Real power generations vs. the continuation parameter, subproblem for the 118 bus system.

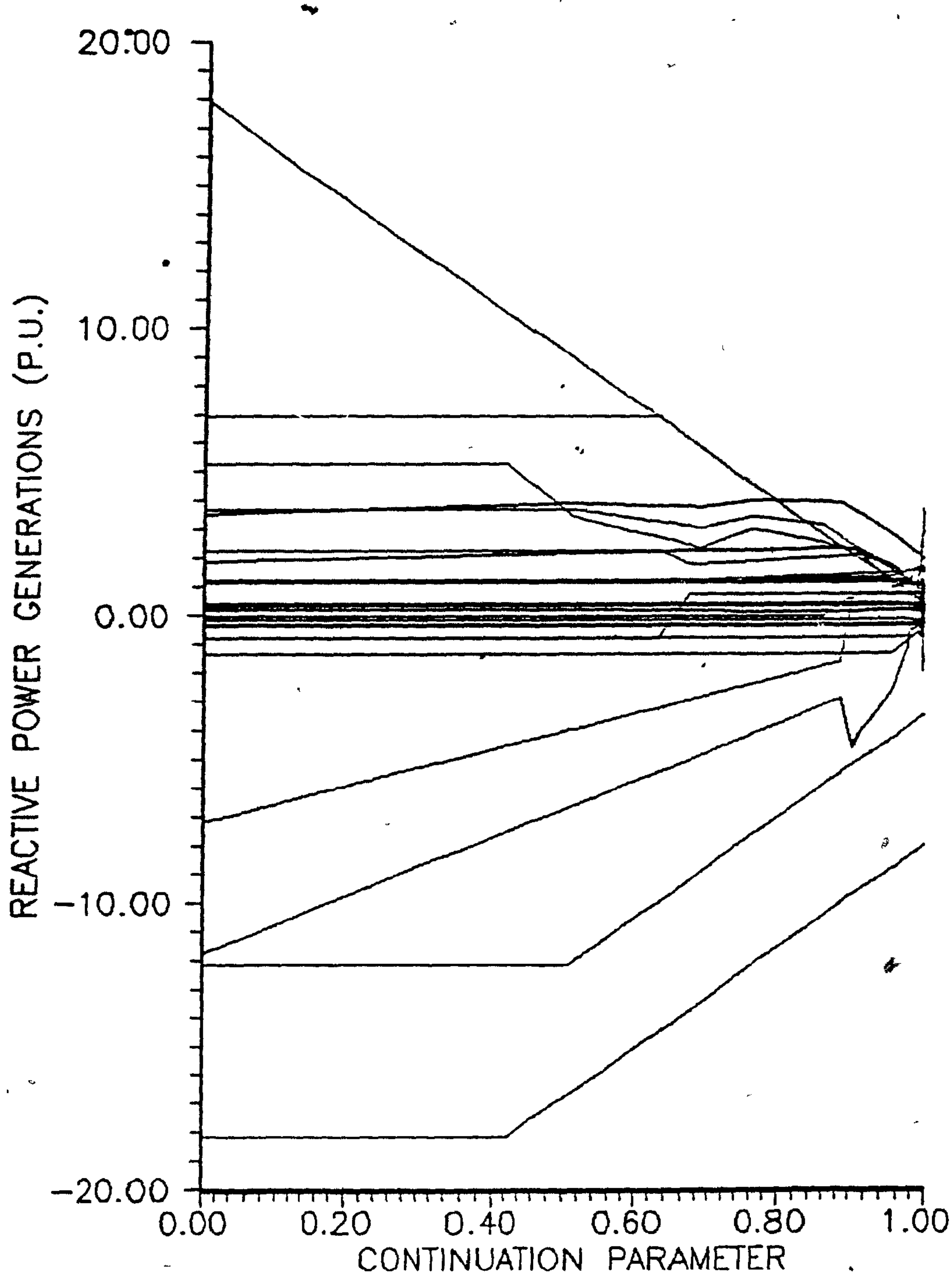


Figure 7.19.b. Reactive power generations vs. the continuation parameter, subproblem for the 118 bus system.

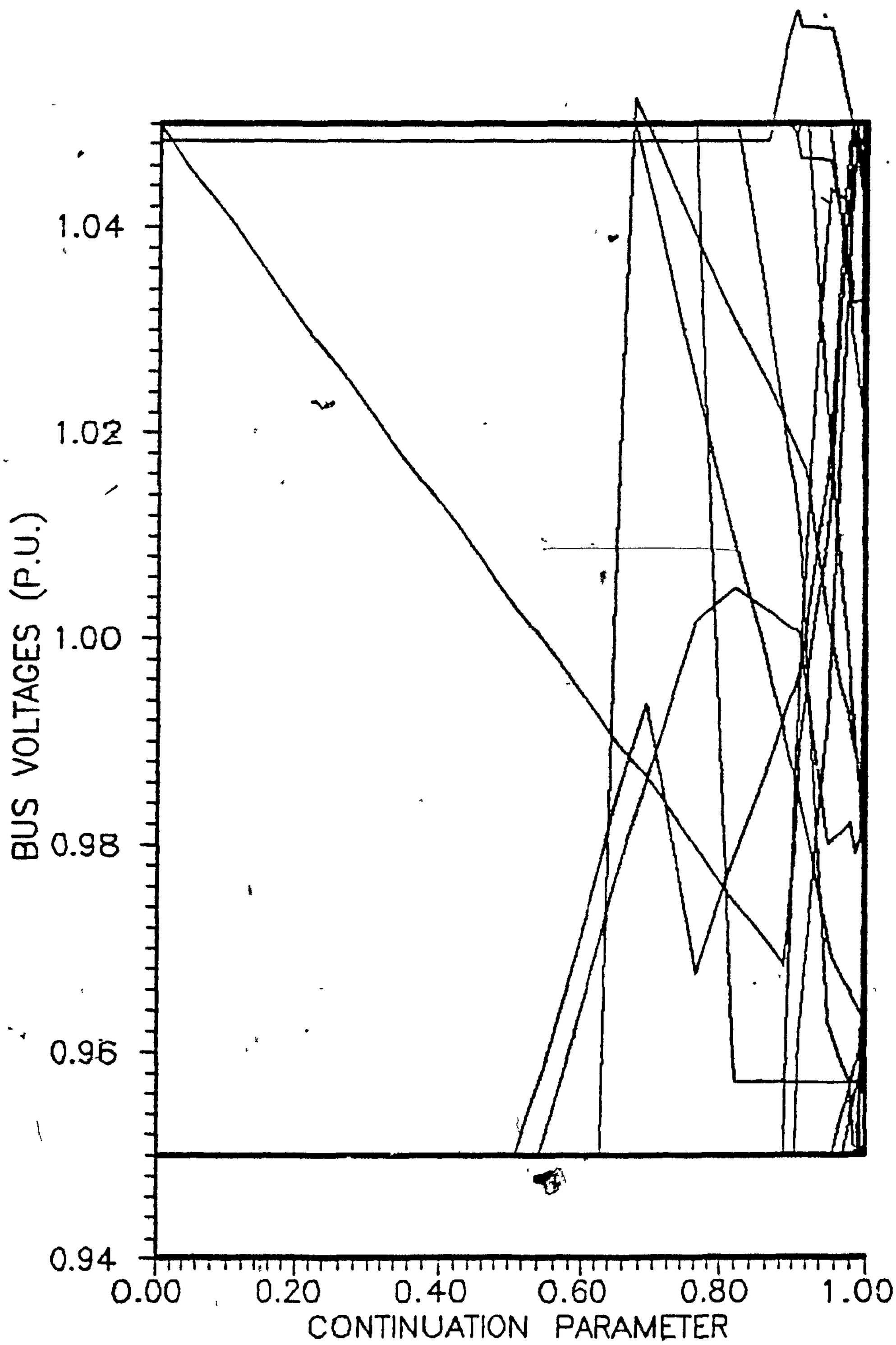


Figure 7.19.c. Selected transparent bus voltage magnitudes vs. the continuation parameter, subproblem for the 118 bus system.

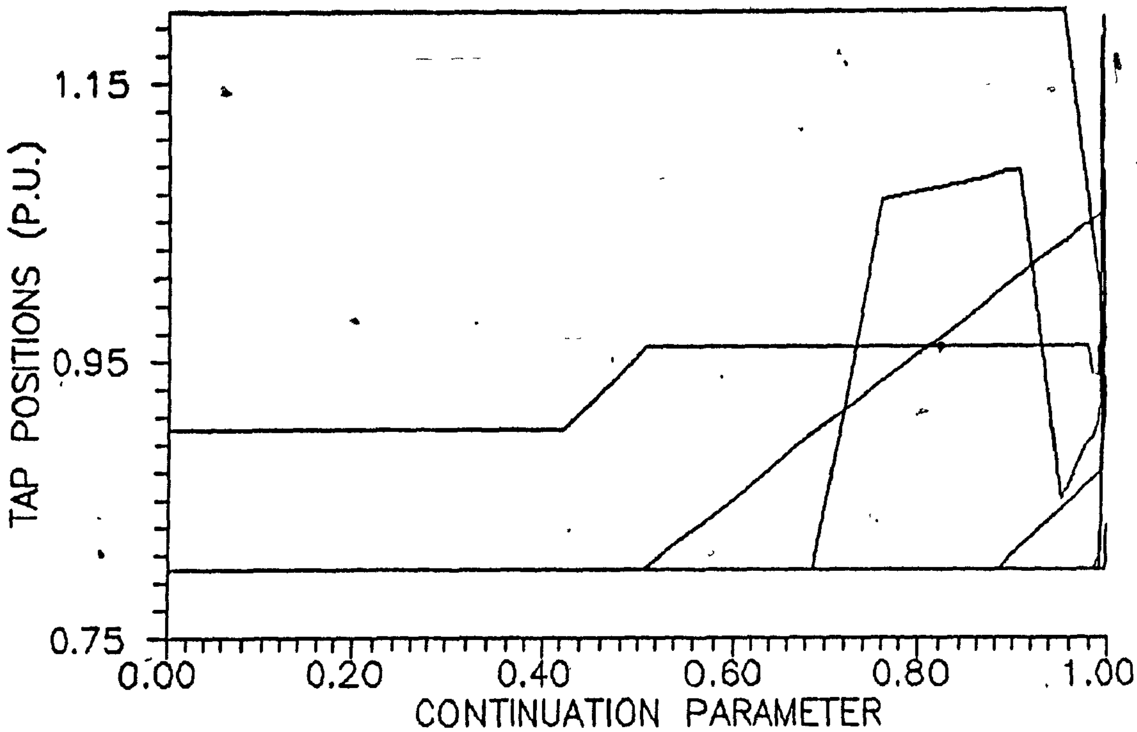


Figure 7.19.d. Variable transformer tap positions vs. the continuation parameter, subproblem for the 118 bus system.

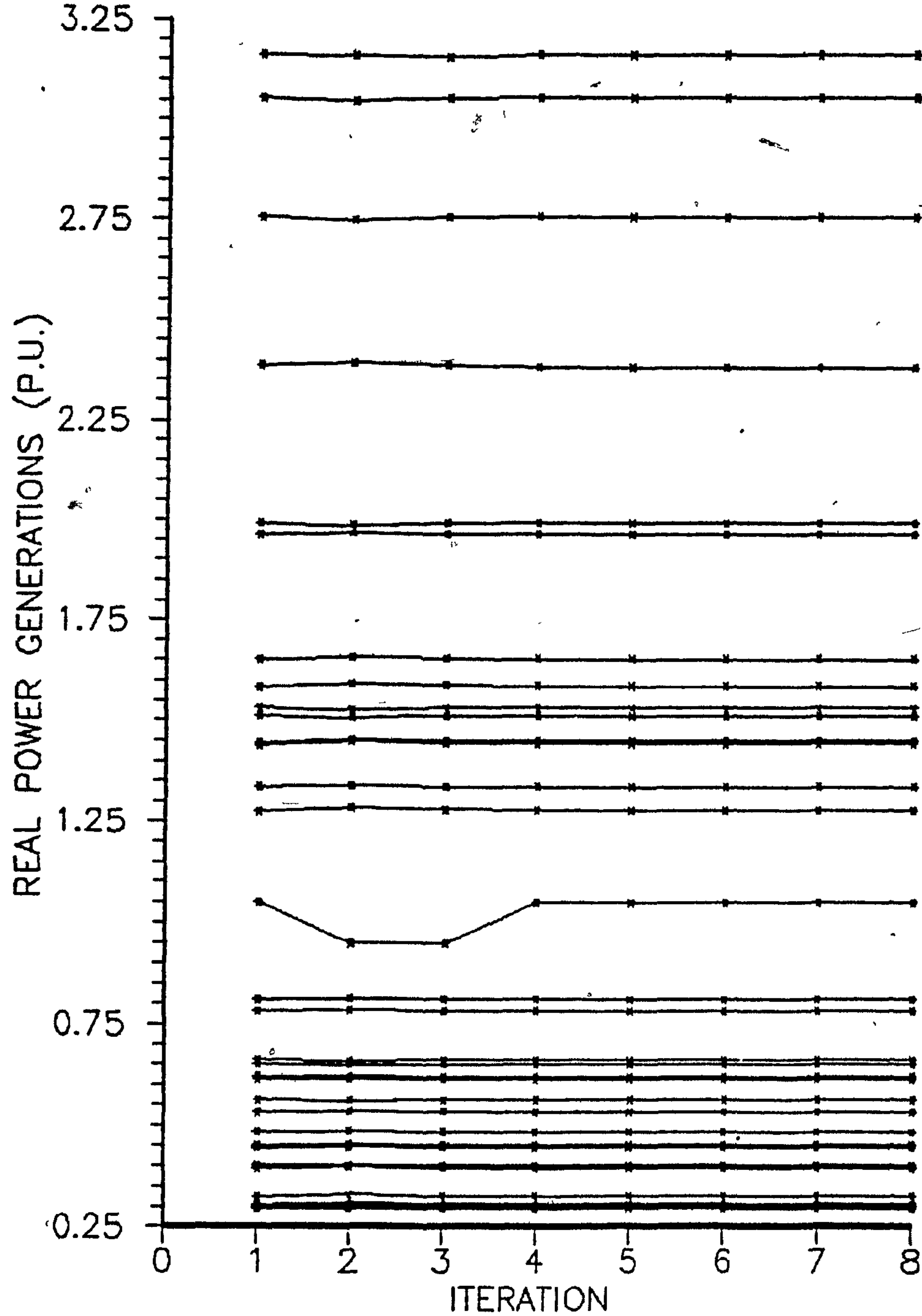


Figure 7.20.a. Subproblem real power generations vs. the major iteration number, solution of the first load of the 118 bus system.

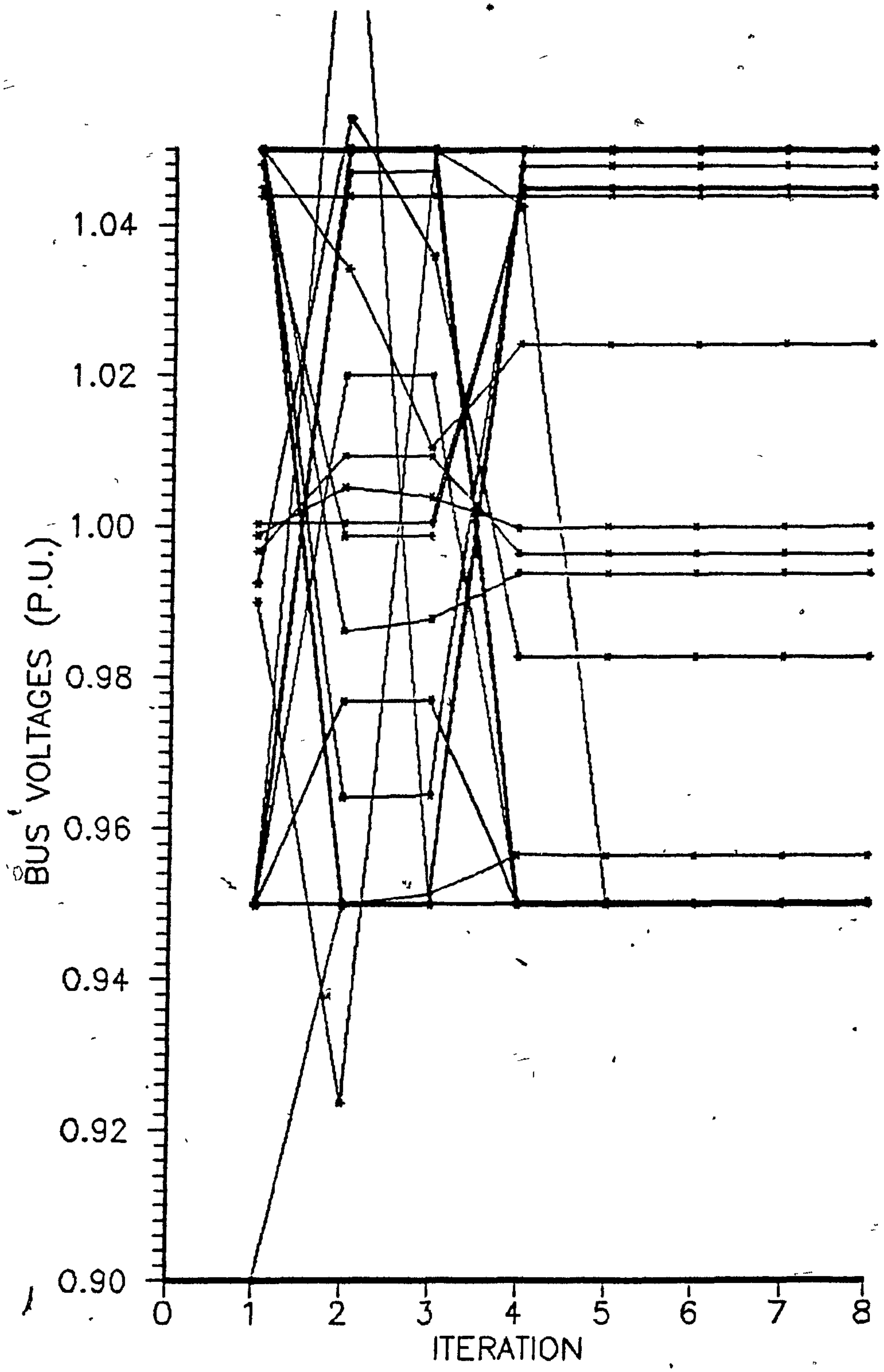


Figure 7.20.b. Subproblem bus voltage magnitudes vs. the major iteration number, solution of the first load of the 118 bus system.

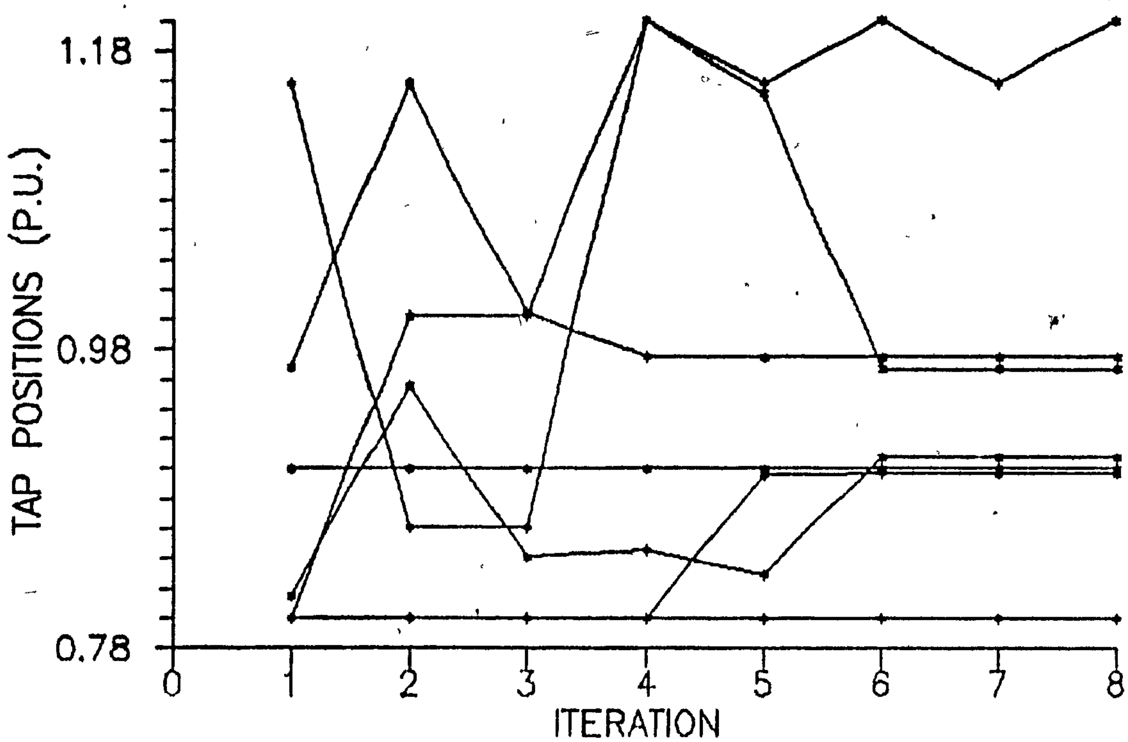


Figure 7.20.c. Subproblem variable tap positions vs. the major iteration number, solution of the first load of the 118 bus system.

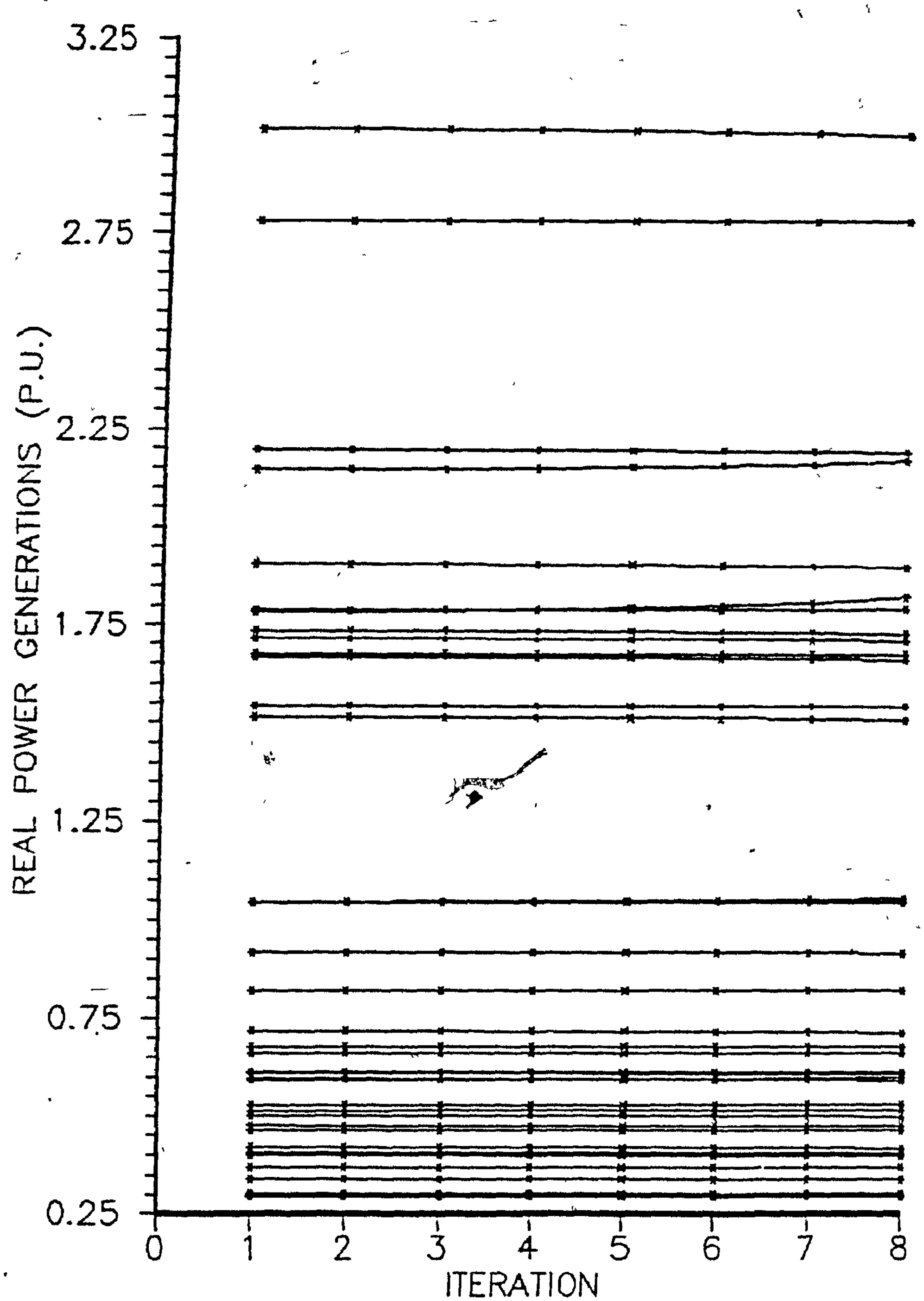


Figure 7.21.a. Feasible real power generations vs. the major iteration number, solution of the first load of the 118 bus system.

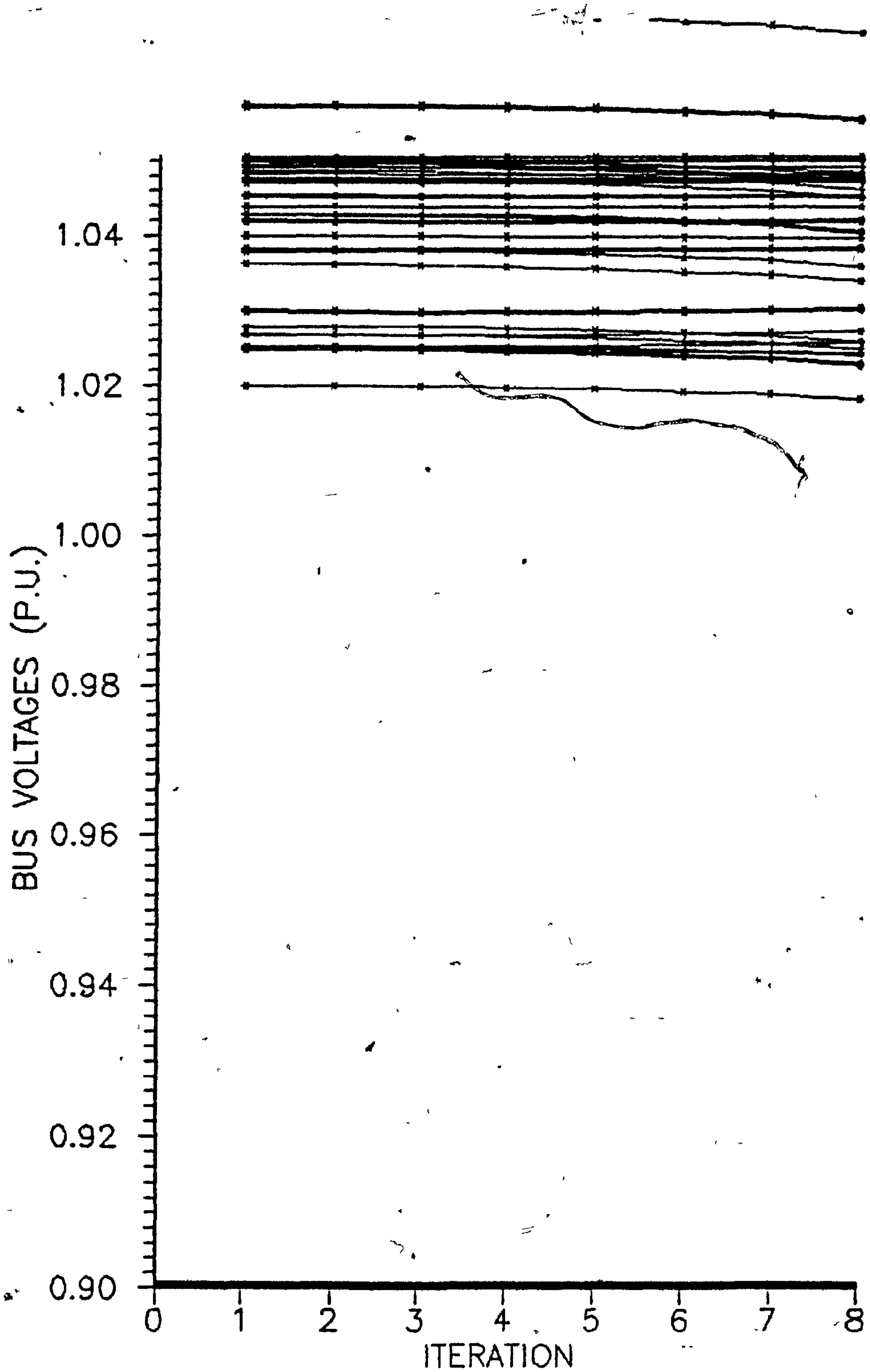


Figure 7.21.b. Feasible bus voltage magnitudes vs. the major iteration number, solution of the first load of the 118 bus system.

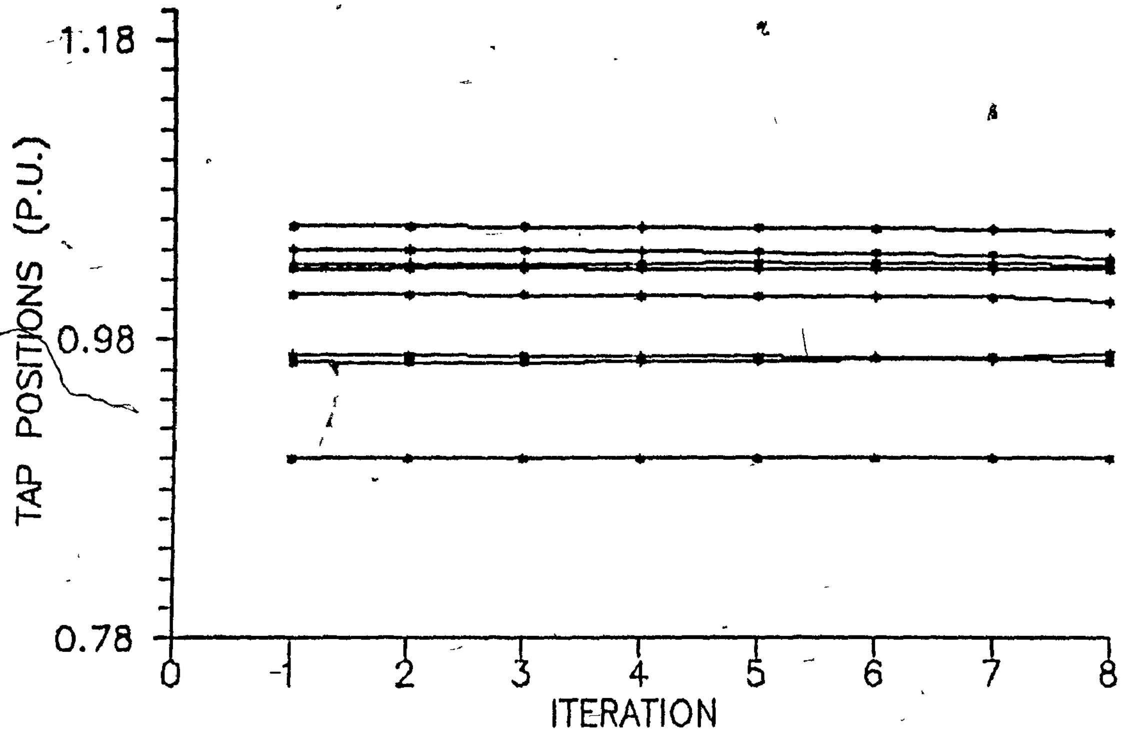


Figure 7.21.c. Feasible variable tap positions vs. the major iteration number, solution of the first load of the 118 bus system.

7.6 A Summary of the General Behaviour of the OPF Algorithm

This section regroups the general results and comments made in the previous descriptive sections 7.2 to 7.5. It discusses the general behaviour of our OPF algorithm as observed in the results, in some cases confirming ideas presented in the previous chapters, and in others describing unforeseen difficulties.

In our tests the real power generations and the costs of the initial guesses were often close to those of the optimal solution. The resulting small improvements in the convergence criteria made it difficult to assess the rate of convergence, but it was most probably linear in all cases. However, some tests systems showed better convergence than others. Most of the progress in solving the optimization was made in the first major iteration or two; that is best illustrated in the 30 bus test, where the initial guess is farthest from the optimal solution. With the chosen tolerances though, 6 to 9 major iterations were required to reach solutions deemed accurate enough.

In studying the results, it is felt that the closeness of the initial guess to the optimal solution causes the algorithm to act sluggishly, and did not help the optimization process. That is explained by the small step sizes which are imposed on the optimization right from the start, and possibly by the ill-conditioning which, as we have often conjectured, accompanies the optimum.

Typically, the first four to six subproblems in the initial nonlinear program started with different sets of independent variables, and produced different violations which caused relatively many breakpoints (i.e.,usually 20 to 30 breakpoints). After that the active set settled down, although a few breakpoints were often required in subsequent subproblems. Even if the initial active set is chosen close to the optimal active set, it seems that the algorithm needs a few major iterations to settle down. Looking at the other end of the solution process, our tests indicate that had more major iterations been allowed in the solutions to the initial loads, the subproblem's active sets would have settled to some definitive active set;

this was seen when the 6 bus test was extended. In some of our tests, the active set finally settled in the load-tracking mode.

Judging from our results and from empirical results in linear and quadratic programming, the number of breakpoints in a typical subproblem compared favorably to the number of iterations of a linear or quadratic program. The results indicate that the number of breakpoints is independent of the system size, but is linked to the closeness of the subproblem's active set and its final active set.

The subproblem solutions encountered few problems on the smaller systems, but ran into serious ill-conditioning problems in the 118 system. The problem is linked to the effect of the system sparsity on the computation of sensitivity coefficients. That idea is developed in section 7.8.(i) and in Appendix 7.5.

The subproblem solutions exhibit some noticeable characteristics. First, the breakpoints always occur in pairs. The first is caused by a variable reaching its limit and the second occurs almost immediately after, when another variable comes off its bound as a form of compensation. Second, the majority of the breakpoints occur near the end of the continuation process, as the variables are being "squeezed" back into the feasible region. The typical solution trajectories for the variables are described in the following. The trajectories of the real power generations were mostly flat, as could be expected for the Q-V redispatching problem which this turns out to be. The most violated dependent variables followed their moving bounds, and some of the others, less violated, returned to the feasible region on their own. The first transparent variables to be released from their imposed bounds moved slowly, but those leaving towards the end of the continuation process often moved sharply. Many of those released from one bound ended up on the other bound.

Although the 118 bus test was plagued with ill-conditioning, the solution trajectories of its subproblem, described in fig. 7.18, were similar to those in the previous tests. The difficulties manifested themselves in the larger number of breakpoints. A breakpoint involving a variable often caused large in the previous tests. The difficulties manifested themselves in the larger number of breakpoints. A breakpoint involving a variable often caused large sudden changes in the neighboring variables or their Lagrange multipliers; as a result many more variables than usual were involved in the active set. Another observed problem was that some variables were forced on and off their bounds a few times, even though the movements of the variables from the bounds were small. Despite these difficulties the continuation algorithm found its optimal solution within a reasonable number of breakpoints for a problem of this size.

The computation of feasible load flow points using a Newton-Raphson algorithm with the standard "slack bus" formulation was very efficient and reliable throughout the tests. The subproblem solutions were usually close to the load flow manifold, so that the Newton-Raphson solver rarely required more than 4 iterations to converge. The slackless load flow was called upon a few times when the standard load flow could not provide a cheaper feasible point. It also performed well.

The reduction of the convergence criteria and of the step size were described as being linear, but with some tests showing faster convergence than others. Two types of behaviour are observed. The 10 bus and 118 bus tests started very close to the optimum. As a result, the step size decreased abruptly and subsequent improvements were very slow. The 6 and 30 bus tests started far enough from the optimum for the step size to decrease progressively. Consequently the improvements in the convergence criteria were less restricted. The number of required major iterations was somewhat similar in the 4 tests, despite the convergence characteristics, but the latter behaviour would be preferable to achieve convergence to tighter tolerances.

The progression of the variables and the objective function from one major iteration to the next is summarized in the following. We noted that the real power generations reached their optimal values quickly. The corresponding costs of the load flow feasible points always decreased (by design), and the subproblem solution costs were usually seen to form progressively improving lower bounds on the optimal cost. The other variables, which have little effect on the cost, are more difficult to evaluate. Their values from the subproblems can oscillate from one major iteration to the next. The step size acts to damp out those large swings in the sequence of load flow feasible points, allowing the algorithm to eventually declare convergence.

The coordination between the subproblem solutions and the load flow feasible points, and their eventual convergence to a single optimum are not necessarily guaranteed. Two aspects of the problem are (1) the convergence of the numerical values and (2) the convergence of the active sets. These aspects are discussed below.

The convergence of the load flow feasible variables to an optimal solution is achieved by moving in the directions designated by the subproblem solutions, but only as far as allowed by a step size. When the step size is reduced quickly, very little improvement is achieved from one major iteration to the next. Then the computed values cannot be declared optimum without leaving a relatively large gap between the subproblem solution and the corresponding load flow feasible point. This can be seen in the results of the 10 and 118 bus systems. To avoid such small step sizes, our algorithm allows for increases in the step size during the process, but few increases were ever implemented. A new study of the step size computation procedure could be worthwhile to improve upon the present situation.

The problem of the coordination between the active sets of the subproblem and of the load flow feasible point is more complex. As was discussed previously, some variables of the subproblem solutions are likely to go to a bound even though their optimal values in the nonlinear problem are nowhere near the bound. The provisions for transparent variables with small sensitivity coefficients in the subproblem did not alleviate the problem, probably because the heuristics designed to handle the problem are incomplete or inefficient. In our results we often observed large differences in the active sets between the computed load flow optimum and the last subproblem solution.

If the optimization converges well (as in the 6 bus and 30 bus cases) and the subproblems cease to produce breakpoints, the sequence of subproblem solutions should eventually coincide with the sequence of load flow feasible points. However in most cases the optimization would not iterate that long, so that in practice their active sets rarely coincide.

To confirm the convergence properties claimed in the previous paragraph, an extended 6 bus test was allowed to run for 80 major iterations. In that test a long series of subproblems was generated without breakpoints. The movement of the load flow feasible points towards the stable sequence of subproblem solutions was slow due to the small step size. At the end of the series, the two groups of variables and their active sets coincided. Then later, in the load-tracking mode, the subproblem modified its active set, with two variables switching bounds. The active set of the load flow feasible solutions did not follow suit, and the active sets of the two groups never had a chance to coincide again.

Finally, the load-tracking scheme produced some very fast solutions to sequences of increasing loads. Average solution times for the 4 tests in the load-tracking mode were from 15% to 30% of the timings for the initial solutions, a substantial gain in speed. The real power generation schedules produced by the load tracking were very smooth. Schedules for the other variables, although less smooth, were quite good. This shows that the continuation principle can be very useful in quickly and accurately solving a sequence of nonlinear optimization problems. The present implementation has been unreliable in certain tests, but its problems are not linked to the continuation principle. This will be discussed in section 7.8.

7.7 Comparison of Results from Various Programs for the 30 Bus System

Our results for the 30 bus system are compared to those published by Ponrajah in his Ph.D. thesis [Ponrajah 1987] and to those of [Alsac and Stott 1974]. Ponrajah's work also includes the optimal solution for this system obtained using the general optimization program MINOS [Murtagh & Saunders 1983]; part of this solution is also repeated here.

No one solution is considered as a reference in our comparison. Its main purpose is of course to validate our results, but we are also looking for general difficulties of the optimization process, which we feel might have occurred in the other programs also.

The optimal values of the more important variables from the four different sources are given in Table 7.20. In brief, here are the major points:

- The optimal costs are very close to each other, with a maximum discrepancy of 0.047%. Hence the four results can be considered equally accurate. The best results might have been obtained with more iterations.
- Despite the similar optimal costs, there are some non-negligeable differences in the values of the real power generations. Our results stand slightly apart from the others. For example, the maximum discrepancies between our results and the others for units 3 and 4 are almost 2%.
- For the other variables, there are some large differences on some quantities and very little on others. Based on the discussions of the previous section and considering that different strategies were used to compute the quantities, these differences had to be expected.
- Differences in voltages are rather small, and no one solution stands apart from the rest. Voltages V29 and V30 in the last column are marked with asterisks; their values are much higher than the values to which they are compared, but only because their upper limits were set higher.

TABLE 7.20 - COMPARISON OF RESULTS FOR THE 30 BUS SYSTEM					
Variable	Status	This program	Ponrajah	MINOS	Alsac & Stott
Real Power Generations (MW)					
Unit(1)	Free	175.96	176.05	176.11	176.26
Unit(2)	Free	49.10	48.84	48.84	48.84
Unit(3)	Free	21.91	21.52	21.52	21.51
Unit(4)	Free	21.76	22.16	22.20	22.15
Unit(5)	Free	12.10	12.25	12.26	12.14
Unit(6)	Lwr bnd	12.00	12.00	12.00	12.00
Optimal Fuel Costs ($/hr)					
		802.31	802.22	802.60	802.40
Reactive Power Generations (MVar)					
Unit(1)	Free	-15.20	-13.94	-14.68	n.a.
Unit(2)	Free	19.37	30.32	29.92	n.a.
Unit(3)	Free	29.56	29.89	30.15	n.a.
Unit(4)	Free	32.75	36.48	36.66	n.a.
Unit(5)	Free	9.78	14.43	14.74	n.a.
Unit(6)	Free	15.76	8.44	8.97	n.a.
Bus Voltages at Generation Buses (P.U.)					
V(25)	Upp bnd	1.050	1.050	1.050	1.050
V(26)	Free	1.036	1.038	1.038	1.038
V(27)	Free	1.011	1.010	1.011	1.011
V(28)	Free	1.017	1.019	1.019	1.019
V(29)	Free	1.037	1.050	1.050	1.091*
V(30)	Free	1.044	1.050	1.050	1.091*
Tap positions (P.U.)					
T(1)	Free	1.019	1.005	1.002	1.003
T(2)	Free	0.978	0.956	0.954	0.960
T(3)	Free	1.028	1.100	1.100	1.047
T(4)	Free	0.987	1.037	1.035	0.942
Shunt Controller Admittances (S)					
B(1)	Free	0.028	0.050	0.050	-
B(2)	Free	0.049	0.050	0.050	-
B(3)	Free	0.034	0.050	0.050	-
B(4)	Upp bnd	0.050	0.050	0.050	-
B(5)	Free	0.042	0.050	0.050	-
B(6)	Free	0.042	0.050	0.050	-
B(7)	Upp bnd	0.050	0.037	0.037	-
B(8)	Upp bnd	0.050	0.050	0.050	-
B(9)	Free	0.032	0.029	0.029	-

- The values of the reactive powers, shunt admittances and taps also show some large discrepancies on about half of the variables. Only the tap positions are given by all four sources and again, no one result stands apart from the rest.

The four programs are basically driven by the same criterion, to find the minimum cost. Hence it is only normal that they found the same optimal costs. A bit surprisingly, the optimal real power generations show some small but non-negligeable discrepancies. That indicates that the objective function is rather flat near the optimum, even as a function of the real power generations.

The remainder of the comparison tends to confirm the idea that the optimal values of variables which are not directly cost-related are difficult to compute. It also shows that their values are not critical in minimizing fuel costs. In view of the difficulties in their computation, the comparison tends to justify the recent prevailing strategy in OPF, which is simply to locate feasible Q's, V's and passive control settings.

7.8 Numerical and Algorithmic Difficulties Encountered in the Program

This section presents some of the unforeseen difficulties encountered in the program's operation. Basically, two important problems are the most often responsible for program failure: (1) for large sparse systems, ill-conditioning of the subproblem; and (2) inappropriate modifications to the set of independent variables, which can occur in various stages of the program. In the following, both problems are analyzed and remedies for the problems are sketched.

i) Ill-conditioning of the subproblem

The ill-conditioning of the optimality equations $Ax=b$ in the subproblem stems numerically from two problems. For one, the sensitivity coefficients used to model the dependent constraints are too often 4 to 6 orders of magnitude smaller than the coefficients of the power balance equation. Appendix 7.5 traces the root causes of this problem. Another problem is that neighboring dependent variables are often expressed as almost similar functions. Physically, this corresponds to the fact that the effects of reactive powers and voltages are very localized in the network. Hence their sensitivities with respect to independent variables situated far away in the network are naturally very small. This ill-conditioning seems to be a normal occurrence, and is an inherent problem of sparse systems when using the "compact" load flow model.

An illustration from the 118 bus system is given in Table 7.21, with 8 functional constraints present. The columns in the table are the rows of the G matrix of eq. 4.10. The values in each column of the table are mostly of the same order of magnitude, but the orders of magnitude of each column range from 10^{-1} to 10^{-9} for dependent constraints, compared to values of about one for real power generations in the power balance equation. The right-hand sides of the constraints are all of the same order of magnitude however, so that scaling is inappropriate. As a result, the computed slopes of the solution trajectories have very large values. The variables then move very quickly onto or off of their bounds. The problem is that the ill-conditioning often causes moves in the wrong direction. As witnessed earlier, some dependent variables follow their moving bounds closely in the continuation process, but the algorithm often moves them in and out of the active set at great computational expense. Also, the changes in the transparent variables are exaggerated. Another problem is that the continuation process virtually stalls at a value of the continuation parameter. For example, in the 118 bus test documented in Table 7.19, 54 breakpoints were required to increase the continuation parameter from 0.995 to 1.000.

A post-mortem analysis was carried out on two runs of the 118 bus test to monitor the ill-conditioning in the optimization process, by computing the condition number of the A matrix at each breakpoint. It was computed as the quotient of the largest to the smallest singular values of the A matrix; these values were computed using the LINPACK subroutine DSVDC [Dongarra et.al. 1979]. The results are shown in fig. 7.22 a. and b. The first is the easily solved case described in Table 7.18, and the second is a run that failed. We note that in both cases the condition number jumped at least an order of magnitude when a constraint was added to the active set, and similarly it dropped when a constraint was released. However at some point the release of a constraint hardly lowered the condition number. This could serve as a sign to give special treatment to the last activated constraint. Our program did no such thing, and from there the continuation process encountered many breakpoints and slowed considerably.

In the case which was easily solved, illustrated in fig. 7.22.a. the process eventually found an active set with a low condition number, and soon after it reached its solution. Still, 11 breakpoints were required to move the continuation parameter from Θ=0.9708 to Θ=1.00.

In the case that eventually failed, illustrated in fig. 7.22.b., the condition numbers remained high. After 54 iterations, the release of a constraint brought no relief to an incredibly high condition number of more than 10^{20}. At that point the program could not proceed normally so it rearranged the partition of independent/ dependent variables. It then proceeded with another 29 iterations, of which only the first few are shown in the figure. The condition numbers after the change were similar to those observed before the change. The program then failed.

Remedies to the ill-conditioning problem in the subproblem are more algorithmic than mathematical in nature. The suggestions made in Appendix 4.2 on degeneracy would have to be implemented, to monitor the near-singularity of the optimality equations and to manage the "offending" dependent constraints separately from the others. Computing condition numbers on-line to detect offending constraints would be prohibitively time-consuming. Instead, simpler techniques are available, for example based on the study of the diagonal element entering the Cholesky factorization. If this value is too small, the new constraint could be set aside, and would not enter the constraint submatrix. Presently all the needed elements for this study are in place in our program, so it would not be difficult to implement. Some study would be required though to determine the appropriate heuristics. A seemingly similar scheme is implemented in the nonlinear programming subroutine VE05 of the Harwell Libraries [Hopper 1977].

TABLE 7.21 - COEFFICIENTS OF THE FUNCTIONAL CONSTRAINTS IN THE 118 BUS TEST

INDEP. VAR.	FUNCTIONAL CONSTRAINTS							
	Power Balance Eq	J59	V64	V30	V112	V86	J172	V63
P 107	0 8987E+00	-.3699E+00	0.4258E-04	0.3500E-04	- 3311E-04	- 8775E-03	0.4338E-09	- 5999E-07
P 103	0.8751E+00	-.3602E+00	0 4146E-04	0.3408E-04	-.3224E-04	0.3458E-03	0 4224E-09	-.5842E-07
P 100	0.9062E+00	-.3730E+00	0 4293E-04	0.3529E-04	-.3338E-04	- 2796E-08	0.4374E-09	-.6049E-07
P 90	0 8975E+00	- 3582E+00	0 4120E-04	0.3387E-04	-.2893E-04	-.2133E-08	0 6618E-09	-.5873E-07
P 89	0.9002E+00	- 3585E+00	0.4124E-04	0.3390E-04	-.2875E-04	-.2115E-08	0 6848E-09	-.5884E-07
P 87	0 8979E+00	-.3496E+00	0.4019E-04	0.3304E-04	- 2573E-04	-.1846E-08	0 1251E-01	-.5785E-07
P 85	0.9120E+00	-.3551E+00	0.4082E-04	0.3356E-04	- 2613E-04	- 1875E-08	0 9264E-09	- 5875E-07
P 80	0 9853E+00	-.4292E+00	0 4947E-04	0.4066E-04	-.4503E-04	-.1106E-08	0.2470E-09	-.6826E-07
P 76	0.1019E+01	- 2079E+00	0 2272E-04	0 1868E-04	0 1163E-03	- 6305E-09	0.1566E-09	- 6145E-07
P 74	0 1010E+01	- 1069E+00	0 9885E-05	0.8126E-05	0.2602E-03	-.3618E-09	0.8938E-10	-.7279E-07
P 69	0 1000E+01	0 0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0 0000E+00	0 0000E+00	0 0000E+00
P 66	0 9943E+00	0 1050E+00	- 2452E-03	-.2653E-03	- 2032E-03	- 3377E-09	0.7570E-10	-.1470E-06
P 65	0 9949E+00	0.1181E+00	0.1193E-03	0.9808E-04	- 2159E-03	- 3753E-09	0.8407E-10	- 1411E-06
P 62	0.1015E+01	0.1117E+00	-.7575E-03	- 1028E-02	-.2118E-03	-.3577E-09	0 8016E-10	-.1480E-06
P 61	0.1014E+01	0 1127E+00	-.8406E-03	-.1192E-02	- 2127E-03	- 3605E-09	0.8079E-10	-.1474E-06
P 59	0.1037E+01	0 1116E+00	- 1809E-02	- 1289E-02	- 2139E-03	-.3581E-09	0 8027E-10	- 1524E-06
P 56	0.1063E+01	0 1053E+00	-.1254E-02	- 9180E-03	- 2105E-03	- 3411E-09	0.7648E-10	- 1604E-06
P 55	0 1062E+01	0 1056E+00	- 1282E-02	- 9369E-03	-.2106E-03	- 3419E-09	0 7666E-10	-.1600E-06
P 54	0 1063E+01	0 1049E+00	- 1232E-02	- 9028E-03	- 2102E-03	-.3402E-09	0 7628E-10	- 1606E-06
P 49	0.1016E+01	0.8962E-01	- 4781E-03	- 3930E-03	- 1905E-03	- 2945E-09	0.6609E-10	- 1583E-06
P 46	0.1019E+01	0.8279E-01	- 4130E-03	- 3395E-03	-.1604E-03	- 2760E-09	0.6198E-10	-.1664E-06
P 42	0.1031E+01	0 8976E-01	-.2936E-03	- 2414E-03	- 2769E-03	- 3194E-09	0.7196E-10	- 2859E-06
P 36	0 9477E+00	0 8093E-01	- 7161E-04	-.5886E-04	-.2589E-03	- 3144E-09	0 7110E-10	-.3946E-06
P 26	0.8829E+00	0 5447E-01	-.2535E-04	- 2084E-04	0.3043E-02	-.2846E-09	0.6508E-10	- 4784E-06
P 25	0 8775E+00	0 4849E-01	- 2439E-04	- 2005E-04	0.2618E-02	- 2793E-09	0.6406E-10	- 4536E-06
P 12	0 8965E+00	0 6623E-01	-.3081E-04	- 2532E-04	0.2882E-02	- 2954E-09	0.6717E-10	- 6479E-06
P 10	0 8592E+00	0 6399E-01	- 2781E-04	- 2286E-04	0 3431E-02	-.2836E-09	0 6449E-10	- 4296E+01
P 8	0.8816E+00	0 6567E-01	- 2853E-04	- 2346E-04	0 3521E-02	-.2910E-09	0 6617E-10	- 7097E-06
P 6	0.8850E+00	0.6552E-01	- 2994E-04	- 2461E-04	0 3031E-02	- 2917E-09	0 6634E-10	- 6593E-06
P 4	0 8735E+00	0 6480E-01	- 2918E-04	- 2399E-04	0 3143E-02	- 2881E-09	0 6550E-10	- 6668E-06
P 1	0 8839E+00	0.6540E-01	- 3004E-04	- 2469E-04	0 2976E-02	- 2913E-09	0 6625E-10	-.6531E-06
V 32	0.1471E-01	0 6909E-03	- 1656E-05	- 1361E-05	- 2441E-01	- 4453E-11	0 1024E-11	0.3253E-07
T 4	- 1098E+00	- 1021E-01	0.2972E+00	0 4587E+00	0 2109E-04	0.3332E-10	-.7475E-11	0 1687E-07
V 18	0 1508E-01	0 2342E-02	- 4245E-05	- 3490E-05	-.1102E+00	- 5306E-11	0 1163E-11	0 1723E-06
T 1	0 2527E-02	- 9839E-03	0 1131E-06	0 9298E-07	- 7240E-07	- 5196E-11	-.3856E+00	-.1628E-09
V 105	- 1588E-02	0 6532E-03	- 7521E-07	- 6179E-07	0 5846E-07	0 5930E+00	- 7664E-12	0 1059E-09
V 8	0 4124E-01	0 3068E-01	0 2694E-04	0.2215E-04	- 3436E+00	- 3488E-10	0.7070E-11	0 7285E+02
V 59	0 3208E-01	- 3965E-01	- 4628E+00	- 1863E+00	0 3505E-04	0 1121E-09	- 2496E-10	-.2457E-07
V 34	- 5177E-01	0 1648E-01	- 5552E-03	- 4564E-03	- 1074E+00	0 7375E-10	- 1802E-10	0 7280E-06
V 77	0 4382E+00	0 1196E+00	-.1447E-04	- 1189E-04	0 8839E-04	0 5848E-09	-.1782E-09	0 3433E-08
T 2	- 1240E-01	0 1433E+02	- 8767E-04	- 7206E-04	0.1833E-03	- 2519E-09	0 5595E-10	0 9822E-07

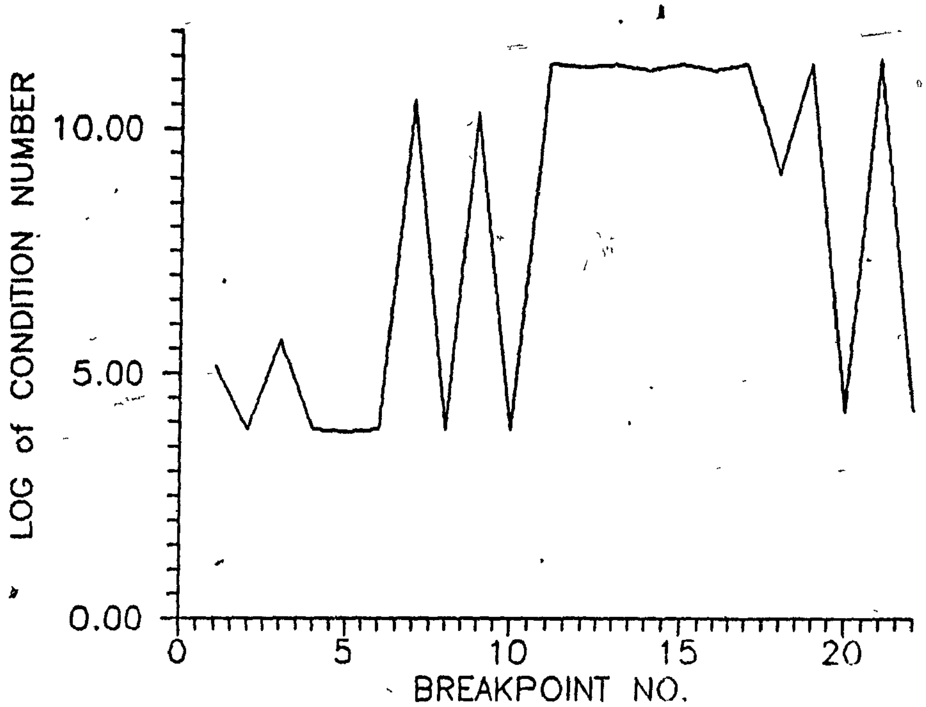


Figure 7.22.a Condition number vs. subproblem iteration, first case of the 118 bus system.

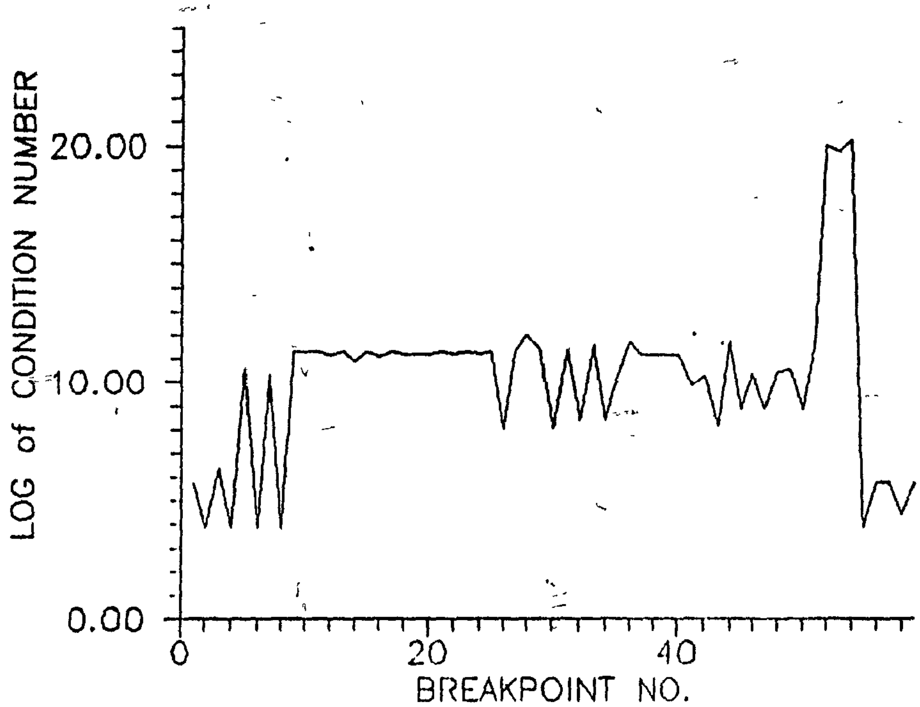


Figure 7.22.b Condition number vs. subproblem iteration, second case of the 118 bus system.

ii) Inappropriate changes in the set of independent variables

There are four circumstances in our OPF algorithm which call for a change to the set of independent variables. Presently, two of these are not foolproof, and can lead to cycling in the program. In a first case, the change is summoned in the subproblem when the continuation method stalls. A change in the set of independent variables often results in better-conditioned functional constraints, and the continuation process continues. In a second case, the change is sometimes needed in the load flow computation when varying the load. As in most load flow solvers, when a dependent variables violates its bound it is made independent and set to that bound, in order to maintain bound feasibility. In the process a previously independent variable is made dependent, and the load flow computation is repeated. Both cases have been observed to work well most of the time. Unfortunately in its present form, the program allows variables to cycle endlessly in and out of the set of independent variables when no adequate set can be formed.

This is an combinatorial problem. It could best be solved by developing a better set of rules for the swapping variables, which would possibly keep a better record of past transactions. These are mostly heuristics, and a careful study would be needed to develop them.

7.9 Conclusion: A General Assessment

In many aspects, our OPF algorithm has shown much promise. Its performance was quite fast and accurate in the tests presented in this chapter. Also, taking into account the discussion concerning differences between the subproblem and the OPF solutions, it is felt that the subproblem could be put to good use as a fast dispatching tool. The most impressive results were those of the load-tracking step, which solved the OPF for a sequence of loads in quick succession.

More work is needed to make the program more reliable. We have advanced some ideas to explain the difficulties observed in the tests. Changes in the program we can now suggest are those discussed in the previous section. One more change is suggested: that the program solve the optimality equations in their most general form (eq. 4.4) instead of the partitioned form presently being used (eq. 4.15-4.17). That would avoid the occasional repartitioning of the independent/dependent variables, allow for pivoting to improve numerical properties, and allow to simplify present data structure problems. The present quick updating schemes used in the subproblem could easily be adapted for use here. The behaviour of this Version 2.0 should then be much improved.

CHAPTER VIII

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

8.1 Conclusions

This thesis has presented a new solution methodology for the optimal power flow problem based on continuation methods. It has produced a large collection of analytical results for various tasks of the optimal power flow problem, and a detailed numerical study for the minimum fuel costs task.

A large part of the analysis has been devoted to studying the use of the continuation method as a tool for solving the OPF subproblem. The subproblem, in turn, is inserted into the larger successive quadratic programming strategy for the nonlinear problem. The subproblems for three tasks, minimum fuel cost, minimum loss and load shedding, were analyzed in detail. New formulations were proposed for the latter two tasks, and various continuation strategies were explored for the solutions of all three tasks. Additions to the subproblem analysis, such as the inclusion of ramp constraints, post-contingency redispatching, and bus incremental costs, were also proposed, but in less detail. Some concepts new to power systems optimization were introduced and exploited; most notable are the transparent variables and the search for fold lines in the load flow manifold. This subproblem structure was originally conceived as an extension to the real power dispatching problem, and it could be used on its own as a real-reactive dispatching tool.

The analysis also proposed a new set of rules to aid convergence of the nonlinear OPF problem. The algorithm presented in the thesis ensures descent of the objective function from one iteration to the next. This particular set of numerical tactics includes step size calculations in various positions, including in a Newton-Raphson solver.

A second application of the continuation principle in the algorithm suggested that closely spaced loads be fed to the nonlinear program, to produce solution trajectories of the dispatchable variables. This was motivated by the idea that the computation of a solution trajectory, either leading to a desired load or following a forecasted load trajectory, could be very fast.

One variant of the general OPF proposed in the analysis was put to the test in a series of numerical simulations. A computer program was written for the economic dispatch task, based on the varying limits strategy to solve the subproblem. This program implements sparsity and efficient data structure techniques to increase speed and to reduce memory requirements. Also, various numerical tactics were introduced to enhance the robustness of the algorithm. Detailed results from four test systems, ranging in size from 6 to 118 buses (32 to 531 electrical variables), give a clear picture of the behaviour of the algorithm. Among the important observations, we note the following:

(1) the number of breakpoints in a subproblem solution is relatively small and independent of its size.
(2) the rules for convergence ensure descent, are robust, and usually require little computation.
(3) the algorithm becomes very fast after a few iterations, because it can avoid the continuation procedure in the subproblem, and because iterative processes are fed excellent initial guesses.
(4) execution speeds in solving for a single load, for these four test systems, increased linearly with size.
(5) the load-tracking scheme produces optimal solutions very quickly, because it profits from the information of the previous solution. Results show that the increases in load hardly effect the optimal settings of the reactive powers, voltages and passive controls.

These and other observations indicate that the algorithm shows much promise for the quick solution of OPF problems.

The program encountered a serious numerical problem in some runs of the 118 bus test, due to ill-conditioning and degeneracy in the optimality conditions. This stems from an inherent problem with the compact load flow model adopted for this study, and not the continuation process per se. The program was not equipped to handle this problem; it is felt, however, that the problem can be remedied, as it has been in other well-known optimization packages.

8.2 Recommendations for Future Research

The following recommendations cover four large areas of research. These are: (1) ways of improving the present OPF program, (2) the development of software packages for other OPF tasks, (3) the use of the latest ideas in continuation method theory, and (4) a fundamental study of the properties of the load flow equations.

8.2.1 Improvements to the Present Program

Many of these improvements were suggested in the main body of the thesis.

To improve the robustness of the subproblem, three points were made at the end of Chapter 7:

- To add a mechanism isolating the functional constraints which cause degeneracy from the other active functional constraints.

- To study a mechanism which avoids the creation of ill-suited sets of independent variables.

- To solve the optimality equations of the subproblem in the form of eq. 4.4 instead of eq. 4.14-4.17, to allow for pivoting of the variables, thereby avoiding the occasional reorganizations of the partitions of the variables.

To improve the execution speed of the real power dispatch algorithm, found in the initial, simple problem of the subproblem, it was suggested to convert from a binary search to a secant search algorithm. The improvement in overall speed of the OPF algorithm would be small, but probably little effort is required to make this change.

To improve the speed of the computation of sensitivity coefficients, a sparse vector solver should be added to the program.

With the experience gained in observing the program, some adjustments could be made, with the hope of avoiding lengthy computations. For example:

- Rules for the imposition of auxiliary bounds tried for the anti-zigzagging scheme could be improved. If variables near fold lines can be identified and their behaviour better understood, they could be removed from the standard optimization procedure. It is the oscillation of these variables in standard procedures which slows down the optimization the most.

- Convergence criteria and convergence tolerances should be studied in greater detail, to determine when it is worthwhile to stop iterating. In some of our tests the values of the objective function converged much faster than the other convergence criteria. Because of the slow convergence of those other criteria, the optimization proceeded with more iterations.

- The convergence characteristics of the algorithm as a function of the initial guess should be studied in greater detail. The impression gained from our present numerical experience is that it is disadvantageous to start the algorithm very close to the solution.

- With the knowledge that the variables other than the P's vary little in the load-tracking loop, it might be advantageous to process dependent constraints outside the subproblem, in the nonlinear loop. That could save time, because the monitored dependent constraints would likely remain feasible from one load-tracking iteration to the next. If so, no additional processing would be required.

8.2.2 The Development of Software Packages for Other OPF Tasks

Chapters 4 and 5 provide the analysis for many new applications of the continuation method in solving OPF subproblems. These include:

- The solution of the economic dispatch subproblem by the varying load strategy. This is potentially a faster solution technique than the one tested in this thesis, once it is initialized with an optimal solution for some load. The best coordination between the subproblem solution trajectories and the nonlinear solution trajectories remains to be determined. One reasonable strategy would be to use the solutions of the subproblem as the optimal dispatches as long as their mismatches with the corresponding load flow points are small. When this mismatch becomes too large, the nonlinear information would be updated and the subproblem would be restarted. Numerical problems remain to be seen, but ill-conditioning observed in the varying limits strategy, due to the insensitivity of reactive powers and voltages, should again be present in this problem.

- Bus incremental costs, which can easily be formed from the solutions of the economic dispatch subproblem when using the varying load strategy.

- The solution of the minimum loss problem, formulated as a parametric linear program. For this subproblem strategy to be effective in nonlinear optimization, it is important to detect transparent variables on their fold lines; otherwise, the linear program will send too many variables to their bounds. The preferred solution strategy is the varying load strategy.

- The solution of the load shedding problem, using the new varying load strategy. This formulation minimizes a norm of the difference between the varying unsatisfied (forecasted) demand and the feasible satisfiable (unknown) load. The solution process proposes to compute the best load and at the same time the optimal generation dispatch for that load.

- The coordination of normal dispatching tasks and the load shedding task, to create an optimal operating schedule for any demand, feasible or infeasible.

- The inclusion of dynamic ramp constraints in the economic dispatch subproblem. This would provide a look-ahead capability for the algorithm in cases where the loads increase rapidly.

- Suggestions for the study of variations in the system parameters, for post-contingency redispatching. Often, the computation of a post-contingency optimal dispatch is not helped by the knowledge of the pre-contingency situation. The continuation method could take advantage of this information. Changes handled via the bounds on the variables are easily processed. Others which involve "left-hand-side" variations are more difficult. Discrete variations of these parameters are suggested, to see if tracking the optimal solution from the pre-contingency to the post-contingency state is worthwhile.

The parameter variations of the previous point would also be useful in expansion planning.

8.2.3 A Better Use of Continuation Techniques in the Nonlinear Problem

First, we suggest that state of the art techniques in continuation methods be incorporated into implementation of the OPF. These techniques, which were developed in the last decade and referred to in the introduction, can bypass some problems of singularity in the system equations. In particular, these techniques could be useful in handling the left-hand-side perturbations.

Secondly, we suggest a systematic study of homotopy strategies in the nonlinear loop of the solution procedure. The outer load-tracking loop, as initially presented in Chapter 3, is an example where the known solution to one nonlinear problem is linked to that of the desired nonlinear problem. However, the solution to the initial, simple problem might be difficult to obtain in most cases. To simplify the solution of the initial problem, many parameters could be relaxed. For example, a load flow feasible point could be made the exact optimum for some artificial OPF problem, with bounds and loads relaxed. Then the parameters would be varied, continuously or discretely, until they reintegrate their original values, and the desired OPF problem would be solved. As in other applications of the varying limits strategy, this technique could be very advantageous if a good initial guess is furnished. A possible advantage over the solution strategy developed in this thesis is that the nonlinear solution trajectories would probably be smooth, without any of the oscillations which hamper the present strategy.

8.2.4 A Study of the Load Flow Equations

The analysis of some more fundamental properties of the load flow manifold is a difficult task, but it could provide valuable information for use in optimization. We can suggest the following topics:

- The determination of connected regions of local convexity in the load flow manifold. This could explain convergence properties, or aid in identifying good initial guesses.

- The determination of convexity properties of the load flow feasible regions, as seen from different projection spaces. In particular, properties of the fold lines, whose projections form the load flow boundaries, would be useful in setting the heuristic rules referred to earlier for setting transparent variables.

- A comparison of the numerical properties of the OPF solutions in the two main formulations, the compact formulation used in this thesis and the sparse formulation with independent state variables. These could be compared for the simplicity of their numerical structures, the conditioning of the load flow Jacobians used in the subproblems, the behaviour of transparent variables, the convergence of the nonlinear algorithm, and the effectiveness of the tactics used to aid convergence of the nonlinear algorithm. Of course, to make such a comparison, experience with both types of programs would be necessary.

- Finding all solutions to the load flow equations or to an OPF problem, for a small system. This could show the disposition and the closeness of the multiple solutions. The solution trajectories could be computed, given a load trajectory. A perturbation analysis of the eigenfunctions could be useful in determining convexity properties of the load flow manifold.

- The analysis of various models of second order information for the load flow equations. Convergence problems in the Newton-Raphson solver are due to the inadequacy of the linear model, which cannot represent curvature. Some second order information, albeit simplified, could approximate the curvature of the load flow equations. This procedure would improve the robustness of the algorithm for solutions near a feasibility boundary, or in determining the least-squares solutions of infeasible scheduled injections.

REFERENCES

Power Systems

Adielson T., "Determination of an Optimal Power Flow by Iterative Suboptimizations", Proc. of 4th PSCC Conf.,vol.2 ,paper 2.1/9,1972.

Aldrich J.F., Happ H.H., Lever J.F., "Multi-Area Dispatch", IEEE Trans. on PAS, vol.90 ,pp.2661-2670 ,1971.

Aldrich J.F., Fernandes R.A., Vicks L.W., Happ H.H., "Benifits of Voltage Scheduling in Power Systems", IEEE T-PAS,Vol. PAS-99,No.5,Sep./Oct. 1980.

Allen J.C.J., Peddle R.A., "The Effect of Dynamic Electronic Pricing on Power System Monitoring and Control",Proc. of Second Int. Conference on Power System Monitoring and Control,Durham, July 1986 ,pp.177-182.

Alsac O., Stott B., "Optimal Load Flow with Steady-State Security", IEEE Trans. on PAS, vol.93, pp.745-751, 1974.

Alsac O., Stott B., "Decoupled Algorithms in Optimal Load Flow Calculations", paper 75545-4, IEEE PES Summer Meeting, 1975.

Alvarado F.L., "Penalty Factors from Newton's Method",IEEE Trans. on PAS, vol.97 , no.6 ,pp.2031-2040 ,1978.

Aoki K., Nishikori A., "An Algorithm for Constrained Load Flow" IEEE Trans. on PAS, vol.103, no.5, pp.963-973, 1984.

Aoki K., Satoh T., "Economic Dispatch with Network Security Constraints Using Parametric Quadratic Programming", IEEE Trans. on PAS,vol.101 ,no.12, pp.4548-4556 ,1982.

Aoki K., Satoh T., "New Algorithm for Classic Economic Dispatch", IEEE Trans. on PAS, vol.103 ,no.6 ,pp.1423-1431 ,1984.

Bechert T.E., Kwatny H.G., "On the Optimal Dynamic Dispatch of Real Power", IEEE Trans. on PAS, vol.91, pp.889-898, 1972.

Biggs M.C., Laughton M.A., "Optimal Electric Power Scheduling,a Large Nonlinear Test Program Solved By Recursive Quadratic Programming", Math. Progr.,vol.13, pp.167-182 ,1970.

Billinton R., Sachdeva S.S., "Optimal Real and Reactive Power Operation in a Hydrothermal System". IEEE Trans. on PAS,vol.91 ,pp.1405-1411 ,1972.

Blanchon G., Dodu J.C., Merlin A., "Developing a New Tool for Real-Time Control in Order to Coordinate the Regulation of Reactive Power and the Voltage Schedule in Large-Scale EHV Power Systems",Symp. CIGRE-IFAC,paper 209-01, Florence ,1983.

Blanchon G., Dodu J.C., Merlin A., "New Developments of the Reactive Power Flow Optimization Model used at EDF", from Proc. 8th PSCC conf.,pp.427-434, Helsinki 1984.

Bottero M.H., Galiana F.D., Fahmideh-Vojdani A.R., "Economic Dispatch Using the Reduced Hessian",IEEE Trans. on PAS,vol.101 ,no.10 ,pp.3679-3688 ,1982.

Brownlee W.R., "Coordination of Incremental Fuel Costs and Incremental Transmission Losses by Fuction of Voltage Phase Angles", AIEE Trans., vol.73, Pt.III-a ,pp.529-541 ,June 1954.

Burchett R.C., Happ H.H., Vierath D.R., Wirgau K.A., "Advanced Optimal Power Flow Methodology", Canadian Communication and Power Conference,1980.

Burchett R.C., Happ H.H., Vierath D.R., Wirgau K.A., "Developments in Optimal Power Flow", Proc. of PICA conf., 1981.

Burchett R.C., Happ H.H., Vierath D.R., Wirgau K.A., "Developments in Optimal Power Flow", IEEE Trans. on PAS,vol.101 ,no.2 ,pp.406-414 ,1982.

Burchett R.C., Happ H.H, Wirgau K.A., "Large Scale Optimal Power Flow", IEEE Trans. on PAS,vol.101 ,no.10 ,pp.3722-3732 ,1982.

Burchett R.C., Happ H.H., Vierath D.R., "Quadratically Convergent Optimal Power Flow", IEEE Trans. on PAS,vol.103 ,no.11 ,pp.3267-3276 ,1984.

Cadogan J.B., Eisenberg L., "Sulfur Oxide Emissions Management for Electric Power Systems", IEEE Trans. on PAS,vol.96 ,pp.393-401 ,1977.

Cahn C.R., "The Determination of Incremental and Total Loss Formulas from Functions of Voltage Phase Angles", AIEE Trans.,vol.74 ,Pt.III ,pp.161-176 , April 1955.

Calderon L.R., "Short Term Coordination and Fast Recoordination of Hydrothermal Systems: A New Approach Using The Continuation Method", Ph.D. Thesis, Dept. of Electrical Eng., McGill University, Montreal 1985.

Calderon L.R., Galiana F.D., "Continuous Solution Simulation in the Short Term Hydrothermal Coordination Problem", IEEE T-PWRS vol.2, no.3, pp.737-743, 1987.

Calvert J.F., Sze T.W., "A New Approach to Loss Minimization in Electric Power System", AIEE Trans.,vol.76 ,pt.III ,pp.1439-1446 ,1958.

Caramanis M., Stremel J.P., Charny L., "Unit Size and Economics of Scale in Capacity Expansion with an Efficient, Real Number Representation of Capacity Additions", IEEE T-PAS, vol.103, no.3, pp.506-515, 1984.

Carpentier J., "Contribution à l'étude du dispatching économique", Bulletin de la Société Française des Electriciens,ser.8,vol.3, pp.431-447 ,1962.

Carpentier J., Cassapoglou C., Hensgen C., "Differential Injections, A Method for General Resolution of Problems of Economic Dispatching Without Complete Variables Using the Generalized Reduced Gradient Method", Proc. of Conf. by Hellenic Operational Res. Soc., Athans,Nov. 4-8 1968.

Carpentier J., "Results and Extensions of the Methods of Differential Injections", Proc. of 4th PSCC Conf.,vol.2 ,paper 2.1/8 ,1972.

Carpentier J., "System Security in the Differential Injections Method for Optimal Load Flows", Proc. of PSCC, 1975.

Carpentier J., Cotto G., Niederlander P., "New Concepts for Automatic Generation Control in Electric Power Systems Using Parametric Quadratic Programming", Proc. IFAC Symp. on Real Time Digital Control Applications", Guadalajara, pp.445-460, January 1983.

Carpentier J.L., Cotto G., "Modern Concepts for Security Control in Electric Power System", Proc. of CIGRE, paper 102-01, Florence, Italy, 1983.

Carpentier J.L., "Optimal Power Flows: Uses, Methods and Developments", Proc. of IFAC Conf., 1985.

Carpentier J., "CRIC, a New Active Reactive Decoupling Process in Load Flows, Optimal Power Flows and System Control", Proc. of IFAC Conf. pp.65-70, Beijing, August 1986.

Carpentier J., "Towards a Secure and Optimal Automatic Operation of Power Systems", keynote speech at PICA Conf., Montreal, 1987.

Chamorel P.A., Germond A.J., "An Efficient Constrained Power Flow Technique Based on Active - Reactive Decoupling and Linear Programming", IEEE Trans. on PAS, vol.101 , no.1 , pp.158-167 , 1982.

Chan M.L., "Integrating Load Management into Energy Management System's Normal Operations", IEEE T-PAS, vol.105, no.11, pp.152-157, 1986.

Chan S.M., Schweppe F., "A Generation Reallocation and Load Shedding Algorithm", IEEE Trans. on PAS, vol.98 , no.1 , pp.26-34 , 1979.

Chan S.M., Yip E., "A Solution of the Transmission Limited Dispatch Problem by Sparse Linear Programming", IEEE Trans. on PAS, vol.98 , no.3 , pp.1044-1053 , 1979.

Contaxis G.C., Papadias B.C., Delkis C., "Decoupled Power System Security Dispatch", IEEE Trans. on PAS, vol.102 , no.9 , pp.3049-3056 , 1983.

Contaxis G.C., Delkis C., Korres G., "Decoupled Optimal Load Flow Using Linear or Quadratic Programming", IEEE Trans. on PS,vol.PWRS-1 ,no.2 ,pp.1-7 ,1986.

Dayal G., Grigsby L.L., Hasdorf L., "Quadratic Programming for Optimal Active And Reactive Power Dispatch using Special Techniques for Reducing Storage Requirements", paper A 76 388-9,IEEE PES Summer Meeting, 1976.

Debs A.S., Benson A.R., "Security Assessment in Power Systems", Engineering Foundation Conference on Systems Engineering for Power: Status and Prospects, Henniker, N.H., pp.144-176, August 1975.

Decker G.L., Brooks A.D., "Valve Point Loading of Turbines", AIEE Trans.,vol. 77 ,Pt.III ,pp.481-486 ,1958.

Delson J.K., "Controlled Emission Dispatch", IEEE PES Winter Meeting, 1974.

Deo D., "Economic Load Dispatch of Interconnected Power Sytem by Linear Programming", J. Inst. Engr. (India),Elect. Eng. Div.,vol.53 ,pt.EL 3,pp.117-122 ,1973.

Dersin P., Levis A.H., "Feasibility Sets for Steady-State Loads in Electric Power Networks", IEEE Trans. on PAS,vol.101 ,no.1,pp.60-70 ,1982.

Desrochers M., Turgeon A., Ferland J., "Development Scheme for River Valleys", T-PWRS vol.1, no.1, pp.89-95, 1986.

Dhar R.N., Mukherjee P.K., "Reduced Gradient Method for Economic Dispatch", Proc. of IEE,vol.120 ,no.5 ,pp.608-610 ,1973.

Dhar R.N., "Computer Aided Power System Operation & Analysis", McGraw-Hill 1982.

Di Liacco T.E., "Real Time Control of Power Systems", Proc. of IEEE, vol.62. no.7, pp.884-891, 1974.

Di Liacco T.E., "An Overview of Power System Control Centres",Energy Control Center Design, IEEE Tutorial Course 77TU0010-9-PWR, pp5-16, 1977.

Dillon T.S., Morsztyn K., Tun´T., "Sensitivity Analysis of the Problem of Economic Dispatch of Thermal Power Systems", Int. J. of Control,vol.22 ,no.2, pp.229-248, 1975.

Dillon T.S., "Rescheduling,Constrained Participation Factors and Parameter Sensitivity in The Optimal Power Flow Problem", IEEE Trans. on PAS,vol.100, no.5 ,pp.2628-2634 ,1981.

Dillon T.S., "Computational Methods for Solution of the Optimal Power Flow Problem", from the fourth International Symposium on Large Engineering Systems, Calgary 1982.

Dodu J.C., Merlin A., "Improvements to the MEXICO Model for Probabilistic Planning Studies", Int. J. Power and Energy System, vol.1, no.1, pp.46-56, April 1979.

Dommel H.W., Tinney W.F., "Optimal Power Flow Solutions", IEEE Trans. on PAS, vol.87 ,pp.1866-1876 ,1968.

Dopazo J.F., Klitin O.A., Stagg G.W., Watson M., "An Optimization Technique for Real and Reactive Power Allocation", Proc. of IEEE ,vol.55 ,no.11, pp.1877-1885, 1967.

Ejebe G.C., Puntel W.R., Wollenberg B.F., "A Load Curtailment Algorithm for the Evaluation of Power Transmission System Adequacy", IEEE PES Summer Meeting, paper 77 505-1 ,1977.

Elangovan S., "A New Approach for Real Power Minimization", Proc. of IEE, vol.130,pt.c ,no.6 ,pp.295-299 ,1983.

El Abiad A.H., Jaimes F.J., "A Method for Optimal Scheduling of Power and Voltage Magnitude", IEEE Trans. on PAS,vol.88 ,pp.413-422 ,1969.

El-Hawary M.E., Feehan T., "Economic Active Reactive Dispatch - Theory and Computational Experience", Proc. Canadian Communications and Power Conf. Montreal, Oct. 1978 ,pp.386-388.

El-Hawary M.E., Christensen G.S., "Optimal Economic Operation of Electric Power Systems", Academic Press 1979.

El-Hawary M.E., Tsang D.H., "The Hydrthermal OPtimal Load Flow, A Practical Formulation and Solution Techniques Using Newton's Approach", IEEE PES Winter Meeting, paper 86 WM 062-4, 1986.

El-Kady M.A., Bell B.D., Carvalho V.F., Burchett R.C., Happ H.H., Vierath D.R., "Assessment of Real Time Optimal Voltage Control", IEEE T-PAS Vol.. PWRS-1,No.2, May 1986.

El-Shibini M.A., Dayeh M.B., "Reactive Power Optimization Using Modified Linear Programming", paper C 75 024-5, IEEE PES Winter Meeting, 1975.

Fahmideh-Vojdani A., "Analysis and Continuous Simulation of Secure-Economic Operation of Power Systems", Ph.D. Thesis, Dept. of Electrical Eng., McGill University, 1982.

Fahmideh-Vojdani A., Galiana F.D., "The Continuation Method and its Application in System Planning and Operation", paper 102-04,Proc. of CIGRE, Florence, Italy, 1983.

Farghal S.A., Tantawy M.A., Abopu Hussein M.S., Hassan S.A., Abou Elela A.A., "Fast Technique for Power System Security Assessment using Sensitivity Parameters of Linear Programming", IEEE Trans. on PAS,vol.103 ,no.5 ,pp.946-953 ,1984.

Fernandes R.A., Happ H.H., Wirgau K.A., "The Application of Optimal Power Flow for Improved System Operation", Proc. of 6th PSCC Conf.,vol.1 ,pp.465-472, Darmstadt Germany, 1978.

Fernandes R.A., Lange F., Burchett R.C., Happ H.H., Wirgau K.A., "Large Scale Reactive Power Planning", IEEE T-PAS vol.102, no.5, pp.1083-1088, 1983.

Fischetti M.A., "Electric Utilities: Poised for Deregulation", IEEE Spectrum, vol.23, no.5, pp34-40, 1986.

Fischl R., "Optimal System Expansion: A Critical Review", Engineering Foundation Conference on Systems Engineering for Power: Status and Prospects, Henniker, N.H., pp.233-260, August 1975.

Franchi L., Garzillo A., Innorta M., Marannino P., Marchese V., "Centralized Control of Real Power for Thermal Units by a Parametric Linear Programming Algorithm", Proc. of IFAC Conf. on Automatic Control in Power Genration Distribution and Protection", pp.51-60, Pretoria, Sept. 1980.

Franchi L., Innorta M., Marannino P., Sabelli C., "Evaluation of Economy and/or Security Oriented Objective Functions for Reactive Power Scheduling in Large Scale Systems", Proc. of PICA Conf.,Houston,pp.277-284 ,1983.

Galiana F.D., Banakar M.H., "Realizability Inequalities for Security Constrained Load Flow Variables", IEEE Trans. on CAS, vol.29, no.11, pp.767-771, 1982.

Galiana F.D., Fahmideh-Vojdani A., Huneault M., Juman M., "Optimal Power System Dispatch Through the Continuation Method: Variation of Functional Inequality Limits", Proc. of IEEE International Symposium on Circuits and Systems, California, May 1983.

Garver L.L., Van Horne P.R., Wirgau K.A., "Load Supplying Capability of Generation-Transmission Networks", IEEE Trans. on PAS,vol.98 ,no.3 ,pp.957-962, 1979.

Gent M.R., Lamont J.W., "Minimum Emission Dispatch", IEEE TRans. on PAS, vol.90, no.6, pp.2650-2660, 1971.

George E.E., "Intrasystem Transmission Losses", Trans. AIEE,vol.62 ,pp.153-158, Mar. 1943.

George E.E., Page H.W., Ward J.B., "Coordination of Fuel Cost and Transmission Loss by Use of the Network Analyser to Determine Plant Incremental Loss Formulae", AIEE Trans.,vol. 68 ,pt.II ,pp.1152-1163 ,1949.

Ghoneim M.Z., Saleh M.H., Askourah M.A.N., Abdel Rahman S.A., "Estimation of Optimum Load Shedding in Transmission Systems Using Mathematical Programming Techniques", paper A77 595-2, IEEE PES Summer Meeting, 1977.

Ghoudjehbaklou H., Puttgen H.B., "Optimization Topics Related to Small Power Producing Facilities Operating under Energy Spot Pricing Policies", paper 86 SM 302-4 ,presented at IEEE PES Summer Meeting, July 1986.

Glimn A.F., Kirchmayer L.K., Stagg G.W., "Analysis of Losses in Interconnected Systems", AIEE Trans.,vol.71 ,pt.3 ,pp.796-808, 1952.

Glimn A.F., Habermann jr. R., Kirchmayer L.K., Thomas R.W., "Automatic Digital Computer Applied to Generation Scheduling", AIEE Trans.,vol.73 ,pt.III-B, pp.1267-1275 ,1954.

Glimn A.F., Kirchmayer L.K., Skiles J.J., "Improved Method of Interconnecting Transmission Loss Formulas", AIEE Trans.,vol.77 ,pt.III ,pp.755-760 ,1958.

Gross G., Luini J.F., "Effective Control of Convergence of the Newton Loadflow", Proc. of PICA Conf.,New Orleans ,1975 ,pp.41-48.

Gruhl J., Schweppe F., Ruane M., "Unit Commitment Scheduling of Electric Power Systems", Engineering Foundation Conference on Systems Engineering for Power: Status and Prospects, Henniker, N.H., August 1975.

Hajdu L.P., Peschon J., Tinney W.F., Piercy D.S., "Optimum Load-Shedding Policy for Power Systems", IEEE Trans. on PAS,vol.87 ,pp.784-795 ,1968.

Hale H.W., "Power Losses in Interconnected Transmission Networks", AIEE Trans, vol.71 ,pt.III ,pp.973-998 ,1952.

Hano I., Tamura Y., Narita S., Matsomoto K., "Real Time Control of System Voltage and Reactive Power", IEEE Trans. on PAS,vol.88 ,no.10 ,pp.1544-1559, 1969.

Happ H.H., "Optimal Power Dispatch", IEEE Trans. on PAS,vol.93 ,pp.820-830 , 1974.

Happ H.H., "Optimal Power Dispatch", from Systems Engineering for Power: Status and Prospects, U.S. Energy Research and Development Administration, Henniker, New Hampshire 1975.

Happ H.H., "Optimal Power Dispatch - A Comprehensive Survey", IEEE Trans. on PAS,vol.96 ,pp.841-854 ,1977.

Happ H.H., Wirgau K.A., "Static and Dynamic VAR Compensation in System Planning", IEEE Trans. on PAS,vol.97 ,pp.1564-1578 ,1978.

Hill E.F., Stevenson jr. W.D., "A New Method of Determining Loss Coefficients", IEEE Trans. on PAS,vol.87 ,pp.1548-1553 ,1968.

Hobson E.T., "Network Constrained Reactive Power Control Using Linear Programming", IEEE Trans. on PAS,vol.99 ,no.3 ,pp.868-877 ,1980.

Hobson E., Fletcher D.L., Stadlin W.O., "Network Flow Linear Programming Techniques and Their Application to Fuel Scheduling and Contingency Analysis", IEEE Trans. on PAS,vol.103 ,no.7 ,pp.1684-1691 ,1984.

Horton J.S., Grigsby L.L., "Voltage Optimization using Combined Linear Programming and Gradient Techniques", IEEE Trans. on PAS,vol.103 ,no.7, pp.1637-1643 ,1984.

Housos E., Irisarri G.D., "Real and Reactive Power System Security Dispatch using a Variable Weights Method", IEEE Trans. on PAS,vol.102 ,no.5 ,pp.1260-1268 ,1983.

Hughes A., Jee G., Hsiang P., Shoults R.R., Chen M.S., "Optimal Reactive Power Planning", IEEE T-PAS, vol. 100, no.5, pp 2189-2196, 1981.

Huneault M., Fahmideh-Vojdani A., Juman M., Calderon R., Galiana F.D., "The continuation Method in Power System Optimization: Applications to Economy-Security Functions", IEEE Trans. on PAS,vol.104 ,no1 ,pp.114-124, 1985.

Huneault M., Galiana F.D., "A Survey and Analysis of the OPF Literature", to be presented.

Ihura Y., Gross G., "Efficient Large-Scale Hydro System Scheduling with Forced Spill Conditions", IEEE PES winter meeting, Dallas, 1984.

Innorta M., Marannino P., "Advance Dispatch Procedures for the Centralized Control of Real Power", Proc. of PICA conf.,pp.188-194, 1985.

Innorta M., Marannino P., Granelli G.P., Montagna M., Silvestri A., "Security Constrained Dynamic Dispatch of Real Power for Thermal Groups", Proc. of PICA Conf.,Montréal ,1987.

Isoda H., "On Line Load Dispatching Method Considering Load Variation Characteristics and Response Capabilities of Thermal Units", IEEE Trans. on PAS,vol.101 ,no.8 ,pp.2925-2930 ,1982.

Jarjis J., "Load Flow Feasibility Conditions in Power Networks", Ph.D.Thesis, Dept. of Electrical Eng., McGill University, Montreal 1980.

Juman J., "Economic Disparch with Transmission Limits Using the Continuation Method", Master's Thesis, Dept. of Electrical Eng., McGill University 1983.

Kaltenbach J.C., Peschon J., Gerhig E.H., "A Mathematical Technique for the Expansion of Electric Power Transmission Systems", IEEE Trans. on PAS, vol.89, no.1, pp.113-119, 1970.

Kaltenbach J.C., Hajdu L.P., "Optimal Corrective Rescheduling for Power System Security", IEEE Trans. on PAS,vol.90 ,pp.843-851 ,1971.

Kerr R.H., Kirchmayer L.K., "Theory of Economic Operation of Interconnected Areas", AIEE Trans.,vol.78 ,pt.III ,pp.647-653 ,1959.

Khan M.A., Kuppusamy, "Optimum Load Curtailment under Emergency Conditions using Constant Matrices", paper A79 113-2,IEEE PES Winter Meeting, 1979.

Kirchmayer L.K., Stagg G.W., "Analysis of Total and Incremental Losses in Transmission Systems", AIEE Trans.,vol.70 ,pt.II ,pp.1197-1205 ,1951.

Kirchmayer L.K., Stagg G.W., "Evaluation of Methods of Co-ordinating Incremental Fuel Costs and Incremental Transmission Losses", AIEE Trans., Vol.71, Part III, pp 513-520 ,1952.

Kirchmayer L.K., "Economic Operation of Power Systems", John Wiley & Sons, 1958.

Kirchmayer L.K., "Economic Control of Interconnected Systems", John Wiley & Sons, 1959.

Kishore A., Hill E.F., "Static Optimization of Reactive Power Sources by use of Sensitivity Parameters", IEEE Trans. on PAS,vol.90 ,pp.1166-1173 ,1971.

Kohli N.P., Kohli J.C., "Optimal Capacitor Allocation by 0-1 Programming", IEEE PES Summer Meeting, paper A75 476-2, San Francisco, 1975.

Kondragunta J., Walker L.N., "Optimal Long Term Acquisition of Coal for Fuel Blending", IEEE T-PAS, vol.103, no.5, pp.100-1007, 1984.

Krogh B.H., "Allocation of Generation Ramping to Reduce Transmission Line Overloads", IEEE Trans. on PAS,vol.102 ,no.7 ,pp.2110-2115 ,1983.

Krogh B., Javid S.H., Chow J.H., "Multistage Rescheduling of Generation, Load

Shedding and Short Term Transmission Capacity for Emergency State Control", IEEE Trans. on PAS,vol.102 ,no.5 ,pp.1466-1472 ,1983.

Kumai K., Ode K., "Power System Voltage Control by Using a Process Control Computer", IEEE Trans. on PAS,vol.87 ,pp.1985-1990,1968.

Lamont J.W., Gent M.R., "Environmentally Oriented Dispatching Techniques", Proc. of PICA Conf., pp.421-427 ,1973.

Lauer G.S., Bertsekas D.P., "Solution of Large-Scale Optimal Unit Commitment Problems" IEEE T-PAS, vol.101, no.1, pp.79-86, 1982.

Lee K.Y., Park Y.M., Ortiz J.L., "A Unified Approach to Optimal Real and Reactive Power Dispatch", IEEE PES Summer Meeting, paper 84 SM 572-4, Seattle 1984.

Lee K.Y., Ortiz J.L., Park Y.M., Pond L.G., "An Optimization Technique for Reactive Power Planning of Subtransmission Networks Under Normal Operation", IEEE T-PWRS, vol.1, no.2, pp.153-159, 1986.

Lee S.T.Y., Hicks K.L., Hnyilicza E., "Transmission Expansion Planning by Branch and Bound Integer Programming with Optimal Cost Capacity Curves", IEEE Trans. on PAS, vol.93, no.5, pp.1390-1400, 1974.

Lee T.H., Thorne D.H., Hill E.F., "A Transportation Method for for Economic Dispatching - Application and Comparison", IEEE Trans. on PAS,vol.99 ,no.6, pp.2373-2385 ,1980.

Lee T.H., Thorne D.H., Hill E.F., "Modified Minimum Cost Flow Dispatching Method for Power System Application", IEEE Trans. on PAS,vol.100 ,no.2, pp.737-744, 1981.

Lescoeur B., Galland J.B., "Tariffs and Load Management: The French Experience", paper 86 SM 312-3 ,presented at IEEE PES Summer Meeting, July 1986.

Levin N., Zahavi J., "Optimal Mix Algorithms with Existing Units", IEEE T-PAS, vol.103, no.5, pp.954-962, 1984.

Lipowski J.S., Charalambous C., "Solution of Optimal Load Flow Problem by Modified Recurcive Quadratic Programming Method", Proc. of IEE,vol.128 ,pt.c, no.5, pp.288-294, 1981.

Luo J.S., Hill E.F., Lee T.H., "Bus Incremental Costs and Economic Dispatch", IEEE Trans. on PS,vol.PWRS-1 ,no.1 ,pp.161-167 ,1986.

Mansour S.Y., Kelly D.H., Koval D.O., "Imapct of Reactive Power Flow on the Optimum Economic Operation of Power Systems", Proc. of Canadian Society for Electrical Engineering, pp.179-194, May 1984.

Maria G.A., Findlay J.A., "A Newton Optimal Power Flow Program for Ontario Hydro EMS", paper 86 SM 326-3 ,presented at IEEE PES Summer Meeting, July 1986.

Medicherla T.K.P., Billinton R., Sachdeva M.S., "Generation Rescheduling and Load Shedding to Alleviate Line Overloads-Analysis", IEEE Trans. on PAS, vol.98 ,no.6 ,pp.1876-1884 ,1979.

Merlin A., "On The Optimal Generation Planning in The Large Transmission System (The MAYA Model)", Proc. of 4th PSCC Conf.,vol.2 ,paper 2.1/6 ,1972.

Miller R.H., "Power System Operation", second edition, McGraw-Hill 1983.

Moskalev A.G., "Principals of the Most Economical Distribution of Active and Reactive Loads in Automatically Controlled Power Systems", Elektrichestvo, vol.12 ,pp.24-33 ,1963.

Mota-Palomino R., Quintana V.H., "Sparse Reactive Power Scheduling by a Penalty Function -Linear Programming Technique", IEEE Trans. on PWRS,vol.1 , no.3 ,pp.31-39 ,1986.

Nabona N., Ferris L.L., "Optimization of Economic Dispatch Through Quadratic and Linear Programming", Proc. of IEE,vol.120 ,no.5 ,pp.574-580 ,1973.

Narita S., Hamman M.S.A.A., "A Computational Algorithm for Real-Time Control of System Voltage and Reactive Power,Part I-Problem Formulation; Part II-Algorithm of Optimization", IEEE Trans. on PAS,vol.90 ,no.6 ,pp.2495-2508, 1971.

Nicholson H., Sterling M.J.H., "Optimum Dispatch of Active and Reactive Generation by Quadratic Programming", IEEE Trans. on PAS,vol.92 ,pp.644-654,1973.

Ohuchi A., Kaji I., "A Branch and Bound Algorithm for Start up and Shut Down Problems of Thermal Generating Units", Electr. Eng. in Japan, vol.95, no.5, pp.54-61, 1975.

Oyama T., "Theoretical Aspects of an Electricity Marginal Cost Model", paper 86 SM 304-0 ,presented at IEEE PES Summer Meeting, July 1986.

Padiyar K.R., Aggarwal R.P., Jose T.L., "Reactive Power Control using Dual Interval Programming", Proc. of Second Int. Conference on Power System Monitoring and Control,Durham, July 1986 ,pp.162-165.

Palaniswamy K.A., Misra K.B., Sharma J., "Quasi-Newton Method for Optimal Load Flow Solution under Emergency Mode of Operation", Electric Power System Research,vol.4 ,no.4 ,pp.289-296, 1981.

Palaniswamy K.A., Sharma J., Misra K.B., "Optimum Load Shedding Taking Into Account of Voltage and Frequency Characteristics of Loads", IEEE Trans. on PAS,vol.104 ,no.6 ,pp.1342-1348 ,1985.

Pang C.K., Sheble C.B., Albuyeh F., "Evaluation of Dynamic Programming Based Methods and Multiple Area Representation for Thermal Unit Commitments", IEEE T-PAS, vol.100, no.3, pp.1212-1218, 1981.

"Optimal Power Flow Software", "Optimal Power Flow by Newton's Metod", and

"Optimal Power Flow by Successive Linear Programming", PCA Corp., company advertisement, 1985.

Peach S., "Hydro-Quebec System Restoration Synthesis", presented at the IEEE PES 1984 Workshop on Real Time Monitoring and Control of Power Systems, Montreal, October 1984.

Peschon J., Piercy D.S., Tinney W.F., Tveit O.J., Cuenod M., "Optimum Control of Reactive Power Flow", IEEE Trans. on PAS,vol.87 ,no.1 ,pp.40-48 ,1968.

Peschon J., Bree D.W., Hajdu L.P., "Optimal Solutions Involving System Security", Proc.of PICA Conf.,pp.210-218 ,May 1971.

Peschon J., Dommel H., Powell W., Bree D.W., "Optimal Power Flow for Systems with Area Interchange Controls", IEEE Trans. on PAS,vol.91 ,pp.889-895 ,1972.

Podmore R., "Economic Power Dispatch with Line Security Limits", IEEE Trans. on PAS,vol.93 ,no.1 ,pp.289-295 ,1974.

Podmore R., Virmani S., Athay T., "Transient Energy Stability Analysis", in System Engineering for Power, Engineering Foundation Conference, Henniker New Hampshire, August 1977.

Ponrajah R.A., Galiana F.D., "Derivation and Applications of Optimum Bus Incremental Costs in Power System Operation and Planning", IEEE T-PAS, vol.104, no.12, pp.3416-3422, 1985.

Ponrajah R., "The Minimum Cost Optimal Power Flow Problem Solved Via the Restart Homotopy Continuation Method", Ph.D. Thesis, Dept. of Electrical Eng., McGill University, Montreal, 1987.

Ponrajah R.A., Galiana F.D., "The Minimum Cost Optimal Power Flow Problem Solved Via the Restart Homotopy Continuation Method", presented at IEEE PES Winter Meeting, New York 1988.

Quintana V.H., Lipowski J., "Optimal Load Flow via a Quadratic-Programming Approach", in Large Engineering Systems, O.P.Malik, M.C.Sing(eds.),1982.

S.Ramaiyer, K.Ramachadran, S.Hariharan, "New Technique for Optimal Reactive Power Allocation for Loss Minimization in Power Systems", Proc. of IEE, vol.130, pt.c, no.4, pp.178-182, 1983.

Reid G.F., Hasdorff L., "Economic Dispatch Using Quadratic Programming", IEEE Trans. on PAS,vol.92 ,pp.2015-2023 ,1973.

Ringlee R.J., Williams D.D., "Economic System Operation Considering Valve Throttling Losses, II- Distribution of System Loads by the Method of Dynamic Programming", AIEE Trans,vol.82 ,pp.615-620 ,1963.

Romano R., Quintana V.H., Lopez R., Valadez V., "Constrained Economic Dispatch of Multiarea Systems using Dantzig-Wolfe Decomposition Principle", IEEE Trans. on PAS, vol.100 ,no.4 ,pp.2127-2137 ,1981.

Ross D.W., Kim S., "Dynamic Economic Dispatch of Generation", IEEE Trans. on PAS,vol.99 ,no.12 ,pp.2060-2068 ,1980.

Rutz W.L., Mirchandani P.B., Wicks F.E., Becker M., "A Reliability Treatment Applied to the Analysis of Optimal Generation Plant Size", IEEE T-PAS vol.102, no.8, pp.2569-2577, 1983.

Sachdev M.S., Ibrahim S.A., "A Simulation Technique for Studying Real and Reactive Power Flow Patterns", IEEE Trans. on PAS,vol.94 ,no.6 ,pp.2092-2100, 1975. Correction 76 vol.1 ,p.374.

Sasson A.M., "Combined Use Of The Powell and Fletcher-Powell Nonlinear Programming Methods for Optimal Load Flows", IEEE Trans. on PAS,vol.88 , pp.1530-1537 ,1969.

Sasson A.M., "Nonlinear Programming Solutions for The Load Flow Minimum Loss and Economic Dispatch Problems", IEEE Trans. on PAS,vol.88 ,pp.399-409 ,1969.

Sasson A.M., Aboytes F., Cardenas R., Gomez F., Viloria F., "A Comparison of Power Systems Static Optimization Techniques", Proc.of PICA Conf.,pp.329-337, May 1971.

Sasson A.M., Viloria F., Aboytes F., "Optimal Load Flow Solution Using the Hessian Matrix", Proc.of PICA Conf., pp.203-209 ,May 1971.

Sasson A.M., Viloria F., Aboytes F., "Optimal Load Flow Solution Using The Hessian Matrix", IEEE Trans. on PAS,vol.92 ,no.1 ,pp.31-41 ,1973.

Sasson A.M., Merril H.M., "Some Applications of Optimization Techniques to Power System Problems", Proc. of IEEE,vol.62 ,no.7 ,pp.959-972 ,1974.

Savulescu S.C., "Qualitative Indices for the System Voltage and Reactive Power Control", IEEE Trans. on PAS,vol.95 ,pp.1413-1421 ,1976.

Scheidt J.L., "A Survey of Power System Control Centres Justifications", IEEE T-PAS, vol.98, no.1, pp.135-140, 1979.

Seelke jr. J.L., "Assessing Financial Benifits (Avoided Revenue Requirements) due to Load Management Program", IEEE T-PAS vol.101, no.10, pp.3892-3901, 1982.

Shen C.M., Laughton M.A., "Determination of Optimum Power-System Operating Conditions Under Constraints", Proc. of IEE,Vol.116 ,pp.225-239 ,1969.

Shen C.M., Laughton M.A., "Power System Load Scheduling with Security Constraints using Dual Linear Programming", Proc. of IEE,vol.117 ,no.11 , pp.2117-2127 ,1970.

Shoults R.R., Chen M.-S., "Reactive Power Control by Least Squares Minimization", IEEE Trans. on PAS,vol.95 ,no.1 ,pp.325-334 ,1976.

Shoults R.R., "A Simplified Economic Dispatch Algorithm Using Decoupled Networks", IEEE PES Summer Meeting, paper A77 738-8, Mexico City 1977.

Snyder jr. W.L., Powell jr. H.D., Rayburn J.C., "Dynamic Programming Approach to Unit Commitment", IEEE T-PWRS vol.2., no.2, pp.339-350, 1987.

Somuah C.B., Schweppe F.C., "On Line Computer Control of a Power System During an Emergency", IEEE Trans. on PAS,vol.100 ,no.5 ,pp.2619-2627 ,1981.

Squires R.B., "Economic Dispatch of Generation Directly from Power System Voltages and Admittances", AIEE Trans. vol.79 ,pt.III ,pp.1235-1244,1961.

Steinberg M.J., Smith T.S., "Economic Loading of Power Plants and Electric Systems", McGraw-Hill 1943.

Stott B., Marinho J.L., "Linear Programming for Power System Network Security Applications", IEEE Trans. on PAS,vol.98 ,no.3 ,pp.837-848 ,1979.

Stott B., Alsac O., Marinho J.L., "The Optimal Power Flow Problem", from Electric Power Problems:The Mathematical Challenge,A.M.Erisman et.al. (editors),pp.327-351,SIAM 1980.

Stott B., Alsac O., "Experience with Successive Linear Programming for Optimal Rescheduling of Active and Reactive Power", paper 104-01,Proc. of CIGRE, Florence,Italy,1983.

Stott B., Alsac O., Monticelli A.J., "Security Analysis and Optimization", Proc. of IEEE, pp.1623-1644, December 1987.

Subramanian D.K., "Optimum Load Shedding Through Programming Techniques", IEEE Trans. on PAS,vol.90 ,no.1 ,pp.89-95 ,1971.

Sullivan R.L., "Power System Planning", McGraw-Hill 1977.

Sun D.I., Ashley B., Brewer B., Hughes A., Tinney W.F., "Optimal Power Flow by Newton Approach", IEEE Trans. on PAS,vol.103 ,no.10 ,pp.2864-2880 ,1984.

Sze T.W., Garnett J.R., Calvert J.F., "Some Applications of a New Approach to Loss Minimaztion in Electrical Utility Systems", AIEE Trans.,vol.77 ,pt.III, pp.1577-1585 ,1959.

Talukdar S.N., Wu F.F., "Computer Aided Dispatch for Electric Power Systems", Proceedings of the IEEE,Vol. 69,No.10,October 1981.

Talukdar S.N., Giras T.C., Kalyan V.K., "Decomposition for Optimal Power Flows", IEEE Trans. on PAS,vol.102 ,no.12 ,pp.3877-3884 ,1983.

Tudor J.R., Lewis W.A., "Transmission Losses and Economy Loading by Use of The Admittance Constants", IEEE Trans. on PAS,vol.82 ,pp.676-683 ,1963.

Turgeon A., "Optimal Short-Term Hydro Scheduling from the Principle of Progressive Optimality", Water Resources Research, vol.17, pp.481-486, June 1981.

Van Meeteren H.P., Frame J.G., Fudeh H.R., "Optimal Power Flow in Energy Management Systems", Proc. of Second Int. Conference on Power System Monitoring and Control,Durham, July 1986 ,pp.296-300.

Van Ness J.E., "A Note on Incremental Loss Computation", AIEE Trans.,vol.81 , Pt.III ,pp.735-739 ,Feb. 1963.

Van C.Warrington A.R., "Protective Relays - Their Theory and Practice", Vol.1 1968, Vol.2 1974, Chapman and Hall.

Vemuri S., Ranjit Kumar A.B., Hackett D.F., Eisenhauer J., Lugtu R., "Fuel Resource Scheduling, Part I - Overview of an Energy Management Problem", IEEE PES Winter Meeting, 1984.

Waight J.G., Albuyeh F., Bose A., "Scheduling of Generation and Reserve Margin Using Dynamic and Linear Programming", IEEE T-PAS, vol.100, no.5, pp.2226-2230, 1981.

Waight J.G., Bose A., Sheble G.B., "Generation Dispatch with Reserve Margin Constraints using Linear Programming", IEEE Trans. on PAS,vol.100 ,no.1 , pp.252-258 ,1981.

Wells D.W., "Method for Economic Secure Loading of a Power System", Proc. of IEE,vol.115 ,no.8 ,pp.1190-1194 ,1968.

Wenyuan Li , "An On-Line Power Dispatch Method with Security", Electric Power System Research,vol.9 ,no.2 ,pp.173-182, 1985.

Wirgau K.A., "Voltage Scheduling in System Operations", NAPSIC - North Central Regional Meeting, Minneapolis 1979.

Wollenberg B.F., Stadlin W.O., "A Real Time Optimum for Security Dispatch", IEEE Trans. on PAS,vol.93 ,pp.1640-1649 ,1974.

Wood A.J., Wollenberg B.F., "Power Generation Operation & Control",John Wiley & Sons 1984.

Wu F.F., Gross G., Luini J.F., Lock P.M., "A Two Stage Approach to Solving Large Scale Optimal Power Flows", Proc. of PICA conf., Cleveland 1979.

Yamayee Z.A., "Maintenance Scheduling: Description, Literature Survey, and Interface with Overall Operations Scheduling", IEEE T-PAS, vol.101, no.8, pp.2770-2779, 1982.

J.Zaborsky, G.Huang, S.Y.Lin, "Control of Reactive Power and Voltage in Emergencies", Automatica, vol.21 ,no.3 ,pp.237-246 ,1985.

Zhang G.R., Brameller A., "On-Line Security Constrained Economic Dispatch and Reactive Power Control Using Linear Programming", from proc. 8th PSCC conf.,pp.396-400, Helsinki 1984.

Zhang G., "On-Line Network Constrained Reactive Power Control Using an Incremental Reactive Current Model", Proc. of Second Int. Conference on Power System Monitoring and Control,Durham, July 1986 ,pp.156-161.

Zhang G., "Security Constrained Economic Dispatch and Reactive Power Control using Successive Linear Programming", Proc. of IFAC conf. ,pp. 524-529 ,1986.

Zhuang F., Galiana F.D., "Towards a More Rigorous and Practical Unit Commitment by Lagrangian Relaxation, Proc. of the PICA Conf., pp.385-392, May 1987.

Optimization

Abadie J., Carpentier J., "Generalization of the Wolfe Reduced Gradient Method to the Case of Nonlinear Constraints", in Optimization, R.Fletcher (ed.), Academic Press 1969, pp.37-47.

Avriel M., "Nonlinear Programming,Analysis and Methods", Prentice-Hall 1976.

Bartholomew-Biggs M.C., "Recursive Quadratic Programming Methods for Nonlinear Constraints", in Nonlinear Optimization 1981, M.J.D.Powell (ed.), pp.213-222, Academic Press 1982.

Bazaraa M.S., Shetty C.M., "Nonlinear Programming: Theory and Algorithms", John Wiley & Sons 1979.

Beale E.M.L., "On Quadratic Programming", Naval Res. Logistics Quarterly, vol. 6, pp.227-243, 1959.

Ben Israel A., Ben Tal A., Zlobec S., "Optimality in Nonlinear Programming, A Feasible Directions Approach", John Wiley & Sons 1981.

Biggs M.C., "Constrained Minimization Using Recursive Equality Quadratic Programming", in Numerical Methods for Nonlinear Optimization, F.A. Lootsma (ed.), Academic Press 1972.

Biggs M.C., "Constrained Minimization Using Recurcive Quadratic Programming: Some Alternative Subproblem Formulations", in Towards Global Optimization, L.C.W. Dixon and G.P.Szego (eds.), North-Holland 1975.

Boot J.G.C., "Quadratic Programming", North-Holland 1964.

Chvatal V., "Linear Programming", Freeman 1983.

Dantzig G.B., "Linear Programming and Extensions", Princeton University Press 1963.

Dembo R.S., Tulowitzki U., "Sequential Truncated Quadratic Programming Methods", in Numerical Optimization 1984, P.T.Boggs, R.H.Byrd, R.B.Schnabel (eds.), pp.83-101 ,SIAM 1985.

Fiacco A.V., McCormack G.P., "Nonlinear Programming: Sequential Unconstrained Minimization Techniques", John Wiley & Sons 1968.

Fiacco A.V. (ed.), "Mathematical Programming With Data Perturbations I", Marcel Dekker 1982.

Fiacco A.V. (ed.), "Sensitivity, Stability and Parametric Analysis", North-Holland 1984.

Fletcher R., "An Exact Penalty Function For Nonlinear Programming With Inequalities", Math. Progr., vol.5, pp.129-150, 1973.

Fletcher R., "An Ideal Penalty Function for Constrained Optimization", in Nonlinear Programming 2, O.L. Mangasarian, R.R. Meyer, S.M. Robinson (eds.), Academic Press 1975.

Fletcher R. "Practical Methods of Optimization", vol.2, John Wiley & Sons, 1981.

Fletcher R., "Methods for Nonlinear constraints",in Nonlinear Optimization 1981, M.J.D.Powell (ed.), pp.185-212 , Academic Press 1982.

Gal T., "Linear Parametric Programming-A Brief Survey", in Math. Programming Study 21,Sensitivity,Stability,and Parametric Analysis, A.V.Fiacco (ed.), pp.43-68 ,North-Holland 1984.

Gass S.I., Saaty T.L., "The Computational Algorithm for the Parametric Objective Function", Naval Res. Logistics Quarterly, vol.1, pp.39-45, 1955.

Gill P.E., Murray W., Wright M.H., "Practical Optimization", Academic Press 1981.

Griffith R.E., Stewart R.A., "A Nonlinear Programming Technique for the Optimization of Continuous Processing Systems", Management Science, vol.7, pp.379-392, 1961.

Han S.P., "A Globally Convergent Method for Nonlinear Programming", J. of Optimization Theory and Applications,vol.22 ,no.3 ,pp.297-309 ,1977.

Huard P., "Convergence of the Reduced Gradient Method" in Nonlinear Programming 2, O.L. Mangasarian, R.R. Meyer, S.M.Robinson (eds.), Academic Press 1975, pp.29-54.

Kuhn H.W., Tucker A.W., "Nonlinear Programming", in Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, J.Neyman (ed.), Univ. of California Press 1951.

Lasdon L.S., Waren A.D., "Generalized Reduced Gradient Software for Linearly and Nonlinearly Constrained Problems", in Design and Implementation of Optimization Software, H.Greenberg (ed.), Sijthoff and Noordhoff 1978.

Lasdon L.S., "Reduced Gradient Methods", in Nonlinear Optimization 1981, M.J.D.Powell (ed.), pp.243-250 , Academic Press 1982.

Lasdon L.S., "Nonlinear Programming Algorithms - Applications,Software, and Comparisons",in Numerical Optimization 1984, P.T.Boggs, R.H.Byrd, R.B.Schnabel (eds.),pp.41-70 ,SIAM 1985.

Lemaréchal C., Mifflin (eds.), "Nonsmooth Optimization", Pergammin Press 1979.

Luenberger D.G., "Introduction to Linear and Nonlinear Programming", first edition, Addison-Wesley 1973.

Luenberger D.G., "Introduction to Linear and Nonlinear Programming", second edition, Addison-Wesley 1984.

Mangasarian O.L., "Nonlinear Programming", McGraw-Hill 1969.

Meyer R.R., "Sufficient Conditions for the Convergence of Monotonic Mathematical Programming Algorithms", J. Computer and Systems Sciences", vol.12, pp.108-121, 1976.

Murtaugh B.A., Saunders M.A., "Large-Scale Linearly Constrained Optimization", Math. Prog. 14 ,pp.41-72 ,1978.

Murtaugh B.A., Saunders M.A., "MINOS/AUGMENTED User's Guide", Tecnical report SOL 80-14, Stanford University, June 1980.

Murtaugh B.A., Saunders M.A., "A Projected Lagrangian Algorithm and its Implementation for Sparse Nonlinear Constraints", Math. Progr. Study vol.16, pp.84-117, 1982.

Murtaugh B.A., Saunders M.A., "MINOS 5.0 User's Guide", Tecnical report SOL 83-20, Stanford University, December 1983.

Murty K.G., "Linear Programming", John Wiley & Sons 1983.

Nash S.G., "Solving Nonlinear Programming Problems using Truncated-Newton Techniques", in Numerical Optimization 1984, P.T.Boggs, R.H.Byrd, R.B.Schnabel (eds.),pp.119-136 ,SIAM 1985.

Palacios-Gomez F., Lasdon L., Engquist M., "Nonlinear optimization by Successive Linear Programming", Management Sci., vol.28, no.10, pp.1106-1120, 1982.

Pierre D.A., Lowe M.J., "Mathematical Programming via Augmented Lagrangians",Addison-Wesley 1975.

Polak E., "Computational Methods in Optimization", Academic Press 1971.

Powell M.J.D., "Algorithms for Nonlinear Constraints That Use Lagrangian Functions",Math. Progr. 14 ,pp.224-248 ,1978.

Rosen J.B. "The Gradient Projection Method for Nonlinear Programming, Part II - Nonlinear Constraints", SIAM J. Appl. Math. vol.9 pp.514-532, 1961.

Scales L.E., "Introduction to Nonlinear Optimization,MacMillan 1985.

Slater M., Lagrange Multipliers Revisited: A Contribution to Nonlinear Programming", Cowles Commission Discussion Paper, Math.403, Nov. 1950.

Sorenson D.C., "Trust Region Methods for Unconstrained Optimization", in Nonlinear Optimization 1981 ,M.J.D.Powell (ed.), pp.29-38 , Academic Press 1982.

Van de Panne C., "Methods for Linear and Quadratic Programming", North Holland Publ. Co. 1975.

Wolfe P., "The Simplex Method for Quadratic Programming", Econometrica, vol.27, pp.382-398, 1959.

Wolfe P., "Methods of Nonlinear Programming", in Nonlinear Programming, J. Abadie (ed.), North-Holland 1967.

Wolfe P., "Convergence Conditions for Ascent Methods", SIAM Review,vol.11 no.2 ,pp.226-236 ,1969.

Zangwill W.I., "Nonlinear Programming: A Unified Approach", Prentice-Hall 1969.

Zlobec S., "course notes for Optimization I, McGill University 1984.

Zoutendijk G., "Mathematical Programming Methods", North-Holland 1960.

Continuation Methods

Allgower E., Georg K., "Simplicial and Continuation Methods for Approximating Fixed Points, SIAM Review, vol.22, pp.28-85, 1980.

Bank B., Guddat J., Klatte D., Kummer B., Tammer K., "Nonlinear Parametric Optimization", Acakemie-Verlag, Berlin 1982.

Chow S.N., Malet-Paret J., Yorke Y.A., "Finding Zeroes of Maps: Homotopy Methods that are Constructive with Probability One", Math. Comp. vol.32, pp.887-899, 1978.

Davidenko D.F., "On a New Method of Numerically Integrating a System of Nonlinear Equations", Doklady Academie Nauk, SSSR 88, pp.601-604, 1953.

Deist F., Sefor L., "On Newton's Method and Nonlinear Equationsby Parameter Variation", Computer J., vol.10, pp.78-82, 1967.

Eaves B.C., "Homotopies for the Computation of Fixed Points", Math. Progr. vol.3, pp.1-22, 1972.

Eaves B.C., "A Short Course in Solving Equations With PL Homotopies", SIAM-AMS Proceedings. Vol.9, pp.73-143, 1976.

Eaves C.B. (ed.), "Homotopy Methods and Global Convergence", Plenum 1983.

Fiacco A.V. (ed.), Mathematical Programming with Data Perturbations I, Marcel Dekker, 1982.

Fiacco A.V. (ed.), Sensitivity, Stability and Parametric Analysis, Math. Progr. Study 21, North holland, 1984.

Ficken F., "The Continuation Method for Functional Equations", Communications on Pure and Applied Math., vol.4, pp.435-456, 1951.

Fink J.P., Rheinboldt W.C., "Folds on the Solution Manifold of a Parameterized Equation", SIAM J. Numer. Anal., vol.23, no.4, pp.693-706, 1986.

Freudenstein F., Roth B., "Numerical Solution of Systems of Nonlinear Equations", J. Ass. Comp. Machinery, vol.10, pp.550-556, 1963.

Friedrichs K.O., "Lectures on Functional Analysis", Inst. for Math. and Mech., New York University, Chapter VII, 1950.

Garcia C.B., Zangwill W.I., "Finding all Solutions To Polynomial Systems and Other Systems of Equations", MAth. Progr. vol.16. pp.159-176, 1979.

Garcia C.B., Zangwill W.I., "Pathways to Solutions, Fixed Points, and Equilibria", Prentice Hall 1981.

Gfrerer H., Guddat J., Wacker H-J. , "A Globally Convergent Algorithm Based on Imbedding and Parametric Optimization", Computing, vol.30, pp.225-252, 1983.

Guddat J., Wacker H., Zulehner W., "On Imbedding and Parametric Optimization-A Concept of a Globally Convergent Algorithm for Nonlinear Optimization Problems", in Math. Programming Study 21,Sensitivity,Stability,and Parametric Analysis, A.V.Fiacco (ed.), pp.79-96 ,North-Holland 1984.

Hackl J., "Solution of Optimization Problems with Nonlinear Restrictions via Continuation Methods", in Continuation Methods, H.-J. Wacker (ed.), Academic Press 1978.

Houthakker H.S., "The Capacity Method of Quadratic Programming", Econometrica, vol.28 ,no.1 ,pp.62-87 ,1960.

Hu S.-T., "Homotopy Theory", Academic Press, 1959.

Keller H., "Global Homotopies and Newton Methods", in Recent Advances in Numerical Analysis, C.van de Boor, G.H.Golub, (eds.), Academic Press 1978.

Kubicek M., "Dependence of Solutions of Nonlinear Systems on a Parameter", ACM Trans. Math. Software, vol.2, pp.98-107, 1976.

Lahaye E., "Une méthode de résolution d'une catégorie d'équations transcendantes", C.R. Acad. Sci., Paris, no.198, pp.1840-1842, 1934.

Lahaye E., "La résolution des systèmes d'équations transcendantes", Acad. roy. Belge, Bull. Cl. Sci., vol.5, pp.805-822, 1948.

Meyer G., "On Solving Nonlinear Equationswith a One-Parameter Operator Imbedding", SIAM J. on Num. Anal., vol.5, pp.739-752, 1968.

Morgan A., "Solving Polynomial Systems Using Continuation for Engineering and Scientific Problems", Prentice Hall 1987.

Rheinboldt W.C., Burkardt J.V., "A Locally Parameterized Continuation Process", ACM Trans. Math. Software, vol.9, no.2, pp.215-235, 1983.

Rheinboldt W.C., Burkardt J.V., "Algorithm 596: A Program for a Locally Parameterized Continuation Process", ACM Trans. Math. Software, vol.9, pp.236-241, 1983.

Saigal R., "On the Convergence Rate of Algorithms for Solving Equations That Are Based on Methods of Complementary Pivoting", Math. Operations Res., vol.2, pp.108-124, 1977.

Saigal R., "An Efficient Procedure for Traversing Large Pieces in Fixed Point Algorithms", in Homotopy Methods and GLobal Convergence, B.C.Eaves, F.J.Gould, H.-O.Peitgen, M.J.Todd (eds), Plenum 1983, pp.239-248.

Saigal R., Todd M.J., "Efficient Acceleration Techniques for Fixed Point Algorithms", SIAM J. Numer. Anal., vol.15, pp.997-1007, 1978.

Scarf H., "The Approximation of Fixed Points of a Continuous Mapping", SIAM J. Appl. Math. vol.15, pp.1328-1343, 1967.

Schauder J., uber lineare elliptische Differentialgleichungen 2ter Ordnung", Math Z., vol.38, pp.257-282, 1934.

Watson L.T., Fenner D., "Algorithm 555, Chow-Yorke Algorithm for Fixed Points or Zeroes of C^2 Maps[C5]", ACM Trans. Math. Software, vol.6, pp.252-259, 1980.

Watson L.T., Holzer S.M., Hansen M.C., "Tracking Nonlinear Equilibrium Paths by a Homotopy Method", Nonlinear Anal., Theory, Methods & Appl., vol.7, no.11, pp.1271-1282, 1983.

Pure and Applied Mathematics

Chillingsworth D.R.J., "Differential Topology With a View to Applications", Pitman 1976.

Milnor J.W., "Topology from a Differentiable Viewpoint", University Press of Virginia 1965.

Rockafellar R.T., "Convex Analysis", ch.23-26, Princeton Un. Press 1970.

Spivak M., "Calculus on Manifolds", Benjamin/Cummings 1965.

Computational Techniques

Dahlquist G., Bjorck A., "Numerical Methods", Prentice-Hall 1974.

Dennis jr. J.E., Schnabel R.B., "Numerical Methods for Unconstrained Optimization and Nonlinear Equations", Prentice-Hall 1983.

Dongarra J.J., Moler C.B., Bunch J.R., Stewart G.W., "LINPACK User's Guide", SIAM 1979.

Duff I.S., "A Survey of Sparse Matrix Research", Proc. of IEEE, vol.65, no.4, pp.500-535, 1977.

Duff I.S., "Direct Methods for Solving Sparse Systems of Linear Equations", SIAM J. Sci. Stat. Comput., vol.5, no.3, pp.605-619, 1984.

Evans D.J., "Iterative Methods for Sparse Matrices", in Sparsity and its Applications", D.J. Evans (ed.), Cambridge University Press 1985.

George A., Liu J.W., "Computer Solution of Large Sparse Positve Definite Systems", Prentice-Hall 1981.

George A., Ng E., "Orthogonal Reduction of Sparse Matrices to Upper Triangular Form Using Householder Transformations", SIAM J. Sci. Stat. Comp., vol.7, no.2, pp.460-477, 1986.

Golub G.H., Van Loan C.F., "Matrix Computations", Johns Hopkins Un. Press 1983.

Gomez A., Franquelo L.G., "An Efficient Ordering Algorithm to Improve Sparse Vector Methods", IEEE PES Winter Meeting, paper 88 WM 165-3, 1988.

Hellerman E., Rarick D., "Reinversion with the Preassigned Pivot Procedure", Math. Progr. vol.1, pp.195-216, 1971.

Hopper M.J., "Harwell Subroutine Library, A Catalogue of Subroutines (1973) Supplement no.2 ", AERE Harwell, August 1977.

Markowitz H.M., "The Elimination Form of the Inverse and its Application to Linear Programming", Management Science vol.3, pp.255-269, 1957.

Paige C.C., Saunders M.A., "LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares", ACM Trans. Math. Software, vol.8, no.1, pp43-71, 1982.

Pissanetsky S., "Sparse Matrix Technology", Academic Press 1984.

Stewart G.W., "Introduction to Matrix Computations", Academic Press 1973.

Tinney W.F., Walker J.W., "Direct Solutions of Sparse Network Factorizations by Optimally Ordered Triangular Factorization", Proc. of IEEE, vol.55, no. 11, pp.1801-1809, 1967.

Tinney W.F., Brandwajn V., Chan S.M., "Sparse Vector Methods", IEEE Trans. on PAS, vol.104, no.2, pp.295-301, 1985.

Computing and Data Structures

Aho A.V., Hopcroft J.E., Ullman J.E., "Data Structures and Algorithms", Addison-Wesley 1985.

Dongarra J.J., "Performance of Various Computers Using Standard Linear Equations Software in a Fortran Environment", Simulation vol.49, no.2, pp.51-62, 1987.

AN EXTENDED BIBLIOGRAPHY OF OPF AND DISPATCHING PUBLICATIONS

This is the list of references used to build the literature survey of Chapter 2, and classified in Appendix 2.3.

1930

Estrada H., "Economic Load Allocation", Electrical World, Oct.11 1930.

1931

Hahn G.R., "Load Division by the Increment Method", Power, vol.73, pp.910-911, 1931.

1933

Steinberg M.J., Smith T.H., "Incremental Loading of Generating Stations", Electr. Eng., vol.52 ,pp.674-678 ,Oct. 1933.

1934

Steinberg M.J., Smith T.S., "The Theory of Incremental Rates and Their Practical Application of Load Division", Electr. Eng., vol.53, pt.I, pp.432-435, March 1934; pt.II, pp.571-584, April 1934.

1943

George E.E., "Intrasystem Transmission Losses", Trans. AIEE, vol.62, pp.153-158, Mar. 1943.

Steinberg M.J., Smith T.S., "Economic Loading of Power Plants and Electric Systems",McGraw-Hill 1943.

1949

George E.E., Page H.W., Ward J.B., "Coordination of Fuel Cost and Transmission Loss by Use of the Network Analyser to Determine Plant Incremental Loss Formulae", AIEE Trans., vol. 68, pt.II, pp.1152-116, 1949.

1950

Ward J.B., Eaton J.R., Hale H.W., "Total and Incremental Losses in Power Transmission Networks", Electr. Eng., vol.69, p.451, May 1950.

Ward J.B., Eaton J.R., Hale H.W., "Total and Incremental Losses in Power Transmission Networks", AIEE Trans., vol.69, pt.I, pp.626-632, 1950.

1951

Kirchmayer L.K., Stagg G.W., "Analysis of Total and Incremental Losses in Transmission Systems", AIEE Trans., vol.70, pt.II, pp.1197-1205, 1951.

1952

Glimn A.F., Kirchmayer L.K., Stagg G.W., "Analysis of Losses in Interconnected Systems", AIEE Trans., vol.71, pt.3, pp.796-808, 1952.

Hale H.W., "Power Losses in Interconnected Transmission Networks", AIEE Trans, vol.71, pt.III, pp.973-998, 1952.

Kirchmayer L.K., Stagg G.W., "Evaluation of Methods of Co-ordinating Incremental Fuel Costs and Incremental Transmission Losses", AIEE Trans., Vol.71, Part lII, pp.513-520, 1952.

1954

Brownlee W.R., "Coordination of Incremental Fuel Costs and Incremental Transmission Losses by Fuction of Voltage Phase Angles", AIEE Trans., vol.73, Pt.III-a, pp.529-541, June 1954.

Glimn A.F., Habermann jr. R., Kirchmayer L.K., Thomas R.W., "Automatic Digital Computer Applied to Generation Scheduling", AIEE Trans., vol.73, pt.III-B, pp.1267-1275, 1954.

Harder E.L., "Economic Load Dispatching",Westinghouse Engr.,vol.14 ,no.6, pp.194-200, 1954.

Travers R.H., Harker D.C., Long R.W., Harder E.L., "Loss evaluation - Part III; Economic Dispatch Studies of Steam-Electric Generating Systems", AIEE Trans., vol.73, Pt.III-b, pp.1091-1104, 1954.

1955

Cahn C.R., "The Determination of Incremental and Total Loss Formulas from Functions of Voltage Phase Angles", AIEE Trans., vol.74, Pt.III, pp.161-176, April 1955.

Early E.D., Smith G.L., Watson R.E., "A General Transmission Loss Equation", AIEE Trans., vol.74, Pt.III, pp.510-520, June 1955.

1956

Shipley R.B., Hochdorf M., "Exact Economic Dispatching - Digital Computer Solution", AIEE Trans., Part III, Vol.75, pp.1147-1153, 1956.

1958

Calvert J.F., Sze T.W., "A New Approach to Loss Minimization in Electric Power System", AIEE Trans., vol.76, pt.III, pp.1439-1446, 1958.

Decker G.L., Brooks A.D., "Valve Point Loading of Turbines", AIEE Trans., vol.77, Pt.III, pp.481-486, 1958.

Glimn A.F., Kirchmayer ,L.K. Skiles J.J., "Improved Method of Interconnecting Transmission Loss Formulas", AIEE Trans., vol.77, pt.III, pp.755-760, 1958.

Kirchmayer L.K., "Economic Operation of Power Systems", John Wiley & Sons, 1958.

Lubisich P.C., "Penalty Factors from Power-System Equations", AIEE Trans, vol.77, pt.III, pp.494-501, 1958.

1959

Fukao T., Yamazak T., Kimura S., "An Application of Dynamic Programming to Economic Operation Problem of a Power System", Electric Technical Journal of Japan, vol.5, no.2, pp.64-68, 1959.

George E.E., "A New Method of Making Transmission Loss Formulas Directly from Digital Power Flow Studies", AIEE Trans, vol.78, pt.III-B, pp.1567-1572, 1959.

Kerr R.H., Kirchmayer L.K., "Theory of Economic Operation of Interconnected Areas", AIEE Trans., vol.78, pt.III, pp.647-653, 1959.

Kirchmayer L.K., "Economic Control of Interconnected Systems", John Wiley & Sons, 1959.

Miller A.R., Koen jr H.R., Deliyannides J.S., "The Use of Power Transfer Equations to Derive Economic Co-ordination relationships Expressed as Functions of Voltage Phase Angles",AIEE Trans.,vol.78 ,pt.III ,pp.747-754, 1959.

Sze T.W., Garnett J.R., Calvert J.F., "Some Applications of a New Approach to Loss Minimaztion in Electrical Utility Systems", AIEE Trans., vol.77, pt.III, pp.1577-1585, 1959.

Watson R.E., Stadlin W.O., "The Calculation of Incremental Transmission Losses and The General Transmission Loss Equation", AIEE Trans., vol.78, pt.III-A, pp.12-18, 1959.

1960

Fisher M.J., "Power System Load Allocation Using a New Equation for Transmission Losses", AIEE Trans PAS, pp.507-513, 1960.

Schmidt W.S., "Economic Combination of Units for Weekend Loads", Electric Light and Power, pp.61-65, April 1960.

1961

Hayward A.P., "Economic Scheduling of Generation by Valve Points", AIEE Trans., vol.80, pt.III, pp.963-965, 1961.

Squires R.B., "Economic Dispatch of Generation Directly from Power System Voltages and Admittances", AIEE Trans., vol.79, pt.III, pp.1235-1244, 1961.

1962

Blodgett D.G., Hissey T.W., Falk A.K., Schultz W.B., "Application of an On-Line Digital Computer for Dispatch and Control of the Detroit Edison System", AIEE Winter General Meeting, paper CP-2-247, 1962.

Carpentier J., "Contribution à l'étude du dispatching économique", Bulletin de la Société Française des Electriciens, ser.8, vol.3, pp.431-447, 1962.

Light F.H., Gille J.A., "Economic Operation at Valve Cracking Points", AIEE Winter General Meeting, New York 1962.

1963

Carpentier J., Siroux J., "L'optimisation de la production à l'Electricité de France", Bull. de la Société Française des Electriciens, ser.8, vol.3, March 1963.

Happ H.H., Ille W.B., Reisinger R.H., "Economic System Operation Considering

Valve-Loop Throttling Losses, I- Method of Computing Valve-Loop Heat Rates on Multi-Valve Turbines", AIEE Trans., vol.82, pt.III, pp.609-615, April 1963.

Moskalev A.G., "Principals of the Most Economical Distribution of Active and Reactive Loads in Automatically Controlled Power Systems", Elektrichestvo, vol.12, pp.24-33, 1963.

Ringlee R.J., Williams D.D., "Economic System Operation Considering Valve Throttling Losses, II - Distribution of System Loads by the Method of Dynamic Programming", AIEE Trans, vol.82, pp.615-620, 1963.

Smith H.M., Tong S.Y., "Minimizing Power Transmission Losses by Reactive -Volt-Ampere Control", IEEE Trans. on PAS, vol.82, pp.542-544, August 1963.

Tudor J.R., Lewis W.A., "Transmission Losses and Economy Loading by Use of The Admittance Constants", IEEE Trans. on PAS, vol.82, pp.676-683, 1963.

Van Ness J.E., "A Note on Incremental Loss Computation", AIEE Trans., vol.81, Pt.III, pp.735-739, Feb. 1963.

Walker T.R., Hutchins W.H., "Accuracy of Loss Formula Coefficients on the TVA Power System", AIEE Trans., p.858 ,1963.

1964

Anstine L.T., Henderson J.H., Kramer F.A., Light F.H., Pall G.A., Reed F.M., Stewart H.G., Wetter C.H., "Application of Digital Computer Techniques to the Economic Scheduling of Marginal Generating Units for the Pensylvania, New Jersey, Maryland Interconnection", IEEE Trans. on PAS, vol.83, pp.316-320, 1964.

Fukada S., Umezu T., Sekine Y., "The Present Status of the Economic Operation of the Power System in Japan", CIGRE Report no.308, 1964.

Happ H.H., Hohenstein J.F., Kirchmayer L.K., Stagg G.W., "Direct Calculation of Transmission Loss Formula -II", IEEE Trans. on PAS, vol.83, pp.702-707, 1964.

1965

Krumm L.A., "Generalization of the Gradient Method of Optimization of Mode of Operation of Integrated Power Systems", Isvestia Academy Nauk, USSR Energetica and Transport, no.3, pp.3-16, 1965.

1966

Cuenod M., Durling A., Pun L., "Optimum Fitting Method, and its Application to Dynamic Economic Dispatch of Power Systems", Third IFAC Conf. London, June 1966.

1967

Dopazo J.F., Klitin O.A., Stagg G.W., Watson M., "An Optimization Technique for Real and Reactive Power Allocation", Proc. of IEEE , vol.55, no.11, pp.1877-1885, 1967.

Gamn A.Z., Krumn L.A., Sher I.A., "Optimizing the Power Flow in a Large Power System by a Gradient Method with Tearing into Subsystems", Elektrichevsko, no.1, pp.21-29, 1969.

Happ H.H., "Multi-Computer Configurations and Diakoptics: Dispatch of Real Power Pools", Proc. of PICA conf., pp.95-107, 1967.

Long R.W., Barrios J.R., "A Parametric Equation Approach to the Economic Dispatch Problem", IEEE Trans. on PAS, vol.86, pp.141-149, 1967.

Norimatsu T.A.T., Deido T., "Optimum Allocation of Active and Reactive Generations", Proc. of PICA Conf., pp.167-176, 1967.

Roth J.E., Ambrose Z.C., Schuppin L.A., Gassert J.D., Hunt D.M., Williams D.D., Wood W., Martin L.W., "Economic Dispatch of Pennsylvania-New Jersey-Maryland Interconnetion System Generation of a Multi-Area Basis", Proc. of PICA Conf., pp.117-126, 1967.

1968

Benthall T.P., "Automatic Load Scheduling in a Multi Area Power System", Proc. of IEE, pp.592-596 ,1968.

C.Brewer, et.al., "Performance of a Predictive Automatic Load Dispatching System",Proc. of IEE, pp.1577-1586 ,1968.

Carpentier J., Cassapoglou C., Hensgen C., "Differential Injections, A Method for General Resolution of Problems of Economic Dispatching Without Complete Variables Using the Generalized Reduced Gradient Method", Proc. of Conf. by Hellenic Operational Res. Soc., Athans,Nov. 4-8 1968.

Dommel G.W., Tinney W.F., "Optimal Power Flow Solutions"IEEE Trans. on PAS, vol.87, pp.1866-1876, 1968.

Hajdu L.P., Peschon J., Tinney W.F., Piercy D.S., "Optimum Load-Shedding Policy for Power Systems", IEEE Trans. on PAS, vol.87, pp.784-795, 1968.

Hano I., Chiba A., Umezu T., Ode K., "Optimal Control of Voltage and Reactive Power in Power Systems", CIGRE paper 32-03 ,1968.

Hill E.F., Stevenson jr. W.D., "A New Method of Determining Loss Coefficients", IEEE Trans. on PAS, vol.87, pp.1548-1553, 1968.

Hill E.F., Stevenson jr. W.D., "An Improved Method for Determining Incremental Loss Factors from Power System Admittances and Voltages", IEEE Trans. on PAS, vol.87, pp.1419-1425, 1968.

Kumai K., Ode K., "Power System Voltage Control by Using a Process Control Computer", IEEE Trans. on PAS, vol.87, pp.1985-1990, 1968.

Peschon J., Piercy D.S., Tinney W.F., Tveit O.J., Cuenod M., "Optimum Control of Reactive Power Flow", IEEE Trans. on PAS, vol.87, no.1, pp.40-48, 1968.

Peschon J., Piercy D.S., Tinney W.F., Tveit O.J., "Sensitvity in Power Systems", IEEE Trans. on PAS, vol.87, pp.1687-1696, 1968.

Wells D.W., "Method for Economic Secure Loading of a Power System", Proc. of IEE, vol.115, no.8, pp.1190-1194, 1968.

1969

Ariatti F., Grohmann D., Venturini D., "Method for Economic Load Dispatching in a Thermal Power System", Proc. of 3th PSCC Conf., Rome, Italy ,1969.

El Abiad A.H., Jaimes F.J., "A Method for Optimal Scheduling of Power and Voltage Magnitude", IEEE Trans. on PAS, vol.88, pp.413-422, 1969.

Farrara E., Galiero R., "Application of Linear Programming and Special Loadflow Algorithms to Optimum Scheduling Computations", Proc. of 3th PSCC Conf., Rome, Italy, 1969.

Fink L.H., Kwatny H.G., McDonald J.P., "Economic Dispatch of Generation Via Valve-Point Loading", IEEE Trans. on PAS, vol.88, pp.805-811, 1969.

Hano I., Tamura Y., Narita S., Matsomoto K., "Real Time Control of System Voltage and Reactive Power", IEEE Trans. on PAS, vol.88, no.10, pp.1544-1559, 1969.

Happ H.H., Undrill J.M., "MultiComputer Computation of Power Flow and Economic Dispatch", Proc. of 3th PSCC Conf., Rome, Italy ,1969.

Happ H.H., "Multicomputer Configurations and Diakoptics - Dispatch of Real Power in Power Pools", IEEE Trans. on PAS, vol.88, pp.764-772, 1969.

Meyer W.S., Albertson V.D., "Loss Formula Computation by Optimal Ordering Techniques Which Exploit The Sparsity of The Network Admittance Matrix, Pt. I and II", Midwest Power Symposium ,1969.

Sasson A.M., "Combined Use Of The Powell and Fletcher-Powell Nonlinear

Programming Methods for Optimal Load Flows", IEEE Trans. on PAS, vol.88, pp.1530-1537, 1969.

Sasson A.M., "Nonlinear Programming Solutions for The Load Flow Minimum Loss and Economic Dispatch Problems", IEEE Trans. on PAS, vol.88, pp.399-409, 1969.

Sasson A.M., Cory B.J., "Non-Linear Programming Techniques for the Load Flow, Minimum Loss and Economic Dispatchıns Problems", Proc. of 3th PSCC Conf., Rome, Italy, 1969.

Shen C.M., Laughton M.A., "Determination of Optimum Power-System Operating Conditions Under Constraints", Proc. of IEE, Vol.116, pp.225-239, 1969.

Sullivan R.L., Elgerd O.I., "Minimally Proportioned Reactive Generation Control Via Automatic Tap Changing Transformers", Proc. of PICA Conf., Denver Colo., pp.233-241, 1969.

Taylor D.G., "A Linear Programming Method Suitable for Power Flow Problems", Proc. of 3rd PSCC Conference, Rome, 1969, paper OM2.

1970

Bokay B., Racz L., "Calculation of Minimum Reactive Power in Optimizing the Operation of an Energy System", Symposium on the Application of Operational Research, pp.25-27, 1970.

G.Daniel, "Some Aspects of the Optimization of Electric Power Systems", Symposium on the Application of Operational Research,pp25-27 ,1970. [1]

Dillon T., Morsztyn K., Granborg B.S.M., "Application of the Sequential Unconstrained Minimization Technique for Nonlinear Programming to the Optimization of a Thermal Power System", Proc. of 3ed Hawaii International Conf., pp.949-952 ,1970.

Dodu J.C., Merlin A., "A Solution Method for the Economic Dispatch Problem in

The Direct Current Approximation (Maya Model)", Bull. EDF Div. Etud. et Rech. B , no.3, pp.5-24, 1970.

Friedlander G.D., "Power, Pollution, and the Imperiled Environment - I, II", IEEE Spectrum, pp.65-70, Dec. 1970.

Jaimes F.J., El Abiad A.H., "Nonlinear Programming and Optimal Scheduling of Power System Operations", Midwest Power Symposium 1970.

Mikami H., Nagasako K., Nakano S., Hirakoji Y., Tsuboi H., "All Digital Economic Load Dispatching and Control System", Hitachi Rev., vol.19, no.11, pp.405-414, 1970.

Olesnicky R.M.A., "A New Approach for Economic Load Scheduling", Proc. of IEE, pp.1388-1389, 1970.

Ramamoorty M., Rao G., "Economic Load Scheduling of Thermal Power Systems using the Penalty Function Approach", IEEE Trans. on PAS, vol.89, pp.2075-2078, 1970.

Shen C.M., Laughton M.A., "Power System Load Scheduling with Security Constraints using Dual Linear Programming", Proc. of IEE, vol.117, no.11, pp.2117-2127, 1970.

<u>1971</u>

Aldrich J.F., Happ H.H., Lever J.F., "Multi-Area Dispatch", IEEE Trans. on PAS, vol.90, pp.2661-2670, 1971.

Aldrich J.F., Happ H.H., Lever J.F., "Power Dispatch in Multi-Areas", Proc. of the American Power Conf., vol.33, pp.1084-1093 ,1971.

Gent M.R., Lamont J.W., "Minimum Emission Dispatch", IEEE Trans. on PAS, vol.90, pp.2650-2660, 1971.

Gladys H., "Criteria for Economic Load Distribution Among Interconnected Power Systems", vol.47, no.2, pp.73-77, 1971.

Gladys H., "Consideration of Network Losses in the Presence of Economic Load Distribution Among Interconnected Power Systems", vol.47, no.3,pp. 106-109, 1971.

Gungor R.B., Tsang N.F., Webb B., "A Technique for Optimizing Real and Reactive Power Schedules", IEEE Trans. on PAS, vol.90, pp.1781-1790, 1971.

Happ H.H., "The Inter-Area Matrix: A Tie Line Flow Model for Power Pools", IEEE Trans on PAS, vol.90, no.1, pp.36-45, 1971.

Kaltenbach J.C., Hajdu L.P., "Optimal Corrective Rescheduling for Power System Security", IEEE Trans. on PAS, vol.90, pp.843-851, 1971.

Kishore A., Hill E.F., "Static Optimization of Reactive Power Sources by use of Sensitivity Parameters", IEEE Trans. on PAS, vol.90, pp.1166-1173, 1971.

Meyer W.S., Albertson V.D., "Improved Loss Formula Computation by Optimally Ordered Elimination Techniques", IEEE Trans. on PAS, vol.90, no.1, pp.62-69, 1971.

Narita S., Hamman M.S.A.A., "A Computational Algorithm for Real-Time Control of System Voltage and Reactive Power,Part I-Problem Formulation; Part II-Algorithm of Optimization", IEEE Trans. on PAS, vol.90, no.6, pp.2495-2508, 1971.

Peschon J., Bree D.W., Hajdu L.P., "Optimal Solucions Involving System Security", Proc.of PICA Conf., pp.210-218, May 1971.

Sasson A.M., Viloria F., Aboytes F., "Optimal Load Flow Solution Using the Hessian Matrix", Proc.of PICA Conf., pp.203-209 ,May 1971.

Subramanian D.K., "Optimum Load Shedding Through Programming Techniques", IEEE Trans. on PAS, vol.90, no.1, pp.89-95, 1971.

Thanikachalam A., Tudor J.R., "Optimal Rescheduling of Power for System Reliability", IEEE Trans. on PAS, vol.90, pp.2186-2192, 1971.

<u>1972</u>

Adielson T., "Determination of an Optimal Power Flow by Iterative Suboptimizations", Proc. of 4th PSCC Conf., vol.2, paper 2.1/9, 1972.

Bechert T.E., Kwatny H.G., "On the Optimal Dynamic Dispatch of Real Power", IEEE Trans. on PAS,vol.91, pp.889-898, 1972.

Billinton R., Sachdeva S.S., "Optimal Real and Reactive Power Operation in a Hydrothermal System", IEEE Trans. on PAS, vol.91, pp.1405-1411 ,1972.

Brewer C., Revington A.M., "Linear Programming for Optimizing Generation and Immediate Spare: Development and Application", <u>IEE Conf. Proc. on Computers in Power System Operation and Control</u>, Bournemouth, England, pp.115-134, 1972.

Carpentier J., "Results and Extensions of the Methods of Differential Injections", Proc. of 4th PSCC Conf., vol.2, paper 2.1/8, 1972.

Cory B.J., Henser P.B., "Economic Dispatch with Security using Nonlinear Programming", Proc. of 4th PSCC Conf., paper 2.1/5, 1972.

Daniels H., Chen M.S., "An Optimization Technique and Security Calculations for Dispatch Computers", IEEE Trans. on PAS, vol.91, pp.883-888, 1972.

Dillon T., Morsztyn K., "Active and Reactive Load Scheduling in a Thermal Power System in the Presence of Tap-Changing Transformers Using Nonlinear Programming", Inst. Eng. Aust. Electr. Eng. Trans. , EE1 (2),pp.67-70 ,1972.

Jolissaint C.H., Arvantidis N.V., Luenberger D.G., "Decomposition of Real and Reactive Power Flows", IEEE Trans. on PAS, vol.91, pp.661-670 ,1972.

Merlin A., "On The Optimal Generation Planning in The Large Transmission System (The MAYA Model)", Proc. of 4th PSCC Conf., vol.2, paper 2.1/6, 1972.

Peschon J., Dommel H., Powell W., Bree D.W., "Optimal Power Flow for Systems with Area Interchange Controls", IEEE Trans. on PAS, vol.91, pp.889-895, 1972.

Sekine Y., Kawakami J., "Optimal Line Dispatching of Transmission Lines and Transformers", Proc. of 4th PSCC Conf., vol.2, paper 2.1/3, 1972.

Sullivan R.L., "Minimum Pollution Dispatching", IEEE PES Summer Meeting, 1972.

Sullivan R.L., "Controlling Generator MVAR Loadings Using a Static Optimization Technique", IEEE Trans. on PAS, vol.91, pp.906-910, 1972.

Velghe J., Perterson N.,"Optimal Control of Real and Reactive Power Flows Under Constraints", Proc. 4th.PSCC, Grenoble,Paper2.1/14,Sep.1972. [3],[4]

1973

Billinton R., Sachdeva S.S., "Real and Reactive Power Optimization by Suboptimum Techniques", IEEE Trans. on PAS, vol.92, no.3,.pp.950-956, 1973.

Carpentier J.L., "Differential Injection Method, A General Method for Secure Optimal Load Flows", Minnesota,Paper TP VIII E(a),June 4-6,1973.

Dension O.J., Repper N.D., Ringlee R.J., "Direct Economic Dispatch", paper C 73 098-1, IEEE PES Winter Meeting, 1973.

Deo D., "Economic Load Dispatch of Interconnected Power Sytem by Linear Programming", J. Inst. Engr. (India), Elect. Eng. Div., vol.53, pt.EL 3, pp.117-122, 1973.

Dhar R.M., Mukherjee P.K., "Reduced Gradient Method for Economic Dispatch", Proc. of IEE, vol.120, no.5, pp.608-610, 1973.

Duran H., "A Simplex-Like Method for Solving the Optimum Power Flow Problem", Proc. of PICA Conf., 1973.

Friedmann P.G., "Power Dispatch Strategies for Emission and Environmental Control", Proc. of Instrument Society of America, vol.16, 1973.

Glavitsch H., "Economic Load Dispatching and Corrective Rescheduling using On-Line Information of the System State", Proc. of PICA Conf., 1973.

Kwatny H.G., Bechert T.E., "On the Structure of Optimal Area Controls in Electric Power Networks", IEEE Trans. on Automatic Control, vol.AC-18, no.2, pp.167-172, 1973.

Lamont J.W., Gent M.R., "Environmentally Oriented Dispatching Techniques", Proc. of PICA Conf., pp.421-427 ,1973.

Nabona N., Ferris L.L., "Optimization of Economic Dispatch Through Quadratic and Linear Programming", Proc. of IEE, vol.120, no.5, pp.574-580, 1973.

Nanda J., Arora D.B., "Linear Programming Application to Real Power Dispatch with Running Spare Capacity Constraints", IEE-IERE Proc. (India),vol.11, no.6, pp.241-246, 1973.

Nicholson H., Sterling M.J.H., "Optimum Dispatch of Active and Reactive Generation by Quadratic Programming", IEEE Trans. on PAS, vol.92, pp.644-654, 1973.

Patton A.D., "Dynamic Optimal Dispatch for Real Power for Thermal Generating Units", Proc. of 8th PICA conf., pp.403-411, Minneapolis, Min., 1973.

Podmore R., "A Simplified and Improved Method for Calculating Loss Formulas", Proc. of PICA Conf., 1973.

Ramamoorty M., "Reduced Gradient Method for Economic Dispatch", Proc. of IEE, vol.120, no.11, pp.1419-1420, 1973.

Reid G.F., Hasdorff L., "Economic Dispatch Using Quadratic Programming", IEEE Trans. on PAS, vol.92, pp.2015-2023, 1973.

Sasson A.M., Viloria F., Aboytes F., "Optimal Load Flow Solution Using The Hessian Matrix", IEEE Trans. on PAS, vol.92, no.1, pp.31-41, 1973.

1974

Abou Taleb N., Megahed I., Moussa A., Zaky A., "A New Approach to the Solution of the Economic Dispatch Problem", IEEE PES Winter Meeting, paper C 74 167-3, 1974.

Alsac O., Stott B., "Optimal Load Flow with Steady State Security", IEEE Trans. on PAS, vol.93, no.3, pp.745-751, 1974.

Cameron II D.E., Koehler J.E., Ringlee R.J., "A Study Mode Multi-Area Economic Dispatch Program", IEEE PES Winter Meeting, 1974.

Dayal G., Hasdorff L., "Optimal Load Flow Solution Using The Ricochet Gradient Method", IEEE PES Winter Meeting, paper C74 019-6, 1974.

Delson J.K., "Controlled Emission Dispatch",IEEE PES Winter Meeting, 1974.

T.E.Dy Liacco, "Real Time Control of Power Systems",IEEE Proc.,vol.62 ,no.7, pp.884-891 ,July 1974.

Ferrer T.A., Morst R.L., "Effluent Charges - A Price on Pollution", Atmos. Environment, vol.8, pp.657-667, 1974.

Finnegan O.E., Fouad A.A., "Economic Dispatch with Pollution Constraints", IEEE PES Winter Meeting, 1974.

Graf D., Schneider K.D., "Economically Optimum Reactive Power Characteristics of the East German Power System", Energietechnik, vol.24, no.2, pp.65-69, 1974.

Happ H.H., "Optimal Power Dispatch", IEEE Trans. on PAS, vol.93, pp.820-830, 1974.

Mukherjee P.K., Dhar R.N., "Optimal Load-Flow Solution by Reduced-Gradient Method", Proc. of IEE, vol.121, no.6, pp.481-487, 1974.

Nanda J., Arora D.B., "Application of Linear Programming for Economic Dispatch with Line Flow Constraints", J. Inst. Eng. (India), vol.54, pt.El 5, pp.166-171, 1974.

Podmore R., "Economic Power Dispatch with Line Security Limits", IEEE Trans. on PAS, vol.93, no.1, pp.289-295, 1974.

Rashed A.M.H., Kelly D.H., "Optimal Load Flow Solution Using Lagrangian Multipliers and the Hessian Matrix", IEEE Trans. on PAS, vol.93, no.5, pp.1292-1297, 1974.

Suhakar D.P., Sarma M.S., "On-Line Optimal Load Scheduling of Power Systems Using Modified Fletcher-Powell Method", 5th Pittsburg Conf. on Modeling and Simul., pt.2, p.651, 1974.

Wollenberg B.F., Stadlin W.O., "A Real Time Optimum for Security Dispatch", IEEE Trans. on PAS, vol.93, pp.1640-1649, 1974.

1975

Alsac O., Stott B., "Decoupled Algorithms in Optimal Load Flow Calculations", IEEE PES Summer Meeting, paper 75545-4, 1975.

Cadogan J.B., Eisenberg L., "Environmental Control of Electric Power Systems", ISA Trans., vol.14, pp.214-224, 1975.

Dejax P.J., Gazis D.C., "Optimal Dispatch of Electric Power with Ambient Air Quality Constraints", IBM Watson Research Center, Yorktown Heights, N.Y., 1975.

Dillon T.S., Morsztyn K., Tun T., "Sensitivity Analysis of the Problem of Economic Dispatch of Thermal Power Systems", Int. J. of Control, vol.22, no.2, pp.229-248, 1975.

Dodu J.C., Merlin A., "The Application of Linear Programming to the Planning of Large-Scale Power Transmission Systems: The Mexico Program", Proc. 5th PSCC Conf., Cambridge, England, 1975.

Eisenberg L., Cadogan J., "Evaluation of Environmental Dispatch Strategies", presented at ISA Annual Conf., Milwaukee, 1975.

El-Shibini M.A., Dayeh M.B., "Reactive Power Optimization Using Modified Linear Programming", IEEE PES Winter Meeting, paper C 75 024-5, 1975.

Happ H.H., Nour N.E., "Multi-Areas Network Modelling for Power Pools", IEEE PES Summer Meeting, 1975.

IEEE Working Group on Operating Economics, "Economy Security Functions in Power Systems Operations",paper 75CH0969-6-P5,IEEE PES Winter Meeting.

Jain J.K., Rao N.D., "A Rapid and Reliable Method for On Line Economic Dispatch", Proc. of PICA Conf., 1975.

Khan M.A., Pai M.A., "Securty Constrained Optimization of Power Sytems", Proc. of PICA Conf., pp.61-67, 1975.

Kohli N.P., Ray L.M., "Active and Reactive Power Optimization by Least Squares", IEEE PES Winter Meeting, paper C 75 053-4, 1975.

Lamont J.W., Sim K., Hamilton E.P., "A Multi-Area Environmental Dispatching Algorithm", Proc. of PICA Conf., 1975.

Pai M.A., Paranjothi S.R., "Optimal Power Flow with Security Constraints using Successive Linear Programming", IEEE PES Winter Meeting, paper A 75-455-6, 1975.

Ruane M.F., Schweppe F.C., Gruhl J., "Economic-Environmental System Planning for Electric Power Systems", from Systems Engineering for Power: Status and Prospects, U.S. Energy Research and Development Administration, Henniker, New Hampshire 1975.

Sachdev M.S., Ibrahim S.A., "A Simulation Technique for Studying Real and Reactive Power Flow Patterns", IEEE Trans. on PAS, vol.94, no.6, pp.2092-2100, 1975. Correction 76 vol.1 ,p.374.

Schweitzer P.F., "A Computer Simulation Model for Controlling Electric Utility Air Quality Perturbations", Proc. of Summer Computer Simulation Conf. 1975.

Sjelvgren D.V., Bubenko J.A., "Decomposition Technique in a Security Related Optimal Power Flow", Proc. 5th PSCC Conf., Cambridge, England, 1975.

Song K.Y., Lee K.J., "Study on the Application of Optimum Load Shedding", J. of Korean Inst. of Elect. Eng., vol.24, no.2, pp.84-91, 1975.

Spare jr. J.H., "A Multilevel Approach to Economic Dispatch with Security Constraints for Interconnected Power Systems", Ph.D. Dissertation, Un. of Pennsylvania, 1975.

Sullivan R.L., Hilson D.W., "Computer-Aided Ambient Air Quality Assessment for Generation System Planning", Proc. of PICA Conf., pp.247-251, 1975.

Wadhwa C.L., "Modified Variable Metric Algorithm for Optimal Load Flows", IEEE PES Winter Meeting, paper C 75 101-1, 1975.

Wadhwa C.L., "Optimal Loadflow Using Self Scaling Variable Metric Algorithm", IEEE PES Summer Meeting, paper A 75 438-2, 1975.

Zahavi J., Eisenberg L., "Economic - Environmental Power Dispatch", IEEE Trans. Syst. Man. Cyber., vol.5, no.5, pp.485-489, 1975.

1976

Dayal G., Grigsby L.L., Hasdorf L., "Quadratic Programming for Optimal Active And Reactive Power Dispatch using Special Techniques for Reducing Storage Requirements", IEEE PES Summer Meeting, paper A 76 388-9, 1976.

Kothari D.P., Mital R.K., "Comparison of Economic Dispatch and Minimum Emission Dispatch", Electr. India, vol.16, no.1, pp.9-12, 1976.

Saeed M., Shah A., Hill E.F., "Optimal Load Dispatching using Approximate Linear Programming", Joint Power Generation Conference, paper 76 A76-6296, 1976.

Savulescu S.C., "Qualitative Indices for the System Voltage and Reactive Power Control", IEEE Trans. on PAS, vol.95, pp.1413-1421, 1976.

Shoults R.R., Chen M.-S., "Reactive Power Control by Least Squares Minimization", IEEE Trans. on PAS, vol.95, no.1, pp.325-334,1976.

Wahda D.B., Nanda J., "New Approach to Modified Coordination Equations for Economic Load Dispatch", Proc. of IEE, vol.123, no.9, pp.923-925, 1976.

1977

Adler R.G., Fischl R., "Security Constrained Economic Dispatch with Participation Factors Based on Worst Case Bus Loading Variations", IEEE Trans. on PAS, vol.96, pp.347-356, 1977.

Barcelo W.R., Lemmon W.W., Keon H.R., "Optimization of the Real Time Dispatch with Constraints for Secure Operation of Bulk Power Systems", IEEE Trans. on PAS, vol.96, pp.741-757, 1977.

Biggs M.C., Laughton M.A., "Optimal Electric Power Scheduling,a Large Nonlinear Test Program Solved By Recursive Quadratic Programming", Math. Progr., vol.13, pp.167-182, 1977.

Cadogan J.B., Eisenberg L., "Sulfur Oxide Emissions Management for Electric Power Systems", IEEE Trans. on PAS, vol.96, pp.393-401, 1977.

Ejebe G.C., Puntel W.R., Wollenberg B.F., "A Load Curtailment Algorithm for the Evaluation of Power Transmission System Adequacy", IEEE PES Summer Meeting, paper 77 505-1, 1977.

Ghoneim M.Z., Saleh M.H., Askourah M.A.N., Abdel Rahman S.A., "Estimation of

Optimum Load Shedding in Transmission Systems Using Mathematical Programming Techniques", IEEE PES Summer Meeting, paper A77 595-2, 1977.

Hobson E., Stott B., "Linear Programming for Power System Real-Time Control Calculations", Proc. of IFAC Conf., Melbourne, 1977.

Megahed I., Abou Taleb N., Iskaandrani M., Moussa A., "A Modified Method for Solving the Economic Dispatch Problem", IEEE Trans. on PAS, vol.96, pp.124-133, 1977.

Pai M.A., Paranjothi S.R., "Preventive Resheduling of Real and Reactive Powers in Power Systems using Dual LP Formulation", IFAC Symp. on Automatic Control and Protection in Power Systems, Melbourne, 1977.

Shoults R.R., "A Simplified Economic Dispatch Algorithm Using Decoupled Networks", paper A77 738-8, IEEE PES Summer Meeting, Mexico City, 1977.

1978

Alvarado F.L., "Penalty Factors from Newton's Method", IEEE Trans. on PAS, vol.97, no.6, pp.2031-2040, 1978.

Bala jr. J.L., Thanikachalam A., "An Improved Second Order Method for Optimal Load Flow", IEEE Trans. on PAS, vol.97, no.4, pp.1239-1244 ,1978.

El-Hawary M.E., Feehan T., "Economic Active Reactive Dispatch - Theory and Computational Experience", Proc. Canadian Communications and Power Conf. Montreal, Oct. 1978, pp.386-388.

Fernandes R.A., Happ H.H., Wirgau K.A., "System Loss Reduction by Coordinated Tap and Generator Voltage Adjustment", Proc. American Power Conf., Vol.40, pp.1390-1393, 1978.

Fernandes R.A., Happ H.H., Wirgau K.A., "Optimal Reactive Power Flow: Loss Reduction for Improved System Operations", Proc. Canadian Communications and Power Conf., Montreal, Oct. 1978, pp.324-327.

Fernandes R.A., Happ H.H., Wirgau K.A., "The Application of Optimal Power Flow for Improved System Operation", Proc. of 6th PSCC Conf., vol.1, pp.465-472, Darmstadt Germany, 1978.

Fischl R., Wasley R.G., "Efficient Computation of Optimal Load Flow Sensitivities", Proc. Canadian Communications and Power Conf. Montreal, Oct. 1978 ,pp.401-404.

Galiana F.D., Vojdani A.F., "Analytic Solutions of The Economic Dispatch Problem", Proc. Canadian Communications and Power Conf. Montreal, Oct. 1978, pp.389-391.

Khan M.A., Kuppusamy K., "Optimum Load Curtailment Under Emergency Conditions using Constant Matrices", IEEE PES Summer Meeting, paper A78 549-8, 1978.

Malik J.N., Rao G.K., "Economic Load Dispatch Solution Using Fast Decoupled Load Flow", Proc. IEE, vol.125, no.4, pp.334-335, 1978.

Mamandur K.R.C., Berg G.J., "Economic Shift in Electric Power Generation with Line Flow Constraints", IEEE Trans. on PAS, vol.97, no.5, pp.1618-1626, 1978.

Semlyen A., "A New Approach for Loss Formula Calculation", Proc. Int. Symp. on Simulation, Modelling and Decision in Energy Systems, Montreal, 1978.

Stott B., Hobson E., "Power System Security Control Calculations Using Linear Programming Parts I and II", IEEE Trans. on PAS, vol.97, pp.1713-1731, 1978.

<u>1979</u>

Chan S.M., Schweppe F., "A Generation Reallocation and Load Shedding Algorithm", IEEE Trans. on PAS, vol.98, no.1, pp.26-34, 1979.

Chan S.M., Yip E., "A Solution of the Transmission Limited Dispatch Problem by Sparse Linear Programming", IEEE Trans. on PAS, vol.98, no.3, pp.1044-1053, 1979.

Galiana F.G., Vojdani A.F., "A Comparison of the Classical and Modified Coordination Equations in Economic Dispatch", paper A 79 079-5, IEEE PES Winter Meeting, 1979.

Garver L.L., Van Horne P.R., Wirgau K.A., "Load Supplying Capability of Generation-Transmission Networks", IEEE Trans. on PAS, vol.98, no.3, pp.957-962, 1979.

Grigsby L.L., Rahman S., "The Use Of Linear Programming in Generation Control", Proc. Control of Power Systems Conference & Exposition, Colledge Station, Texas, March 1979.

Ilic M., Calovic M., "Multicriteria Approach to Optimal Operation of Power Systems", Electric Power System Research, vol.2, no.2, pp.145-153, 1979.

Khan M.A., Kuppusamy, "Optimum Load Curtailment under Emergency Conditions using Constant Matrices",paper A79 113-2,IEEE PES Winter Meeting, 1979. [7]

Khan M.A., Kuppusamy, "Security Constrained Dispatch Through Successive Relinearization using Constant Matrices", IEEE PES Winter Meeting, paper A 79 085-2, 1979.

Lugtu R., "Security Constrained Dispatch", IEEE Trans. on PAS, vol.98, pp.270-274, 1979.

Medicherla T.K.P., Billinton R., Sachdeva M.S., "Generation Rescheduling and Load Shedding to Alleviate Line Overloads-Analysis", IEEE Trans. on PAS, vol.98, no.6, pp.1876-1884, 1979.

Nanda J., Stanek E.K., Bijwe P.R., "A Novel Piecewise Fast Decoupled Load Flow Algorithm and its Application to Economic Load Flow Dispatching", paper A 79 069-6, IEEE PES Winter Power Meeting, 1979.

Prada R.B., Cory B.J., "On-Line Secure and Economic Dispatch", Electric Power System Research, vol.2, no.3, pp.189-198, 1979.

Rashed A.M.H., Moussa A., Gamal El-Din M.Y., "Steady State Optimal Load Shedding Solutions in Power System", IEEE PES Summer Meeting, pp.1-6,1979.

Shoults R.R., Grady M.W., Helmick S., "An Efficient Method for Computing Loss Formula Coefficients Based on the Method of Least Squares", IEEE Trans. on PAS, vol.97, no.6, pp.2144-2152, 1979.

Stott B., Marinho J.L., "Linear Programming for Power System Network Security Applications", IEEE Trans. on PAS, vol.98, no.3, pp.837-848, 1979.

Stott B., Marinho J.L., Alsac O., "Review of Linear Programming Applied to Power System Rescheduling ", Proc. of PICA Conf., 1979.

Vojdani A.F., Galiana F.D., "Economic Dispatch with Generation Constraints", Proc. of the 18th. IEEE conference on Decision and Control, Fort Lauderdale, Florida, Dec. 1979.

Wirgau K.A., "Voltage Scheduling in System Operations",NAPSIC - North Central Regional Meeting, Minneapolis, 1979.

Wirgau K.A., "Reactive Power Dispatching", Electric Forum, vol.5, no.1, pp.6-9, 1979.

Wu F.F., Gross G., Luini J.F., Lock P.M., "A Two Stage Approach to Solving Large Scale Optimal Power Flows", Proc. of PICA conf., Cleveland 1979.

1980

Aldrich J.F., Fernandes R.A., Vicks L.W., Happ H.H., "Benifits of Voltage Scheduling in Power Systems", IEEE Trans. on PAS, vol.99, no.5, pp.1701-1712, 1980.

Burchett R.C., Happ H.H., Vierath D.R., Wirgau K.A., "Advanced Optimal Power Flow Methodology", Canadian Communication and Power Conference,1980.

Hobson E.T., "Network Constrained Reactive Power Control Using Linear Programming", IEEE Trans. on PAS, vol.99, no.3, pp.868-877, 1980.

Lee T.H., Thorne D.H., Hill E.F., "A Transportation Method for for Economic Dispatching - Application and Comparison", IEEE Trans. on PAS, vol.99, no.6, pp.2373-2385, 1980.

Ross D.W., Kim S., "Dynamic Economic Dispatch of Generation", IEEE Trans. on PAS, vol.99, no.12, pp.2060-2068, 1980.

1981

Burchett R.C., Happ H.H., Vierath D.R., Wirgau K.A., "Developments in Optimal Power Flow", Proc. of PICA conf., 1981.

Dillon T.S., "Rescheduling,Constrained Participation Factors and Parameter Sensitivity in The Optimal Power Flow Problem", IEEE Trans. on PAS, vol.100, no.5, pp.2628-2634, 1981.

Happ H.H., Wirgau K.A., "Optimal Dispatch and Voltage Scheduling in Power Systems", Proceedings of the International Symposium on Circuits and Systems, 1981.

Lee T.H., Thorne D.H., Hill E.F., "Modified Minimum Cost Flow Dispatching Method for Power System Application",IEEE Trans. on PAS,vol.100 ,no.2, pp.737-744 ,1981.

Lipowski J.S., Charalambous, "Solution of Optimal Load Flow Problem by Modified Recurcive Quadratic Programming Method", Proc. of IEE, vol.128, pt.c, no.5, pp.288-294, 1981.

Mamandur K.R.C., Chenoweth R.D., "Optimal Control of Optimal Power Flow for Improvements in Voltage Profiles and for Real Power Loss Minimization", IEEE Trans. on PAS, vol.100, no.7, pp.3185-3194, 1981.

Medicherla T.K.P., Billinton R., Sachdeva M.S., "Generation Rescheduling and Load Shedding to Alleviate Line Overloads-System Studies", IEEE Trans. on PAS, vol.100, no.1, pp.36-42, 1981.

Palaniswamy K.A., Misra K.B,. Sharma J., "Quasi-Newton Method for Optimal Load Flow Solution under Emergency Mode of Operation", Electric Power System Research, vol.4, no.4, pp.289-296, 1981.

Romano R., Quintana V.H., Lopez R., Valadez V., "Constrained Economic Dispatch of Multiarea Systems using Dantzig-Wolfe Decomposition Principle", IEEE Trans. on PAS, vol.100, no.4, pp.2127-2137, 1981.

Shoults R.R., Sun D.T., "Optimal Power Flow Based on P-Q Decomposition", PICA Conference Record, 1981.

Somuah C.B., Schweppe F.C., "On Line Computer Control of a Power System During an Emergency", IEEE Trans. on PAS, vol.100, no.5, pp.2619-2627, 1981.

Talukdar S.N., "A Fast and Robust Variable Metric Method for Optimum Power Flows", PICA Conference Record,1981.

Vojdani A.F., "Analytic Approach to Economic Dispatch", M.Eng. Thesis, Dept. of Electrical Eng., McGill University, 1981.

1982

Aoki K., Satoh T., "Economic Dispatch with Network Security Constraints Using Parametric Quadratic Programming", IEEE Trans. on PAS, vol.101, no.12, pp.4548-4556, 1982.

Bottero M.H., Galiana F.D., Fahmideh-Vojdani A.R., "Economic Dispatch Using the Reduced Hessian", IEEE Trans. on PAS, vol.101, no.10, pp.3679-3688, 1982.

Bui R.T., Ghaderpanah S., "Real Power Rescheduling and Security Assessment", IEEE Trans. on PAS, vol.101, no.8, pp.2906-2915, 1982.

Burchett R.C., Happ H.H., Vierath D.R., Wirgau K.A., "Developments in Optimal Power Flow", IEEE Trans. on PAS, vol.101, no.2, pp.406-414 ,1982.

Burchett R.C., Happ H.H., Wirgau K.A., "Large Scale Optimal Power Flow ", IEEE Trans. on PAS, vol.101, no.10, pp.3722-3732, 1982.

Chamorel P.A., Germond A.J., "An Efficient Constrained Power Flow Technique Based on Active - Reactive Decoupling and Linear Programming", IEEE Trans. on PAS, vol.101, no.1, pp.158-167, 1982.

Dersin P., Alexander H.L., "Feasibility Sets for Steady-State Loads in Electric Power Networks", IEEE Trans. on PAS, vol.101, no.1, pp.60-70, 1982.

Divi R., Kesavan H.K., "A Shifted Penalty Function Approach for Optimal Load Flow", IEEE Trans. on PAS, vol.101, no.9, pp.3502-3512, 1982.

Elacqua A.J., Corey S.L., "Security Constrained Dispatch at The New York Power Pool", IEEE Trans. on PAS, vol.101, no.8, pp.2876-2884, 1982.

Fox B., McCracken H., "Load Curtailment Model for Economic Loading of Thermal Generation", Proc. of IEE, vol.129, pt.c, no.6, pp.233-238, 1982.

Housos E.C., Irisarri D.D., "A Sparse Variable Metric Optimization Method Applied to the Solution of Power System Problems", IEEE Trans. on PAS, vol.101, no.1, pp.195-202, 1982.

Isoda H., "On Line Load Dispatching Method Considering Load Variation Characteristics and Response Capabilities of Thermal Units", IEEE Trans. on PAS, vol.101, no.8, pp.2925-2930, 1982.

Mamandur K.R.C., "Emergency Adjustments to VAR control Variables to Alleviate Over-Voltages,Under-Voltages,and Generator VAR Limit Violations", IEEE Trans. on PAS, vol.101, no.5, pp.1040-1047, 1982.

Quintana V.H., Lipowski J.,"Optimal Load Flow via a Quadratic-Programming Approach", in <u>Large Engineering Systems</u>, O.P.Malik, M.C.Sing(eds.),1982.

Shoults R.R., Sun D.T.-W., "Optimal Power Flow Based upon P-Q Optimization", IEEE Trans. on PAS, vol.101, no.2, pp.397-405, 1982.

Talukdar S.N., Giras T.C., "A Fast and Robust Variable Metric Method for Optimum Power Flows", IEEE Trans. on PAS, vol.101, no.2, pp.415-420, 1982.

Vojdani A.F., Galiana F.D., "Economic Dispatch with Non Convex Cost Functions and its application to Valve Point Loading", Large Scale System Symposium, Sponsored by the IEEE,1982.

1983

Blanchon G., Dodu J.C., Merlin A., "Developing a New Tool for Real-Time Control in Order to Coordinate the Regulation of Reactive Power and the Voltage Schedule in Large-Scale EHV Power Systems", Symp. CIGRE-IFAC, paper 209-01, Florence, 1983.

Carpentier J.L., Cotto G., "Modern Concepts for Security Control in Electric Power System", Proc. of CIGRE, paper 102-01, Florence, Italy, 1983.

Chandrashekhar K.S., Hill D.J., "Dynamic Security Dispatch: Basic Formulation", IEEE Trans. on PAS, vol.102, no.7, pp.2145-2154, 1983.

Contaxis G.C., Papadias B.C., Delkis C., "Decoupled Power System Security Dispatch", IEEE Trans. on PAS, vol.102, no.9, pp.3049-3056, 1983.

Elangovan S., "A New Approach for Real Power Minimization", Proc. of IEE, vol.130, pt.c, no.6, pp.295-299, 1983.

Elfstrom L., Sjelvgren D., Paulsson E., Bubenko J., "Optimal Reactive Power Flow for Preventive Rescheduling for Hydro-Thermal Power Systems", Symp. CIGRE-IFAC, paper 104-04, Florence, 1983.

Fahmideh-Vojdani A., Galiana F.D., "The Continuation Method and its Application in System Planning and Operation", Proc. of CIGRE, paper 102-04, Florence, Italy, 1983.

Franchi L., Innorta M., Marannino P., Sabelli C., "Evaluation of Economy and/or Security Oriented Objective Functions for Reactive Power Scheduling in Large Scale Systems", Proc. of PICA Conf., Houston, pp.277-284, 1983.

Galiana F.D., Fahmideh-Vojdani A., Huneault M., Juman M., "Optimal Power System Dispatch Through the Continuation Method: Variation of Functional Inequality Limits", Proc. of IEEE International Symposium on Circuits and Systems, California, May 1983.

Glavitsch H., Spoerry M., "Quadratic Loss Formula for Reactive Dispatch", Proc. of PICA Conf., Houston, pp.27-33, 1983.

Housos E., Irisarri G.D., "Real and Reactive Power System Security Dispatch using a Variable Weights Method", IEEE Trans. on PAS, vol.102, no.5 , pp.1260-1268, 1983.

Irving M.R., Sterling M.J.H., "Economic Dispatch of Active Power and Constraint Relaxation", Proc.of IEE, vol.130, pt.c, no.4, pp.172-177, 1983.

Jamshidian F., Liu K.C., Lugtu R., "Penalty Factors Calculations Incorporating Interchange Constraints", Proc.of PICA Conf., Houston, pp.95-99, 1983.

Krogh B.H., "Allocation of Generation Ramping to Reduce Transmission Line Overloads", IEEE Trans. on PAS, vol.102, no.7, pp.2110-2115, 1983.

Krogh B., Javid S.H., Chow J.H., Multistage Rescheduling of Generation, Load Shedding and Short Term Transmission Capacity for Emergency State Control", IEEE Trans. on PAS, vol.102,no.5, pp.1466-1472, 1983.

Ramaiyer S., Ramachadran K., Hariharan S., "New Technique for Optimal Reactive Power Allocation for Loss Minimization in Power Systems", Proc. of IEE, vol.130, pt.c, no.4, pp.178-182, 1983.

Roy L., Rao N.D., "A New Algorithm for Real-Time Optimal Dispatch of Active and Reactive Generation Retaining Nonlinearity", IEEE Trans. on PAS, vol.102, no.4, pp.832-842, 1983.

Stott B., Alsac O., "Experience with Successive Linear Programming for Optimal Rescheduling of Active and Reactive Power", Proc. of CIGRE, paper 104-01, Florence, Italy, 1983.

Talukdar S.N., Giras T.C., Kalyan V.K., "Decomposition for Optimal Power Flows", IEEE Trans. on PAS, vol.102, no.12, pp.3877-3884, 1983.

Xia Zu-Zhi, Si Zhong-Ren, "Automatic Generation and Economic Dispatch Control with Security Constraints",paper 104-03,Proc of CIGRE,Florence, 1983. [1]

1984

Aoki K., Satoh T., "New Algorithm for Classic Economic Dispatch", IEEE Trans. on PAS, vol.103, no.6, pp.1423-1431, 1984.

Blanchon G., Dodu J.C., Merlin A., "New Developments of the Reactive Power Flow Optimization Model used at EDF", from Proc. 8th PSCC conf., pp.427-434, Helsinki, 1984.

Burchett R.C., Happ H.H., Vierath D.R., "Quadratically Convergent Optimal Power Flow", IEEE Trans. on PAS, vol.103, no.11, pp.3267-3276, 1984.

Carpentier J., "A Link between Short Term Scheduling and Dispatching: 'Seperability of Dynamic Dispatch' ", Proc. 8th PSCC conf., pp.391- 395, Helsinki, 1984.

Chamorel P.-A., Germond A.J., "Hierarchal Optimization of Reactive Power with Linear Programming", Proc. 8th PSCC conf., pp.409-417, Helsinki, 1984.

Doi A., Tadeka S., Uemura K., "A New Efficient Method for Integrated Control of Voltage and Reactive Power in Power Systems", Proc. 8th PSCC Conf, pp.434-438, Helsinki, 1984.

Farghal S.A., Tantawy M.A., Abopu Hussein M.S., Hassan S.A., Abou Elela A.A., "Fast Technique for Power System Security Assessment using Sensitivity

Parameters of Linear Programming", IEEE Trans. on PAS, vol.103, no.5 , pp.946-953, 1984.

Hobson E., Fletcher D.L., Stadlin W.O., "Network Flow Linear Programming Techniques and Their Application to Fuel Scheduling and Contingency Analysis", IEEE Trans. on PAS, vol.103, no.7, pp.1684-1691, 1984.

Horton J.S., Grigsby L.L., "Voltage Optimization using Combined Linear Programming and Gradient Techniques", IEEE Trans. on PAS, vol.103, no.7, pp.1637-1643, 1984.

Huneault M., Calderon R., Galiana F.D., "Fast Secure Economic Dispatch using Continuation Methods", Proc. of CEA conference, Halifax N.S.,May 1984.

Landqvist A., Bubenko sr. J.A., "A Generalized Reduced Gradient Methodology for Optimal Reactive Power Flows", from Proc. 8th PSCC conf., pp.418-425, Helsinki 1984.

Lee K.Y., Park Y.M., Ortiz J.L., "Optimal Real and Reactive Power Dispatch", Electric Power System Research, vol.7, no.3, pp.201-212, 1984.

Lin C.E., Viviani G.L., "Hierarchal Economic Dispatch Device for Piecewise Quadratic Cost Functions", IEEE Trans. on PAS, vol.103, no.6, pp.1170-1175, 1984.

Luo J.S., HIll E.F., Lee T.H., "Power System Economic Dispatch via Network Approach", IEEE Trans. on PAS, vol.103, no.6, pp.1242-1248, 1984.

Mansour S.Y., Kelly D.H., Koval D.O., "Impact of Reactive Power Flow on the Optimum Economic Operation of Power Systems", Proc. of Canadian Society for Electrical Engineering, pp.179-192, May 1984.

Mota-Palomino R., Quintana V.H., "A Penalty-Function Linear Programming Method for Constrained Economic Dispatch", IEEE Trans. on PAS, vol.103, no.6, pp.1414-1422, 1984.

Peralta W.H., Serrano B.R., Rivera J.F., "Optimum Elimination of Inadmissible Operating States", Electric Power Systems Research, vol.7, pp.255-269, 1984.

Sun D.I., Ashley B., Brewer B., Hughes A., Tinney W.F., "Optimal Power Flow by Newton Approach", IEEE Trans. on PAS, vol.103, no.10, pp.2864-2880, 1984.

Zhang G.R., Brameller A., "On-Line Security Constrained Economic Dispatch and Reactive Power Control Using Linear Programming", Proc. 8th PSCC conf., pp.396-400, Helsinki 1984.

1985

Chandrashekar K.S., "Online Correction Dispatch Algorithm for Dynamic Security", Proc. of IEE, vol.132, pt.c, no.1, pp.20-22, 1985.

Finley L.A., Standish T.R., Philips R.C., "Optimizing System Performance Through Dynamic Load Shed Scheduling", IEEE Trans. on PAS, vol.104, no.6, pp.1286-1289, 1985.

Huneault M., Fahmideh-Vojdani A., Juman M., Calderon R., Galiana F.D., "The continuation Method in Power System Optimization: Applications to Economy-Security Functions", IEEE Trans. on PAS, vol.104, nol, pp.114-124, 1985.

Innorta M , Marannino P., "Advance Dispatch Procedures for the Centralized Control of Real Power", Proc. of PICA conf., pp.188-194, 1985.

Lee K.Y., Park Y.M., Ortiz J.L., "A Unified Approach to Optimal Real and Reactive Power Dispatch", IEEE Trans. on PAS, vol.104, no.5, pp.1147-1153, 1985.

Lim S.S., Ahmed N.U., "Dynamic Economic Dispatch under System Availability Constraints and Transmission Losses", Electric Power System Research, vol.9, pp.63-72, 1985.

Lim S.S., Ahmed N.U., Schenk K.F., "Dynamic Optimal Economic Dispatch under System Availability Constraints", Electric Power System Research, vol.9, pp.11-27, 1985.

Palaniswamy K.A., Sharma J., Misra K.B., "Optimum Load Shedding Taking Into Account of Voltage and Frequency Characteristics of Loads", IEEE Trans. on PAS, vol.104, no.6, pp.1342-1348, 1985.

Van den Bosh P.P.J., "Optimal Static Dispatch with Linear, Quadratic, and Non-Linear Functions of the Fuel Costs', IEEE PES Winter Meeting, Paper 85 WM 085-6, 1985.

Van den Bosh P.P.J. ,"Optimal Dynamic Dispatch Owing to Spinning-Reserve and Power Rate Limits", IEEE PES Winter Meeting, Paper 85 WM 086-4, 1985.

Wenyuan L., "An On-Line Power Dispatch Method with Security", Electric Power System Research, vol.9, no.2, pp.173-182, 1985.

Xu G., Galiana F.D., Low S., "Decoupled Economic Dispatch Using the Participation Factors Load Flow", IEEE Trans. on PAS, vol.104, no.6, pp.1377-1384, 1985.

Zaborsky J., Huang G., Lin S.Y., "Control of Reactive Power and Voltage in Emergencies", Automatica, vol.21, no.3, pp.237-246, 1985.

1986

Backlund Y., Sjelvgren D., "An Extended Generalized Gradient Methodology with Second Order Information for Optimal Power Flows", Proc. of Second Int. Conference on Power System Monitoring and Control, Durham, July 1986, pp.301-306.

Bright J.M., Demaree K.D., Britton J.P., "Reactive Security and Optimality in Real-Time", Proc. of IFAC conf., pp.468-473, 1986.

Boming Z., Niande X., Shiying W., Erkeng Y., Saosien S., "A Fast Loop-Based Algorithm for Computing Loss Formula Coefficients", Proc. of IFAC conf., pp.518-523, 1986.

Broussole F., Dubost L., "A Real-Time Secure Economic Dispatch Model with Provisional Overload", Proc. of IFAC conf., pp.530-534, 1986.

Carpentier J.L., " 'CRIC',A New Active Reactive Decoupling Process in Load Flows Optimal Power Flows and System Control", Proc. of IFAC 1986.

Cernic P.M., "Multiarea Economic Dispatch Using The Gradient Method", Proc. of IFAC conf., pp.512-517, 1986.

Contaxis G.C., Delkis C., Korres G., "Decoupled Optimal Load Flow Using Linear or Quadratic Programming", IEEE Trans. on PWRS, vol.1, no.2, pp.1-7 1986.

El-Kady M.A., Bell B.D., Carvalho V.F., Burchett R.C., Happ H.H., Vierath D.R., "Assessment of Real Time Optimal Voltage Control", IEEE Trans. on PWRS, vol.1, No.2, pp.98-107, 1986.

Fox B., McCracken H., "Determination of Implied Load Curtailment Costs for Economic Loading Calculations", Proc. of Second Int. Conference on Power System Monitoring and Control, Durham, July 1986, pp.172-176.

Granville S., Periera M.V.F., Monticelli A., "A Decomposition Approach to Reactive Sources Optimization", Proc. of IFAC conf.,pp.496-501, 1986.

Hassan Ghoudjehbaklou, Hans Bjorn Puttgen, "Optimization Topics Related to Small Power Producing Facilities Operating under Energy Spot Pricing Policies", paper 86 SM 302-4 ,presented at IEEE PES Summer Meeting, July 1986. [23]

Hon T.K., Sterling M.J.H., Cheng D.T.Y., "Emergency Rescheduling by a Network Flow Method", Proc. of Second Int. Conference on Power System Monitoring and Control, Durham, July 1986, pp.113-118.

Maria G.A., Findlay J.A., "A Newton Optimal Power Flow Program for Ontario Hydro EMS", paper 86 SM 326-3 ,presented at IEEE PES Summer Meeting, July 1986.

Monticelli A., Periera M.V.F., Granville S., "Security Constrained Optimal Power Flow with Post-Contingency Corrective Rescheduling", IEEE PES Winter Meeting, paper 86 WM 085-05, New York 1986.

Mota-Palomino R., Quintana V.H., "Sparse Reactive Power Scheduling by a Penalty Function -Linear Programming Technique", IEEE Trans. on PWRS, vol.1, no.3, pp.31-39, 1986.

Padiyar K.R., Aggarwal R.P., Jose T.L., "Reactive Power Control using Dual Interval Programming", Proc. of Second Int. Conference on Power System Monitoring and Control, Durham, July 1986, pp.162-165.

Van Meeteren H.P., Frame J.G., Fudeh H.R., "Optimal Power Flow in Energy Management Systems", Proc. of Second Int. Conference on Power System Monitoring and Control, Durham, July 1986, pp.296-300.

Zhang G., "On-Line Network Constrained Reactive Power Control Using an Incremental Reactive Current Model", Proc. of Second Int. Conference on Power System Monitoring and Control, Durham, July 1986, pp.156-161.

Zhang G., "Security Constrained Economic Dispatch and Reactive Power Control using Successive Linear Programming", Proc. of IFAC conf., pp. 524-529, 1986.

1987

Innorta M., Marannino P., Granelli G.P., Montagna M., Silvestri A., "Security Constrained Dynamic Dispatch of Real Power for Thermal Groups", Proc. of PICA Conf., Montréal, 1987.

APPENDIX 2.1

SOME BASIC RESULTS FROM NONLINEAR PROGRAMMING

Nonlinear programming theory investigates general procedures for optimising a nonlinear algebraic objective function, subject to restrictions on the values of the variables, expressed as nonlinear algebraic functions. Most work in this field falls into one of two categories: the study of optimality conditions and that of convergence properties. The former has to do with the characterization of solutions of nonlinear programming problems, in the form of necessary and sufficient conditions for a solution. These cannot provide solutions as such, but they do serve two purposes: to make recognizable an optimal solution, and to provide a goal for numerical techniques. Various sets of optimality conditions have been proposed [Ben Israel et.al. 1981], with some superceding others, and ranging in their generality. The best known, the Kuhn-Tucker conditions [Kuhn & Tucker 1951, Mangasarian 1969], will be stated here, with a quick view towards numerical techniques.

Convergence theory studies the ability of a solution process, or algorithm, to attain an optimal solution. The algorithms are usually iterative processes, due to the presence of nonlinearities in the functions of interest, and due to the necessity to search for the active constraints. Proper convergence rules ensure that as the process advances, the iterates tend towards an optimal solution. They produce new iterates which are better than their predecessors from which they are generated, in the sense defined by the objective function.

Usually in the optimization literature, convergence analysis is provided with new algorithms. There are however some general rules. The construction of the OPF algorithm presented in this thesis is based on such rules.

This appendix is divided into three parts. A first part quickly presents the nonlinear programming problem, using a compact notation, and gives two important remarks on limitations in the theory. In a second part, the optimality conditions are stated in two forms, the primal and the dual. A third part reviews briefly the notion of iterative algorithm and conditions for convergence.

A2.1.1 A nonlinear programming problem

A scalar objective function f(x) is minimized, subject to restrictions on the choice of values of x. A general form for a nonlinear minimization problem, denoted NLP, is written symbolically as

$$\min_{x} \; f(x) \quad \text{s.t.} \quad g(x) \le 0 \qquad \text{(NLP)}$$

where x is the vector of variables
f(x) is a scalar objective function
g(x) is a vector of equality and inequality constraints

A restriction usually placed on functions f and g in practice is that they be smooth [Avriel 1976]. That is due to differentiability requirements in many methods. In some methods, this requirement has been relaxed, allowing for continuous functions [Lemarechal & Mifflin 1979]. In those methods subgradients [Rockafellar 1970] replace gradients at points of nondifferentiability.

A restriction on the claim to optimality, for techniques based on the upcoming optimality conditions, is that solutions are only local optima. That is because these conditions are based on local information. Then global convergence eluded to in a later section is only towards a local optimum.

A2.1.2 Optimality Conditions in Nonlinear Programming

The necessary and sufficient conditions are presented, for minimization only, in nonlinear programming. The two well-known formulations of the optimality conditions, the primal and the dual, are considered. These are non-constructive conditions as such, but they suggest different solution strategies, which are introduced in Appendix 2.2.

a. The primal formulation of the optimality conditions

This is a general statement about the regions defined by functions f and g. Its formulation is as follows.

Let

x^* be a feasible point, i.e. $g(x^*) \le 0$.

F be a region of descent, constructed from x^*:

$F = \{x^* + \alpha d \mid f(x^* + \alpha d) \le f(x^*)$, for α sufficiently small $\}$

d is called a direction of descent

α is a step size

G be the feasible region, constructed from x^*:

$G = \{x^* + \alpha d \mid g(x^* + \alpha d) \le 0$, for α sufficiently small $\}$

d is called a feasible direction

Then the primal optimality condition states that a feasible point x^* is an optimal solution of problem NLP if and only if the intersection of sets F and G is empty [Ben Israel et.al. 1981]. This is a necessary and sufficient condition.

Stated in words, x^* is an optimal solution if and only if there exists no direction d emanating from x^* which is both a direction of descent and a feasible direction.

b. The dual formulation of the optimality conditions

The dual formulation can be generated from the primal, through the use of any one of the theorems of the alternative [Zlobec 1984, Bazaraa & Shetty 1979] and differential information . They impose conditions which are mathematically equivalent to the primal conditions, but in terms of an augmented set of variables (x,λ). The vector of λ, called Lagrange multipliers, is of the same dimension as $g(x)$. The dual formulation is as follows [Mangasarian 1969]:

A necessary condition for the point x^* to be an optimal solution of problem NLP is that there exist Lagrange multipliers λ^* satisfying these conditions:

1. The optimality condition.
   Define the Lagrangian function $\mathcal{L} = f(x) + \lambda^T g(x)$. Then the gradient of $\mathcal{L}$ with respect to x, evaluated at x^*, vanishes:
   $$\nabla_x \mathcal{L}(x^*) = 0$$
2. The feasibility conditions.
   $$g(x^*) \leq 0 \quad \text{(primal feasibility)}$$
   $$\lambda^* \geq 0 \quad \text{(dual feasibility)}$$
3. The complementary slackness condition.
   $$\lambda^*_i g_i(x^*) = 0$$

Stated in words, a necessary condition for x^* to be an optimal solution of problem NLP is that, at that point, the negative of the gradient of the objective function lies within the cone formed by the convex combination of active constraint gradients.

These are generally referred to as the first order Kuhn-Tucker optimality conditions. They are necessary but not sufficient conditions for a minimum. A sufficient condition for minimization is provided by second order information. It states that for the point x^* obtained from the first order conditions to be a minimum, the Hessian of the Lagrangian with respect to x evaluated at (x^*, λ^*) must be positive definite:

> For all nonzero vectors u, of dimension dim(x), belonging to the null space of the active constraint Jacobian, $\nabla_x g(x^*)$, we have
> $$u^T \nabla_{xx} \mathcal{L}(x^*, \lambda^*)\, u > 0$$

This is the second order Kuhn-Tucker sufficiency condition.

Limitations on problem NLP which exclude the use of the dual formulation of the Kuhn-Tucker conditions are called constraint qualifications. Several have been formulated, but possibly the most general and the simplest to verify is Slater's condition [Slater 1950, Zlobec 1984]. To validate the use of the Kuhn-Tucker conditions it requires that there exist a feasible point where all inequality constraints are strictly satisfied.

These are more restrictive optimality conditions than the primal conditions, since they require differentiability of the functions f and g. The advantage of these conditions is that the Lagrange multipliers provide a better assurance of optimality, compared to the converging sequence of the objective function in the primal formulation.

A2.1.3 Convergence of Nonlinear Programming Algorithms

Methods for solving nonlinear programming problems basically search out a solution described by the optimality conditions. They start with an estimate of the solution, or initial guess. Then, ideally as the process advances, new estimates get closer to the solution. A solution is reached when a sequence of estimates converges. In primal-based methods, values of the objective function usually form the sequence; in dual-based methods, it can be the objective or the solution estimates of the optimality equations.

The process described above is an iterative descent algorithm [Bazaraa & Shetty 1979, Luenberger 1984]. It is iterative, in that it generates a sequence of points, each new point being computed from its predecessor. Figure A2.1.1 illustrates the general structure of the iterative algorithm. The driving process A which links x_k to x_{k+1} can be quite general: a simple relationship, an equation, a set of equations, or even an algorithm in itself. It is a descent method in that values of the objective function associated with the points of the sequence are diminishing as the process advances.

If the algorithm can converge to the solution regardless of the initial guess, then it is said to be globally convergent. This is an essential property of any successful solution algorithm. An important feature of a globally convergent algorithm is that at each iteration, it narrows down the search for future iterates. That is illustrated in figure A2.1.2. The shrinking region retains the desirable property which drives the process; in this case, it delimits the region of lower values of the objective. Note that the shrinking region in the illustration need not only represent the system variables; it could represent also, for example, a choice of possible combinations of active sets, or other important considerations. Convergence of a process corresponds to narrowing down the search until only the solution remains. That might occur in a finite number of iterations, in which case the algorithm is said to terminate, or it might approach a solution at the limit.

General rules for global convergence have been proposed to guide in the construction of algorithms [Zoutendijk 1960, Wolfe 1969, Zangwill 1969, Polak 1971, Huard 1975, Meyer 1976] . Maybe the best known in optimization are those of Zangwill. His Convergence Theorem A serves as a basis for the convergence of the OPF algorithm developed in this thesis.

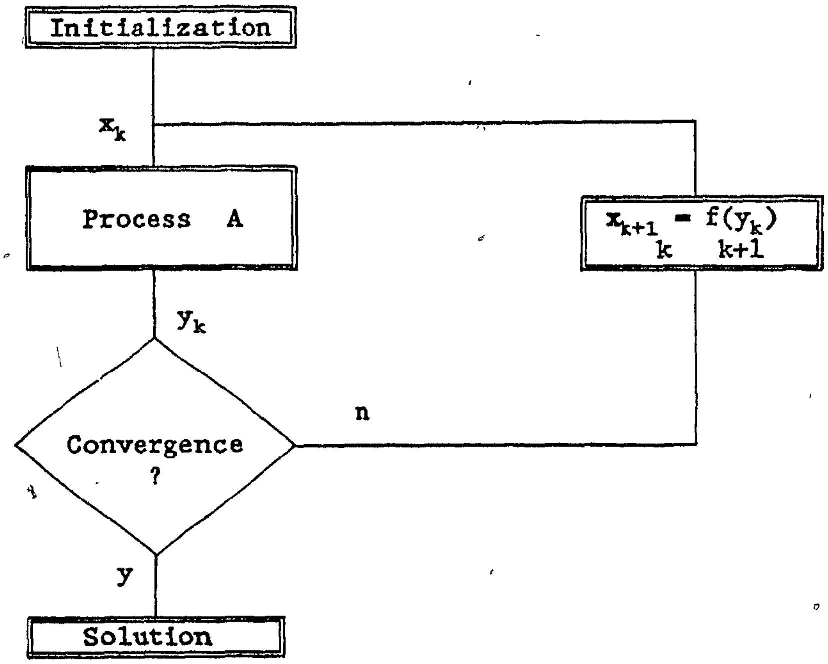


Figure A2.1.1. An iterative process.

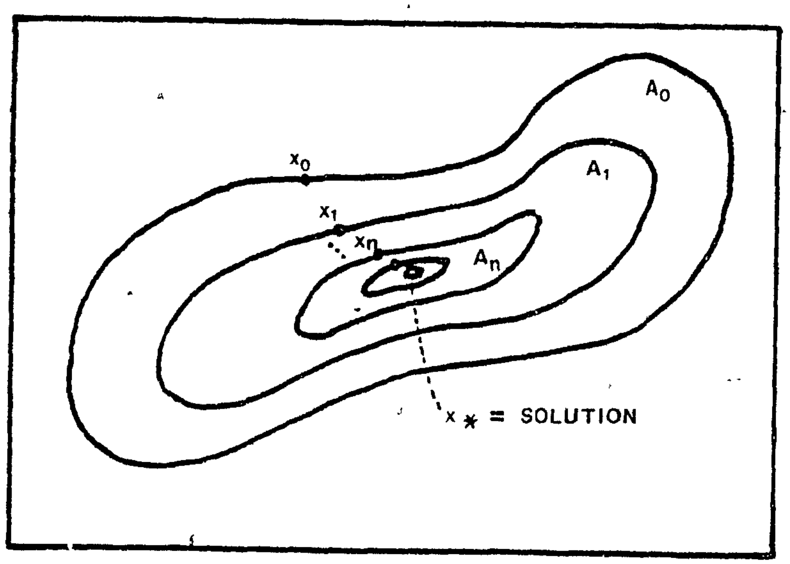


Figure A2.1.2. Search process of a globally convergent algorithm.

APPENDIX 2.2

A SURVEY OF STANDARD NONLINEAR PROGRAMMING METHODS USED IN OPF

The development of numerical algorithms for nonlinear programming has seen a steady flow of proposals, since its inception in the 1940's. There have emerged, however, a few well-delimited categories, based on the overall strategies of the methods. These are briefly presented in this section.

Although the basic ideas which define these categories differ, often the details in the implementation overlap. Hence, no classification structure can be suggested to separate all the methods. The only major partition for classification is based on the choice of formulation of the optimality conditions. As seen in Appendix 2.1, two formulations of the optimality conditions are prevalent, called the primal and the dual [Luenberger 1973, Ben Israel et.al 1981]. Numerical methods based on the primal approach act directly on the optimization; those of the dual act indirectly, by introducing the Lagrange multipliers.

Aside from general considerations, precise techniques which make up the algorithms can often be partitioned into two groups. For example:

- Search directions can be established using first order or second order approximations.
- Subproblems can be single-staged or multi-staged.
- Constraints are handled using exact or penalty techniques.
- The partition of independent variables can be fixed at the outset or updated to fit the problem.

These techniques are independent of each other, so that different combinations are possible. Numerical optimization methods can be characterized by the combination of these and other criteria.

A2.2.1 Primal-Based methods

a) Feasible Direction Methods

Algorithms based directly on the primal formulation were developed, for the general nonlinear problem, mostly in the 1950's-60's. They are called feasible direction methods [Zoutendijk 1960, Zangwill 1969].

A basic algorithm for these methods is as follows. A direction d emanating from a feasible point x_0 is computed from approximations of the objective function f(x) and the active constraints g(x). Then a step size α^* is computed to minimize an approximation of f(x) along direction d. If in the process of increasing α from 0 to α^* a constraint becomes active, at $\alpha=\alpha'$ say, then $x_0 + \alpha' d$ replaces x_0 as an estimate of the solution; if not, a full step with $\alpha=\alpha^*$ is taken. That completes one iteration. The process is repeated until the sequence of points thus generated converges.

The best known feasible direction methods for nonlinear programming are gradient methods [Rosen 1961, Wolfe 1967], and in particular, the reduced gradient method (GRG) [Abadie & Carpentier 1969, Lasdon & Warren 1978]. In all gradient methods, linear approximations of the objective and the constraints are used. In GRG, the constraints are handled exactly, by updating the independent/dependent partition of variables after each iteration. Variables at a limit are always made independent, so that they are expressed as simple bounds.

Special cases of feasible direction methods, developed in the late 1940's - 1950's, are linear programming (LP), for problems with functions f and g linear [Dantzig 1963], and early simplex-type techniques for quadratic programming (QP), with functions f quadratic and g linear [Boot 1964]. Both of these special cases avoid nonlinear equations in their solution process, so that they terminate when the right active set is found. This property makes them ideal as subproblems, used repeatedly to approximate nonlinear problems, in methods to be described later.

In linear programming, the edges of the feasible region serve as descent directions, leading from one vertex to another, always with a lower value of the objective function. This procedure, called the simplex method, reaches an optimum because for linear programs, the optimum is known to be at a vertex. This method is very fast; because of the linearities in all the functions and the very simple search direction, each iteration of the process requires little computation.

Primal and dual formulations of linear programming have been developed [Murty 1983]. The primal, as in other feasible direction methods, acts only on the actual problem variables. Because of the linearities, the dual is formulated exclusively in terms of the Lagrange multipliers. Numerical algorithms to solve either formulation are basically similar. In practice, the number of iterations to solve the problem is linked to the number of constraints. Hence, the dual is advantageous when the constraints in the primal outnumber the variables. In the dual, the roles are reversed.

The first methods of quadratic programming follow the same simplex-type approach [Beale 1959, Wolfe 1959]. These have been superceded, in recent works, by dual-based methods.

b) Successive Linear Programming

Successive linear programming (SLP) solves nonlinear programs by generating a sequence of linear programs. Initial conditions for a given linear program are provided by the solution of the previous linear program, according to some rules. The sequence of solutions to the linear programs should converge to the solution of the nonlinear problem.

This method was first proposed in the early 1960's [Griffith & Stuart 1961], but then attracted limited interest until the late seventies [Palacios-Gomez et.al. 1982]. Results from production codes during this period often suffered from unreliable performances. Developed by practitioners, the methods lacked the necessary theory for enforcing convergence. A major disadvantage is that solutions of the subproblems are situated at vertices of the feasible region, even though the optimum of the nonlinear problem need not be. More recent implementations suggest rules to enhance convergence by manipulating bounds of the subproblems [Lasdon 1985].

c) Penalty Function Methods

In these methods, no explicit search for active constraints is carried out. Instead, search directions are computed with penalty terms added to the objective function, taking into account the violated constraints. That reduces excursions outside the feasible region. Solution techniques for unconstrained optimization [Dennis & Schnabel 1983] can then be applied. After each solution of an unconstrained problem, coefficients of the penalty terms are updated, to better restrict the "illegal" excursions. That constitutes one iteration. An optimal solution is found when a sequence of values of the augmented objective function converges, and the constraint violations are deemed small enough. These are the easiest techniques to implement, but also the least reliable, because of constraint violations and because of unavoidable ill-conditioning [Gill et.al 1981].

The best known work on penalty methods is the Sequential Unconstrained Minimization Technique (SUMT) [Fiacco & McCormick 1968]. It proposes some interior (barrier) and exterior penalty functions. In OPF implementations, the most common is the quadratic exterior penalty function [Sasson 1969a, Housos & Irisarri 1982]. Solution of the ensuing unconstrained quadratic problem can be carried out in one of three ways:

- In Newton methods, the required gradient and Hessian are computed exactly [Avriel 1976].

- In Quasi-Newton methods (variable metric methods), the Hessian is first approximated, and then updated using incoming information [Avriel 1976]. This is useful for small problems with a dense Hessian.

- In conjugate gradient methods, the optimality equations are solved by this iterative solver. This is advantageous for very large problems, because the equations are handled one by one.

Pure penalty methods, such as SUMT, have been proposed sporadically for OPF. However, many proposed methods apply penalty functions to some constraints. In many cases, the independent variables are handled with exact bounds, while dependent constraint violations are added to the objective function via penalty functions.

A2.2.2 Dual-Based Methods

The development of numerical optimization techniques over the past twenty years or so has concentrated, for the most part, on solving the dual formulation. The optimal solution is characterized by a set of nonlinear equations in (x,λ). Solution techniques can be placed roughly into two groups:

- Those which apply primal methods directly to solve the Kuhn-Tucker optimality conditions.

- Those which exploit the properties of the optimality conditions, and especially the Lagrange multipliers.

In OPF implementations, gradient or mixed gradient-penalty methods make up most of the first group. In the second group, dual-based quadratic programming and successive quadratic programming will be considered.

a) Quadratic Programming Based on the Kuhn-Tucker Conditions

Recent methods efficiently solve the Kuhn-Tucker conditions for a quadratic program. Using the active set defined by the estimate of the optimum, the first order optimality conditions are generated. They form a set of linear equations in the variables and the Lagrange multipliers. Different strategies exist [Van de Panne 1975, Gill et.al 1981] to compute the solution. If the solution is completely feasible, it is the optimal solution; if not the active set is updated and the process is repeated. The process terminates with the true active set and the optimum.

Quadratic programming offers some of the simplicity of linear programming in manipulating linear constraints, and a greater precision in pinpointing the optimum. For this reason, successive quadratic programming has become more popular than successive linear programming as a method for solving nonlinear programs.

b) Successive Quadratic Programming

These are also known as recursive quadratic programming or Lagrange-Newton methods. Nonlinear programs can be solved by generating a sequence of quadratic programming approximations, the solutions of which converge to that of the nonlinear problem. Much theoretical work was produced from the late 1960's to the mid 70's [Biggs 1972 & 1975, Fletcher 1973 & 1975, Han 1977, Powell 1978]. They seek the best choice of subproblem and rules for enforcing convergence.

At the optimum, the subproblem and the original problem share the same optimality conditions. In all the proposed subproblems, linearized constraints replace the nonlinear constraints and all the bounds are retained. Different quadratic objective functions are suggested:

- A quadratic approximation of the objective function. This case is called the "Newton strategy" by Murtaugh and Saunders [Murtaugh & Saunders 1980]. It is seldom mentioned in the theory, due to problems with limited convergence. This classification is convenient though, since many OPF implementations prior to the popularization of SQP are of this type.

- A quadratic approximation of the Lagrangian or of a modified Lagrangian [Murtaugh & Saunders 1980]. The extra terms, linear in the Lagrange multipliers and in the active constraints, are called exact penalty functions because their effect disappears at the optimum. They monitor the curvature of the constraints to ensure descent (for minimization). This is the usual formulation in these methods.

- An augmented quadratic approximation of the Lagrangian or modified Lagrangian [Murtaugh & Saunders 1982]. An added exact penalty term with adjustable coefficients, quadratic in the active constraints, discourages excursions from the nonlinear feasible region.

The methods corresponding to the last two objectives are called projected Lagrangian methods. These methods are not restricted to SQP, although that now seems to be the trend. Earlier projected Lagrangian methods [Murtaugh & Saunders 1978] solved subproblems with general objectives and linearized constraints using reduced gradient methods.

Convergence properties have been verified, as in Newton methods, when the initial guess is close "enough" to the solution. Techniques to enhance convergence are:

- The use of a step size along a search vector linking the subproblem's expansion point and its solution [Han 1977]. This approach is used in the algorithm proposed in this thesis.

- The adjustment of penalty coefficients, in the augmented Lagrangian objective [Murtaugh & Saunders 1982].

Note that reference to augmented Lagrangian objectives here should not be confused with the augmented Lagrangian method [Pierre & Lowe 1975]. The latter is much like the penalty methods, but in which only exact penalty functions are deployed. The method was developed more or less in parallel with projected Lagrangian methods. So far it has not been used in OPF.

A2.2.3 Parametric Programming

Parametric linear programming [Murty 1983, Gal 1984] and parametric quadratic programming [Houtthaker 1960, Van de Panne 1975] have been available almost as long as LP and QP, although they have only recently been applied in power systems. Applications of the continuation method in optimization falls into these categories. Given an optimal solution, these methods efficiently track the optimal solution trajectories, following changes in some system parameters. Early works [Gass & Saaty 1955] limited themselves to a single "region of stability", where the active set remains constant. Methods were soon developed, for LP and QP, to update the active set and pursue solutions over a wide range of active sets.

Algorithms have been proposed for general nonlinear programs [Hackl 1978, Gfrerer et. al. 1983, Guddat 1984]. This is an active area for research, since many different algorithmic structures have yet to be tried. The major difficulties are the efficient handling of nonlinearity and the detection of changes in the active set. In this thesis, the solution of a nonlinear program is proposed by SQP, using, amongst other things, the continuation method to solve the subproblems.

A2.2.4 Present Trends in Optimization

Here we present the opinions stated in some recent review papers and textbooks on nonlinear programming methods [Fletcher 1982, Bartholomew-Biggs 1982, Lasdon 1982, Gal 1984, Lasdon 1985, Scales 1985].

Successive quadratic programming - projected Lagrangian methods now seem to be the most popular. Their advantages are that they are numerically efficient, converge quickly near the optimum, require fewer function and gradient evaluations, and they need not satisfy equality constraints at each iteration. A major disadvantage is that global convergence is uncertain when weak restrictions are placed on constraint violations.

Successive linear programming is advantageous for problems with linear or near linear constraints, and with solutions at a vertex. However, convergence can be slow due to inherently inefficient convergence control mechanisms. This method is very popular in the petrochemical industry, but has not caught on so much in power systems.

Reduced gradient methods are efficient for linear or near linear constraints, are generally very reliable for any problem, and are generally slow. Two reasons for the slowness are that at each iteration the nonlinear information is updated and feasibility of the operating point is maintained. That requires much computation.

Penalty methods are simple to implement, but are plagued with numerical difficulties, and do not assure a feasible solution. For these reasons, these methods have attracted little interest of late.

Parametric programming is still relatively unknown to practitioners, although it has received in recent years much attention in the applied optimization literature [Fiacco 1982, Eaves 1983, Fiacco 1984]. Recent developments in OPF using parametric programming look promising; they are discribed in Chapter 2.

New numerical techniques have been suggested to enhance reliability or to increase speed. They could be added to most of the previous algorithms. Some of these are:

- Scaling the variables to reduce ill-conditioning of linear computations [Gill et.al. 1981].

- The use of trust regions [Sorenson 1982, Dennis & Schnabel 1983]. These are simple constraints (a box or hypersphere) added to discourage large excursions from the nonlinear feasible region. The size of the region can be adjusted at each iteration.

- The relaxing of constraints to solve subproblems which would otherwise be infeasible.

- The use of truncated computations, to speed up computation [Dembo & Tulowitzki 1984, Nash 1984].

The first three techniques have made their way into some QPF algorithms.

APPENDIX 2.3

LISTING OF PUBLICATIONS IN THE OPF LITERATURE

Publications are listed for each branch of the OPF literature, as depicted in fig. 2.1. Superscripts indicate cross-referencing to other branches; this practice has been kept to a minimum. An added listing, not given its own branch in Chapter 2, is redispatching.

A - ECONOMIC DISPATCH BY INCREMENTAL LOADING

Estrada(1930)	Hahn(1931)	Steinberg(1933)
Steinberg(1934)	Steinberg(1943)	

B - ECONOMIC DISPATCH - CLASSICAL EICC/LOSS FORMULAE

George(1943)	George(1949)	Ward(1950 a,b)
Kirchmayer(1951)	Glimn(1952)*A	Hale(1952)
Kirchmayer(1952)	Ward(1953)	Brownlee(1954)
Glimn(1954)	Harder(1954)	Travers(1954)
Cahn(1955)	Early(1955)	Shipley(1956)
Kirchmayer(1958)	Lubisich(1958)	George(1959)
Watson(1959)	Fischer(1960)	Schmidt(1960)
Blodgett(1962)	Moskalev(1963)	Tudor(1963)
Van Ness(1963)	Walker(1963)	Anstine(1964)
Happ(1964b)	Happ(1967)	Long(1967)
Roth(1967)	Hill(1968a,b)	Akhtar(1969)
Ariatti(1969)	Happ(1969a,b)	Meyer(1969)
Mikami(1970)	Olesnicky(1970)	Gungor(1971)
Meyer(1971)	Dension(1973)	Podmore(1973)
Happ(1974)	Jain(1975)	Wahda(1976)
Adler(1977)	Shoults(1977)*K	Alvarado(1978)
El-Hawary(1978)	Galiana(1978)	Malik(1978)
Mamandur(1978)	Selmyen(1978)	Galiana(1979)
Nanda(1979)	Shoults(1979)	Vojdani(1979)
Vojdani(1981)	Isoda(1982)	Glavitsch(1983)
Krogh(1983a)	Aoki(1984)	Lin(1984)

Mansour(1984) Wenyuan(1985)*Q Xu(1985)
Boming(1986)

C - CLASSICAL EICC/INTERCONNECTED SYSTEMS

Glimn(1952)*B Glimn(1958) Kerr(1959)
Kichmayer(1959) Miller(1959) Aldrich(1971a,b)
Gladys(1971a,b) Happ(1971) Cameron(1974)
Happ(1975b) Jamshidian(1983)

D - OTHER TECHNIQUES/INTERCONNECTED SYSTEMS

Peschon(1972a) Deo(1973) Kwatny(1973)
Spare(1975) Romano(1981)

E - VALVE POINT LOADING

Decker(1958) Hayward(1961) Light(1962)
Happ(1963) Ringlee(1963)*T Fink(1969)
Vojdani(1982)

F - ENVIRONMENTAL DISPATCH

Friedlander(1970) Gent(1971) Sullivan(1972a)
Friedman(1973) Lamont(1973) Delson(1974)
Ferrer(1974) Finnegan(1974) Cadogan(1975)
Dejax(1975) Eisenberg(1975) Lamont(1975)
Ruane(1975) Schweitzer(1975) Sullivan(1975)
Zahavi(1975) Kothari(1976) Cadogan(1977)

G - ECONOMIC DISPATCH BY LAGRANGIAN TECHNIQUES

Squires(1961)

H - KT CONDITIONS - SUCCESSIVE APPROXIMATION SOLVER

Carpentier(1962) Carpentier(1963)

I - GRADIENT METHODS

Fukada(1964) Krumm(1965) Gamn(1967)
Carpentier(1968) Dommel(1968) Dhar(1971)
Peschon(1971) Carpentier(1972) Velghe(1972)*M

Carpentier(1973a)	Dhar(1973)	Ramamoorty(1973)
Alsac(1974)	Dayal(1974)	Mukherjee(1974)
Podmore(1974)[*Q]	Rashed(1974)	Alsac(1975)
Sachdev(1975)[*Y]	Barcelo(1977)	Bala(1978)
Ilic(1979)	Prada(1979)	Wu(1979)
Burchett(1980)	Burchett(1981)	Roy(1983)
Landquist(1984)	Backlund(1986)	Cernic(1986)

J - SUCCESSIVE LINEAR PROGRAMMING

Farrara(1969)	Abou Taleb(1974)	Pai(1975)
Megahed(1977)	Khan(1979b)	Mamandur(1982)
Stott(1983)	Peralta(1984)	PCA(1985)[*L]
Van Meeteren(1986)[*L]		

K - SUCCESSIVE QUADRATIC PROGRAMMING - NEWTON STRATEGY

Peschon(1968a)[*T]	El-Abiad(1969)	Shen(1969)
Jaimes(1970)	Nabona(1973)[*Q]	Nicholson(1973)[*Q,T]
Suhakar(1974)	Dillon(1975)	Wadhwa(1975a,b)
Shoults(1977)[*B]	Dillon(1981)[*R]	Talukdar(1981)
Talukdar(1982)	Contaxis(1986)[*O,Q]	Maria(1986)[*O,Q]

L - SUCCESSIVE QUADRATIC PROGRAMMING - PROJECTED LAGRANGIAN

Biggs(1977)	Lipowski(1981)	Aoki(1982)
Burchett(1982a,b)	Burchett(1984)	Sun(1984)
PCA(1985)[*J]	Van Meeteren(1986)[*J]	

M - PQ DECOMPOSITION

Dopazo(1967)	Norimatsu(1967)	Adielson(1972)[*U]
Billinton(1972)[*T]	Jolissaint(1972)	Velghe(1972)[*I]
Billinton(1973)	Sjelvgren(1975)	Shoults(1981)
Chamorel(1982)	Shoults(1982)	Contaxis(1983)[*Q,U]
Housos(1983)[*N]	Talukdar(1983)	K.Lee(1984)
Lee(1985)	Carpentier(1986)	

N - SUMT

Sasson(1969a,b,c)	Dillon(1970)	Ramamoorty(1970)

Sasson(1971)	Dillon(1972)	Sasson(1973)
Kohli(1975)	Fischl(1978)	Divi(1982)
Housos(1982)	Housos(1983)[*M]	

O - ECONOMIC DISPATCH BY LINEAR PROGRAMMING

Benthall(1968)	Wells(1968)	Taylor(1969)
Dodu(1970)	Shen(1970)	Thanikachalam(1971)
Brewer(1972)	Merlin(1972)	Duran(1973)
Nanda(1973)	Nanda(1974)	Wollenberg(1974)[*Q]
Dodu(1975)	Khan(1975)	Saeed(1976)
Hobson(1977)	Pai(1977)[*U]	Stott(1978)
Grigsby(1979)	Stott(1979a)	Elaqua(1982)
Fox(1982)[*A']	Irving(1983)	Farghal(1984)
Mota-Palomino(1984)	Zhang(1984)[*Y]	Van den Bosch(1985a)
Broussole(1986)	Contaxis(1986)[*K,Q]	Maria(1986)[*K,Q]

P - ECONOMIC DISPATCH BY NETWORK TECHNIQUES

Lee(1980)	Lee(1981)	Hobson(1984)
Luo(1984)	Hon(1986)	

Q - ECONOMIC DISPATCH BY QUADRATIC PROGRAMMING

Nabona(1973)[*K]	Nicholson(1973)[*K,V]	Reid(1973)
Podmore(1974)[*I]	Wollenberg(1974)[*O]	Dayal(1976)
Lugtu(1979)	Bottero(1982)	Quintana(1982)
Contaxis(1983)[*K,U]	Wenyuan(1985)[*B]	Contaxis(1986)[*K,O]
Maria(1986)[*K,O]		

R - ECONOMIC DISPATCH USING PARAMETRIC PROGRAMMING

Dillon(1981)[*K]	Blanchon(1983)[*U]	Vojdani(1983)
Galiana(1983)	Blanchon(1984)[*U]	Carpentier(1984)
Huneault(1984)	Huneault(1985)	Innorta(1985)
Innorta(1987)[*T]		

S - ECONOMIC DISPATCH BY DYNAMIC PROGRAMMING

Fukao(1959)	Ringlee(1963)[*E]	

T - DYNAMIC DISPATCH

Cuenod(1966) Bechert(1972) Patton(1973)
Ross(1980) Lim(1985a,b) Van den Bosch(1985b)
Innorta(1987)*R

U - REACTIVE-VOLTAGE DISPATCHING,OTHER THAN V,W

Sullivan(1969) Kishore(1971) Sullivan(1972b)
Graf(1974) El-Shibini(1975) Pai(1977)*O
Wirgau(1979b) Blanchon(1983)*R Franchi(1983)
Elfstrom(1983) Blanchon(1984)*R Chamorel(1984)
Bright(1986) Granville(1986) Mota-Palomino(1986)
Padiyar(1986) Zhang(1986a,b)

V - REACTIVE-VOLTAGE DISPATCHING THROUGH MIN. LOSS

Smith(1963) Hano(1968) Kumai(1968)
Peschon(1968)*K Hano(1969) Bokay(1970)
Narita(1971) Billinton(1972)*M Nicholson(1973)*J,Q
Savulescu(1976) Mamandur(1981) Elangovan(1983)
Franchi(1983)*U Doi(1984) Horton(1984)
El-Kady(1986)*Z

W - REACTIVE-VOLTAGE DISPATCHING THROUGH MIN. SLACK GENERATION

Adielson(1972)*M Fernandes(1978a,b,c) Wirgau(1979a)
Aldrich(1980) Happ(1981) Contaxis(1983)*M,Q
Ramalyer(1983)

X - MINIMUM LOSS

Calvert(1958) Sze(1959) Sasson(1969b,c)*N

Y - MINIMUM DEVIATIONS

Kaltenbach(1971) Daniels(1972) Shoults(1977)
Khan(1979a) Hobson(1980) Krogh(1983b)*A'
Zhang(1984)*O

Z - MINIMUM OVERLOADS

Sekine(1972) Sachdev(1975)*I,N Shoults(1976)

A' - MINIMUM LOAD SHEDDING

Hajdu(1968)	Subramanial(1971)	Song(1975)
Ejebe(1977)	Ghoneim(1977)	Khan(1978)
Chan(1979a,b)	Medicherla(1979)	Rashed(1979)
Medicherla(1981)	Palaniswamy(1981)	Fox(1982)*O
Krogh(1983b)*Y	Finlay(1985)	Palaniswamy(1985)
Fox(1986)		

B' - MAXIMUM LOAD,LOADABILITY REGION

Garver(1979)	Dersin(1982)

RESCHEDULING

Peschon(1968b)	Kaltenbach(1971)*Y	Thanikachalam(1971)
Cory(1972)	Glavitsch(1973)	Sachdev(1975)*I,N,Z
Shoults(1976)*Z	Pai(1977)*O,U	Mamandur(1978)*B
Medicherla(1979)*A'	Stott(1979b)	Dillon(1981)*K,R
Medicherla(1981)*A'	Palaniswamy(1981)*A'	Somuah(1981)
Bui(1982)	Elfstrom(1983)*U	Krogh(1983b)*Y,A'
Meliopoulos(1983)	Chandrashekar(1985)	Zaborsky(1985)
Hon(1986)*P	Monticelli(1986)	

APPENDIX 2.4

ENUMERATION OF PROBLEMS AND SOLUTION TECHNIQUES FOR OPF

This is an exhaustive list of the elements which make up the problems of OPF and its subsets, as well the elements of the solution techniques.

A - TASK

1. Economic Dispatch
2. Minimum Loss
3. Minimum Reactive Power
4. Minimum Violations
5. Minimum Deviation
6. Minimum Load Shedding
7. Maximum Load

B - PROBLEM FORMULATION STRUCTURE

1. Full OPF + Tracking of Input Parameters
2. Full OPF + Dynamic Constraints
3. Full OPF
4. Nonlinear OPF, No Dependent Injection

---

5. Security Dispatch + Tracking of Input Parameters
6. Security Dispatch + Dynamic Constraints
7. Full Objective, Linear Constraints
8. Full Objective, Linear Constraints, No Dependent Injection
9. Linear Objective, Linear Constraints
10. Linear Objective, Linear Constraints, No Dependent Injection

---

11. Full Objective, DC Load Flow + Tracking of Input Parameters
12. Full Objective, DC Load Flow + Dynamic Constraints
13. Full Objective, DC Loadflow
14. Full Objective, DC Loadflow, No Dependent Injection

---

15. Equal Incremental Cost Criterion + Tracking Input Parameters
16. EICC + Dynamic Constraints
17. EICC With Static Loss Model
    a. Quadratic in Independent Variables
    b. Quadratic in P and Q
    c. Quadratic in P
    d. Linear in Independent Variables
    e. Linear in P and Q
    f. Linear in P
18. EICC With Dynamic Loss Model
    a. Linear in Independent Variables
    b. Linear in P and Q
    c. Linear in P
19. Lossless Power Balance (Incremental Loading)

C - COORDINATES

1. Polar Coordinates
2. Rectangular Coordinates

D - CHOICE OF VARIABLES

1. Complete set
2. Injections
    a. Real and Reactive Power Generations, Line Flows
    b. Real and Reactive Power Generations
    c. Real Power Generations, Line Flows
    d. Real Power Generations
    e. Reactive Power Generations, Line Flows
    f. Reactive Power Generations
3. States
    a. Voltage Magnitude and Phase Angles (or rectangular coord.) + i. - vi.
    b. Voltage Magnitudes + i. - vi.
    c. Voltage Phase Angles + i. - vi.

The items i. to vi. refer to the types of equipment listed on the next page.

i. Taps, Shifters, Shunts
ii Taps, Shunts
iii. Shifters
iv. Taps
v. Shunts
vi. Other Combination

E - SET OF INDEPENDENT VARIABLES

1. Fixed Partition
   a. States
   b. Injections
   c. Bus Voltages, Line Currents
   d. P-V at generation busses, P-Q at load buses, V at slack bus
   e. Other
2. Dynamic Partition

F - OUTPUT OF SUBPROBLEM

1. Solution to an LP or QP, All Variables of Interest Included
2. Solution to an LP or QP, Not All Variables of Interest Included
3. Search Direction and Step Size
4. Search Direction

G - CHOICE OF SUBPROBLEM STRUCTURE

1. Linear
   a. Gradient Methods
   b. Linear Programming
   c. Projected Lagrangian using gradient method
2. Quadratic
   a. Quadratic Programming
   b. Successive Quadratic Programming
      i. Newton strategy
      ii. Projected Lagrangian
3. SUMT, followed by particular techniques for solution
4. Real - Reactive Decomposition, followed by particular structures for the subproblems

H - DETAILS OF SUBPROBLEM STRUCTURE

1. Objective
   a. Nonlinear in Injections
   b. Full Quadratic in Injections
   c. Full Linear in Injections
   d. Quadratic Approximation in States
   e. Augmented Quadratic Approximation in States
      i. SUMT
      ii. Augmented or Projected Lagrangian
   f. Linear Approximation in States
2. Load Flow Constraint Linearization
   a. All Injections vs. States, Jacobian Matrix
   b. All Injections vs. States, Approximate Jacobian
   c. Independent Injections vs. States, Jacobian Matrix
   d. Independent Injections vs. States, Approximate Jacobian
   e. Real Power Generations and Line Flows vs. Phase Angles (and Shifters), Jacobian Matrix
   f. Real Power Generations and Line Flows vs. Phase Angles (and Shifters), DC Load Flow
   g. Reactive Power Generations vs. Voltage Magnitudes (,Taps,and Shunts), Jacobian
   h. Real Power Generations vs. Phase Angles, Jacobian
   i. Real Power Generations vs. Phase Angles, DC Load Flow
   j. Lossy Power Balance
   l. Lossless Power Balance
3. Bounds on States
   a. Contain Phase Angles
   b. Do Not Contain Phase Angles
   c. No Bound on States
4. Reserve Constraints
   a. Present
      i. Static
      ii. Dynamic
   b. Absent

5. Ramp Constraints
    a. Present
        i. Static
        ii. Dynamic
    b. Absent
6. Frequency Constraints
    a. Present
        i. Expressed as a function of real power
        ii. Expressed as such in a control loop
    b. Absent
7. Environmental constraints
    a. Present
    b. Absent

I - SOLUTION TECHNIQUES FOR SUBPROBLEM

1. Newton Method
2. Quasi - Newton
    a. BFGS Update
    b. Fletcher-Powell
    c. Davidon-Fletcher-Powell
    d. Han-Powell
3. Gradient, Reduced Gradient
4. Conjugate Gradient
5. Linear Programming
    i. Simplex
    ii. Dual - Simplex
    iii. Dantzig - Wolfe
    iv. Parametric LP, Continuation Method (see 8)
6. Quadratic Programming
    i. Beale
    ii. Wolfe
    iii. Dantzig - Wolfe
    iv. Thiel Van-de-Panne
    v. Gill-Murray-Wright
    vi. Parametric QP, Continuation Method (see 8)
7. Network Techniques

8. Continuation Method
    a. Varying Load
    b. Varying Limit
9. Integer Programming
10. Dynamic Programming

J - EQUALITY CONSTRAINT STRUCTURE

1. Equality Constraints Present, Including Load Flow Equations
2. Active Set Method With (Generalized) Power Balance Equation

K - HANDLING OF CONSTRAINTS

1. Independent Constraints
    a. Lagrange Multipliers
    b. Penalty Functions
2. Dependent Variables
    a. Lagrange Multipliers
    b. Penalty Functions
3. Primal Tableau-Type Method

L - POSITION OF DEPENDENT CONSTRAINTS

1. Inside Subproblem
2. Outside Subproblem

M - CONVERGENCE CRITERION

1. For Subproblems
    a. Termination
    b. Convergence of Sequence
2. For Nonlinear Problem
    a. Closeness to Solution of Optimality Conditions
    b. Convergence of Sequence
    c. All Constraints Satisfied

N - ITERATION RULES FOR NONLINEAR PROBLEM

1. Choice of Variables to Keep
    a. States ,i. or ii.
    b. Independent Variables ,i. or ii.

i. Steplength used

*. Adequate decrease

**. Optimal Decrease

ii. Steplength Not Used

2. Feasibility maintained at each iteration?

a. yes

b. no

3. Position of Nonlinear Iteration

a. After a Single-Stage Subproblem

b. After a Multistaged Subproblem

APPENDIX 3.1

THE LOAD FLOW EQUATIONS: FORMULATION AND USEFUL PROPERTIES

A3.1.1 Formulation of the Load Flow Equations

Consider a bus i of a power system connected to adjacent buses j through transmission lines. Each line is modelled in steady-state as a pi circuit. Shunt branches have identical imaginary admittances of $\underline{y}_{ij} = j\underline{b}_{ij}$ and the series branch has an admittance of y_{ij}. A shunt compensation device with imaginary admittance $y_{ci} = jb_{ci}$ can be connected to bus i. Tap and phase shifting transformers can be seen in analysis as particular cases of a transformer with a complex tap ratio, denoted a_{ij}. The tap changing transformer exhibits a variable modulus t and fixed zero phase angle, while the phase shifting transformer maintains a fixed unit modulus and has a variable phase angle ϕ.

The load flow equations needed in this work express real and reactive power and line current injections (P,Q,J_t) versus the complex bus voltage components (V,δ) and passive network controls (b_c,t,ϕ). Expressions for the first two types of injections can be found in most textbooks on power systems, but they are rarely given for the line current injection. Expressions for the three types of injections are given below.

The real and reactive power injections are components of the complex apparent power S. This injection is the sum of the apparent powers sent to the transmission lines, plus the reactive compensation. For bus i, it is written

$$S_i = v_{i\ bus}\,[I_{i\ comp.}]^* + \sum_{j\epsilon Ai} v_{i\ line}\,[\,I_{ij\ line} + I_{ij\ shunt}]^* \qquad (A3.1.1)$$

where v_i is the complex bus voltage

$I_{i\ comp}$ is the shunt compensation current

$I_{ij\ line}$ and $_{shunt}$ are the line series and shunt branch currents

$*$ indicates the complex conjugate.

The set A_i contains the indices of buses adjacent to i. Replacing line voltages by bus voltages, the appropriate expression for S_i is developed:

$$S_i = v_i\,[v_i y_{ci}]^* + \sum_{j\epsilon Ai} a_{ij} v_i\,[(a_{ij}v_i - a_{ji}v_j)y_{ij} + v_i \underline{y}_{ij})]^* \tag{A3.1.2.a}$$

$$= -jV_i^2 b_{ci} + \sum_{j\epsilon Ai} (|a_{ij}V_i|^2 (y_{ij} + \underline{y}_{ij})^* - a_{ij}a_{ji}^* V_i V_j^* y_{ij}^*) \tag{A3.1.2.b}$$

In eq. A3.1.2.a, the term between brackets in the summation is the line current I_{ij}. The line current injection is taken to be $J_{t.}i_j = I_{ij}(I_{ij})^*$. The three required injections can then be written:

$$P_i = Re(S_i) = \sum_{j\epsilon Ai} \{ |a_{ij}V_i|^2 |y_{ij}| \cos(\Omega_{ij}) - K_{ij}\cos(\Phi_{ij}) \} \tag{A3.1.3}$$

$$Q_i = Im(S_i) = -V_i^2 b_{ci} - \sum_{j\epsilon Ai} \{ |a_{ij}V_i|^2 (|y_{ij}|\sin(\Omega_{ij}) + \underline{b}_{ij}) + K_{ij}\sin(\Phi_{ij}) \} \tag{A3.1.4}$$

$$J_{t.ij} = \{ |a_{ij}V_i|^2 |y^+_{ij}|^2 + |a_{ji}V_j|^2 |y_{ij}|^2 - 2K_{ij}|y^+_{ij}|\cos(\Phi_{ij} + \Omega^+_{ij}) \} \tag{A3.1.5}$$

where $K_{ij} = |a_{ij}|\ |a_{ji}|\ |y_{ij}|\ V_i\ V_j$ (A3.1.6.a)

$\Phi_{ij} = (\delta_i - \delta_j) + (\phi_{ij} - \phi_{ji}) - \Omega_{ij}$ (A3.1.6.b) and

Ω_{ij} is the phase angle of the series admittance y_{ij}.

$|y^+|$ and Ω^+ are the modulus and phase angle of the sum of the line series and shunt admittances.

δ_i and δ_j are voltage phase angles at buses i and j.

ϕ_{ij} and ϕ_{ji} are phase shifter angles on line ij placed at buses i and j respectively.

If passive network controls are absent from a bus or from a line, then default values are inserted in the load flow equations. The taps take on values of one in the scaled per unit system, and shifters and shunt compensation admittance take on values of zero.

A3.1.2 Useful Properties of the Load Flow Equations

Topological properties of the load flow equations useful for optimization will be stated in this section. These are basic qualities of the equations, and are invariant under coordinate transformations. Definitions of terms used here can be found in [Chillingsworth 1976] or [Bazaraa & Shetty 1979].

Properties of the function

- The load flow equations define a continuous manifold.
- The feasible region of the optimization is the intersection of the load flow manifold and the inequality-feasible hyperbox. It forms a closed and bounded (hence compact) set. That is important in convergence theorems.
- The feasible region is assumed connected. Then the optimal solution need not be computed on many different (as yet undetected) segments of the manifold.
- The feasible region is not convex in general. Locally convex regions would be of great interest though. Each one of these regions is a connected set where the Hessian of the load flow equations is positive definite. It is known that the flat voltage profile is situated in a locally convex region [Galiana & Banakar 1982].

Properties of the derivative

- The derivative of the load flow equations is continuous. The load flow Jacobian is the derivative of the injections expressed in terms of the states.
- The load flow equations are differentiable "almost everywhere", in the sense of Sard's theorem [Milnor 1965]. That means that regions where the Jacobian loses full rank are of lower dimension than the load flow manifold. That makes those regions very "small".
- Differentiability is in doubt when the Jacobian loses full rank. However, singular points of the Jacobian are non-degenerate, in the sense of the Morse theorem [Chillingsworth 1976]. That means that the singular points are differentiable, in a topological sense, and derivatives can be computed in a different coordinate system. Along with the continuity property, that makes the load flow manifold smooth everywhere.
- Derivatives are computed in coordinate systems other than the states using the implicit function theorem and the chain rule [Spivak 1965].

The dimension of the load flow manifold

The dimension of the load flow manifold, denoted dim(m), is one less than the number of states. The load flow equations admit a null space of dimension one. That is easily seen in polar coordinates, where all the injections are functions of voltage phase angle differences, but not of any angle alone. To avoid carrying the null space solutions, one phase angle is designated reference angle and set to zero.

The dimension of the manifold is of practical importance because it is also the dimension of the basis in the subproblem. Most but not all combinations of dim(m) independent variables form a basis. More on infeasible bases is given in section 3.4.1.

APPENDIX 3.2

FORMULATION OF THE LOAD FLOW JACOBIAN

The load flow Jacobian, denoted J, is the derivative of the vector of injections with respect to the vector of states. It is a matrix with as many rows as there are injections and as many columns as there are states. Element J_{ij} is the derivative of the i^{th} injection versus the j^{th} state. Here are the expressions of the components of the Jacobian:

- w.r.t. the near bus (i) and far bus (j) voltage magnitudes:

$$\frac{\partial P_i}{\partial V_i} = \frac{P_i}{V_i} + V_i |a_{ij}|^2 |y_{ij}| \cos \Omega_{ij} \quad (a); \qquad \frac{\partial P_i}{\partial V_j} = -\frac{K_{ij}\cos(\Phi_{ij})}{V_j} \quad (b)$$

$$\frac{\partial Q_i}{\partial V_i} = \frac{Q_i}{V_i} - V_i |a_{ij}|^2 |y_{ij}| \sin \Omega_{ij} \quad (c); \qquad \frac{\partial Q_i}{\partial V_j} = -\frac{K_{ij}\sin(\Phi_{ij})}{V_j} \quad (d)$$

$$\frac{\partial J_{ij}}{\partial V_i} = \frac{2}{V_i}\left[J_{ij} - |a_{ji}|^2 V_j^2 |y_{ij}|\right] \quad (e)$$

$$\frac{\partial J_{ij}}{\partial V_j} = \frac{2}{V_j}\left[J_{ij} - |a_{ij}|^2 V_i^2 |y_{ij}|\right] \quad (f)$$

(A3.2.1)

- w.r.t. the near bus (i) and far bus (j) voltage phase angles:

$$\frac{\partial P_i}{\partial \delta_i} = K_{ij} \sin(\Phi_{ij}) \quad (a); \qquad \frac{\partial P_i}{\partial \delta_j} = -K_{ij} \sin(\Phi_{ij}) \quad (b)$$

$$\frac{\partial Q_i}{\partial \delta_i} = -K_{ij} \cos(\Phi_{ij}) \quad (c); \qquad \frac{\partial Q_i}{\partial \delta_j} = K_{ij} \cos(\Phi_{ij}) \quad (d)$$

$$\frac{\partial J_{ij}}{\partial \delta_i} = K_{ij}\, y^{+} \sin(\Phi_{ij} + \Omega^{+}_{ij}) \quad (e); \qquad \frac{\partial J_{ij}}{\partial \delta_j} = -\frac{\partial J_{ij}}{\partial \delta_i} \quad (f)$$

(A3.2.2)

- w.r.t. the near (ij) and far (ji) variable taps:

$$\frac{\partial P_i}{\partial a_{ij}} = \frac{P_i}{a_{ij}} + |a_{ij}||y_{ij}|V_i^2\cos\Omega_{ij} \quad (a); \qquad \frac{\partial P_i}{\partial a_{ji}} = -\frac{K_{ij}\cos(\Phi_{ij})}{a_{ji}} \quad (b)$$

$$\frac{\partial Q_i}{\partial a_{ij}} = \frac{Q_i}{a_{ij}} - |a_{ij}||y_{ij}|V_i^2\sin\Omega_{ij} \quad (c); \qquad \frac{\partial Q_i}{\partial a_{ji}} = -\frac{K_{ij}\sin(\Phi_{ij})}{a_{ji}} \quad (d)$$

$$\frac{\partial J_{ij}}{\partial a_{ij}} = \frac{2}{a_{ij}}[J_{ij} - a_{ji}^2V_j^2|y_{ij}|] \quad (e)$$

$$\frac{\partial J_{ij}}{\partial a_{ji}} = \frac{2}{a_{ji}}[J_{ij} - a_{ij}^2V_i^2|y_{ij}|] \quad (f)$$

(A3.2.3)

- w.r.t. the near (ij) and far (ji) phase shift phase angles:

$$\frac{\partial P_i}{\partial \phi_{ij}} = -\frac{\partial P_i}{\partial \delta_j} \quad (a) \quad ; \quad \frac{\partial P_i}{\partial \phi_{ji}} = \frac{\partial P_i}{\partial \delta_j} \quad (b)$$

$$\frac{\partial Q_i}{\partial \phi_{ij}} = -\frac{\partial Q_i}{\partial \delta_j} \quad (c) \quad ; \quad \frac{\partial Q_i}{\partial \phi_{ji}} = \frac{\partial Q_i}{\partial \delta_j} \quad (d)$$

$$\frac{\partial J_{ij}}{\partial \phi_{ij}} = \frac{\partial J_{ij}}{\partial \delta_i} \quad (e) \quad ; \quad \frac{\partial J_{ij}}{\partial \phi_{ji}} = \frac{\partial J_{ij}}{\partial \delta_j} \quad (f)$$

(A3.2.4)

- w.r.t. the near shunt compensation admittance:

$$\frac{\partial Q_i}{\partial b_{ci}} = -V_i^2 \qquad (A3.2.5)$$

APPENDIX 3.3

TRANSFORMATION OF THE LOAD FLOW JACOBIAN FOR USE IN THE OPF SUBPROBLEM

The modifications implemented in the subproblem formulation will now be described mathematically. They amount to making (1) the proper choice of partition of variables and (2) a transformation of the load flow Jacobian.

The rows and columns of the load flow Jacobian are partitioned along the lines of the independent/dependent injections and states. The number of independent injections is made equal to the number of dependent states. Besides that, one dependent injection is labelled the manifold injection. The reader is referred to Appendix 3.4 for the details of the nomenclature for these partitions. The load flow Jacobian then takes on the more detailed form

$$\begin{bmatrix} y_b \\ y_m \\ y_d \end{bmatrix} = \begin{bmatrix} J_{bd} & J_{bb} \\ J_{md} & J_{mb} \\ J_{dd} & J_{db} \end{bmatrix} \begin{bmatrix} x_d \\ x_b \end{bmatrix} \qquad \text{(A3.4.1)}$$

Hence the Jacobian is partitioned into six parts depending on the status of the (y,x) pairs. The first subscript pertains to the injection, the second to the state.

The change of algebraic basis is performed by having the y_b and the x_d subvectors swap places. Then the right hand side vector of variables would be made up of the desired basis. Simple algebraic manipulations lead to the new formulation

$$d' = G'b' \qquad \text{(A3.4.2)}$$

or more explicitly,

$$\begin{bmatrix} x_d \\ y_m \\ y_d \end{bmatrix} = \begin{bmatrix} G_{xy} & G_{xx} \\ G_{my} & G_{mx} \\ G_{yy} & G_{yx} \end{bmatrix} \begin{bmatrix} y_b \\ x_b \end{bmatrix} \qquad \text{(A3.4.3)}$$

The components of the G′ matrix are expressed in terms of the components of J.

$$G_{xy} = J_{bd}^{-1} \quad (a) \qquad G_{xx} = - J_{bd}^{-1}J_{bb} \quad (d)$$

$$G_{my} = J_{md}J_{bd}^{-1} \quad (b) \qquad G_{mx} = J_{mb} - J_{md}J_{bd}^{-1}J_{bb} \quad (e)$$

$$G_{yy} = J_{dd}J_{bd}^{-1} \quad (c) \qquad G_{yx} = J_{db} - J_{dd}J_{bd}^{-1}J_{bb} \quad (f)$$

(A3.4.4)

The elements of G′ are the sensitivity coefficients, used in many linear models. It should be noted that eq. A3.4.1 and A3.4.3 are mathematically equivalent.

The equation linking y_m to the independent variables is retained as the linear model of the load flow manifold. It is the generalized power balance equation. This equation will be presented from here on as

$$g_0^T b = 0 \qquad (A3.4.5)$$

with

$$g_0^T = \begin{bmatrix} -G_{my}^T & 1 & -G_{mx}^T \end{bmatrix} \qquad (A3.4.6)$$

The remaining rows of G′ form the functional representation of the dependent variables, and together form a matrix denoted G_1'. Hence $d = G_1'b'$. A final modification brings the formulation to its definite form. A zero column vector is concatenated to G_1', at least conceptually, in order to express d as a function of b. The augmented matrix is the G_1 matrix referred to in the subproblem formulation. Since bounds are placed on dispatchable quantities d_g, the final expression is written in terms of that vector. It is written

$$d_g = d_0 + G_1 b \qquad (A3.4.7)$$

with

$$d_0 = d_1 + d_e \qquad (A3.4.8)$$

$$G_1 = \begin{bmatrix} G_y & 0 & G_x \end{bmatrix} \qquad (A3.4.9)$$

APPENDIX 3.4

NOMENCLATURE FOR THE OPF AND ITS SUBPROBLEM

Some basic nomenclature has already been defined, when presenting the OPF. Some new nomenclature and notation are needed in formulating and solving the subproblems, to keep track of the various partitions placed on the variables and on the corresponding sets of coefficients. Special attention should be paid to this Appendix, since it will be assumed in the subsequent text that the new nomenclature and notation are understood.

Recall that the load flow variables are naturally partitioned into injections (y) and states (x); their components were enumerated in section 3.3.1. Independently, variables are partitioned mathematically as independent (denoted b, for basis) or dependent (d). One injection is labelled the manifold variable, and one state is the reference state. The crossing of the two partitions results in the following sets of variables:

- seen as in injections/states :

$$y = \begin{bmatrix} y_b \\ y_m \\ y_d \end{bmatrix} \quad (a), \qquad x = \begin{bmatrix} x_d \\ x_r \\ x_b \end{bmatrix} \quad (b) \qquad (A3.4.1)$$

with subscripts b,d denoting independent, dependent variables,
m denoting the manifold injection,
r denoting the reference state.

The reference state being set identically to zero, it will be ignored from here on. It will be chosen a voltage phase angle.

- seen as independent/dependent :

$$b = \begin{bmatrix} y_b \\ y_m \\ x_b \end{bmatrix} \quad (a), \qquad d = \begin{bmatrix} x_d \\ y_d \end{bmatrix} \quad (b) \qquad (A3.4.2)$$

The inclusion of manifold injection y_m in vector b is retained because it will simplify notation, even though it isn't independent. Independent variables are denoted with subscript b, and together form vector b'.

The three types of load flow variables listed below are regrouped under a common notation, but distinguished by subscripts g, l, and e. Subscript g is used in a general sense to designate generation, for all variables. This is proposed to unify notation; all dispatchable variables have been affixed that subscript. Subscript l denotes the bus load variables. Subscript e is needed in the subproblem to denote the expansion point for all load flow quantities. The notation without subscript is reserved for excursions from the expansion point, used in the formulation of the linearized load flow equations. The relations between the variables are given by the expressions

$$y = y_g - (y_l + y_e) \quad \text{(A3.4.3.a)}$$

$$x = x_g - x_e \quad \text{(A3.4.3.b)}$$

or

$$b = b_g - (b_l + b_e) \quad \text{(A3.4.4.a)}$$

$$d = d_g - (d_l + d_e) \quad \text{(A3.4.4.b)}$$

Similar notation can be applied to subsets of y/x or b/d.

A partition separates inactive variables from those at a bound. Superscripts I and A are assigned for this purpose.

A partition separates those independent variables which appear in the objective function from those which do not. The latter are called transparent and are denoted t . For example, in economic dispatch, independent variables other than real power generations are transparent. The partitions would be designated by subscripts p and t.

Lagrange multipliers in the subproblem solution are considered in two groups. Those associated with independent variables are denoted μ , and those with functional constraints λ . In particular, the generalized power balance equation is an equality constraint, and is always active. Its Lagrange multiplier is denoted λ_0 . The remaining Lagrange multipliers of λ are denoted by the λ_1 vector. Together, all Lagrange multipliers for the static OPF are denoted Λ.

All the coefficients of the functional constraints, obtained through manipulations on the full load flow Jacobian matrix, are denoted g. Elements g_{ij} form row vectors g_i^T which in turn make up the G matrix. The row vector of coefficients for the generalized power balance equation is denoted g_0^T and the remaining part of G is denoted G_1. The partition of rows of G according to the active/inactive status of the dependent variables will not require additional notation, since the context will always distinguish between the two.

Aside from superscript M and m for upper and lower bounds, superscript lim will be used to designate the group of variables at their known limits, without specifying which limit.

Other notation will be defined as it appears. As a final note, we will try to keep notation as simple as possible, even though many partitions have been defined, by avoiding multiple superscripts and subscripts whenever possible. This will be done by dropping superscripts or subscripts where the context allows. Prior notice will be given before the simplification is implemented.

APPENDIX 4.1

PARTITIONS AND DIMENSIONS OF VARIABLES, COEFFICIENT MATRICES AND VECTORS IN THE OPTIMALITY CONDITIONS

Here are the partitions of the variables, coefficient matrices and vectors needed for the optimality conditions of economic dispatch, minimum loss and minimum load shedding. They are split along the active/inactive and the P_g/transparent partitions. The notation has already been defined in Appendix 3.4, except the term H which is defined in this Appendix.

- Partitions for the independent variables:

$$b = \begin{bmatrix} p^A \\ p \\ t^A \\ t \end{bmatrix} = \begin{bmatrix} P_g^A - (P_l + P_e)^A \\ P_g - (P_l + P_e) \\ t_g^A - (t_l + t_e)^A \\ t_g - (t_l + t_e) \end{bmatrix} \qquad \text{(A4.1.1)}$$

The subscripts b assigned to the loads and expansion points in eq. A4.1.1 are not meant to indicate that they are at a bound; rather they show that the associated dispatchable variables are at a bound.

- Partition of the arrays of the objective function parameters:

$$a = \begin{bmatrix} a^A \\ a \end{bmatrix} \qquad \text{(A4.1.2.a)}$$

and

$$B = \begin{bmatrix} B^A & \\ & B \end{bmatrix} \qquad \text{(A4.1.2.b)}$$

- Partition of the dependent constraints:

$$G = \begin{bmatrix} g_0^T \\ G_1 \end{bmatrix} = \begin{bmatrix} g_{0p}^{AT} & g_{0p}^T & g_{0t}^{AT} & g_{0t}^T \\ G_{1p}^A & G_{1p} & G_{1t}^A & G_1^t \end{bmatrix} \qquad \text{(A4.1.3)}$$

- Partition of the indexing matrices R_b, R_d and R_1:

$$R_b = \begin{bmatrix} R_p & & & \\ & 0_p & & \\ & & R_t & \\ & & & 0_t \end{bmatrix} \qquad \text{(A4.1.4)}$$

$$R_d = \begin{bmatrix} R_d^A & \\ & 0_d \end{bmatrix} \qquad \text{(A4.1.5)}$$

and

$$R_1 = \begin{bmatrix} R_1^A & \\ & 0_1 \end{bmatrix} \qquad \text{(A4.1.6)}$$

The submatrices R_p, R_t, R_d^A and R_1^A contain diagonal elements ±1, and submatrices 0_p, 0_t 0_d and 0_1 are zero matrices, all of the appropriate dimensions.

The term H is defined here, in order to simplify notation in the optimality conditions of Chapter 4:

$$H = R_d^A G_1^A$$

This H matrix is made up from the rows of coefficients of the active dependent constraints, premultiplied by the appropriate ±1 index. The partitions of H are identical to those for G_1.

Nomenclature is introduced for the dimensions of all these vectors and matrices, and others about to be defined. The dimensions are then given in Table A4.1.1.

- Nomenclature:

ng,ngI,ngA	number of generations, inactive generations and active generations,respectively;
nt,ntI,ntA	number of transparent variables, inactive transparents, and active transparents, respectively;

and active transparents, respectively;

nl,nlI,nlA number of loads, inactive loads and active loads respectively.

- Dimensions:

TABLE A4.1.1

DIMENSIONS OF PRINCIPAL VECTORS AND MATRICES

Variables	Dimension
P_g, Pg_{lim}	ng x 1
t_g, t_g^{lim}	nt x 1
$P^A, P_g^A, P_l^A, P_e^A, g_{0p}^A, \mu_p, a^A$	ngA x 1
$P^I, Pg^I, P_l^I, P_e^I, g_{0p}, a^I$	ngI x 1
$t^A, t_g^A, t_l^A, t_e^A, g_{0t}^A, \mu_t$	ntA x 1
$t^I, t_g^I, t_l^I, t_e^I, g_{0t}$	ntI x 1
k_l^{lim}	ndA x 1
B^A, R_p	ngA x ngA
$B^I, 0_p$	ngI x ngI
R^t	ntA x ntA
0_t	ntI x ntI
G_{1p}^A, H_p^A	ndA x ngA
G_{1p}, H_p	ndA x ngI
G_{1t}^A, H_t^A	ndA x ntA
G_{1t}, H_t	ndA x ntI
R_d^A	ndA x ndA
0_d	ndI x ndI
R_l^A	nlA x nlA
0_l	nlI x nlI

APPENDIX 4.2

DEGENERACY

Degeneracy occurs when the active constraints outnumber the inactive independent variables. It can mean one of two things: that the optimal solution trajectory has met with a feasibility limit, or it has moved to a vertex of the feasible region. For the latter, the number of faces joined at that vertex is greater than the number of independent variables. It is important to have techniques to distinguish between the two, and for the latter, to reorganize the active set and push forward the optimal solution trajectory. This section proposes some techniques to deal with the degeneracies, for the different situations encountered in the study.

Example

First, a simple illustration of degeneracy is presented. Consider the quadratic program

$$\min_{x_1,x_2} \quad 9.25 - [1 \quad 6] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \tfrac{1}{2} [x_1 \quad x_2] \begin{bmatrix} 2 & \\ & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\text{s.t.} \quad \begin{array}{rcccl} \theta & \leq & x_1 + x_2 & \leq & 1 + 10\theta \\ 0 & \leq & x_1 & \leq & 2 \\ 0 & \leq & x_2 & \leq & 2 \end{array} \qquad \text{(EX1)}$$

The cost contours and the feasible region are drawn at $\theta=0$ in figure A4.2.1.a. The optimal solution can be seen to be $x_1=0$, $x_2=1$ and the functional constraint is at its upper bound. As θ increases, an optimal solution trajectory climbs the $x_1=0$ constraint. The optimal solution trajectory is

$$\begin{array}{ll} x_1 = 0 & \mu_1 = 3 - 20\theta \\ x_2 = 1 + 10\theta & \mu_2 = 0 \\ x_1 + x_2 = 1 + 10\theta & \lambda_1 = 4 - 20\theta \end{array}$$

The situation at $\theta=0.1$ is drawn in figure A4.2.1.b. The optimal solution trajectory has reached the corner of the box, and now three constraints are active in two-dimensional space. This is an example of degeneracy. The values of the variables at $\theta=0.1$ are $x_1=0$, $x_2 = x_1+x_2 = 2$, $\mu_1=1$, $\mu_2=0$, $\lambda_1=2$.

A successful elimination of degeneracy would take x_1 off its lower bound. In the process, variables x move continuously, but the Lagrange multipliers jump. The new optimal solution trajectory is

$$\begin{aligned} x_1 &= 10\theta - 1 & \qquad \mu_1 &= 0 \\ x_2 &= 2 & \mu_2 &= 20\theta - 1 \\ x_1 + x_2 &= 1 + 10\theta & \lambda_1 &= 3 - 20\theta \end{aligned}$$

and the values of the variables at $\theta=0.1$ are $x_1=0$, $x_2 = x_1 + x_2 = 2$, $\mu_1=0$, $\mu_2=1$, $\lambda_1=1$.

For θ between 0.1 and 0.15 the optimal solution trajectory follows the $x_2=2$ bound, to $x_1=0.5$. For θ between 0.15 and 0.25, illustrated in figure A4.2.1.c, the optimal solution is fixed at $x_1=0.5$, $x_2=2$, and the functional constraint is inactive. At $\theta=0.25$, the functional constraint hits its lower bound. For θ between 0.25 and 0.4, the optimal solution trajectory follows the $x_2=2$ bound to $x_1=2$. At $\theta=0.4$, the active set is a single point, the top right-hand corner of the box. This is illustrated in figure A4.2.1.d. Degeneracy cannot be resolved here, because a further increase in θ results in an empty feasible region. The value $\theta=0.4$ represents the feasibility limit for problem EX1. This completes the example.

Four cases of degeneracy and their remedies are considered. A first group of cases, deemed "pathological", could occur when the G constraint matrix for active constraints loses full rank. A second case, used in real power dispatch, frees expensive real power generations from their lower bounds when the solution trajectory would otherwise be blocked. In the third case, a technique for resolving degeneracy is presented for when there is one constraint too many. This problem always accompanies a breakpoint in parametric linear programming (minimum loss). It is also likely to be the most common degeneracy in all the optimization procedures being studied here.

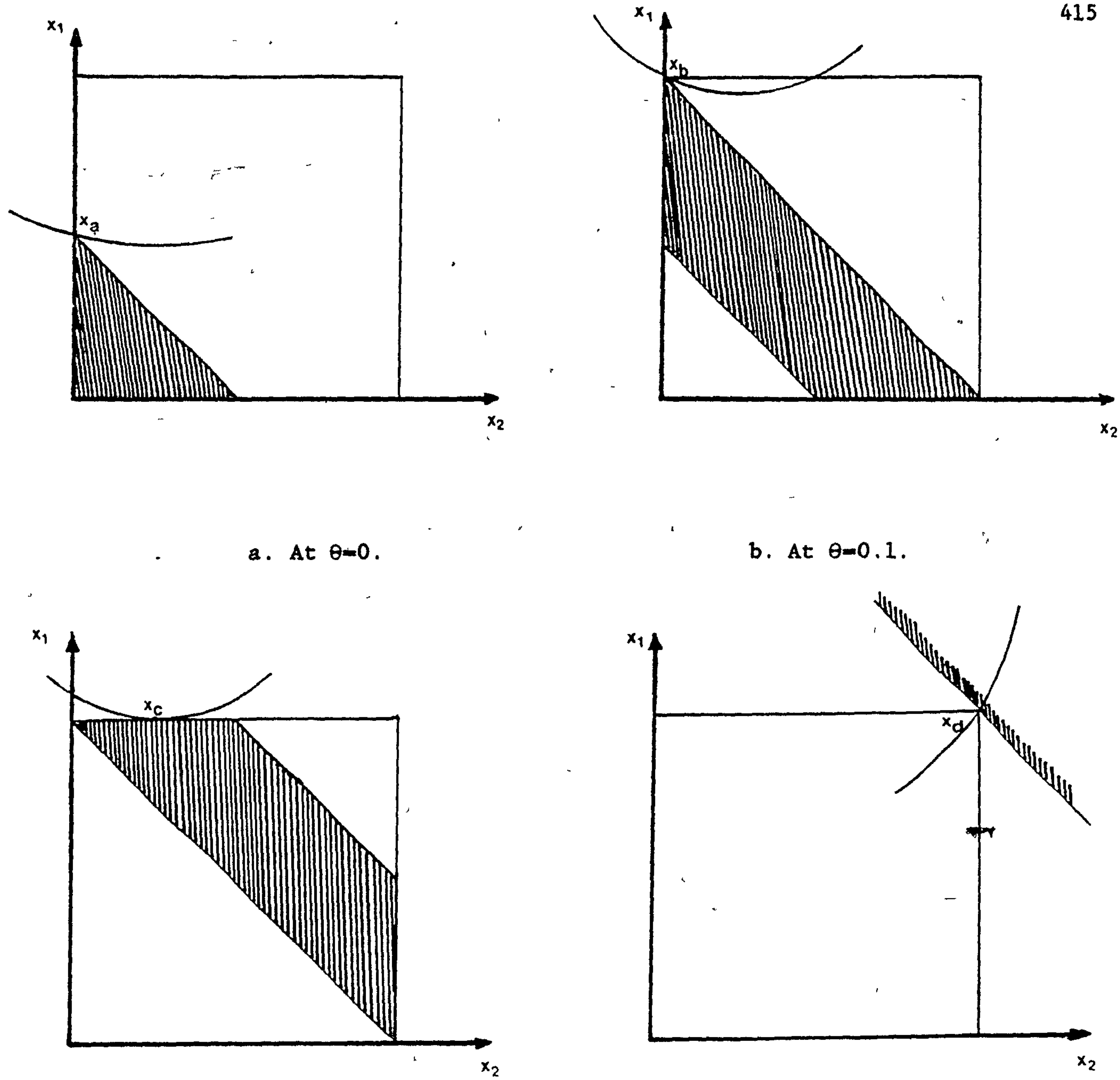
Finally a method is suggested for diffusing degeneracies with too many constraints being added simultaneously to the active set.

All but one of the applications in this section are based on perturbation techniques. They modify the problem slightly, to create artificial facets in the feasible region, in place of the degenerate vertex. The optimal solution trajectory passes over the edges or possibly the vertices of these facets, which are not degenerate. The remaining application, the third, is based on pivoting techniques. This method, prevalent in linear programming, is a combinatorial technique which tries to find an alternative active set. Rules exist to avoid repetition of candidates, or cycling. The case presented here is a simple application of pivoting techniques.

Case 1. Pathological degeneracies

We call pathological the following degeneracy: the particular arrangement of system parameters and system topology are such that for a newly activated constraint, say d_{gi}, the vector of sensitivity coefficients g_i is numerically linearly dependent on the rows already forming the G matrix for active constraints. In the following, a rank deficiency of one will be considered. The case where two or more simultaneously activated constraints are linearly dependent on the previous constraints is virtually impossible, but perturbation techniques suggested further would allow to treat them one at a time.

Figure A4.2.2.a illustrates a pathological degeneracy which occurs when the perturbation function affects only the right-hand-side of the optimality conditions. Some constraints, say C_1 and C_2, are active. In the three dimensional space, up to three active constraints are permitted. As Θ is increased C_1 is raised and C_3 is lowered. The active segment [ab] and the inactive segment [cd] are parallel. At some $\Theta=\Theta'$, all three constraints are active and meet along segments [ab] = [cd]. Once that value of Θ has passed, constraint C_2 is freed. The difficulty is in choosing which constraint must be dropped.



a. At θ=0.

b. At θ=0.1.

c. For 0.15 ≤ θ ≤ 0.25.

d. At θ=0.4.

Figure A4.2.1. Four stages of problem EX1.

The easiest way to resolve this type of degeneracy is to perturb g_i by a small amount, to destroy the parallelism of the intersections of the constraints. The perturbation should be negligeable compared to system quantities but large compared to machine precision. A single element of g_i, say g_{ij}, could be perturbed by $\pm\epsilon$. The sign of ϵ is chosen so that at the degenerate breakpoint, the new constraint remains inactive. At a slightly higher value of θ, a new breakpoint is encountered when the perturbed constraint is activated. All the constraints would likely be active over some small interval of θ. Then a Lagrange multiplier will reduce to zero and the corresponding constraint is dropped. The degenerate breakpoint is replaced by two breakpoints; that is a typical consequence of the perturbation technique.

This case also resolves any problem of parallel constraints. Only one can be active, except for some $\theta=\theta'$.

Figure A4.2.2.b illustrates a pathological degeneracy which occurs when the perturbation affects the left-hand-side of the optimality conditions. This could occur in contingency analysis, when transmission line parameters or the quadratic cost parameters are varied. Constraints C_1 and C_2 are active, and share segment [ab]. As θ is increased, constraint C_3 is rotated. At some $\theta=\theta'$, the three constraint planes meet along [ab]. Once this value of θ has passed, constraint C_2 is freed. Again the difficulty is to determine which constraint is dropped.

Again the case of a single rank deficiency will be resolved. The easiest solution technique would be to perturb the right-hand-side limit of the new constraint by $\pm\epsilon$. The sign of ϵ would be chosen so that at the degenerate breakpoint the new constraint remains inactive. When the constraint is activated, at a slightly higher value of θ, the intersections of the constraints taken all together no longer coincide. Again the leaving constraint is decided by the regular process. The case of suddenly parallel and coinciding constraints can be handled by this technique.

A most pathological case is portrayed in figure A4.2.2.c. The constraints share as intersection the line segments [ab]. As θ increases constraints C_2 and C_3 move together in such a way that their intersection always lies on C_1. The active constraint matrix G is rank deficient independently of θ. This has occured occasionally in our numerical tests. In this case the dependent constraints which cause rank deficiency need not be placed in the cptimality conditions, since the feasible region is accurately represented by the other active constraints. This special constraint is monitored separately until one of the constraints in the optimality conditions is released from the active set. Only then is it resubmitted to the optimality conditions.

Perturbation techniques, such as those presented above, proceed by solving slightly perturbed problems. Tracking of the optimal solutions of the true problem can be resumed once the solution trajectory is pushed far enough beyond the degenerate breakpoint. With the assurance that the proper optimal active set is known, the optimality conditions are recomputed without the perturbations. Since the perturbations are small, the solutions to the unperturbed problem should be very close to those of the perturbed problem.

Case 2. Degeneracy occuring with real power generations at their lower bounds

The second case is an idea suggested for real power dispatch [Fahmideh-Vojdani 1982]. When degeneracy occurs and some real power generations are at a lower bound, as many as needed are released from their bounds. If there is a choice, those with the incrementally cheapest lower bounds are taken first. This is a simple rule, easy to implement, so it can be proposed tentatively when applicable. It can be proven valid on the simple problem (ed) of economic dispatch, with functional dependent constraints omitted. In general though, it can only by considered a heuristic. A simple counterexample illustrates the point. If constraint $x_1 + 3/4x_2 \leq 2-30\theta$ is added to problem EX1, at $\theta=1/30$ three constraints are active; $x_1=0$, $x_1 + x_2 = 4/3$, and $x_1 + 3/4x_2 = 1$. Variable $x_2=4/3$ is inactive. As θ is increased, constraint $x_1 + x_2 = 1+\theta$ is dropped from the active set, and x_1 remains at a lower bound. That shows that functional constraints can be dropped from the active set before lower bound generations.

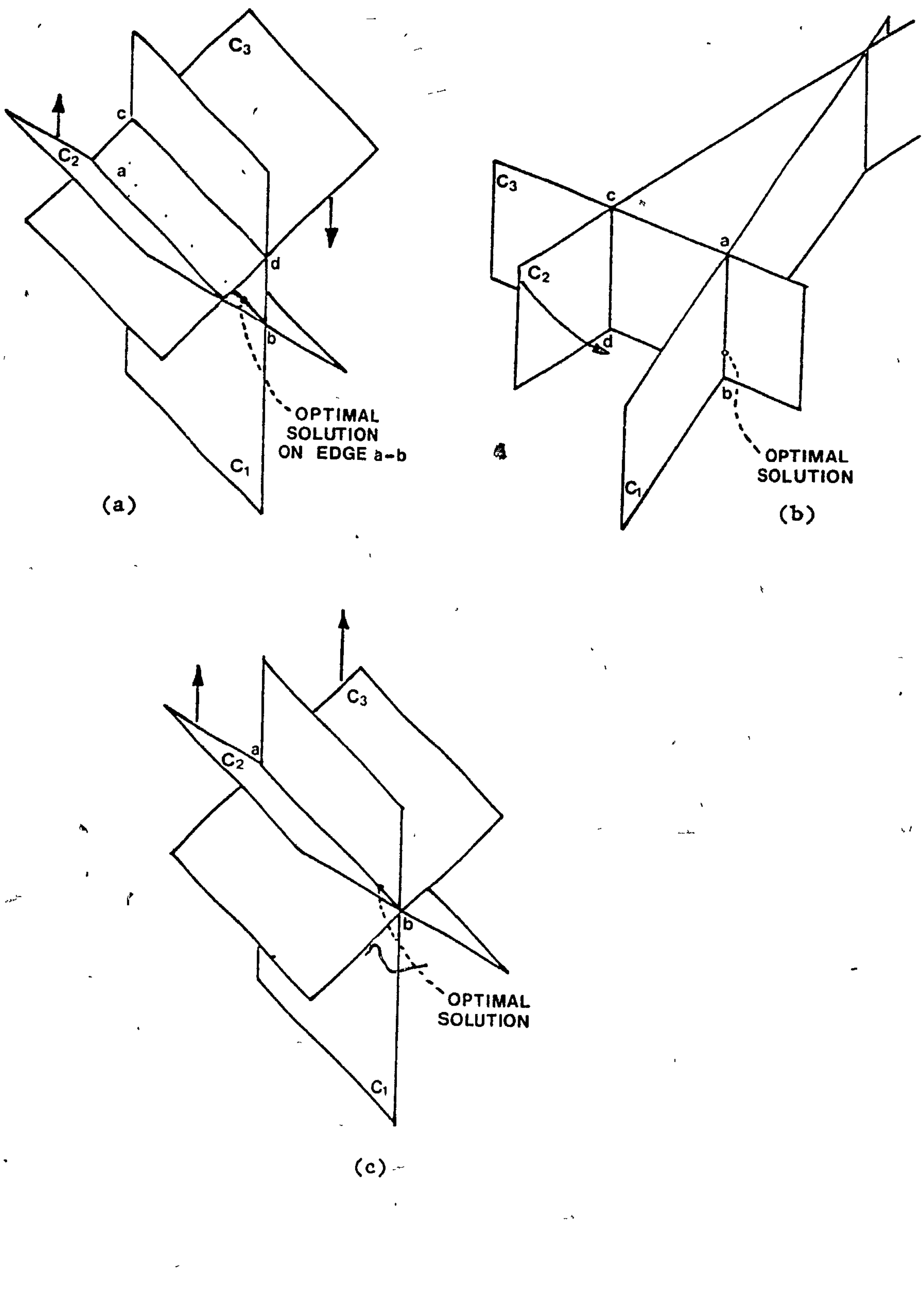


Figure A4.2.2 Three views of degeneracy.

Case 3. Degeneracy due to one constraint too many

When the constraints outnumber the independent variables by one, a pivoting technique can be implemented quite easily. To resolve degeneracy, the load flow variables are held constant, but a new set of Lagrange multipliers is sought which can solve the optimality conditions.

When the new constraint is added to the active set, the constraint matrix A, as in eq.4.11, becomes "long", with one more row than column. Its transpose is "wide". Holding load flow variables constant, it is clear that the set of equations for the Lagrange multipliers allows a manifold of solutions, with one degree of freedom. At θ_i,

$$A^T\Lambda = Bb(\theta_i) + a \qquad (A4.2.1)$$

In the following, subscripts o and n pertain to old and new constraints, respectively. Let the Lagrange multiplier of the newly activated constraint serve as the parameter, with which other Lagrange multipliers are expressed. The expression is

$$\Lambda_o = A_o^{-1} [Bb(\theta_i) + a - A_n\Lambda_n] \qquad (A4.2.2.a)$$
$$= \Lambda_{o0} + \Delta\Lambda.\Lambda_n \qquad (A4.2.2.b)$$

where A_o, Λ_o are the square constraint matrix and the Lagrange multipliers before the addition of the new active constraint.

A_n, Λ_n are the new vector of constraint coefficients and the new Lagrange multiplier.

The new combination of Lagrange multipliers which satisfies eq. A4.2.2 contains all non-negative values, but at least one value is nil. The newly added constraint remains in the active set and the variable whose Lagrange multiplier vanishes is dropped from the active set.

See figures A4.2.3.a and .b for an illustration of the search for a new set of Lagrange multipliers. A unique trajectory of Lagrange multipliers leads to a bounded hyperplane of values, at $\theta=\theta_i$. In fig. a, one Lagrange multiplier is chosen along some opposite edge of the hyperplane, where another Lagrange multiplier is chosen as zero. From there the regular process continues. In fig. b, no other Lagrange multiplier for an inequality constraint can be reached, so the feasibility limit has been reached.

The procedure to find the new Lagrange multiplier is as follows. Allow Λ_n in eq. A4.2.2 to increase, and compute values of Λ_n for which values of elements of Λ_o can drop to zero. The steps are

STEP 1. Set i=1 and $i^*=0$. This procedure is not applied to the generalized power balance equation, an equality constraint.

STEP 2. Set i=i+1. If i>n, the dimension of Λ_o, go to STEP 6.

STEP 3. If $\Delta\Lambda \geq 0$, then Λ_{oi} cannot drop to zero. Return to STEP 2. If not, go to STEP 4.

STEP 4. Compute the value of Λ_n for which $\Lambda_{oi}=0$.

$$\Lambda_{ni} = \frac{\Lambda_{o0i}}{\Delta\Lambda_i} \tag{A4.2.3}$$

STEP 5. Keep track of the smallest value of Λ_{ni}.

$$\Lambda^* = \min_i (\Lambda_{ni}) \tag{A4.2.4}$$

Recall for which i the Λ^* has been obtained, call it i^*.

STEP 6. If $i^*=0$, then no member of Λ_o can go to zero. Notify the user or a control outside the subproblem. STOP. If not, go to STEP 7.

STEP 7. Compute values of the Lagrange multipliers.

$$\Lambda_o = \Lambda_{o0} + \Delta\Lambda . \Lambda^* \tag{A4.2.5.a}$$
$$\Lambda_n = \Lambda^* \tag{A4.2.5.b}$$

and in particular

$$\Lambda(i^*) = 0 \tag{A4.2.5.c}$$

The new Lagrange multipliers have been found. STOP.

Values of Λ_o versus Λ_n are drawn in figure A4.2.4. In this example, Λ_2 i the first to reach zero. New values of all Lagrange multipliers are found on the dotted line.

If no member of Λ_0 drops to zero, a feasibility limit has been reached. Degeneracy can be resolved if at least one member of Λ_0 drops to zero.

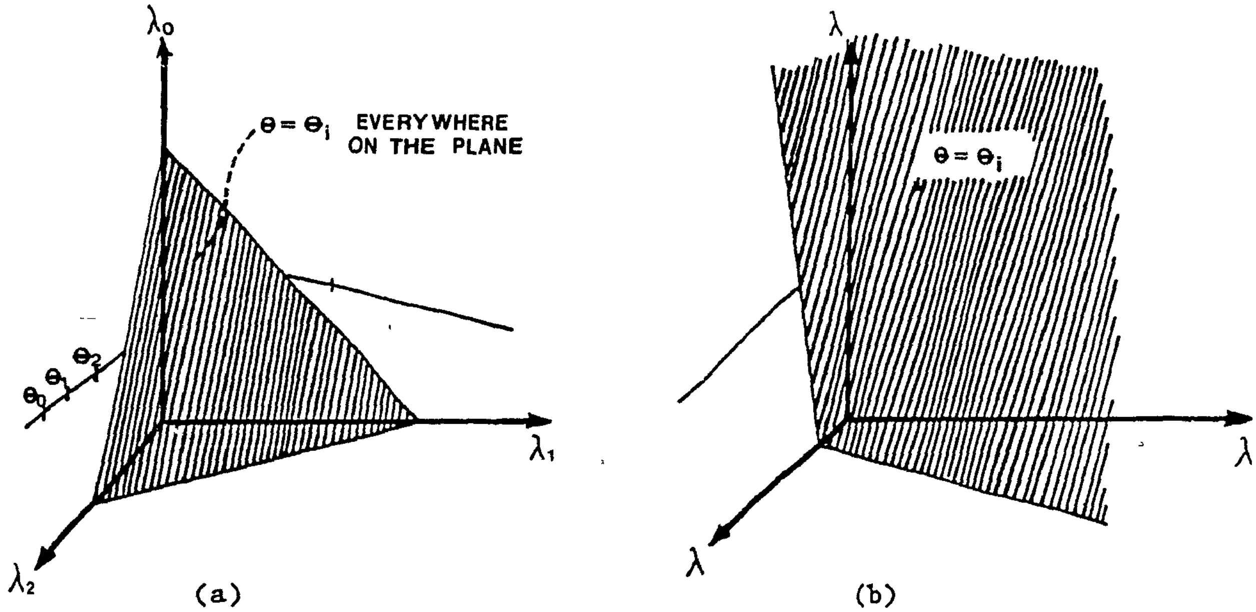


Figure 4.2.3 The trajectory of Lagrange multipliers.

(a) feasible case; (b) infeasible case.

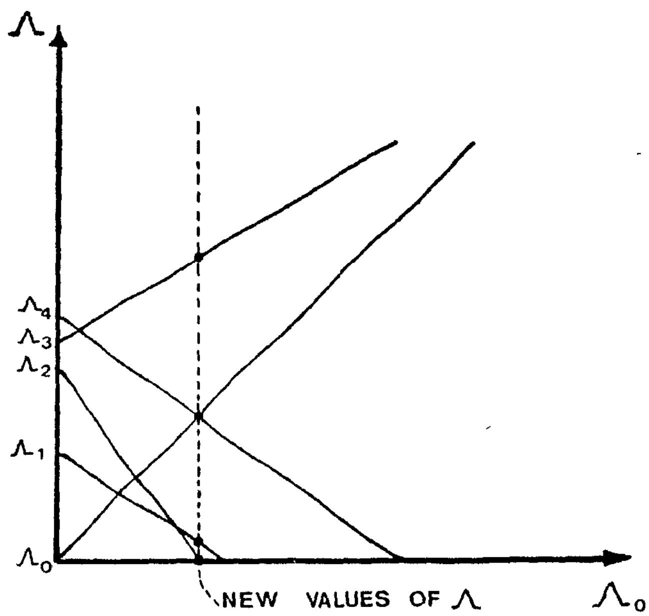


Figure A4.2.4 The new values of the Lagrange multipliers.

Case 4. Degeneracy with more than one constraint too many

The last technique proposed here is used when the constraints outnumber the independent variables by more than one. Such a situation occurs at the beginning of the load shedding problem. The method consists of adding small perturbations to some of the newly added activated constraints, to avoid having them all becoming active together. Let the dimension of the basis be denoted nb. For some $\theta=\theta_i$, many constraints are activated, so that the number of constraints is greater than nb, say nb+m. Then m-1 of the newly activated constraints are perturbed on their right-hand-sides by different small amounts ϵ:

$$g_i^T b = d_{gi} \pm \epsilon_i \qquad \text{(A4.2.6.a)}$$
for dependent variable constraints

and

$$b_{gi} = b_{gi}^{lim} \pm \epsilon_i \qquad \text{(A4.2.6.b)}$$
for independent variable constraints.

Again, the sign of the perturbation is chosen to keep the constraints inactive at the degenerate breakpoint. Not all constraints are perturbed, as in the classical technique, because then the optimal solution trajectory would no longer be optimal. It would have to jump to a new trajectory.

The remaining problem has one constraint too many. Its degeneracy can be tested using the previous pivoting technique. If degeneracy is resolved, then the solution trajectory will proceed, but it is likely to meet one of the perturbed constraints very soon. Some, but not necessarily all of the perturbed constraints will be processed this way. Hence a point with an m dimensional degeneracy is replaced by at most m points with single dimensional degeneracies. If all the degeneracies are resolved, the solution trajectory proceeds. If any one degeneracy is unresolvable, the problem has hit a feasibility limit. If all degeneracies are resolved the perturbations can be dropped and the optimal solution trajectory proceeds. A simplified illustration is presented in figures A4.2.5.a and b. Fig. a shows the initial feasible region with a degenerate vertex. In fig. b the vertex has been "lopped off", thanks to the perturbations, resulting in non-degenerate vertices. The method proposed above is slightly different, in that one-dimensional degeneracies are kept. The solution trajectory moves through the area of degeneracy, and if resolved, moves beyond.

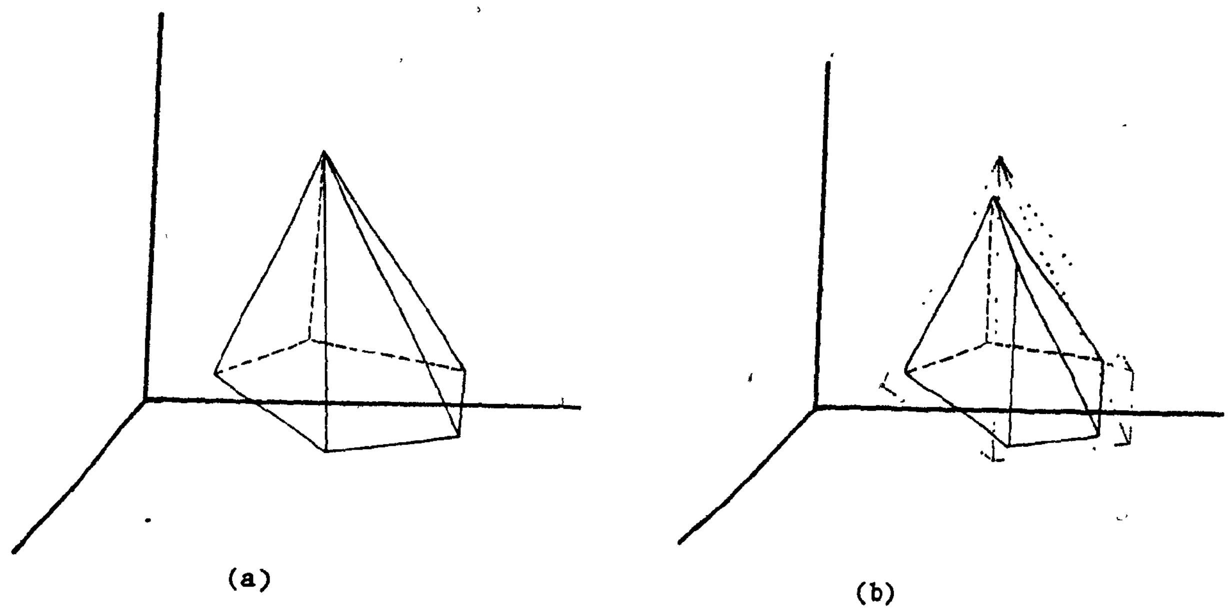


Figure A4.2.5 Break-up of a degenerate vertex.

(a) before ; (b) after.

APPENDIX 7.1

DATA FOR THE 6 BUS SYSTEM

Line Data

Line no.	from	to	G	B_{se}	B_{sh}	J^{max}
1	6	5	0.0000e0	-0.3333e1	.0000e-1	0.2000e1
2	6	4	0.5540e0	-0.2325e1	.1500e-1	0.3000e1
3	6	1	0.4340e0	-0.1827e1	.2100e-1	0.2000e1
4	5	2	0.5770e0	-0.1308e1	.0000e-1	0.3000e1
5	4	3	0.0000e0	-0.7518e1	.0000e-1	0.2000e1
6	4	1	0.5880e0	-0.2582e1	.1500e-1	0.3000e1
7	3	2	0.4550e0	-0.6460e0	.0000e-1	0.2000e1

Tap Data

Tap no.	on line	near bus	T^{min}	T^{max}
1	5	4	0.800	1.200
2	1	6	0.800	1.200

Bus Data

Bus no.	V^{min}	V^{max}	Initial Load P	Initial Load Q
1	0.950	1.050	.0000	.0000
2	0.900	1.100	.0000	.0000
3	0.900	1.100	.5000	.1300
4	0.900	1.100	.6000	.0000
5	0.900	1.100	.7500	.1800
6	0.900	1.100	.5000	.0500

Generation Data

Bus no.	P^{min}	P^{max}	Q^{min}	Q^{max}	Cost coefficients c	a	b
1	0.100	0.920	-0.250	0.250	0.050	0.250	2.500
2	0.500	0.500	-0.300	0.350	0.020	0.500	0.500
3	0.220	1.105	-0.340	0.340	0.010	0.500	1.000
4	0.150	0.990	-0.500	0.600	0.020	0.150	2.000
5	0.330	1.200	-0.100	0.200	0.010	0.300	1.750
6	0.190	1.190	0.000	0.750	0.020	0.300	1.200

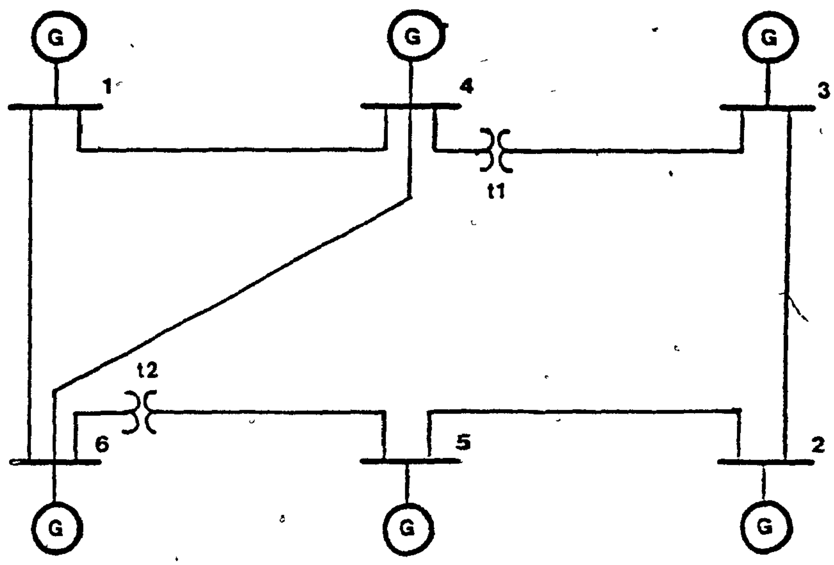


Figure A7.1. One-line diagram of the 6 bus system.

APPENDIX 7.2

DATA FOR THE 10 BUS SYSTEM

Line Data

Line no.	from	to	G	B_{se}	B_{sh}	J^{max}
1	10	4	.9737e1	-.4868e2	.2025e0	.1000e2
2	9	1	.1902e1	-.1252e2	.3037e0	.5530e1
3	8	5	.2600e1	-.7140e1	.2025e0	.6550e1
4	8	2	.3759e1	-.1471e2	.3037e0	.8000e1
5	7	5	.4048e1	-.1983e2	.2025e0	.5082e2
6	7	4	.4048e1	-.1983e2	.1012e0	.1000e2
7	6	4	.3030e1	-.1999e2	.1012e0	.8500e1
8	6	1	.4367e1	-.1461e2	.2025e0	.8145e1
9	5	4	.1902e1	-.1252e2	.3037e0	.8500e1
10	5	3	.3563e1	-.1734e2	.2025e0	.7570e1
11	4	3	.1248e1	-.4901e1	.2025e0	.7000e1
12	4	2	.3609e1	-.1443e2	.3037e0	.8500e1
13	3	1	.1248e1	-.4901e1	.2025e0	.5000e1

Tap Data

Tap no.	on line	near bus	T^{min}	T^{max}
1	2	9	0.900	1.100
2	4	8	0.900	1.100
3	10	5	0.900	1.100
4	9	5	0.900	1.100
5	6	7	0.900	1.100

Bus Data

Bus no.	V^{min}	V^{max}	Initial Load P	Q
1	0.950	1.050	.0000	.0000
1	0.900	1.100	1.500	0.450
2	0.900	1.100	1.000	0.330
3	0.900	1.100	2.500	0.500
4	0.900	1.100	10.000	2.300
5	0.900	1.100	0.000	0.000
6	0.900	1.100	1.000	0.350
7	0.900	1.100	0.000	0.000
8	0.900	1.100	2.500	0.700
9	0.900	1.100	1.000	0.300
10	0.900	1.100	0.000	0.000

<u>Generation Data</u>

Bus no.	P^{min}	P^{max}	Q^{min}	Q^{max}	Cost coefficients c	a	b
4	2.400	6.550	-0.780	3.900	0.025	0.502	0.0346
5	0.800	2.170	-0.660	1.200	0.050	0.648	0.0591
6	0.800	2.160	-0.500	1.500	0.100	0.617	0.0903
7	1.600	4.340	-1.200	2.400	0.020	0.563	0.0334
8	1.200	3.250	-0.790	1.950	0.075	0.589	0.0395
9	0.700	1.800	-0.680	1.550	0.060	0.591	0.1220
10	0.800	3.250	-0.750	2.200	0.090	0.627	0.0406

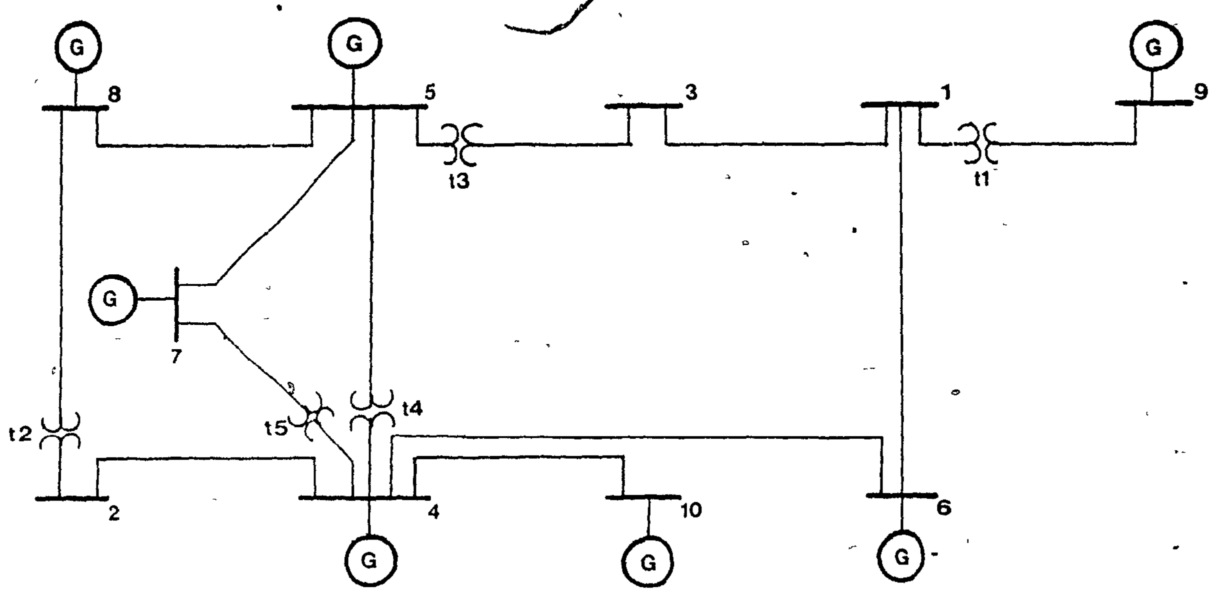


Figure A7.2. One-line diagram of the 10 bus system.

APPENDIX 7.3

DATA FOR THE 30 BUS SYSTEM

Line Data

Line no.	from	to	G	B_{se}	B_{sh}	J^{max}
1	6	5	0.0000e0	-0.3333e1	.0000e-1	0.2000e1
1	30	12	0.0000e0	-0.7143e1	0.0000e0	0.6500e0
2	29	19	0.0000e0	-0.4808e1	0.0000e0	0.6500e0
3	28	24	0.1444e1	-0.4541e1	0.0214e0	0.3200e0
4	28	21	0.6289e1	-0.2201e2	0.0045e0	0.3200e0
5	27	26	0.1136e1	-0.4772e1	0.0209e0	1.3000e0
6	27	22	0.2954e1	-0.7449e1	0.0102e0	0.7000e0
7	26	25	0.5225e1	-0.1565e2	0.0264e0	1.6500e0
8	26	21	0.1686e1	-0.5116e1	0.0187e0	0.6500e0
9	26	20	0.1706e1	-0.5197e1	0.0184e0	0.6500e0
10	25	23	0.1244e1	-0.5096e1	0.0204e0	1.3000e0
11	24	21	0.4363e1	-0.1546e2	0.0065e0	0.3200e0
12	24	1	0.0000e0	-0.2525e1	0.0000e0	0.6500e0
13	23	20	0.8195e1	-0.2353e2	0.0042e0	1.3000e0
14	22	21	0.3590e1	-0.1103e2	0.0085e0	1.3000e0
15	21	20	0.6413e1	-0.2231e2	0.0045e0	0.9000e0
16	21	19	0.0000e0	-0.4808e1	0.0000e0	0.6500e0
17	21	18	0.0000e0	-0.1799e1	0.0000e0	0.3200e0
18	20	12	0.0000e0	-0.3906e1	0.0000e0	0.6500e0
19	19	18	0.0000e0	-0.9091e1	0.0000e0	0.6500e0
20	18	17	0.4116e1	-0.1017e2	0.0000e0	0.3200e0
21	18	15	0.5102e1	-0.1098e2	0.0000e0	0.3200e0
22	18	14	0.2619e1	-0.5401e1	0.0000e0	0.3200e0
23	18	13	0.1785e1	-0.3985e1	0.0000e0	0.3200e0
24	17	16	0.1868e1	-0.4379e1	0.0000e0	0.1600e0
25	16	12	0.1952e1	-0.4104e1	0.0000e0	0.3200e0
26	15	14	0.1677e2	-0.3413e2	0.0000e0	0.3200e0
27	14	6	0.2540e1	-0.3954e1	0.0000e0	0.1600e0
28	13	11	0.5880e1	-0.1176e2	0.0000e0	0.3200e0
29	12	9	0.1520e1	-0.3173e1	0.0000e0	0.3200e0
30	12	8	0.3095e1	-0.6097e1	0.0000e0	0.3200e0
31	11	10	0.3076e1	-0.6219e1	0.0000e0	0.1600e0
32	10	8	0.1808e1	-0.3691e1	0.0000e0	0.1600e0
33	9	8	0.2491e1	-0.2251e1	0.0000e0	0.1600e0
34	8	7	0.1968e1	-0.3976e1	0.0000e0	0.1600e0
35	7	6	0.1461e1	-0.2989e1	0.0000e0	0.1600e0
36	6	4	0.1310e1	-0.2288e1	0.0000e0	0.1600e0
37	5	4	0.1216e1	-0.1817e1	0.0000e0	0.1600e0
38	4	1	0.1969e1	-0.3760e1	0.0000e0	0.1600e0
39	3	2	0.9120e0	-0.1723e1	0.0000e0	0.1600e0
40	3	1	0.9955e0	-0.1881e1	0.0000e0	0.1600e0
41	2	1	0.6875e0	-0.1294e1	0.0000e0	0.1600e0

Shunt Data

Shunt no.	bus	B^{min}	B^{max}
1	3	0.000	0.050
2	6	0.000	0.050
3	7	0.000	0.050
4	8	0.000	0.050
5	12	0.000	0.050
6	13	0.000	0.050
7	15	0.000	0.050
8	17	0.000	0.050
9	18	0.000	0.050

Tap Data

Tap no.	on line	near bus	T^{min}	T^{max}
1	12	24	0.900	1.100
2	18	20	0.900	1.100
3	17	21	0.900	1.100
4	16	21	0.900	1.100

Bus Data

Bus no.	V^{min}	V^{max}	Initial Load P	Q
1	0.950	1.050	.0000	.0000
1	0.950	1.050	.0000	.0000
2	0.900	1.050	.1060	.0190
3	0.900	1.050	.0240	.0090
4	0.950	1.050	.0000	.0000
5	0.950	1.050	.0350	.0230
6	0.950	1.050	.0870	.0670
7	0.950	1.050	.0320	.0160
8	0.950	1.050	.0820	.0250
9	0.950	1.050	.0620	.0160
10	0.950	1.050	.0320	.0090
11	0.950	1.050	.0950	.0340
12	0.950	1.050	.1120	.0750
13	0.950	1.050	.0220	.0070
14	0.950	1.050	.0000	.0000
15	0.950	1.050	.1750	.1120
16	0.950	1.050	.0350	.0180
17	0.950	1.050	.0900	.0580
18	0.950	1.050	.0580	.0200
19	0.950	1.050	.0000	.0000
20	0.950	1.050	.0760	.0160
21	0.950	1.050	.0000	.0000
22	0.950	1.050	.2280	.1090

Bus Data (cont.)

Bus no.	V^{min}	V^{max}	Initial Load P	Q
23	0.950	1.050	.0240	.0120
24	0.950	1.050	.0000	.0000
25	0.950	1.050	.0000	.0000
26	0.950	1.050	.2170	.1270
27	0.950	1.050	.9420	.1900
28	0.950	1.050	.3000	.3000
29	0.950	1.050	.0000	.0000
30	0.950	1.050	.0000	.0000

Generation Data

Bus no.	P^{min}	P^{max}	Q^{min}	Q^{max}	Cost coefficients c	a	b
25	0.500	2.000	-0.200	1.500	0.000	200.0	75
26	0.200	0.800	-0.200	0.600	0.000	175.0	350
27	0.150	0.500	-0.150	0.625	0.000	100.0	1250
28	0.100	0.350	-0.150	0.487	0.000	325.0	167
29	0.100	0.300	-0.100	0.400	0.000	300.0	500
30	0.120	0.400	-0.150	0.847	0.000	300.0	500

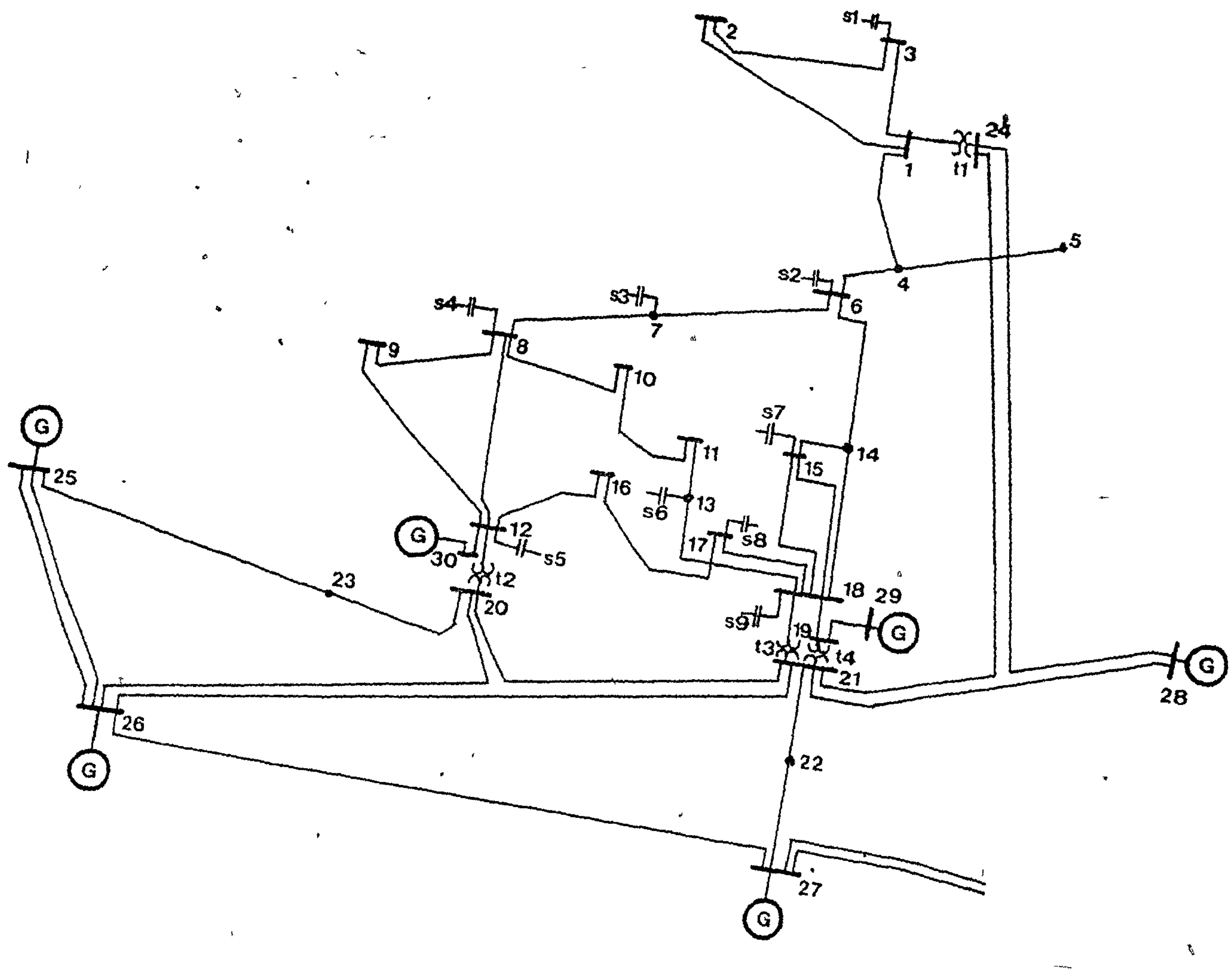


Figure A7.3. One-line diagram of the 30 bus system.

APPENDIX 7.4

DATA FOR THE 118 BUS SYSTEM

Line Data

Line no.	from	to	R	X_{se}	B_{sh}	J^{max}
1	118	076	0.01640D0	0.05440D0	0.01356D0	1.370D0
2	118	075	0.01450D0	0.04810D0	0.01198D0	1.370D0
3	117	012	0.03290D0	0.14000D0	0.03580D0	1.370D0
4	116	068	0.00034D0	0.00405D0	0.16400D0	4.055D0
5	115	114	0.00230D0	0.01040D0	0.00276D0	1.370D0
6	115	027	0.01640D0	0.07410D0	0.01972D0	1.370D0
7	114	032	0.01350D0	0.06120D0	0.01628D0	1.370D0
8	113	032	0.06150D0	0.20300D0	0.05180D0	1.370D0
9	113	017	0.00913D0	0.03010D0	0.00768D0	1.370D0
10	112	110	0.02470D0	0.06400D0	0.06200D0	1.370D0
11	111	110	0.02200D0	0.07550D0	0.02000D0	1.370D0
12	110	109	0.02780D0	0.07620D0	0.02020D0	1.370D0
13	110	103	0.03906D0	0.18130D0	0.04610D0	1.370D0
14	109	108	0.01050D0	0.02880D0	0.00760D0	1.370D0
15	108	105	0.02610D0	0.07030D0	0.01844D0	1.370D0
16	107	106	0.05300D0	0.18300D0	0.04720D0	1.370D0
17	107	105	0.05300D0	0.18300D0	0.04720D0	1.370D0
18	106	105	0.01400D0	0.05470D0	0.01434D0	1.370D0
19	106	100	0.06050D0	0.22900D0	0.06200D0	1.370D0
20	105	104	0.00994D0	0.03780D0	0.00986D0	1.370D0
21	105	103	0.05350D0	0.16250D0	0.04080D0	1.370D0
22	104	103	0.04660D0	0.15840D0	0.04070D0	1.370D0
23	104	100	0.04510D0	0.20400D0	0.05410D0	1.370D0
24	103	100	0.01600D0	0.05250D0	0.05360D0	2.055D0
25	102	101	0.02460D0	0.11200D0	0.02940D0	1.370D0
26	102	092	0.01230D0	0.05590D0	0.01464D0	1.370D0
27	101	100	0.02770D0	0.12620D0	0.03280D0	1.370D0
28	100	099	0.01800D0	0.08130D0	0.02160D0	1.370D0
29	100	098	0.03970D0	0.17900D0	0.04760D0	1.370D0
30	100	094	0.01780D0	0.05800D0	0.06040D0	2.055D0
31	100	092	0.06480D0	0.29500D0	0.07720D0	1.370D0
32	099	080	0.04540D0	0.20600D0	0.05460D0	1.370D0
33	098	080	0.02380D0	0.10800D0	0.02860D0	1.370D0
34	097	096	0.01730D0	0.08850D0	0.02400D0	1.370D0
35	097	080	0.01830D0	0.09340D0	0.02540D0	1.370D0
36	096	095	0.01710D0	0.05470D0	0.01474D0	1.370D0
37	096	094	0.02690D0	0.08690D0	0.02300D0	4.055D0
38	096	082	0.01620D0	0.05300D0	0.05440D0	1.370D0
39	096	080	0.03560D0	0.18200D0	0.04940D0	1.370D0
40	095	094	0.01320D0	0.04340D0	0.01110D0	1.370D0
41	094	093	0.02230D0	0.07320D0	0.01876D0	1.370D0
42	094	092	0.04810D0	0.15800D0	0.04060D0	1.370D0
43	093	092	0.02580D0	0.08480D0	0.02180D0	1.370D0

Line Data (cont.)

Line no.	from	to	R	X_{se}	B_{sh}	J^{max}
44	092	091	0.03870D0	0.12720D0	0.03268D0	1.370D0
45	092	089	0.00799D0	0.03829D0	0.09620D0	1.370D0
46	091	090	0.02540D0	0.08360D0	0.02140D0	1.370D0
47	090	089	0.01638D0	0.06517D0	0.15880D0	1.370D0
48	089	088	0.01390D0	0.07120D0	0.01934D0	1.370D0
49	089	085	0.02390D0	0.17300D0	0.04700D0	1.370D0
50	088	085	0.02000D0	0.10200D0	0.02760D0	1.370D0
51	087	086	0.00000D0	0.20740D0	0.00000D0	2.150D0
52	086	085	0.03500D0	0.12300D0	0.02760D0	1.370D0
53	085	084	0.03020D0	0.06410D0	0.01234D0	1.370D0
54	085	083	0.04300D0	0.14800D0	0.03480D0	1.370D0
55	084	083	0.06250D0	0.13200D0	0.02580D0	1.370D0
56	083	082	0.01120D0	0.03665D0	0.03796D0	1.370D0
57	082	077	0.02980D0	0.08530D0	0.08174D0	1.370D0
58	081	080	0.00000D0	0.03700D0	0.00000D0	4.225D0
59	081	068	0.00175D0	0.02020D0	0.80800D0	2.015D0
60	080	079	0.01560D0	0.07040D0	0.01870D0	1.370D0
61	080	077	0.01088D0	0.03321D0	0.07000D0	2.055D0
62	079	078	0.00546D0	0.02440D0	0.00648D0	1.370D0
63	078	077	0.00376D0	0.01240D0	0.01264D0	1.370D0
64	077	076	0.04440D0	0.14800D0	0.03680D0	1.370D0
65	077	075	0.06010D0	0.19990D0	0.04978D0	1.370D0
66	077	069	0.03090D0	0.10100D0	0.10380D0	1.370D0
67	075	074	0.01230D0	0.04060D0	0.01034D0	1.370D0
68	075	070	0.04280D0	0.14100D0	0.03600D0	1.370D0
69	075	069	0.04050D0	0.12200D0	0.12400D0	1.370D0
70	074	070	0.04010D0	0.13230D0	0.03368D0	1.370D0
71	073	071	0.00866D0	0.04540D0	0.01178D0	1.370D0
72	072	071	0.04460D0	0.18000D0	0.04444D0	1.370D0
73	072	024	0.04880D0	0.19600D0	0.04880D0	1.370D0
74	071	070	0.00882D0	0.03550D0	0.00878D0	1.370D0
75	070	069	0.03000D0	0.12700D0	0.12200D0	1.370D0
76	070	024	0.10221D0	0.41150D0	0.10198D0	1.370D0
77	069	068	0.00000D0	0.03700D0	0.00000D0	6.225D0
78	069	049	0.09850D0	0.32400D0	0.08280D0	1.370D0
79	069	047	0.08440D0	0.27780D0	0.07092D0	1.370D0
80	068	065	0.00138D0	0.01600D0	0.63800D0	4.150D0
81	067	066	0.02240D0	0.10150D0	0.02682D0	1.370D0
82	067	062	0.02580D0	0.11700D0	0.03100D0	1.370D0
83	066	065	0.00000D0	0.03700D0	0.00000D0	6.225D0
84	066	062	0.04820D0	0.21800D0	0.05780D0	1.370D0
85	066	049	0.00900D0	0.04595D0	0.04960D0	2.055D0
86	065	064	0.00269D0	0.03020D0	0.38000D0	6.225D0
87	065	038	0.00901D0	0.09860D0	1.04600D0	2.055D0
88	064	063	0.00172D0	0.02000D0	0.21600D0	4.150D0
89	064	061	0.00000D0	0.02680D0	0.00000D0	6.225D0
90	063	059	0.00000D0	0.03860D0	0.00000D0	2.055D0
91	062	061	0.00824D0	0.03760D0	0.00980D0	2.055D0
92	062	060	0.01230D0	0.05610D0	0.01468D0	1.370D0

Line Data (cont.)

Line no.	from	to	R	X_{se}	B_{sh}	J^{max}
93	061	060	0.00264D0	0.01350D0	0.01456D0	1.370D0
94	061	059	0.03280D0	0.15000D0	0.03880D0	1.370D0
95	060	059	0.03170D0	0.14500D0	0.03760D0	1.370D0
96	059	056	0.04070D0	0.12243D0	0.11050D0	1.370D0
97	059	055	0.04739D0	0.21580D0	0.05646D0	1.370D0
98	059	054	0.05030D0	0.22930D0	0.05980D0	1.370D0
99	058	056	0.03430D0	0.09660D0	0.02420D0	1.370D0
100	058	051	0.02550D0	0.07190D0	0.01788D0	1.370D0
101	057	056	0.03430D0	0.09660D0	0.02420D0	1.370D0
102	057	050	0.04740D0	0.13400D0	0.03320D0	1.370D0
103	056	055	0.00488D0	0.01510D0	0.00374D0	1.370D0
104	056	054	0.00275D0	0.00955D0	0.00732D0	2.055D0
105	055	054	0.01690D0	0.07070D0	0.02020D0	1.370D0
106	054	053	0.02630D0	0.12200D0	0.03100D0	1.370D0
107	054	049	0.03993D0	0.14507D0	0.14680D0	2.055D0
108	053	052	0.04050D0	0.16350D0	0.04058D0	1.370D0
109	052	051	0.02030D0	0.05880D0	0.01396D0	1.370D0
110	051	049	0.04860D0	0.13700D0	0.03420D0	1.370D0
111	050	049	0.02670D0	0.07520D0	0.01874D0	1.370D0
112	049	048	0.01790D0	0.05050D0	0.01258D0	1.370D0
113	049	047	0.01910D0	0.06250D0	0.01604D0	1.370D0
114	049	045	0.06840D0	0.18600D0	0.04440D0	1.370D0
115	049	042	0.03575D0	0.16150D0	0.17200D0	1.370D0
116	048	046	0.06010D0	0.18900D0	0.04720D0	1.370D0
117	047	046	0.03800D0	0.12700D0	0.03160D0	1.370D0
118	046	045	0.04000D0	0.13560D0	0.03320D0	1.370D0
119	045	044	0.02240D0	0.09010D0	0.02240D0	1.370D0
120	044	043	0.06080D0	0.24540D0	0.06068D0	1.370D0
121	043	034	0.04130D0	0.16810D0	0.04226D0	2.055D0
122	042	041	0.04100D0	0.13500D0	0.03440D0	1.370D0
123	042	040	0.05550D0	0.18300D0	0.04660D0	1.370D0
124	041	040	0.01450D0	0.04870D0	0.01222D0	2.055D0
125	040	039	0.01840D0	0.06050D0	0.01552D0	2.055D0
126	040	037	0.05930D0	0.16800D0	0.04200D0	1.370D0
127	039	037	0.03210D0	0.10600D0	0.02700D0	1.370D0
128	038	037	0.00000D0	0.03750D0	0.00000D0	4.150D0
129	038	030	0.00464D0	0.05400D0	0.42200D0	6.225D0
130	037	035	0.01100D0	0.04970D0	0.01318D0	1.370D0
131	037	034	0.00256D0	0.00940D0	0.00984D0	2.055D0
132	037	033	0.04150D0	0.14200D0	0.03660D0	1.370D0
133	036	035	0.00224D0	0.01020D0	0.00268D0	1.370D0
134	036	034	0.00871D0	0.02680D0	0.00568D0	1.370D0
135	034	019	0.07520D0	0.24700D0	0.06320D0	1.370D0
136	033	015	0.03800D0	0.12440D0	0.03194D0	1.370D0
137	032	031	0.02980D0	0.09850D0	0.02510D0	1.370D0
138	032	027	0.02290D0	0.07550D0	0.01926D0	1.370D0
139	032	023	0.03170D0	0.11530D0	0.11730D0	1.370D0
140	031	029	0.01080D0	0.03[illegible]10D0	0.00830D0	1.370D0
141	031	017	0.04740D0	0.15630D0	0.03990D0	1.370D0

Line Data (cont.)

Line no.	from	to	R	X_{se}	B_{sh}	J^{max}
142	030	026	0.00799D0	0.08600D0	0.90800D0	6.225D0
143	030	017	0.00000D0	0.03880D0	0.96000D0	4.225D0
144	030	008	0.00431D0	0.05040D0	0.51400D0	6.225D0
145	029	028	0.02370D0	0.09430D0	0.02380D0	1.370D0
146	028	027	0.01913D0	0.08550D0	0.02160D0	1.370D0
147	027	025	0.03180D0	0.16300D0	0.17640D0	1.370D0
148	026	025	0.00000D0	0.03820D0	0.00000D0	8.000D0
149	025	023	0.01560D0	0.08000D0	0.08640D0	2.055D0
150	024	023	0.01350D0	0.04920D0	0.04980D0	1.370D0
151	023	022	0.03420D0	0.15900D0	0.04040D0	1.370D0
152	022	021	0.02090D0	0.09700D0	0.02460D0	1.370D0
153	021	020	0.01830D0	0.08490D0	0.02160D0	1.370D0
154	020	019	0.02520D0	0.11700D0	0.02980D0	1.370D0
155	019	018	0.01119D0	0.04930D0	0.01142D0	1.370D0
156	019	015	0.01200D0	0.03940D0	0.01010D0	1.370D0
157	018	017	0.01230D0	0.05050D0	0.01298D0	2.055D0
158	017	016	0.04540D0	0.18010D0	0.04660D0	1.370D0
159	017	015	0.01320D0	0.04370D0	0.04440D0	1.370D0
160	016	012	0.02120D0	0.08340D0	0.02140D0	1.370D0
161	015	014	0.05950D0	0.19500D0	0.05020D0	1.370D0
162	015	013	0.07440D0	0.24440D0	0.06268D0	1.370D0
163	014	012	0.02150D0	0.07070D0	0.01816D0	1.370D0
164	013	011	0.02225D0	0.07310D0	0.01876D0	1.370D0
165	012	011	0.00595D0	0.01960D0	0.00502D0	1.370D0
166	012	007	0.00862D0	0.03400D0	0.00874D0	1.370D0
167	012	003	0.04840D0	0.16000D0	0.04060D0	1.370D0
168	012	002	0.01870D0	0.06160D0	0.01572D0	1.370D0
169	011	005	0.02030D0	0.06820D0	0.01738D0	1.370D0
170	011	004	0.02090D0	0.06880D0	0.01748D0	1.370D0
171	010	009	0.00258D0	0.03220D0	1.23000D0	6.225D0
172	009	008	0.00244D0	0.03050D0	1.16200D0	6.225D0
173	008	005	0.00000D0	0.02670D0	0.00000D0	6.225D0
174	007	006	0.00459D0	0.02080D0	0.00550D0	1.370D0
175	006	005	0.01190D0	0.05400D0	0.01426D0	1.370D0
176	005	004	0.00176D0	0.00798D0	0.00210D0	2.055D0
177	005	003	0.02410D0	0.10800D0	0.02840D0	1.370D0
178	003	001	0.01290D0	0.04240D0	0.01082D0	1.370D0
179	002	001	0.03030D0	0.09990D0	0.02540D0	1.370D0

Shunt Data

Shunt no.	bus	B^{min}	B^{max}
1	005	0.000	0.050
2	037	0.000	0.050
3	044	0.000	0.050
4	045	0.000	0.050
5	048	0.000	0.050
6	079	0.000	0.050
7	082	0.000	0.050
8	083	0.000	0.050
9	110	0.000	0.050

Tap Data

Tap no.	on line	near bus	T^{min}	T^{max}
1	51	86	0.800	1.200
2	58	81	0.800	1.200
3	83	65	0.800	1.200
4	89	64	0.800	1.200
5	90	63	0.800	1.200
6	143	30	0.800	1.200
7	148	26	0.800	1.200
8	173	8	0.800	1.200

Phase Shifter Data

Sh. no.	on line	near bus	S^{min}	S^{max}
1	77	68	-0.524	0.524
2	128	38	-0.524	0.524

Bus Data

Bus no.	V^{min}	V^{max}	Initial Load P	Q
1	0.950	1.050	0.5100	0.2700
2	0.950	1.050	0.2000	0.0900
3	0.950	1.050	0.3900	0.1000
4	0.950	1.050	0.3900	0.1200
5	0.950	1.100	0.0000	0.0000
6	0.950	1.050	0.5200	0.2200
7	0.950	1.050	0.1900	0.0200
8	0.950	1.050	0.2800	0.0000
9	0.950	1.100	0.0000	0.0000
10	0.950	1.050	0.0000	0.0000
11	0.950	1.050	0.7000	0.2300
12	0.950	1.050	0.4700	0.1000
13	0.950	1.050	0.3400	0.1600
14	0.950	1.050	0.1400	0.0100
15	0.950	1.050	0.9000	0.3000
16	0.950	1.050	0.2500	0.1000
17	0.950	1.100	0.1100	0.0300
18	0.950	1.050	0.6000	0.3400
19	0.950	1.050	0.4500	0.2500
20	0.950	1.050	0.1800	0.0300
21	0.950	1.050	0.1400	0.0800
22	0.950	1.050	0.1000	0.0500
23	0.950	1.050	0.0700	0.0300
24	0.950	1.050	0.1300	0.0000
25	0.950	1.050	0.0000	0.0000
26	0.950	1.050	0.0000	0.0000
27	0.950	1.050	0.7100	0.1300
28	0.950	1.050	0.1700	0.0700
29	0.950	1.050	0.2400	0.0400
30	0.950	1.100	0.0000	0.0000
31	0.950	1.050	0.3600	0.2700
32	0.950	1.050	0.5900	0.2300
33	0.950	1.050	0.2300	0.0900
34	0.950	1.050	0.5900	0.2600
35	0.950	1.050	0.3300	0.0900
36	0.950	1.050	0.3100	0.1700
37	0.950	1.100	0.0000	0.0000
38	0.950	1.100	0.0000	0.0000
39	0.950	1.050	0.2700	0.1100
40	0.950	1.050	0.6600	0.2300
41	0.950	1.050	0.3700	0.1000
42	0.950	1.050	0.9600	0.2300
43	0.950	1.050	0.1800	0.0700
44	0.950	1.050	0.1600	0.0800
45	0.950	1.050	0.5300	0.2200
46	0.950	1.050	0.0900	0.1000
47	0.950	1.050	0.3400	0.0000

Bus Data (cont.)

Bus no.	V^{min}	V^{max}	Initial Load P	Initial Load Q
48	0.950	1.050	0.2000	0.1100
49	0.950	1.050	0.8700	0.3000
50	0.950	1.050	0.1700	0.0400
51	0.950	1.050	0.1700	0.0800
52	0.950	1.050	0.1800	0.0500
53	0.950	1.050	0.2300	0.1100
54	0.950	1.050	1.1300	0.3200
55	0.950	1.050	0.6300	0.2200
56	0.950	1.050	0.8400	0.1800
57	0.950	1.050	0.1200	0.0300
58	0.950	1.050	0.1200	0.0300
59	0.950	1.050	2.7700	1.1300
60	0.950	1.050	0.7800	0.0300
61	0.950	1.050	0.0000	0.0000
62	0.950	1.050	0.7700	0.1400
63	0.950	1.050	0.0000	0.0000
64	0.950	1.050	0.0000	0.0000
65	0.950	1.100	0.0000	0.0000
66	0.950	1.050	0.3900	0.1800
67	0.950	1.050	0.2[illegible]00	0.0700
68	0.950	1.100	0.0000	0.0000
69	0.950	1.050	0.0000	0.0000
70	0.950	1.050	0.6600	0.2000
71	0.950	1.050	0.0000	0.0000
72	0.950	1.050	0.1200	0.0000
73	0.950	1.050	0.0600	0.0000
74	0.950	1.050	0.6800	0.2700
75	0.950	1.050	0.4700	0.1100
76	0.950	1.050	0.6800	0.3600
77	0.950	1.050	0.6100	0.2800
78	0.950	1.050	0.7100	0.2600
79	0.950	1.050	0.3900	0.3200
80	0.950	1.050	1.3000	0.2600
81	0.950	1.100	0.0000	0.0000
82	0.950	1.050	0.5400	0.2700
83	0.950	1.050	0.2000	0.1000
84	0.950	1.050	0.1100	0.0700
85	0.950	1.050	0.2400	0.1500
86	0.950	1.100	0.2100	0.1000
87	0.950	1.050	0.0000	0.0000
88	0.950	1.050	0.4800	0.1000
89	0.950	1.050	0.0000	0.0000
90	0.950	1.100	1.6300	0.4200
91	0.900	1.050	0.1000	0.0000
92	0.950	1.050	0.6500	0.1000
93	0.950	1.050	0.1200	0.0700
94	0.950	1.050	0.3000	0.1600

Bus Data (cont.)

Bus no.	V^{min}	V^{max}	Initial Load P	Initial Load Q
95	0.950	1.050	0.4200	0.3100
96	0.950	1.050	0.3800	0.1500
97	0.950	1.050	0.1500	0.0900
98	0.950	1.050	0.3400	0.0800
99	0.950	1.050	0.4200	0.0000
100	0.950	1.050	0.3700	0.1800
101	0.950	1.050	0.2200	0.1500
102	0.950	1.050	0.0500	0.0300
103	0.950	1.050	0.2300	0.1600
104	0.950	1.050	0.3800	0.2500
105	0.950	1.050	0.3100	0.2600
106	0.950	1.050	0.4300	0.1600
107	0.950	1.050	0.5000	0.1200
108	0.950	1.050	0.0200	0.0100
109	0.950	1.050	0.0800	0.0300
110	0.950	1.050	0.3900	0.3000
111	0.950	1.050	0.0000	0.0000
112	0.950	1.050	0.6800	0.1300
113	0.950	1.100	0.0600	0.0000
114	0.950	1.050	0.0800	0.0300
115	0.950	1.050	0.2200	0.0700
116	0.950	1.100	1.8400	0.0000
117	0.950	1.050	0.2000	0.0800
118	0.950	1.050	0.3300	0.1000

Generation Data

Bus no.	P^{min}	P^{max}	Q^{min}	Q^{max}	Cost coefficients c	a	b
1	0.700	1.800	-0.230	1.150	0.000	60.73	127.7
4	0.800	2.170	-0.400	1.200	0.000	48.90	78.60
6	0.400	1.080	-999.9	0.750	0.000	69.60	195.6
8	0.800	2.170	-3.500	2.200	0.000	77.30	68.00
10	1.600	4.340	-9.999	9.999	0.000	50.19	45.97
12	0.400	1.080	-0.150	0.750	0.000	80.30	193.2
15	0.300	0.720	-0.800	0.400	0.000	151.3	120.4
18	0.300	0.720	-0.600	1.400	0.000	151.3	120.4
19	0.400	1.080	-0.150	0.750	0.000	136.7	124.6
24	0.300	0.720	-0.400	2.400	0.000	151.3	120.4
25	0.800	2.170	-2.240	7.000	0.000	39.40	78.40
26	1.200	3.240	-6.000	2.250	0.000	63.85	69.99
27	0.300	0.720	-0.080	0.400	0.000	151.3	120.4
31	0.300	0.720	-0.150	0.400	0.000	151.3	120.4
32	0.300	0.720	-0.200	0.400	0.000	151.3	120.4
34	0.400	1.080	-0.400	0.750	0.000	136.7	124.6
36	0.400	1.080	-999.9	999.9	0.000	67.50	206.6
40	0.300	0.720	-999.9	999.9	0.000	151.3	120.4

Generation Data (cont.)

Bus no.	P^{min}	P^{max}	Q^{min}	Q^{max}	Cost coefficients c	a	b
42	0.400	1.080	-999.9	999.9	0.000	136.7	124.6
46	0.300	0.720	-0.080	0.400	0.000	151.3	120.4
49	0.800	2.170	-0.240	1.200	0.000	77.30	68.00
54	0.300	0.720	-08.00	0.400	0.000	151.3	120.4
55	0.400	1.080	-0.150	0.750	0.000	80.30	193.2
56	0.300	0.720	-0.080	0.400	0.000	151.3	120.4
59	0.400	1.080	-0.850	2.000	0.000	67.80	154.6
61	0.400	1.080	-1.650	0.750	0.000	67.80	154.6
62	0.400	1.080	-0.150	0.750	0.000	63.60	201.6
65	0.800	2.160	-8.000	1.500	0.000	46.33	104.1
66	1.200	3.240	-0.450	6.000	0.000	42.13	72.93
69	1.600	4.340	-9.999	9.999	0.000	59.97	39.85
70	0.300	0.720	-999.9	999.9	0.000	151.3	120.4
72	0.300	0.720	-0.080	0.400	0.000	151.3	120.4
73	0.300	0.720	-9.999	9.999	0.000	151.3	120.4
74	0.400	1.080	-0.150	0.750	0.000	80.30	193.2
76	0.400	1.080	-0.150	0.750	0.000	80.30	193.2
77	0.300	0.720	-0.400	0.400	0.000	151.3	120.4
80	1.200	3.250	-999.9	999.9	0.000	31.49	76.87
85	0.400	1.080	-0.150	0.750	0.000	67.80	154.6
87	0.400	1.080	-999.9	999.9	0.000	80.30	193.2
89	1.200	2.250	-0.450	2.200	0.000	58.13	71.76
90	0.800	2.170	-99.99	99.99	0.000	48.90	78.60
91	0.300	0.720	-999.9	999.9	0.000	151.3	120.4
92	0.400	1.080	-9.999	9.999	0.000	136.7	124.6
99	0.300	0.720	-0.080	0.400	0.000	151.3	120.4
100	1.600	4.340	-0.480	2.400	0.000	28.20	46.20
103	1.600	4.340	-0.480	2.400	0.000	36.82	45.98
104	0.400	1.080	-999.9	999.9	0.000	136.7	124.6
105	0.400	1.080	-0.150	0.750	0.000	136.7	124.6
107	0.400	1.080	-0.150	0.750	0.000	67.80	154.6

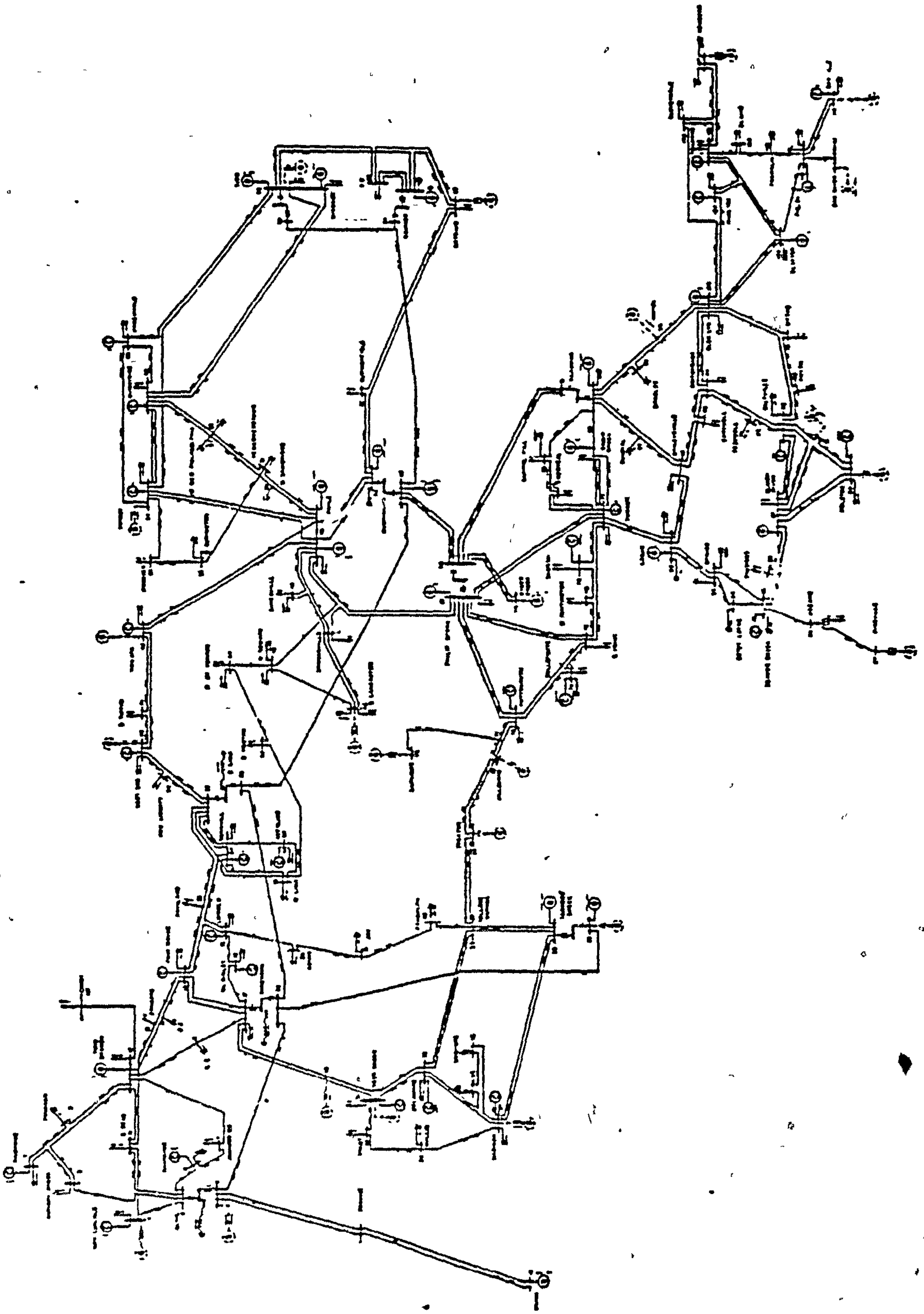


Figure A7.4. One-line diagram of the 118 bus system.

APPENDIX 7.5

THE LINK BETWEEN SYSTEM SPARSITY AND THE SUBPROBLEM ILL-CONDITIONING

We studied the mechanism of computation of the sensitivity coefficients, to show that sparsity and the decoupled nature of the load flow equations result in the very small sensitivity coefficients, and not system size per se. Their computation from elements of the load flow Jacobian, given in Appendix 3.2, requires the solution of a sparse matrix - sparse vector system of linear equations Ax=b. Fig. A7.5. a. to c. show an example of the nonzero pattern of the sparse 186 dimensional matrix in the 118 bus test, and the nonzero patterns of its upper and lower triangular factors computed using subroutine MA28 of the Harwell libraries. The reordered L and U factors regroup variables which have some degree of interaction into many smaller triangular factors. As can be seen, there are many small but relatively decoupled groups of variables. Various right-hand-sides are solved for, corresponding to the different constraints. These hold anywhere from 2 to 12 nonzero values.

In the forward substitution step Ly=b, the few nonzeros in b, denoted b_k, are sure to create nonzero y_k in y. Then they propagate to subsequent variables (i.e. y_{k+i} is nonzero if the nonzero y_k is part of its computation). However, due to the fine segmentation of L, few new nonzeros are actually created.

Beacause of the sparsity patterns of L and U, the x vector fills up with nonzeros in the backward substitution Ux=y. In the forward substitution, the larger number of nonzeros at the bottom of L attracts nonzeros in the last y's. They in turn create nonzeros in the bottom x's. The larger number of nonzeros along the right border of U allow those nonzeros of the bottom x's to propagate almost everywhere.

Our key observation is the following: it seems that by construction, most often the y_k (corresponding to the nonzero b_k) are very small. The derived nonzero y_{k+i} are also very small, because they are computed as the sum of a small number of previously computed small values. Only a few large y_k are computed, corresponding to independent variables in the neighborhood of the dependent variable being modelled. Their influence is usually limited however because they do not propagate far. Then the small nonzero values are transfered to the computation of x in much the same way as described for the y . above.

In summary, when the few y_k in the intermediate solution vector y corresponding to the nonzeros in the right-hand-side b vector are very small, these small values propagate to the solution vector x. This is due to the sparse structures of the A matrix and of the b vector.

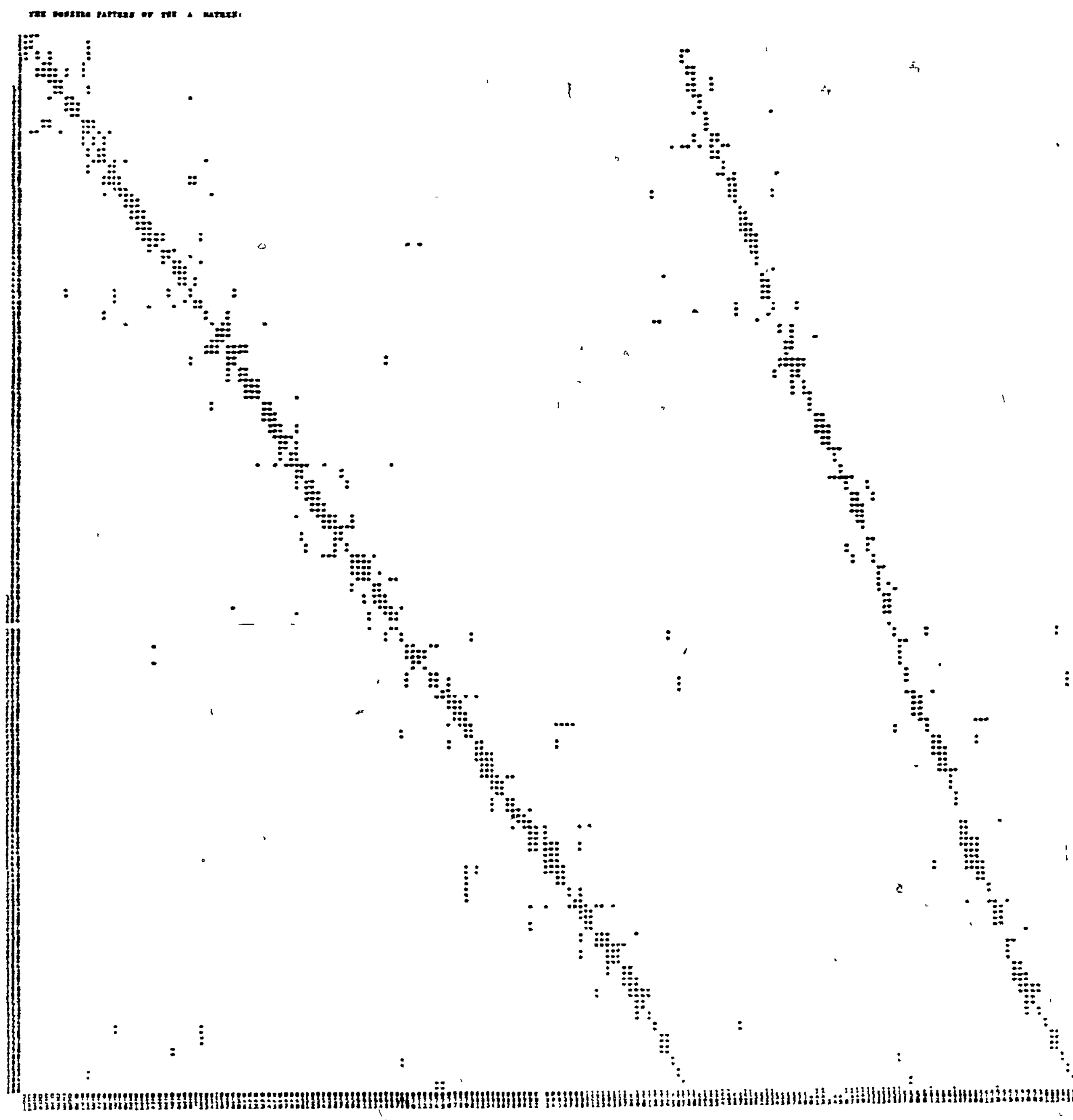


Figure A7.5.a. Nonzero structure of the A matrix.

THE ORDERED LOWER TRIANGULAR FACTOR IS :

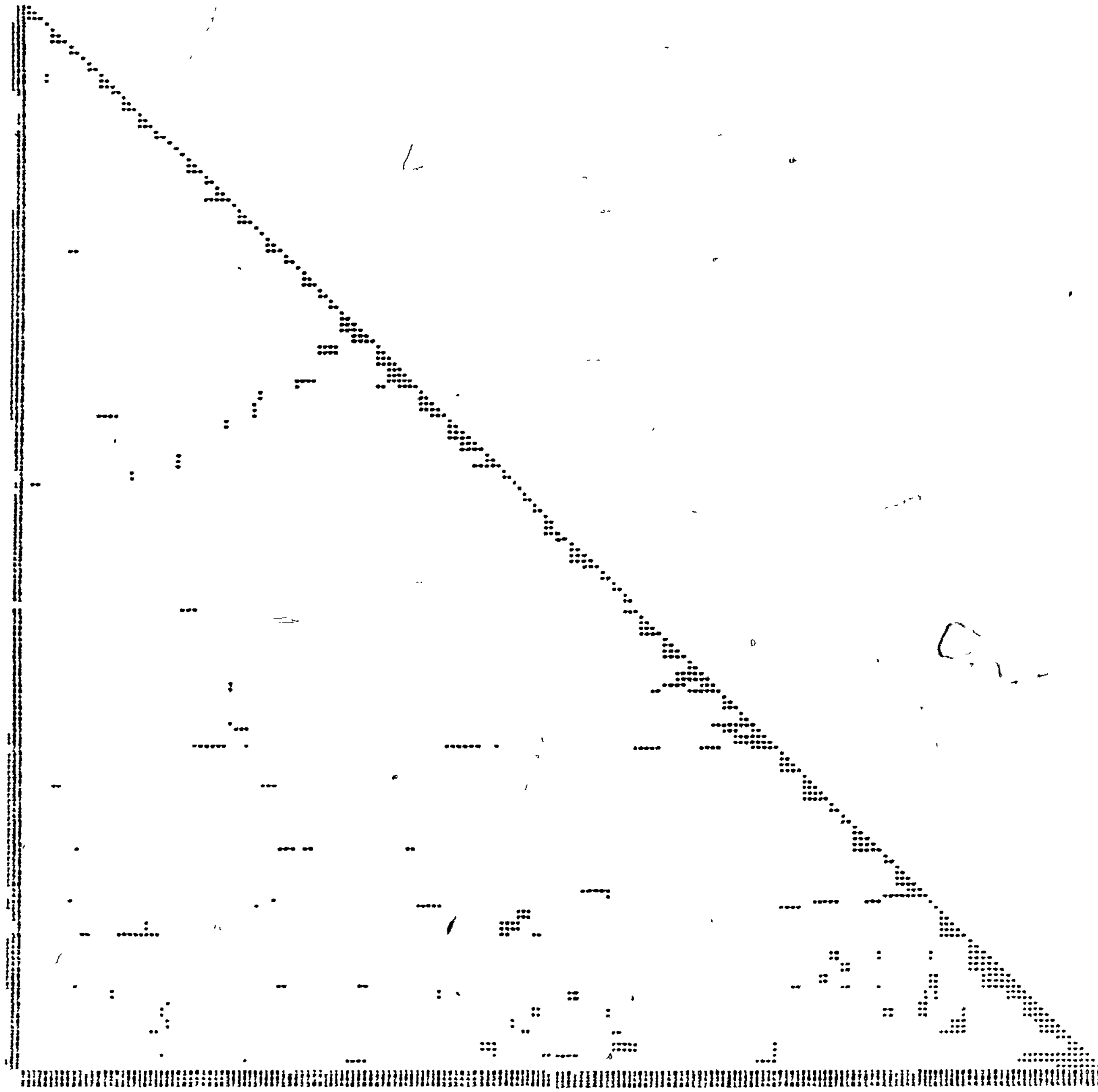


Figure A7.5 b. Nonzero structure of the lower triangular factor L of the A matrix.

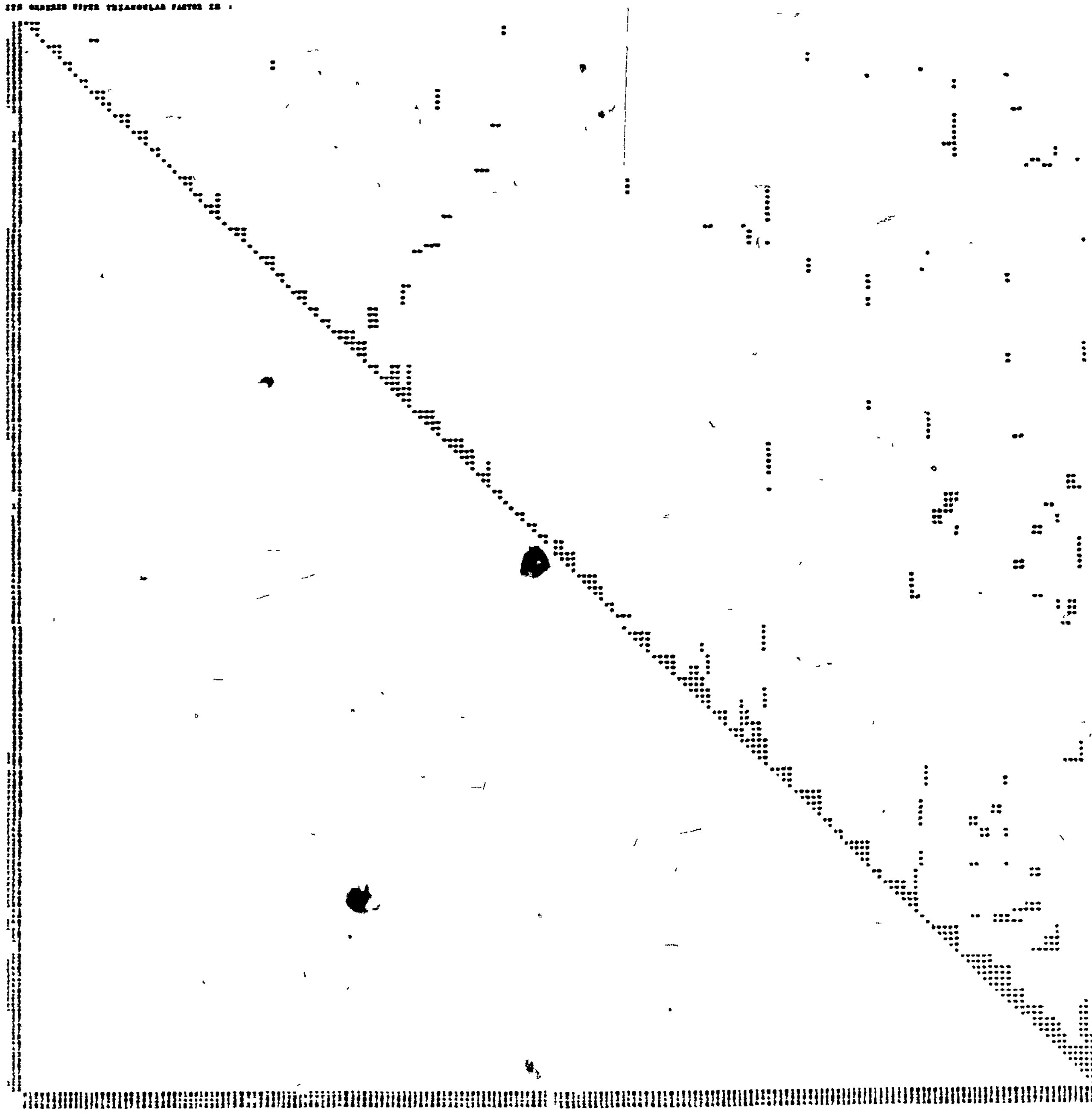


Figure A7.5.c Nonzero structure of the upper triangular factor U of the A matrix.