



National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

AVIS DE MICROFILMAGE

NOTICE OF MICROFILMING

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

A Mathematical Model of
High-Speed Wheel Mobility on Soil for
Application to Aircraft Landing and Take-off

by

Hogan E. Antia



Civil Engineering and Applied Mechanics

McGill University

Montreal, Canada

January 1993

A thesis submitted to the Faculty of Graduate Studies and Research

in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

© Copyright 1993 by Hogan E. Antia



National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file / Votre référence

Our file / Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-87787-1

Canada

Short Title

A MATHEMATICAL MODEL OF HIGH-SPEED WHEEL MOBILITY ON SOIL

Abstract

Motivated by the problem of aircraft and space-shuttle landing on an underlying unpaved soil surface in an emergency, a time-dependent mathematical formulation of high-speed wheel mobility on soil is developed and solved by a control-volume-based finite element method. The wheel-soil contact forces are determined solely on the basis of the physics of the problem. Thus, none of the usual assumptions or simplifications regarding the distribution of interfacial normal and/or shear stresses, albeit in the context of quasi-static wheel mobility, is made. The formulation allows for the incorporation of arbitrary soil constitutive equations, hence strain-rate effects. Soil and wheel inertia are fully accounted for. Consideration is given to the possible existence of a stick-slip phenomenon along the wheel-soil interface.

An object-oriented dynamic wheel-soil interaction C++ computer program is developed and validated. Predictions of soil drag and associated wheel sinkage during simulated aircraft landings are presented. The use of object-oriented programming allows for an easily re-usable and extendable code. Thus the code developed herein can readily be specialized or generalized to solve other practical soil-structure interaction problems.

Résumé

Motivé par la problématique d'atterrissage de la navette spatiale ou des avions sur des surfaces non pavées en cas d'urgence, l'auteur présente un modèle mathématique pour l'historique de la mobilité des roues sur sol à haute vitesse basé sur la méthode des éléments finis à basé de volume de contrôle. Les forces de contact entre la roue et le sol sont déterminées exclusivement en fonction des concepts physiques du problème. Ainsi, aucune supposition n'est nécessaire quant à la distribution des contraintes normales ou de cisaillement, inhérente aux modèles quasi-statiques de mobilité des roues. La formulation permet l'incorporation de relations constitutives générales des sols, et de cette manière, des effets de vitesse de déformation. Les propriétés d'inertie du sol et de la roue sont aussi prises en compte de même que l'existence possible du phénomène de décollement à travers l'interface sol-roue.

Un logiciel orienté objet en langage C++ sur l'interaction dynamique sol-roue a été développé et validé. Des prédictions sur la force de résistance horizontale du sol et sur l'affaissement de la roue pendant la simulation d'atterrissage sont présentées. L'utilisation de la programmation orientée objet permet la création de code à réutilisation et extension relativement simple. Donc, le code développé ici peut être rapidement spécialisé ou généralisé pour l'étude des problèmes pratiques d'interaction sol-structure.

Acknowledgements

I am deeply grateful to all those who, in one way or the other, contributed to the successful completion of this thesis.

Professor Raymond N. Yong, William Scott Professor of Civil Engineering and Applied Mechanics and Director of the Geotechnical Research Center at McGill University, directed my graduate studies and supervised the thesis. His penetrating questions, particularly during the early stages of the investigation, were very illuminating and truly invaluable. His encouragement, support and understanding are also most appreciated.

Research grant for quasi-static wheel mobility tests provided by the Defence Research Establishment Suffield, through contract arrangement with the Geotechnical Research Center, is gratefully acknowledged.

Many thanks to Marcia King who proof-read the thesis and made many useful comments and suggestions.

Finally, I thank my family for their constant support, moral and otherwise.

Table of Contents

Short Title	i
Abstract	ii
Résumé	iii
Acknowledgements	iv
List of Figures	viii
List of Tables	x
List of Symbols	xi
1 Introduction	1
1.1 General	1
1.2 Statement of the Problem	2
1.3 Literature Review	3
1.3.1 Experimentation	3
1.3.2 Empirical Models	3
1.3.3 Mathematical Models	4
1.4 Proposed Mathematical Model	5
1.5 Organization of Thesis	7
References	8
2 Governing Equations	10
2.1 Soil Subsystem	10
2.1.1 Momentum Conservation	10
2.1.2 Mass Conservation	15
2.1.3 Constitutive Equations	19
2.2 Wheel Subsystem	28
2.2.1 Momentum Conservation	28
2.2.2 Mass Conservation	33
2.2.3 Constitutive Equations	34
2.3 Boundary Conditions	34
2.3.1 Inter-Control-Volume Boundary	34
2.3.2 Transmitting Boundary	35
2.3.3 Free-Surface	41
2.3.4 Wheel-Soil Interface	41
References	49

3	Discretization of Governing Equations	52
3.1	Control-Volume Finite Element Method	52
3.2	Domain Discretization	53
3.2.1	Finite Elements and Control Volumes	53
3.3	Interpolation Functions	54
3.4	Numbering Scheme	56
3.4.1	Global	56
3.4.2	Local	58
3.5	Sign convention	60
3.6	Soil Subsystem	62
3.6.1	Momentum Conservation	62
3.6.2	Time Integration	66
3.6.3	Mass Conservation	72
3.6.4	Constitutive Equations	72
3.6.5	Strain-Displacement Equations: Contour Integral Method	75
3.6.6	Hour-Glass Control	77
3.7	Wheel Subsystem	81
3.7.1	Momentum Conservation	81
3.7.2	Time Integration	81
3.7.3	Mass Conservation	81
3.7.4	Constitutive Equations	82
3.8	Boundary Conditions	82
3.8.1	Inter-Control-Volume Boundary	82
3.8.2	Transmitting Boundary	83
3.8.3	Free-Surface	88
3.8.4	Wheel-Soil Interface	88
	References	96
4	Dynamic Grid Generation	98
4.1	Introduction	98
4.2	Regular Domain	99
4.3	Complex Domain	105
4.3.1	Grid-Generation Equations	105
4.3.2	Boundary Conditions	107
4.3.3	Finite-Difference Discretization	108
4.3.4	Solution by Dynamic Relaxation Method	112
	References	124
5	Object-Oriented Program Development	126
5.1	Introduction	126
5.2	Fundamental Concepts	129
5.2.1	Encapsulation	129

5.2.2	Friends	130
5.2.3	Inheritance	132
5.2.4	Virtual Functions	135
5.2.5	Polymorphism	137
5.2.6	Operator Overloading	138
5.2.7	Constructors and Destructors	138
5.3	Program Development	140
5.3.1	Introduction	140
5.3.2	Dynamic Memory Allocation	141
5.3.3	Class Structure for Wheel-Soil Interaction	146
References		150
6	Results	151
6.1	Introduction	151
6.2	Dynamic Grid Generation	152
6.2.1	Inclined Surface	152
6.2.2	Sinusoidal Surface	154
6.2.3	Circular Domain	154
6.2.4	Triangular Domain	154
6.2.5	Discussion	157
6.3	Simulation of Aircraft Landing	158
6.3.1	Problem Parameters	158
6.3.2	Solution	159
References		196
7	Conclusion	197
7.1	Summary and Conclusion	197
7.2	Recommendations for Further Studies	199
7.3	Contributions	200

List of Figures

2.1	Breakdown of body into sub-bodies (material control volumes)	12
2.2	Time-dependent material volume and bounding surface	13
2.3	Three-clement visco-elastic model for soil	21
2.4	Strain rate effect on stress-strain behavior of visco-elastic model	25
2.5	Forces acting on wheel: dynamic equilibrium	30
2.6	Soil thrust, rolling resistance and draw-bar pull	32
2.7	Viscous model of transmitting boundary	40
2.8	Normal and tangential unit vectors at wheel-soil interface	43
2.9	Schematic of stick-slip phenomenon	44
3.1	Discretization of computational domain into: (a) triangular finite elements; (b) polygonal control volumes	54
3.2	Discretization of computational domain into: (a) quadrilateral finite elements; (b) polygonal control volumes	55
3.3	Global numbering scheme for nodes and control volumes	57
3.4	Global numbering scheme for quadrilateral finite elements	58
3.5	Local numbering scheme for nodes of quadrilateral element	59
3.6	Local labelling scheme for control-volume/finite-element intersection	60
3.7	Direction of positive stresses and coordinates	61
3.8	Hour-glass modes for constant-strain quadrilateral element	77
4.1	Packing function variation with intensity factor	103
4.2	Rectangular-mesh generation with intensity factor: (a) 0; (b) .2; (c) .4	104
4.3	Boundary conditions for x and y	111
6.1	Inclined surface domain with packing of grid lines near the surface	153
6.2	Inclined surface domain with packing of grid lines near the surface	153
6.3	Sinusoidal surface domain with packing of grid lines near the surface	155
6.4	Sinusoidal surface domain with packing of grid lines near the surface	155
6.5	Circular domain with no packing of grid lines	156
6.6	Triangular domain with no packing of grid lines	156
6.7	Wheel approaching landing; initial soil grid	164
6.8	Soil drag vs time: soil A	165
6.9	Soil drag vs time: soil B	166
6.10	Soil drag vs time: soil C	167
6.11	Polynomial-fitted soil drag vs time	168

6.12	Wheel sinkage vs time	169
6.13	Translational velocity of wheel vs time	170
6.14	Translational distance of wheel vs time	171
6.15	Rotational velocity of wheel vs time	172
6.16	Wheel-soil contact nodes vs time: soil A	173
6.17	Wheel-soil contact nodes vs time: soil B	174
6.18	Wheel-soil contact nodes vs time: soil C	175
6.19	Wheel-soil contact angle vs time: soil A	176
6.20	Wheel-soil contact angle vs time: soil B	177
6.21	Wheel-soil contact angle vs time: soil C	178
6.22	Wheel-soil contact force at time 0.020 sec: soil A	179
6.23	Wheel-soil contact force at time 0.020 sec: soil B	180
6.24	Wheel-soil contact force at time 0.020 sec: soil C	181
6.25	Grid deformation at time 0.020 sec: soil A	182
6.26	Grid deformation at time 0.020 sec: soil B	183
6.27	Grid deformation at time 0.020 sec: soil C	184
6.28	Contour map of velocity of soil A at time 0.020 sec.	185
6.29	Contour map of velocity of soil B at time 0.020 sec.	186
6.30	Contour map of velocity of soil C at time 0.020 sec.	187
6.31	3D plot of horizontal velocity of soil A at time 0.020 sec.	188
6.32	3D plot of horizontal velocity of soil B at time 0.020 sec.	189
6.33	3D plot of horizontal velocity of soil C at time 0.020 sec.	190
6.34	3D plot of vertical velocity of soil A at time 0.020 sec.	191
6.35	3D plot of vertical velocity of soil B at time 0.020 sec.	192
6.36	3D plot of vertical velocity of soil C at time 0.020 sec.	193
6.37	Effect of wheel radius on soil drag: soil A	194
6.38	Effect of wheel radius on sinkage: soil A	195

List of Tables

5.1	Object Oriented Program Design: Wheel-Soil Interaction	147
6.1	Effect of tolerance level on convergence rate	158
6.2	Soil Parameters	160

List of Symbols

Latin

\vec{a}	Acceleration vector
a_x, a_y	Components of \vec{a} in x and y directions
a_n, a_t	Normal and tangential components of \vec{a}
A	Area
c	Damping coefficient
c_u	Adhesion
c_d	Dilatational wave velocity
c_s	Shear wave velocity
D	Soil drag
DBP	Draw-bar pull
DOS	Direction of slip
$D.O.T$	Direction of travel
E	Young's modulus
E_1, E_2	Young's moduli of viscoelastic model
f	Packing function in i direction
\vec{f}	Force vector

f_x, f_y	Components of \vec{f} in x and y directions
f_n, f_t	Normal and tangential components of \vec{f}
g	Packing function in j direction
\vec{g}	Gravitational acceleration vector
g_x, g_y	Components of \vec{g} in x and y directions
G	Shear modulus
H	Soil thrust
$H(t)$	Heaviside unit function
\vec{i}	Unit vector in x direction
I	Moment of inertia
\vec{j}	Unit vector in y direction
J	Jacobian
ℓ	Length
m	Mass
\vec{n}	Unit normal vector
n_x, n_y	Components of \vec{n} in x and y directions
\vec{p}	Force vector along wheel-soil interface
p_x, p_y	Components of \vec{p} in x and y directions
p_n, p_t	Normal and tangential components of \vec{p}
\vec{P}	Force vector on wheel axle
P_x, P_y	Components of \vec{P} in x and y directions

r	Radius of wheel
R	Rolling resistance
s	slip
S	Scale factor
S	Surface
\bar{t}	Unit tangential vector
t_x, t_y	Components of \bar{t} in x and y directions
t	Time
T	Relaxation time of viscoelastic material
T	Torque
\bar{u}	Displacement vector
u_x, u_y	Components of \bar{u} in x and y directions
\bar{v}	Velocity vector
v_x, v_y	Components of \bar{v} in x and y directions
v_n, v_t	Normal and tangential components of \bar{v}
v_s	Slip velocity
V	Volume
\bar{x}	Position vector
x	Position vector
x, y	Cartesian coordinates
X, Y	Cartesian coordinates (initial)

Greek

α	Angular acceleration
β	Newmark time-integration parameter
γ	Newmark time-integration parameter
ε	Strain tensor
$\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{xy}$	Components of ε
ε	Uniaxial strain
ζ	Direction of slip parameter
η	viscosity of viscoelastic model
Δ	Small change operator
θ	Angle; Circumferential coordinate
θ	Rotation
λ	Lamé elastic constant
μ	Frictional coefficient
ν	Poisson's ratio
π	3.1415926536 ...
ρ	Mass density
$\bar{\sigma}$	Stress vector
σ_x, σ_y	Components of $\bar{\sigma}$ in x and y directions
σ_n, σ_t	Normal and tangential components of $\bar{\sigma}$

σ	Stress tensor
$\sigma_{xx}, \sigma_{yy}, \sigma_{xy}$	Components of σ
σ	Uniaxial stress
τ	Time
ψ	Stress relaxation function
ω	Angular velocity

Calligraphic

\mathcal{F}	Arbitrary function
\mathcal{M}	Conformal module

Mathematical

$\dot{}$	Time derivative; for example, $\dot{u} = du/dt$
\cdot	Dot product
\prime	Derivative with respect to its argument
$ $	Absolute value
$ $	Determinant
\in	Subset
∞	Infinity
\equiv	Equivalent

Others

cs	Control surface
cv	Control volume
e	Finite element
cs.e	Control-surface finite-element intersection
cv.e	Control-volume finite-element intersection

Chapter 1

Introduction

1.1 General

This thesis has been motivated by the problem of aircraft and space-shuttle landing on an underlying unpaved deformable soil surface in an emergency. Aircrafts have traditionally been designed to operate on paved concrete or asphalt runways. Thus, the capability of an aircraft to successfully land on, or take-off from, an unpaved soil surface cannot be taken for granted.

In on-soil landing, the aircraft tires would, at touch-down and during the subsequent ground-roll, penetrate the soil surface and consequently be subjected to a drag force which may greatly exceed that which would be encountered on a paved surface. To insure the structural integrity of the aircraft, and thus avert the potentially huge economic and human cost of a crash, the landing-gear system must be designed to withstand the increased soil drag on the tires.

The primary consideration with regard to aircraft take-off from an unpaved soil surface is the capability of the aircraft power-plant to provide the thrust necessary to overcome the soil drag on the tires and achieve the take-off velocity, subject to such

constraints as the runway length and ground-roll duration.

The accurate prediction of the soil drag is, without doubt, a necessary and fundamental requirement for the analysis and/or design of an aircraft for operational capability (landing, taxiing or take-off) from an unpaved soil terrain. It is noteworthy that, in contrast with the low-speed long-distance steady-state driven wheel mobility characteristic of vehicle operation on soil terrains [1, 2, 3], aircraft ground operation is essentially one of high-speed short-distance transient free-rolling wheel mobility.

Application of a predictive capability of high-speed wheel mobility on soil is not limited to the problem of emergency aircraft or space-shuttle landing. Other applications include forward-area military operations vis-à-vis troop and supply movement, distribution of emergency relief aid to remote regions and prevention of aircraft grounding at war time due to the bombing and cratering of the paved runways.

1.2 Statement of the Problem

This thesis seeks to develop a mathematical model for predicting the soil drag and associated wheel sinkage during transient high-speed wheel mobility on a soil terrain for application to the problem of aircraft landing and take-off from unpaved soil surfaces.

Transient analysis of on-soil wheel mobility is a highly complex, coupled moving-boundary problem. As the wheel advances, contact is established, maintained and eventually broken with various material points (i.e. infinitesimal particles) of the soil surface. The high speed typically encountered in aircraft ground operations makes the wheel and soil inertia as well as the soil strain-rate important considerations.

1.3 Literature Review

1.3.1 Experimentation

In 1964, Boeing Company conducted a series of aircraft taxiing (i.e. uniform speed) operations on the soil terrain at Harper Lake, California. The steady-state soil drag and rut depth were measured at various wheel speeds in the high-speed range typically encountered in aircraft landing and take-off operations. The experimental data [4] indicates a rather complicated variation of soil drag and rut depth with aircraft speed. Further experimental studies under more controllable conditions in a soil bin facility were commissioned by the National Aeronautics and Space Administration (NASA) and the United States Air Force (USAF) in 1967. The results [5, 6] confirmed the test data trend obtained in the Boeing Company studies.

1.3.2 Empirical Models

Richmond *et al.* [4] and Crenshaw [6] developed empirical models based on the concept of dimensionless mobility numbers for predicting soil drag and rut depth in terms of tire speed. Kraft [7] presented linear least-square best-fit equations to available experimental data on soil drag vs rut depth and rut depth vs tire load in the so-called 'region 2 velocity range' where the drag ratio (soil drag/tire load) and rut depth are relatively constant with speed. Pope [8, 9] proposed a model based on rate-controlled plate-bearing tests. Results were, however, presented only for a narrow range of tire speeds.

1.3.3 Mathematical Models

Mathematical models of high-speed wheel mobility on a soil surface are few. In fact, many of the so-called mathematical models are essentially 'curve-fits' of experimental data (e.g. Kraft [7], Crenshaw [6]) or are based on the plate-penetration analogy (e.g. Pope[8, 9], Grahn [11]).

Hovland [12] proposed a mathematical model to determine the contribution of soil inertia to the soil drag. It is based on the concept of a soil wedge being continuously formed and accelerated in front of the wheel. Dagan and Tulin [13] and Karafiath and Sobierajski [14] attempted to solve the dynamic equations of motion of a rigid-plastic soil, subject to prescribed wheel boundary conditions, using the method of characteristics. The formulation of Dagan and Tulin [13] only partially incorporates the soil inertia while that of Karafiath and Sobierajski [14] runs into numerical difficulties at high tire speeds.

The Lockheed-Georgia Co. developed a tire-soil model based on the characterization of the soil as a three-element viscoelastic solid (Crenshaw [10]). Corrective terms were, however, needed to match the qualitative behavior of the predictions with that of the test data. In a fairly recent study, conducted for the United States Air Force by Northrop Corporation, Pi, Yamane and Smith [16] had to modify the Lockheed-Georgia model in order to achieve compatibility between the tire and the soil. Incidentally, they found the modified model to be rather sensitive to the input parameters and consequently deemed it inadequate. The need for a more advanced and reliable model was highlighted. Subsequently, Pi [15] presented a model, based on the quasi-steady motion

of a tire rolling at constant speed on a linear viscoelastic layer (soil), using a kernel function formulation. In this approach, the contact pressure is expressed in terms of the normal and tangential kernel functions, defined as the normal soil surface deformations due, respectively, to a unit-concentrated normal and tangential load moving at *constant* speed. Then, through an iteration process, the soil surface deformations compatible with the given tire load and speed are determined. This model appears to be the most analytically advanced of existing models. Its major drawbacks are the following:

1. wheel is constrained to travel at constant speed.
2. convective soil inertia terms are ignored.
3. arbitrary soil constitutive models cannot be incorporated.

1.4 Proposed Mathematical Model

A comprehensive analytical transient model of dynamic wheel-soil interaction is proposed. Formulation is in terms of the underlying physical principles and laws as embodied in the equations of mass and momentum conservation. (Isothermal conditions are assumed whereupon the energy conservation equation is identically satisfied.)

Soil and wheel inertia are included in full. The formulation allows for the incorporation of arbitrary soil constitutive models. Thus, soil strain-rate effects are readily accommodated. Consideration is also given to the possible existence of the stick-slip phenomenon along the wheel-soil interface.

The wheel-soil interface forces are obtained solely on the basis of the physics of the problem. In other words, no assumptions regarding the distribution of the interfacial shear and/or normal stresses are made. This is particularly significant in light of the overwhelming proportion of researchers who, albeit in the context of quasi-static wheel mobility, presently rely on these simplified assumptions to obtain a solution.

The numerical solution of the governing equations is by a control-volume-based finite element method (CVFEM). This method, which has received the most application in the areas of computational fluid mechanics and heat transfer, enjoys a number of advantages over the conventional finite element method and, in general, yields more accurate results.

An object-oriented programming (OOP) of the control-volume-based finite element method is also proposed. The object-oriented approach is vastly superior to conventional structured programming methods. It results, for example, in the easy re-use and extendability of functional code. Thus arbitrary soil-structure interaction problems can readily be accommodated within the framework of an object-oriented code.

The proposed model is restricted, without loss of generality, to a rigid wheel or highly inflated treadless tire. This restriction has been imposed mainly to facilitate the formulation, implementation and testing of the proposed model without incurring the complexities and uncertainties of any particular tire model [17, 18]. Soil drag and rut depth increase as tire pressure increases. Thus, a rigid wheel would yield an upper bound solution, hence a conservative prediction. It is interesting to note, however,

that a high-pressure tire performs better, i.e. encounters less rolling resistance, than a low-pressure tire on a paved surface [19].

1.5 Organization of Thesis

This thesis is organized into seven (7) chapters as follows:

- 1** introduces the problem and reviews pertinent literature.
- 2** formulates the governing equations and boundary conditions.
- 3** discretizes the governing equations and boundary conditions using a control-volume-based finite element method.
- 4** discretizes the computational domain using body-fitted grid generation techniques within the framework of the dynamic relaxation method.
- 5** proposes an object-oriented programming (OOP) of the wheel-soil interaction and grid-generation equations.
- 6** presents results of numerical computations.
- 7** summarizes and concludes, makes recommendations for further studies and states the contributions of the thesis.

References

- [1] YONG, R. N., FATTAH, E. A. AND SKIADAS, N. *Vehicle Traction Mechanics*. Elsevier Science Publishers, Amsterdam, The Netherlands, 1984.
- [2] KARAFIATH, L. L. AND NOWATZKI, E. A. *Soil Mechanics for Off-Road Vehicle Engineering*. Trans Tech Publishing, Germany, 1978.
- [3] WONG, J. Y. *Theory of Ground Vehicles*. John Wiley, New York, 1978.
- [4] RICHMOND, L. D., BURSKE, N. W., DEBORD, K. J. ET AL. *Aircraft Dynamic Loads from Substandard Landing Sites*. Air Force Flight Dynamics Lab. Report, AFFDL-TR-67-145, The Boeing Co., 1968.
- [5] CRENSHAW, B. M., BUTTERWORTH, C. K., AND TRUESDALE, W. B. *Aircraft Landing Gear Dynamic Loads from Operation on Clay and Sandy Soil*. Air Force Flight Dynamics Lab. Report, AFFDL-TR-69-51, Lockheed-Georgia Co. and IIT Research Institute, 1971.
- [6] CRENSHAW, B. M. Soil/Wheel Interaction at High Speed. *J. Terramechanics*, 8:71-88, 1972.
- [7] KRAFT, D. C. Flootation Performance of Aircraft Tires on Soil Runways. *J. Terramechanics*, 6:47-61, 1969.
- [8] POPE, R. G. The Effect of Sinkage Rate on Pressure Sinkage Relationships and Rolling Resistance in Real and Artificial Clays. *J. Terramechanics*, 6:31-38, 1969.
- [9] POPE, R. G. The Effect of Wheel Speed on Rolling Resistance. *J. Terramechanics*, 8:51-58, 1971.
- [10] CRENSHAW, B. M. *Development of an Analytical Technique to Predict Aircraft Landing Gear/Soil Interaction*. Air Force Flight Dynamics Lab. Report, AFFDL-TR-74-115, Lockheed-Georgia Co., 1975.
- [11] GRAHN, M. Prediction of Sinkage and Rolling Resistance for Off-the-Road Vehicles Considering Penetration Velocity. *J. Terramechanics*, 28(4):339-347, 1991.
- [12] HOVLAND, H. J. Soil Inertia in Wheel-Soil Interaction. *J. Terramechanics*, 10:47-65, 1973.

- [13] DAGAN, G. AND TULIN, M. P. A Study of the Steady Flow of a Rigid-Plastic Clay Beneath a Driven Wheel. *J. Terramechanics*, 6:9-27, 1969.
- [14] KARAFIATH, L. L. AND SOBIERAJSKI, F. S. *Effect of Speed on Tire-Soil Interaction and Development of Towed Pneumatic Tire-Soil Model*. U.S. Army Mobility Systems Lab. Tech. Report, No. 11997 (LL 151), 1974.
- [15] PI, W. S. Dynamic Tire/Soil Contact Surface Interaction Model for Aircraft Ground Operations. *J. Aircraft*, 25:1038-1044, 1988.
- [16] PI, W. S., YAMANE, J. R. AND SMITH, M. J. C. Generic Aircraft Ground Operation Simulation. *27th Structures, Structural Dynamics and Materials Conference*, San Antonio, Texas, May 19-21, 1986.
- [17] TANNER, J. A. (editor) *Tire Modelling*. NASA Conference Publication, CP-2264, 1983.
- [18] CLARK, S. K. (editor) *Mechanics of Pneumatic Tires*. National Bureau of Standards Monograph, No. 122, Washington, D.C, 1971.
- [19] MANNERING, F. L. AND KILARESKI, W. P. *Principles of Highway Engineering and Traffic Analysis*, p 19, Wiley, 1990.

Chapter 2

Governing Equations

The momentum and mass conservation laws, constitutive equations and boundary conditions in dynamic wheel-soil interaction are presented.

2.1 Soil Subsystem

2.1.1 Momentum Conservation

Consider a deforming body B occupying volume $V(t)$ and bounded by surface $S(t)$, at time t . Typically, the body is subject to the following forces:

1) gravitational force

$$\int_{V(t)} \rho \vec{g} dV$$

2) damping force

$$- \int_{V(t)} \rho c \vec{v} dV$$

3) surface force

$$\int_{S(t)} \vec{\sigma} dS$$

where ρ is the mass density, c is the damping factor, \vec{g} is the gravitational acceleration vector, \vec{v} is the velocity vector and $\vec{\sigma}$ is the stress vector.

The momentum of the body is given by

$$\int_{V(t)} \rho \vec{v} dV$$

It follows from Newton's second law that

$$\frac{d}{dt} \int_{V(t)} \rho \vec{v} dV = \int_{V(t)} \rho \vec{g} dV - \int_{V(t)} \rho c \vec{v} dV + \int_{S(t)} \vec{\sigma} dS \quad (2.1)$$

It can be shown [4] that

$$\frac{d}{dt} \int_{V(t)} \rho \mathcal{F} dV \equiv \int_{V(t)} \rho \frac{d\mathcal{F}}{dt} dV$$

where \mathcal{F} is an arbitrary scalar or vector function. Thus eq. (2.1) may be written in the form

$$\int_{V(t)} \rho \frac{d\vec{v}}{dt} dV = \int_{V(t)} \rho \vec{g} dV - \int_{V(t)} \rho c \vec{v} dV + \int_{S(t)} \vec{\sigma} dS \quad (2.2)$$

In *Eulerian-differential* formulations (e.g [7, 12]), the particle acceleration $d\vec{v}/dt$ is often replaced by the *spatial* acceleration $\partial\vec{v}/\partial t$. Using the chain rule of partial differentiation, one obtains the following relationship:

$$\frac{d\vec{v}}{dt} = \frac{\partial\vec{v}}{\partial t} + \vec{v} \cdot \vec{\nabla} \vec{v} \quad (2.3)$$

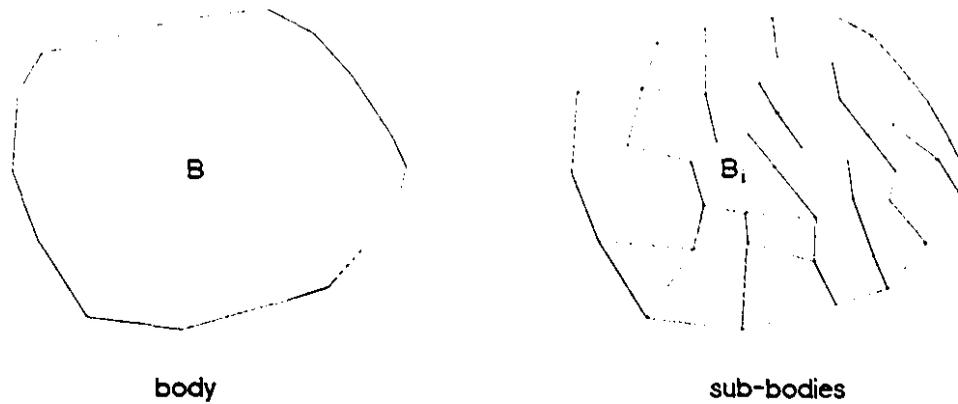


Figure 2.1: Breakdown of body into sub-bodies (material control volumes)

where $\vec{\nabla}$ is the differential vector operator, *dcl.* Thus, the use of $\partial\vec{v}/\partial t$ rather than $d\vec{v}/dt$ essentially ignores the convective inertia terms $\vec{v} \cdot \vec{\nabla}\vec{v}$. The *Lagrangian-integral* formulation, eq. (2.2), incorporates the inertia terms in full.

Let us mentally break down the body B into n smaller, but arbitrary, sub-bodies $B_1, \dots, B_i, \dots, B_n$, as illustrated in Fig. 2.1. These bodies may be conceptualized as *moving* or *material* control-volumes. It will be recalled that in classical fluid mechanics, one generally deals with *spatial* control-volumes which are fixed in space. Let the material volume and bounding surface, at time t , of an arbitrary sub-body B_i be denoted $V_i(t)$ and $S_i(t)$, respectively, as illustrated in Fig. 2.2. The development leading to the momentum conservation equation (2.2) is applicable to each of the sub-bodies.

Thus,

$$\int_{V_i(t)} \rho \frac{d\vec{v}}{dt^2} dV = \int_{V_i(t)} \rho \vec{g} dV - \int_{V_i(t)} \rho c \vec{v} dV + \int_{S_i(t)} \vec{\sigma} dS \quad (2.4)$$

for

$$i = 1, 2, \dots, n$$

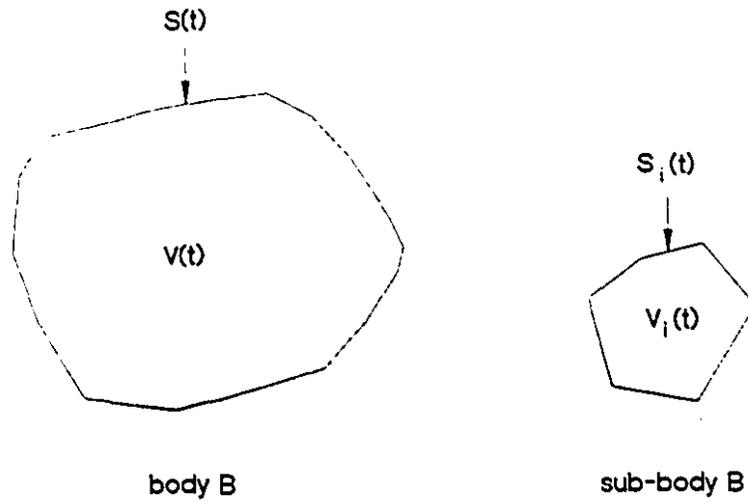


Figure 2.2: Time-dependent material volume and bounding surface

Clearly,

$$\sum_{i=1}^n \int_{V_i(t)} \mathcal{F} dV = \int_{V(t)} \mathcal{F} dV \quad (2.5)$$

for

$$\mathcal{F} = \rho \frac{d^2 \bar{u}}{dt^2}, \quad -\rho c \frac{d\bar{u}}{dt}, \quad \rho \bar{g}$$

By virtue of Newton's Third Law—action and reaction are equal but opposite—the inter-body stress vectors cancel out. Consequently,

$$\sum_{i=1}^n \int_{S_i(t)} \bar{\sigma} dS = \int_{S(t)} \bar{\sigma} dS \quad (2.6)$$

Equations (2.5) and (2.6) constitute what is known as the *conservative property*. It is desirable that the discretized conservation equations possess the conservative property, as explained in the next chapter.

In a two-dimensional formulation in the Cartesian xy plane, an arbitrary vector

quantity \vec{v} may be expressed in the form

$$\vec{v} = v_x \vec{i} + v_y \vec{j} \quad (2.7)$$

where \vec{i} and \vec{j} are the unit vectors along the x and y axes, respectively, and

$$v_x = \vec{v} \cdot \vec{i} \quad (2.8)$$

$$v_y = \vec{v} \cdot \vec{j} \quad (2.9)$$

Thus, expressing \vec{g} and $\vec{\sigma}$ in the form of eq. (2.7) and substituting into eq. (2.4), one obtains

$$\int_{V_i(t)} \rho \frac{dv_x}{dt^2} dV = \int_{V_i(t)} \rho g_x dV - \int_{V_i(t)} \rho c v_x dV + \int_{S_i(t)} \sigma_x dS \quad (2.10)$$

$$\int_{V_i(t)} \rho \frac{dv_y}{dt^2} dV = \int_{V_i(t)} \rho g_y dV - \int_{V_i(t)} \rho c v_y dV + \int_{S_i(t)} \sigma_y dS \quad (2.11)$$

Equations (2.10) and (2.11) are coupled through the mass conservation and constitutive equations.

Kinematics

The particle displacement \vec{u} , velocity \vec{v} and acceleration \vec{a} are related kinematically, as follows:

$$\vec{v} = \frac{d\vec{u}}{dt} \quad (2.12)$$

$$\vec{a} = \frac{d\vec{v}}{dt} \quad (2.13)$$

The *initial conditions* for solving eqs. (2.13) and (2.12) for \vec{v} and \vec{u} , in terms of \vec{a} , are provided in the form:

$$\vec{v}(0) = \vec{v}_0 \quad (2.14)$$

$$\vec{u}(0) = \vec{u}_0 \quad (2.15)$$

The acceleration \vec{a} is determined from the momentum conservation equation.

2.1.2 Mass Conservation

The mass of body B is given by

$$\int_{V(t)} \rho dV \quad (2.16)$$

By the law of conservation of mass

$$\frac{d}{dt} \int_{V(t)} \rho dV = 0 \quad (2.17)$$

Reynolds' Transport Theorem [4, 3] states that for an arbitrary scalar or vector function \mathcal{F} ,

$$\frac{d}{dt} \int_{V(t)} \mathcal{F} dV = \int_{V(t)} \left(\frac{d\mathcal{F}}{dt} + \mathcal{F} \vec{\nabla} \cdot \vec{v} \right) dV \quad (2.18)$$

Thus, substituting ρ for \mathcal{F} in eq. (2.18), the mass conservation equation (2.17) may be expressed in the form

$$\int_{V(t)} \left(\frac{d\rho}{dt} + \rho \bar{\nabla} \cdot \bar{v} \right) dV = 0 \quad (2.19)$$

Clearly, the preceding analysis is applicable to each of the bodies B_1, B_2, \dots, B_n .

Thus,

$$\int_{V_i(t)} \left(\frac{d\rho}{dt} + \rho \bar{\nabla} \cdot \bar{v} \right) dV = 0 \quad (2.20)$$

for

$$i = 1, 2, \dots, n$$

Equations (2.19) and (2.20) are identically satisfied by

$$\frac{d\rho}{dt} + \rho \bar{\nabla} \cdot \bar{v} = 0 \quad (2.21)$$

It can be shown (e.g. [4, 3]) that

$$\bar{\nabla} \cdot \bar{v} = \frac{1}{J} \frac{dJ}{dt} \quad (2.22)$$

where J is the Jacobian. Substituting eq. (2.22) into eq. (2.21), one obtains

$$\frac{d(\rho J)}{dt} = 0 \quad (2.23)$$

Equation (2.23) is a *Lagrangian-differential* form of the mass conservation equation.

Integrating eq. (2.23), one obtains

$$\rho J = \text{constant} \quad (2.24)$$

where J is the *Jacobian* of the current coordinates x, y with respect to the initial coordinates X, Y . In symbolic notation,

$$J = \frac{\partial(x, y)}{\partial(X, Y)} \quad (2.25)$$

$$\frac{\partial(x, y)}{\partial(X, Y)} = \begin{vmatrix} \frac{\partial x}{\partial X} & \frac{\partial x}{\partial Y} \\ \frac{\partial y}{\partial X} & \frac{\partial y}{\partial Y} \end{vmatrix} \quad (2.26)$$

It can be shown [17] that

$$\frac{\partial(x, y)}{\partial(X, Y)} \frac{\partial(X, Y)}{\partial(x, y)} = 1 \quad (2.27)$$

Thus

$$\frac{1}{J} = \frac{\partial(X, Y)}{\partial(x, y)} \quad (2.28)$$

$$= \begin{vmatrix} \frac{\partial X}{\partial x} & \frac{\partial X}{\partial y} \\ \frac{\partial Y}{\partial x} & \frac{\partial Y}{\partial y} \end{vmatrix} \quad (2.29)$$

Substituting

$$X = x - u_x \quad (2.30)$$

$$Y = y - u_y \quad (2.31)$$

into eq. (2.29), we obtain

$$\frac{1}{J} = \begin{vmatrix} 1 - \frac{\partial u_x}{\partial x} & -\frac{\partial u_x}{\partial y} \\ -\frac{\partial u_y}{\partial x} & 1 - \frac{\partial u_y}{\partial y} \end{vmatrix} \quad (2.32)$$

$$= \left(1 - \frac{\partial u_x}{\partial x}\right) \left(1 - \frac{\partial u_y}{\partial y}\right) - \frac{\partial u_x}{\partial y} \frac{\partial u_y}{\partial x} \quad (2.33)$$

If the initial displacement gradients are zero, that is,

$$\frac{\partial u_x}{\partial x}(0) = \frac{\partial u_y}{\partial y}(0) = \frac{\partial u_x}{\partial y}(0) = \frac{\partial u_y}{\partial x}(0) = 0 \quad (2.34)$$

then $J(0) = 1$ and the 'constant' in eq. (2.24) equals the initial density $\rho(0)$ and

$$\rho = \frac{\rho(0)}{J} \quad (2.35)$$

Substituting eq. (2.33) into eq. (2.35), one obtains

$$\rho = \rho(0) \left[1 - \left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_x}{\partial y} \frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial x} \frac{\partial u_y}{\partial y} \right) \right] \quad (2.36)$$

If the displacement gradients are small such that their products are negligible compared to unity, as in ‘small strain’ theory, then

$$\rho = \rho_0 [1 - (\varepsilon_{xx} + \varepsilon_{yy})] \quad (2.37)$$

where

$$\varepsilon_{xx} = \frac{\partial u_x}{\partial x} \quad (2.38)$$

$$\varepsilon_{yy} = \frac{\partial u_y}{\partial y} \quad (2.39)$$

In the sign convention used, strains are positive in extension and negative in compression.

2.1.3 Constitutive Equations

Momentum and mass conservation equations are not sufficient to uniquely determine the response of a material to a given forcing function. This is evident from physical as well as mathematical considerations. The fact that different materials do not have the same mechanical response to an identical forcing function clearly provides a physical basis for an additional equation. Mathematically, the conservation equations are under-determined — there are more unknowns than equations. The additional mathematical and physical requirements for a unique solution are met by the constitutive, i.e. stress-strain, equation of the material.

Soil is a non-linear material with complex stress-strain characteristics. Mathematical analyses of dynamics problems are further complicated by the rate-effects often exhibited by soils. There are two general view-points on the incorporation of the constitutive behavior of soils into a theoretical analysis (Whitman [15]).

One view-point calls for the development and use of a comprehensive mathematical stress-strain model to ensure a complete and accurate analysis. The cap, critical state, Lade's, nested yield-surface, bounding-surface and endochronic models ([18, 16]) fall into this category. In general, these 'advanced models' are not attractive for most practical work because of their complexity and the relatively large number of input parameters required.

The other view-point is that a simple linearized model which simulates the stress-strain features of key importance for the particular problem at hand be used. The advantage of this approach is the ease with which computations can be performed. Furthermore, with much fewer input parameters, the engineer quickly develops a feel for their significance and role. In effect, the engineer's judgement quickly becomes an important part of the overall analysis. For these reasons, the 'simple model' approach is more common among engineering practitioners. The simple models often used include the linear visco-elastic model, linear hysteretic model, bilinear yielding model (Whitman[15]) and the hyperbolic model [18, 16].

Kocher and Summers[10] and others have found that visco-elastic models adequately describe the dynamic stress-strain behavior of soils. Whilst various forms of visco-elastic models are available, we opt for the three-element model consisting of:

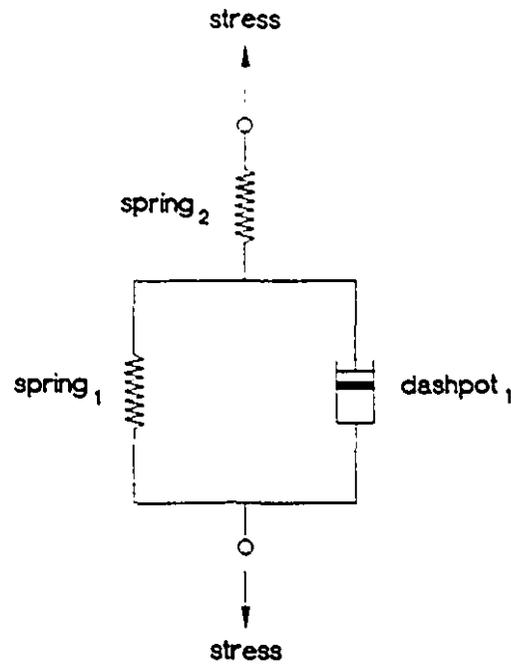


Figure 2.3: Three-element visco-elastic model for soil

1. a spring of modulus E_1 in parallel with a dashpot of viscosity η_1 , and
2. a spring of modulus E_2 in series with 1.

This model, illustrated in Fig 2.3, is known as the standard linear solid and has been used, in the context of wheel-soil interaction, by many researchers (e.g Pi[7], Oida[9], Hiroma and Ota[25]). The model is relatively simple, conceptually and implementationally. Model parameters may be determined using wave propagation theory [11, 10] or in the manner presented by Oida[9].

The stress-strain equation of this model under uniaxial, i.e. one-dimensional, conditions may be written as ([2, 9])

$$(E_1 + E_2)\sigma + \eta_1 \dot{\sigma} = E_1 E_2 \varepsilon + E_2 \eta_1 \dot{\varepsilon} \quad (2.40)$$

where σ and ε are the uniaxial stress and strain, respectively, and the superposed dot denotes time differentiation.

Techniques for solving the first-order ordinary differential equation (2.40) include the Laplace Transform and the integrating factor methods. It is noteworthy that the standard linear solid exhibits a non-linear stress-strain behavior. Consider, for example, a constant rate of straining, represented by the forcing function

$$\varepsilon(t) = \dot{\varepsilon}t \quad (2.41)$$

$$\dot{\varepsilon} = \text{constant}$$

Substituting eq. (2.41) into eq. (2.40), and re-arranging, one obtains

$$\frac{d\sigma}{dt} + \frac{E_1 + E_2}{\eta_1} \sigma = \frac{E_1 E_2}{\eta_1} \dot{\varepsilon}t + E_2 \dot{\varepsilon} \quad (2.42)$$

An integration factor for eq. (2.42) is

$$\begin{aligned} \exp\left(\int \frac{E_1 + E_2}{\eta_1} dt\right) &= \exp\left(\frac{E_1 + E_2}{\eta_1} t\right) \\ &= e^{t/T} \end{aligned} \quad (2.43)$$

where

$$T = \frac{\eta_1}{E_1 + E_2} \quad (2.44)$$

Multiplying eq. (2.42) by $e^{t/T}$, one obtains

$$\frac{d(\sigma e^{t/T})}{dt} = \frac{E_1 E_2}{\eta_1} \dot{\varepsilon} t e^{t/T} + E_2 \dot{\varepsilon} e^{t/T} \quad (2.45)$$

Integrating eq. (2.45) with respect to t yields

$$\sigma e^{t/T} = \frac{E_1 E_2}{\eta_1} \dot{\varepsilon} T (t - T) e^{t/T} + E_2 \dot{\varepsilon} T e^{t/T} + C \quad (2.46)$$

where C is the constant of integration. The initial condition

$$\sigma(0) = 0 \quad (2.47)$$

yields

$$C = \frac{E_1 E_2}{\eta_1} \dot{\varepsilon} T^2 - E_2 \dot{\varepsilon} T \quad (2.48)$$

Substituting eq. (2.48) into eq. (2.46), and re-arranging, one obtains

$$\sigma = E_2 \dot{\varepsilon} \left\{ T \left(1 - \frac{E_1 T}{\eta_1} \right) (1 - e^{-t/T}) + \frac{E_1 T t}{\eta_1} \right\} \quad (2.49)$$

From eq. (2.41),

$$t = \frac{\varepsilon}{\dot{\varepsilon}} \quad (2.50)$$

which, on substituting into eq. (2.49), yields σ in terms of ε :

$$\sigma = E_2 \left\{ T \left(1 - \frac{E_1 T}{\eta_1} \right) \left[1 - \exp \left(-\frac{\varepsilon}{T \dot{\varepsilon}} \right) \right] \dot{\varepsilon} + \frac{E_1 T \varepsilon}{\eta_1} \right\} \quad (2.51)$$

We note that

$$\begin{aligned} 1 - \frac{E_1 T}{\eta_1} &= 1 - \frac{E_1}{E_1 + E_2} \\ &> 0 \end{aligned} \quad (2.52)$$

The slope of the σ vs ε curve is given by

$$\frac{d\sigma}{d\varepsilon} = E_2 \left\{ \left(1 - \frac{E_1 T}{\eta_1} \right) \exp \left(-\frac{\varepsilon}{T \dot{\varepsilon}} \right) + \frac{E_1 T}{\eta_1} \right\} \quad (2.53)$$

The initial slope, at $\varepsilon = 0$, is given by

$$\frac{d\sigma}{d\varepsilon} = E_2 \quad (2.54)$$

Note that it is independent of $\dot{\varepsilon}$. We note also that as $\varepsilon \rightarrow \infty$,

$$\frac{d\sigma}{d\varepsilon} \rightarrow \frac{E_1 T}{\eta_1} = \frac{E_1}{E_1 + E_2} \quad (2.55)$$

A qualitative plot of the stress vs strain vs strain-rate relationship, eq. (2.51), is shown in Fig. 2.4. The stress corresponding to a given strain increases with the strain rate.

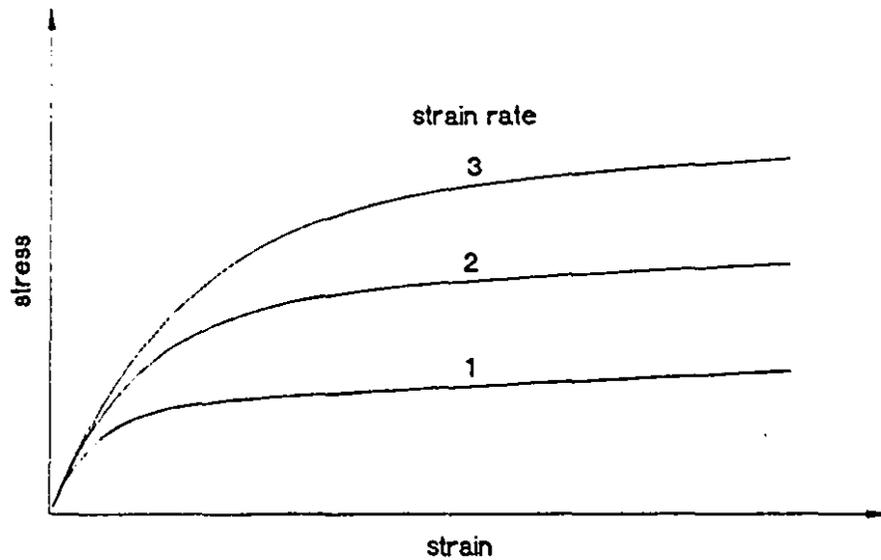


Figure 2.4: Strain rate effect on stress-strain behavior of visco-elastic model

In rate-controlled plate-penetrometer tests conducted by Grahn[31, 32], the pressure-sinkage-velocity curves exhibit a similar trend to the viscoelastic model prediction (Fig. 2.4). The strain rates 1, 2 and 3 indicated in Fig. 2.4 are to be viewed as relative, rather than absolute, quantities. To derive the constitutive relationship under general loading functions, we consider the forcing function

$$\varepsilon(t) = H(t) = \begin{cases} 0 & \text{if } t < 0 \\ 1 & \text{otherwise} \end{cases}$$

It is easy to show that

$$\sigma(t) = \psi(t) = \frac{E_2}{E_1 + E_2} (E_1 + E_2 e^{-t/T}) \quad (2.56)$$

$H(t)$ is the so-called *Heaviside unit function* and T , defined in eq. (2.44), is the re-

relaxation time of the material. $\psi(t)$ —the *stress-relaxation function*—is the stress that must be applied to produce a strain which changes at $t = 0$ from zero to unity and remains unity thereafter.

By virtue of *Boltzmann's Superposition Principle* (e.g [6, 1]), the stress $\sigma(t)$ due to a given arbitrary strain history $\varepsilon(t)$ may be expressed as

$$\sigma(t) = \int_{-\infty}^t \psi(t - \tau) \frac{d\varepsilon(\tau)}{d\tau} d\tau \quad (2.57)$$

On integrating by parts, eq. (2.57) takes the form

$$\sigma(t) = E_2 \left[\varepsilon(t) - \int_{-\infty}^t K(t - \tau) \varepsilon(\tau) d\tau \right] \quad (2.58)$$

where

$$K(t - \tau) = \frac{E_2}{\eta_1} e^{-(t-\tau)/T} \quad (2.59)$$

and $\varepsilon(-\infty) = 0$.

The corresponding stress-strain equations under plane-strain conditions are

$$\sigma_{xx}(t) = \frac{E_2(1 - \nu)}{(1 + \nu)(1 - 2\nu)} \left\{ \varepsilon_{xx}(t) + \frac{\nu}{1 - \nu} \varepsilon_{yy}(t) - \int_{-\infty}^t K(t - \tau) \left[\varepsilon_{xx}(\tau) - \frac{1}{2} \varepsilon_{yy}(\tau) \right] d\tau \right\} \quad (2.60)$$

$$\sigma_{yy}(t) = \frac{E_2(1 - \nu)}{(1 + \nu)(1 - 2\nu)} \left\{ \varepsilon_{yy}(t) + \frac{\nu}{1 - \nu} \varepsilon_{xx}(t) \right\}$$

$$- \int_{-\infty}^t K(t - \tau) \left[\varepsilon_{yy}(\tau) - \frac{1}{2} \varepsilon_{xx}(\tau) \right] d\tau \} \quad (2.61)$$

$$\sigma_{xy}(t) = 2G \left\{ \varepsilon_{xy}(t) - \frac{3(1 - \nu)}{2(1 - 2\nu)} \int_{-\infty}^t K(t - \tau) \varepsilon_{xy}(\tau) d\tau \right\} \quad (2.62)$$

where

$$G = \frac{E_2}{2(1 + \nu)} \quad (2.63)$$

is the shear modulus. Pi[7] treated E_2 , ν and G as three independent parameters because of the relatively low G -value of the soil used. In the derivation of eqs. (2.60)–(2.62), the time-dependence of the volumetric strain is ignored, i.e., the volumetric strain is assumed to be purely elastic ([8, 7]). The unloading behavior of the soil may be modelled by using a smaller value of E_2 .

The strains ε_{xx} , ε_{yy} and ε_{xy} are assumed to be small, in accordance with the linearity requirement of the superposition principle. Thus, the *infinitesimal* strain-displacement equations are applicable:

$$\varepsilon_{xx} = \frac{\partial u_x}{\partial x} \quad (2.64)$$

$$\varepsilon_{yy} = \frac{\partial u_y}{\partial y} \quad (2.65)$$

$$\varepsilon_{xy} = \frac{1}{2} \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right) \quad (2.66)$$

We note, and emphasize, that the wheel-soil interaction model proposed in this thesis is not constrained to the linear visco-elastic model but admits arbitrary consti-

tutive models.

2.2 Wheel Subsystem

2.2.1 Momentum Conservation

Consider a wheel undergoing simultaneous translational and rotational motion on a soil surface under the action of the following force system:

1) wheel-soil contact stress

$$\bar{\sigma}$$

2) gravitational force

$$m\bar{g}$$

3) axle force

$$\bar{P}$$

4) axle torque

$$T$$

The linear momentum of the wheel is

$$m \frac{d\bar{v}}{dt}$$

The equation of linear momentum conservation (Newton's Second Law), governing the translational motion of the wheel, is given by

$$m \frac{d\bar{v}}{dt} = \int_{S(t)} \bar{\sigma} dS + m\bar{g} + \bar{P} \quad (2.67)$$

where $S(t)$ is the wheel-soil contact surface. In component form,

$$m \frac{dv_x}{dt} = \int_{S(t)} \sigma_x dS + mg_x + P_x \quad (2.68)$$

$$m \frac{dv_y}{dt} = \int_{S(t)} \sigma_y dS + mg_y + P_y \quad (2.69)$$

The rotation of the wheel—about an axis perpendicular to the xy plane—is governed by the equation of angular momentum conservation:

$$I \frac{d\omega}{dt} = r \int_{S(t)} \sigma_t dS + T \quad (2.70)$$

where r is the radius of the wheel, I is the moment of inertia of the wheel about its axis of rotation, ω is angular velocity of the wheel and σ_t is the tangential stress:

$$\sigma_t = \bar{\sigma} \cdot \bar{t} \quad (2.71)$$

where \bar{t} is the tangential unit vector. The force system acting on the wheel and the dynamic equilibrium of the wheel, embodied in eqs. (2.68), (2.69) and (2.70), is illustrated in Fig. 2.5. A superposed dot indicates time differentiation; for example, $\dot{u}_x = du_x/dt$.

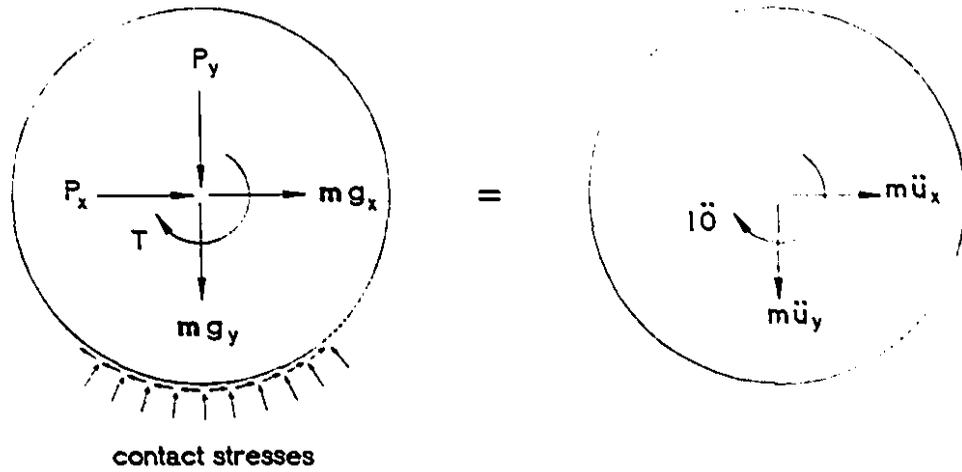


Figure 2.5: Forces acting on wheel: dynamic equilibrium

Let the *direction of travel*, D.O.T, of the wheel be \bar{i} . Thus, a *driving* torque, *braking* torque and *free-rolling* wheel are characterised by T greater than, less than and equal to zero, respectively. The integral of σ_x over those points of the wheel-soil interface where σ_x is positive is referred to as the *soil thrust* H while the integral over those points where σ_x is negative represents the *rolling resistance* R or *soil drag* D , that is,

$$H = \int_{S^+(t)} \sigma_x dS \quad (2.72)$$

$$R = D = \int_{S^-(t)} \sigma_x dS \quad (2.73)$$

where

$$S(t) = S^+(t) + S^-(t) + S^0(t) \quad (2.74)$$

$$\sigma_x > 0 \quad \text{on } S^+(t) \quad (2.75)$$

$$\sigma_x < 0 \quad \text{on } S^-(t) \quad (2.76)$$

$$\sigma_x = 0 \quad \text{on } S^0(t) \quad (2.77)$$

The *draw-bar pull*, DBP , is the force acting on the wheel axle, in the direction opposite the direction of travel of the wheel. Thus,

$$DBP = -P_x \quad (2.78)$$

$$= H - R - m \frac{dv_x}{dt} + mg_x \quad (2.79)$$

The draw-bar pull is a measure of the amount of force that can be pulled along by the wheel. If the wheel is travelling at constant speed ($\dot{v}_x = 0$) in the horizontal plane ($g_x = 0$), eq. (2.79) reduces to

$$DBP = H - R \quad (2.80)$$

Note that the soil thrust, rolling resistance (soil drag) and draw-bar pull are all aligned in the direction of travel of the wheel, as illustrated in Fig. 2.6. The wheel slip s is defined as

$$s = \frac{r\omega - v_x}{r\omega} \times 100\% \quad (2.81)$$

Minimization of the wheel slip is crucial in traditional long-distance off-road mobility where energy loss due to wheel slippages can be considerable.

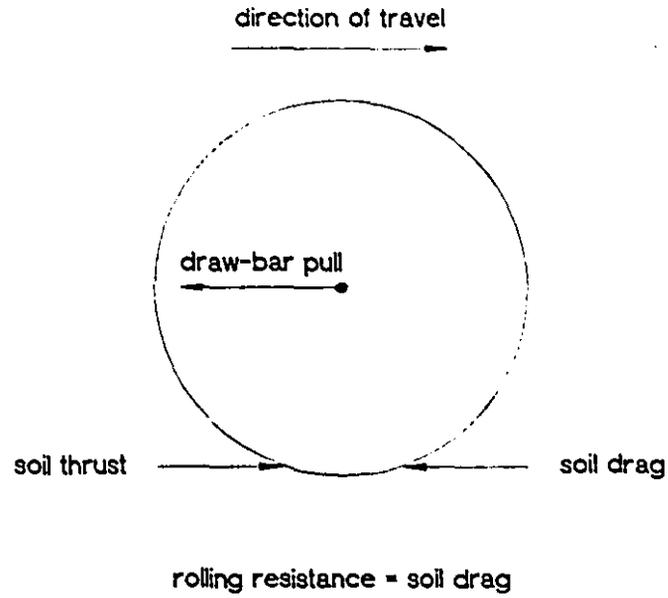


Figure 2.6: Soil thrust, rolling resistance and draw-bar pull

Kinematics

The linear displacement \bar{u} , velocity \bar{v} and acceleration \bar{a} of the wheel are related kinematically, as follows:

$$\bar{v} = \frac{d\bar{u}}{dt} \quad (2.82)$$

$$\bar{a} = \frac{d\bar{v}}{dt} \quad (2.83)$$

The *initial conditions* for solving eqs. (2.83) and (2.82) for \bar{v} and \bar{u} , in terms of \bar{a} , are provided in the form:

$$\bar{v}(0) = \bar{v}_0 \quad (2.84)$$

$$\bar{u}(0) = \bar{u}_0 \quad (2.85)$$

The acceleration \vec{a} is determined from the momentum conservation equation.

The rotation θ , angular velocity ω and angular acceleration α of the wheel are related kinematically, as follows:

$$\omega = \frac{d\theta}{dt} \quad (2.86)$$

$$\alpha = \frac{d\omega}{dt} \quad (2.87)$$

The *initial conditions* for solving eqs. (2.87) and (2.86) for ω and θ , in terms of α , are provided in the form:

$$\omega(0) = \omega_0 \quad (2.88)$$

$$\theta(0) = \theta_0 \quad (2.89)$$

The angular acceleration α is determined from the conservation of angular momentum equation.

2.2.2 Mass Conservation

By the law of mass conservation,

$$\frac{dm}{dt} = 0 \quad (2.90)$$

$$m = \text{constant} \quad (2.91)$$

2.2.3 Constitutive Equations

A *rigid* wheel is considered. Accordingly, a zero strain tensor exists, i.e.

$$\varepsilon_{xx} = \varepsilon_{yy} = \varepsilon_{xy} = 0 \quad (2.92)$$

The state of stress within the wheel is not needed.

2.3 Boundary Conditions

2.3.1 Inter-Control-Volume Boundary

An inter-control-volume boundary necessarily lies inside the computational region. Consider two arbitrary adjacent control-volumes B_1 and B_2 , bounded by S_1 and S_2 , respectively. Their common boundary, or interface, is denoted $S_1 \cap S_2$, where \cap is the usual intersection symbol in set theory. At any point $p \in (S_1 \cap S_2)$,

$$\bar{\sigma}_1 = \sigma \cdot \bar{n}_1 \quad (2.93)$$

$$\bar{\sigma}_2 = \sigma \cdot \bar{n}_2 \quad (2.94)$$

where σ is the stress tensor (continuous across inter-control-volume boundaries) and the subscripts 1 and 2 refer to S_1 and S_2 , respectively. Clearly,

$$\bar{n}_2 = -\bar{n}_1 \quad (2.95)$$

so that

$$\bar{\sigma}_2 = -\bar{\sigma}_1 \quad (2.96)$$

which is in accordance with Newton's Third Law.

In a two-dimensional analysis, in the xy plane,

$$\bar{\sigma} = \begin{Bmatrix} \sigma_x \\ \sigma_y \end{Bmatrix} \quad (2.97)$$

$$\bar{n} = \begin{Bmatrix} n_x \\ n_y \end{Bmatrix} \quad (2.98)$$

$$\sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix} \quad (2.99)$$

Thus

$$\sigma_x = \sigma_{xx}n_x + \sigma_{xy}n_y \quad (2.100)$$

$$\sigma_y = \sigma_{xy}n_x + \sigma_{yy}n_y \quad (2.101)$$

2.3.2 Transmitting Boundary

The complexity of wheel-soil interaction essentially precludes the existence of closed form solutions to practical problems. Consequently, one must resort to a numerical solution procedure, such as the finite element, finite difference or boundary

element method. For practical and economic reasons (e.g. computer storage and execution time), the entire infinite domain cannot be analyzed. It must be truncated around the region of interest—the wheel—to yield a finite computational region.

The *artificial* boundary so obtained should be designed, by way of boundary conditions, to *transmit* the arriving waves. This is because the boundary is, in reality, an internal boundary and thus has no wave-reflecting capability. If waves are reflected back into the computational domain, the mathematical model would be at odds with the actual physical problem and erroneous results would be obtained unless the time-duration of the problem is such that reflected waves do not get 'close' to the wheel. It is worth noting that if the system is so heavily damped that waves are essentially dissipated before arriving at the artificial boundary, then a static-type analysis of the boundary will suffice. For the solution of general wheel-soil interaction problems, a transmitting (synonyms: non-reflecting, radiating, silent) artificial boundary is needed.

Consider the one-dimensional problem of wave propagation in an initially undisturbed, homogeneous, isotropic, elastic half-space generated by the application, on the surface of the half-space, of an arbitrary time-dependent but spatially uniform forcing function [13]. Let the half-space be defined by $y \geq 0$, the surface being $y = 0$. Due to symmetry, lateral motion is inhibited, i.e., the displacement, velocity and acceleration vectors may be expressed as

$$\vec{u} = u\vec{j}$$

$$\vec{v} = v\vec{j}$$

$$\vec{a} = a\vec{j}$$

where u , v and a are functions of y and t only, i.e., $u = u(y, t)$, $v = v(y, t)$ and $a = a(y, t)$. The shear stress σ_{xy} is identically zero.

The equation of motion, i.e. conservation of linear momentum, reduces to

$$\frac{\partial \sigma_{yy}}{\partial y} = \rho \frac{d^2 u}{dt^2} \quad (2.102)$$

if gravity and damping forces are ignored.

The constitutive equation is given by

$$\sigma_{yy} = (\lambda + 2G)\varepsilon_{yy} \quad (2.103)$$

where

$$\varepsilon_{yy} = \frac{\partial u}{\partial y} \quad (2.104)$$

and the Lamé elastic constants λ and G are, in terms of Young's modulus E and Poisson's ratio ν , given by

$$G = \frac{E}{2(1+\nu)} \quad (2.105)$$

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad (2.106)$$

The particle velocity v and acceleration a , expressed in spatial coordinates, are

$$v \equiv \frac{du}{dt} = \frac{\partial u}{\partial t} + v \frac{\partial u}{\partial y} \quad (2.107)$$

$$a \equiv \frac{dv}{dt} = \frac{\partial v}{\partial t} + v \frac{\partial v}{\partial y} \quad (2.108)$$

Solving eqs. (2.107) and (2.108) for v and a , in terms of u , one obtains

$$v = \left(1 - \frac{\partial u}{\partial y}\right)^{-1} \frac{\partial u}{\partial t} \quad (2.109)$$

$$a = \left[\left(1 - \frac{\partial u}{\partial y}\right)^2 \frac{\partial^2 u}{\partial t^2} + 2 \left(1 - \frac{\partial u}{\partial y}\right) \frac{\partial u}{\partial t} \frac{\partial^2 u}{\partial y \partial t} + \left(\frac{\partial u}{\partial t}\right)^2 \frac{\partial^2 u}{\partial u^2} \right] \left(1 - \frac{\partial u}{\partial y}\right)^{-3} \quad (2.110)$$

Using eqs. (2.103), (2.104) and (2.110) in eq. (2.102), the following non-linear wave equation is obtained:

$$\left[c_d^2 - \left(\frac{\partial u}{\partial t}\right)^2 \right] \frac{\partial^2 u}{\partial y^2} = \left(1 - \frac{\partial u}{\partial y}\right)^2 \frac{\partial^2 u}{\partial t^2} + 2 \left(1 - \frac{\partial u}{\partial y}\right) \frac{\partial u}{\partial t} \frac{\partial^2 u}{\partial y \partial t} \quad (2.111)$$

where

$$c_d^2 = \frac{\lambda + 2G}{\rho} \quad (2.112)$$

Equation (2.111) admits a solution of the form

$$u(y, t) = f\left(t - \frac{y}{c_d}\right) \quad (2.113)$$

as can be verified by direct substitution. Clearly, $f(t - y/c_d)$ represents displacement

due to waves travelling in the $+\bar{j}$ direction. Because of the infinite extent of the half-space, motion at any point will be due solely to waves travelling in $+\bar{j}$ direction as no boundary-reflected waves, propagating in the $-\bar{j}$ direction, would be encountered. It is evident, therefore, that if the motion of particles on an artificial boundary satisfy eq. (2.113), that boundary will be a transmitting one.

On differentiating eq. (2.113) with respect to y and with respect to t , one obtains

$$\frac{\partial u}{\partial t} = f' \quad (2.114)$$

$$\frac{\partial u}{\partial y} = \frac{-f'}{c_d} \quad (2.115)$$

where $f' = \partial f / \partial(t - y/c_d)$. On eliminating f' from eqs. (2.114) and (2.115), one obtains

$$\frac{\partial u}{\partial t} + c_d \frac{\partial u}{\partial y} = 0 \quad (2.116)$$

Using eqs. (2.103), (2.104) and (2.109) in eq. (2.116), the following transmitting boundary condition

$$\sigma_{yy} = \frac{-\rho c_d \dot{v}}{(1 - v/c_d)} \quad (2.117)$$

is obtained. If $\bar{n} = \bar{j}$, then the required boundary traction is given by

$$\sigma_y = \frac{-\rho c_d v}{(1 - v/c_d)} \quad (2.118)$$

If $v \ll c_d$, the one-dimensional form of the viscous boundary proposed by Lysmer

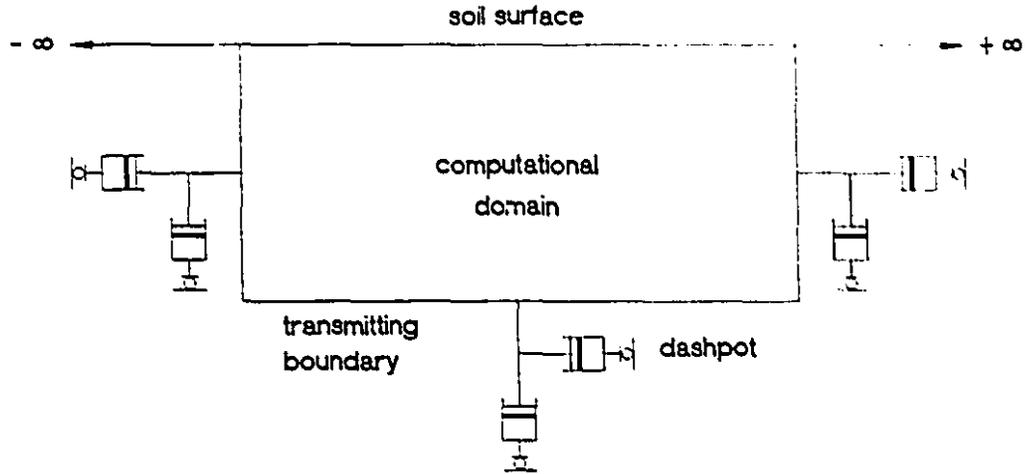


Figure 2.7: Viscous model of transmitting boundary

and Kuhlemeyer [5] is recovered. (The viscous boundary of Lysmer and Kuhlemeyer is undoubtedly one of the most acclaimed; other non-reflecting boundary models can be found in [14] and in references therein). The basic idea of a viscous model of a transmitting boundary is the absorption of the arriving waves by dampers, represented by dashpots, as illustrated in Fig. 2.7. By virtue of eq. (2.118), we consider the following viscous-type transmitting boundary condition for two-dimensional analysis:

$$\sigma_n = \frac{-\rho c_d v_n}{(1 - v_n/c_d)} \quad (2.119)$$

$$\sigma_t = \frac{-\rho c_s v_t}{(1 - v_t/c_s)} \quad (2.120)$$

where c_d is as defined in eq. (2.112) and c_s is

$$c_s^2 = \frac{G}{\rho} \quad (2.121)$$

c_d and c_s are the dilatational and shear wave velocities, respectively, in the medium.

If $v_n \ll c_d$ and $v_t \ll c_s$, or if the convective velocity and acceleration terms are ignored, the boundary equations of Lysmer and Kuhlemeyer are obtained. The transmitting boundary, characterized by eqs. (2.119 and (2.120), will be referred to as a *convective viscous boundary* because of the inclusion of the convective terms.

2.3.3 Free-Surface

The non-contact region of the soil and wheel surfaces constitutes a *free surface*. At each point on a free-surface,

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad (2.122)$$

A non-zero traction due, for example, to wind stresses may be applicable. If such is the case, the zero stress vector in eq. (2.122) is simply replaced with the appropriate value.

2.3.4 Wheel-Soil Interface

The wheel-soil interface is the link between the mobility of the wheel and the associated sub-soil response. The determination of the stresses at the wheel-soil interface is the primary concern, and most difficult problem, in analytical wheel-soil interaction studies. Many researchers (e.g. Hiroma and Ota [25]) simplify the problem by neglecting the tangential component of the stress vector. Others (e.g. Yong and Foda [27]) make

a priori assumptions about the shape of the stress distribution.

Clearly, it should be possible to completely determine, from the physics of the problem, the normal and tangential contact stresses without any assumptions or constraints on the shape or magnitude of the stresses. This is our objective to develop a purely analytical procedure to determine the contact stresses on the basis of the underlying physical laws and principles.

Consider the dynamic contact between the wheel and an arbitrary soil surface particle. Let the wheel particle and soil particle at the contact point be denoted w and s , respectively. The following conditions apply:

$$(\vec{v}_w - \vec{v}_s) \cdot \vec{n} = 0 \quad (2.123)$$

$$(\vec{a}_w - \vec{a}_s) \cdot \vec{n} = 0 \quad (2.124)$$

$$(\vec{\sigma}_w + \vec{\sigma}_s) \cdot \vec{n} = 0 \quad (2.125)$$

$$(\vec{\sigma}_w + \vec{\sigma}_s) \cdot \vec{t} = 0 \quad (2.126)$$

where \vec{n} and \vec{t} are the normal and tangential unit vectors, respectively, at the wheel-soil contact point (Fig. 2.8). Equations (2.123) and (2.124) ensure kinematic compatibility: equality of motion in the normal direction. Equations (2.125) and (2.126) enforce Newton's Third Law.

One more equation, in the tangential \vec{t} direction, is required. The basis of the additional equation is the stick-slip status of the contact point. The stick-slip phenomenon was considered in a recent tribology-based quasi-static model proposed by

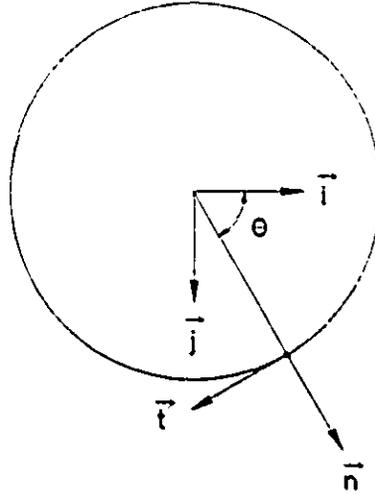


Figure 2.8: Normal and tangential unit vectors at wheel-soil interface

Yong and Foda [27]. A schematic of the stick-slip phenomenon is shown in Fig. 2.9, where w and s denote wheel and soil particles, respectively. At a stick-point, the tangential components of velocity and acceleration are identical:

$$\vec{v}_s \cdot \vec{t} = \vec{v}_w \cdot \vec{t} + r\omega \quad (2.127)$$

$$\vec{a}_s \cdot \vec{t} = \vec{a}_w \cdot \vec{t} + r\alpha \quad (2.128)$$

If the stick-point equations (2.127) and (2.128) result in a shear stress in excess of the shear strength of the interface at the point, slip is deemed to occur. The slip velocity v_s is given by

$$v_s = \vec{v}_w \cdot \vec{t} + r\omega - \vec{v}_s \cdot \vec{t} \quad (2.129)$$

At a slip-point, the shear stress is given by the shear strength of the interface. The simplest and most widely used interface model appears to be the basic Mohr-Coulomb

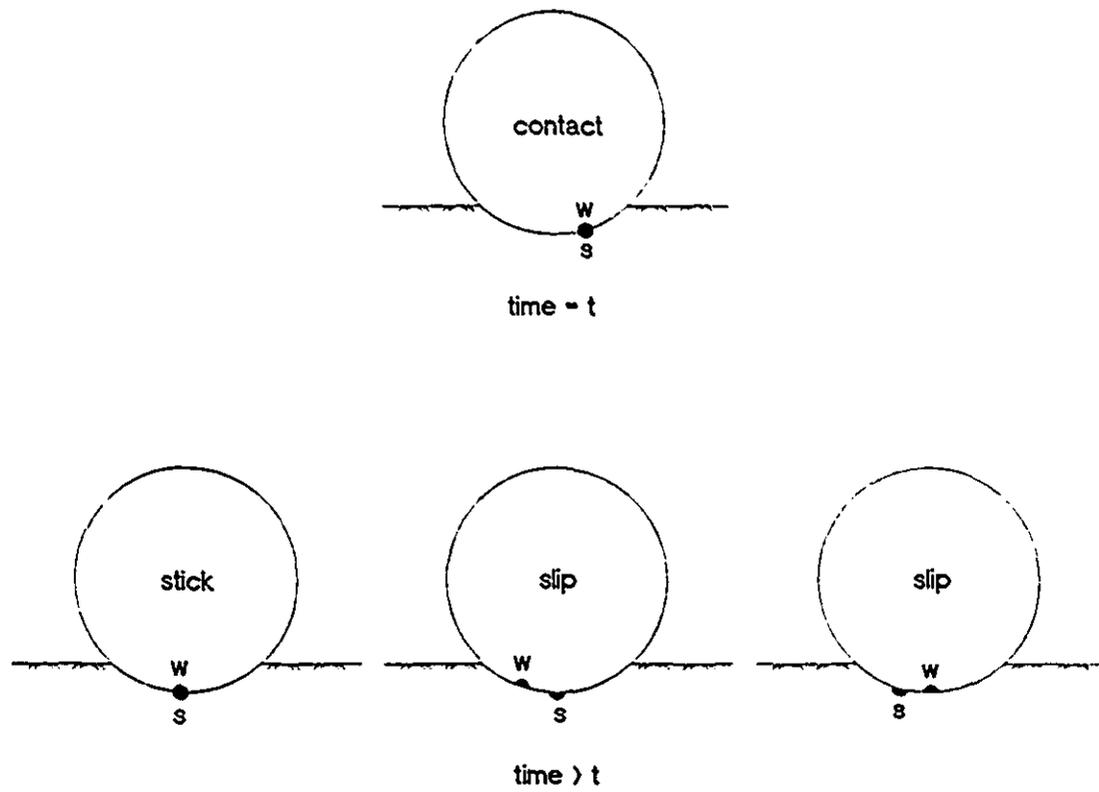


Figure 2.9: Schematic of stick-slip phenomenon

equation, given by

$$\sigma_t = c_a + \mu\sigma_n \quad (2.130)$$

where

σ_t = shear stress

σ_n = normal stress

c_a = adhesion

μ = friction coefficient

Experimental studies aimed at establishing the relationship between the interface shear strength and slip velocity are, for the most part, contradictory and confusing. For example, Payne [19] reported that the slip velocity had little or no effect on the shear strength; Stafford and Tanner [20] concluded that eq. (2.130) was valid, even at high slip velocities, with the friction coefficient increasing logarithmically with slip velocity but the adhesion remaining unchanged. The experimental results of Burchenko *et al.* [30] suggest that adhesion increases with slip velocity up to 2.2–2.8 m/s. Yusu and Dechao [21], using statistical-mechanics-based *rate process theory* [28], derived the following relationship between the shear strength and the slip velocity:

$$\sigma_t = c_a + \mu\sigma_n + A \ln v_s \quad (2.131)$$

where

v_s = slip velocity

A = material constant

Equation (2.131) was validated in the speed range 0.3–1.7 m/s, using a high-speed direct shear device [29]. Equation (2.131) indicates that the net adhesion ($c_a + A \ln v_s$) increases logarithmically with the slip velocity while the friction coefficient is unaffected, in direct contradiction of the results of Stafford and Tanner [20]. McKyes [22] pointed to two drawbacks with eq. (2.131). First, the unit of the coefficient of A , namely, kPa/ln (m/s), is awkward. Second, there is the problem of the logarithmic term, $\ln v$, becoming negative at low speeds and potentially giving a prediction of negative friction. In addition to providing very good fits to the experimental data of Yusu and Dechao [21], the modified equation

$$\sigma_t = c_a + \mu\sigma_n + Av_s \quad (2.132)$$

proposed by McKyes [22] also resolves the above-mentioned problems. Yusu and Dechao [23], in their response to McKyes' comments, strongly defended the nonlinearity of their original equation (2.131), noting that the general trend of a nonlinear equation cannot be captured with one linear equation and, furthermore, that the modified equation (2.132) has no theoretical basis. To resolve the issues raised with regard to negative friction at low speeds and the awkward unit of A , Yusu and Dechao [23]

rewrote their original equation in the form

$$\sigma_t = c_a + \mu\sigma_n + A \ln \left(1 + \frac{v_s}{v_0} \right) \quad (2.133)$$

where v_0 is an arbitrary constant.

Recent studies by Dechao and Yusu [24] revealed that, for a *soil-soil* interface, the linear Mohr-Coulomb model, eq. (2.130), is not valid at high slip rates. They found that the relationship between σ_t and σ_n becomes increasingly non-linear as the shear rate increases. Whether or not this finding applies to *soil-solid* interfaces needs to be thoroughly investigated. Yong and Foda [27], in their tribology-based quasi-static model, used a friction coefficient that decreases with slip velocity. Pi [7] and Ueno *et al.* [26] used a rate-independent friction coefficient.

Pending the resolution of these conflicts and contradictions, we will, following Pi [7] and others, and without loss of generality, implement the Mohr-Coulomb model, eq. (2.130), whose accuracy, to at least first order, is not in question. With reference to the soil particle at an arbitrary wheel-soil contact point, we rewrite eq. (2.130) in the form

$$\vec{\sigma} \cdot \vec{t} = \zeta(c_a + \mu\vec{\sigma} \cdot \vec{n}) \quad (2.134)$$

where

$$\zeta = \frac{v_s}{|v_s|} \quad (2.135)$$

is the direction of slip (DOS) parameter, ensuring that the shear stress acts in a direc-

tion opposite that of the slip velocity.

As the wheel advances, various points on the soil surface establish, maintain and eventually break contact with the wheel. Contact is established between the wheel and an arbitrary soil surface particle if

$$|\vec{x}_s - \vec{x}_w| \leq r \quad (2.136)$$

where \vec{x}_s and \vec{x}_w are the position vectors of the (infinitesimal) soil particle and *center* of the wheel, respectively; r is the radius of the wheel. It is presumed that the wheel-soil interface has zero tensile strength. Thus, contact is broken, i.e. separation occurs, when continued contact requires a tensile normal stress:

$$\vec{\sigma}_s \cdot \vec{n} < 0 \quad (2.137)$$

In general, the stick and slip modes co-exist along the wheel-soil interface. A detailed analysis of the coupled stick-slip equations is presented in chapter 3.

References

- [1] MASE, G. E. *Theory and Problems of Continuum Mechanics*. McGraw-Hill, New York, 1970.
- [2] CHUNG, T. J. *Continuum Mechanics*. Prentice-Hall, Englewood Cliffs, New Jersey, 1988.
- [3] MALVERN, L. E. *Introduction to the Mechanics of a Continuous Medium*. Prentice-Hall, Englewood Cliffs, New Jersey, 1969.
- [4] SERRIN, J. Mathematical Principles of Classical Fluid Mechanics. In Flügge, S. (editor) *Encyclopedia of Physics*, Volume VIII/1: *Fluid Dynamics I*. Springer-Verlag, Berlin, 1959.
- [5] LYSMER, J. AND KUHLEMEYER, R. L. Finite Dynamic Model for Infinite Media. *J. Eng. Mech. Div. ASCE*, 95:859-877, 1969.
- [6] TSCHOEGL, N. W. *The phenomenological Theory of Linear Viscoelastic Behavior*. Springer-Verlag, Berlin, 1989.
- [7] PI, W. S. Dynamic Tire/Soil Contact Surface Interaction Model for Aircraft Ground Operations. *J. Aircraft*, 25:1038-1044, 1988.
- [8] GALIN, L. A. AND SHMATKOVA, A. A. Motion of a rigid Stamp on the Boundary of a Viscoelastic Half-Plane. *PMM (J. Appl. Math. Mech.)*, 32:445-453, 1968.
- [9] OIDA, A. Analysis of Rheological Deformation of Soil by Means of Finite Element Method. *J. Terramechanics*, 21:237-251, 1984.
- [10] KOCHER, M. F. AND SUMMERS, J. D. Wave Propagation Theory for Evaluating Dynamic Soil Stress-Strain Models. *Trans. ASAE*, 31(3):683-691, 1988.
- [11] SHIBUSAWA, S. AND OIDA, A. Dependency of Observation Parameters on Soil Dynamic Parameters in Unconfined Impact Compression Tests. *J. Terramechanics*, 29(3):289-306, 1992.
- [12] ABOUDI, J. The Dynamic Indentation of an Elastic Half-Space by a Rigid Punch. *Int. J. Solids Structures*, 13:995-1005, 1977.
- [13] ACHENBACH, J. D. *Wave Propagation in Elastic Solids*. North-Holland, Amsterdam, 1973.

- [14] COHEN, M. AND JENNINGS, P. C. Silent Boundary Methods for Transient Analysis. In BELYTSCHKO, T. AND HUGHES, T. J. R. *Computational Methods for Transient Analysis*. Elsevier Science Publishers, Amsterdam, The Netherlands, 1983, pp 301-360.
- [15] WHITMAN, R. V. *Soil Dynamics*. Lecture Notes. Massachusetts Institute of Technology, Cambridge, Mass, 1981.
- [16] DESAI, C. S. AND SIRIWARDANE, H. J. *Constitutive Laws for Engineering Materials with Emphasis on Geologic Materials*. Prentice-Hall, Englewood Cliffs, New Jersey, 1984.
- [17] HILDERBRAND, F. B. *Advanced Calculus for Applications*. Prentice-Hall, Englewood Cliffs, New Jersey, 1976.
- [18] YONG, R. N. AND KO, H-Y. *Limit Equilibrium, Plasticity and Generalized Stress-Strain in Geotechnical Engineering*. Proceedings of a Workshop held at McGill University, Montreal, May 28-30, 1980. ASCE, New York, 1981.
- [19] PAYNE, P. C. J. A Field Method for Measuring Soil Metal Friction. *J. Soil Science*, 7:235-241, 1956.
- [20] STAFFORD, J. V. AND TANNER, D. W. Effect of Rate on Soil Shear Strength and Soil Metal-Friction—II. Soil Metal-Friction. *Soil Tillage Res.*, 3:321-330, 1983.
- [21] YUSU, Y. AND DECHAO, Z. Investigation of the Relationship Between Soil-Metal Friction and Sliding Speed. *J. Terramechanics*, 27(4):283-290, 1990.
- [22] MCKYES, E. Discussion: Investigation of the Relationship Between Soil-Metal Friction and Sliding Speed, Yao Yusu and Zeng Dechao, *J. Terramechanics*, 27(4):283-290, 1990. *J. Terramechanics*, 29(2):281-282, 1992.
- [23] YUSU, Y. AND DECHAO, Z. Author's Reply: Investigation of the Relationship Between Soil-Metal Friction and Sliding Speed. *J. Terramechanics*, 29(2):283-285, 1992.
- [24] DECHAO, Z. AND YUSU, Y. Investigation on the Relationship Between Soil Shear Strength and the Shear Rate. *J. Terramechanics*, 28:1-10, 1991.
- [25] HIROMA, T. AND OTA, Y. Analysis of Normal Stress Distribution Under a Wheel Using a Viscoelastic Model of Soils. *Proc. 10th Int. Conf. ISTVS*, Kobe, Japan, 1990.
- [26] UENO, M., HASHIGUCHI, K., NOHSE, Y., KOYAMA, F., UCHIYAMA K. AND IZUMI, H. Analysis of Soil-Wheel Interaction by Elastoplastic Finite Element Method. *Proc. 10th Int. Conf. ISTVS*, Kobe, Japan, 1990.

- [27] YONG, R. N. AND FODA, M. A. Tribology Model for Determination of Shear Stress Distribution along the Tyre-Soil Interface. *J. Terramechanics*, 27:93-114, 1990.
- [28] MITCHELL, J. K. *Fundamentals of Soil Behavior*, pp 291-305. Wiley, New York, 1976.
- [29] YAO, Y. S., ZENG, D.C. *et al.* Developing a High-Speed Direct Shear Device. *J. Beijing Agric. Eng. University*, 8:31-35, 1988 (in Chinese).
- [30] BURCHENKO, P. N., FUMAROV, R. T. AND KASCHAEV, B. A. Determination of Frictional Coefficient of Material in Contact with Soil. *Mechanization and Electrification of Socialistic Agric.*, 12:49-50, 1971 (in Russian). English Translation from Russian by U.S Dept. of Commerce, NTIS, PB-210, 366-T.
- [31] GRAHN, M. The Influence of Penetration Velocity on the Pressure Sinkage Relationship. *Proc. 4th European Conf. ISTVS*, Wageningen, Netherlands, Vol I:37-44, 1989.
- [32] GRAHN, M. Prediction of Sinkage and Rolling Resistance for Off-the-Road Vehicles Considering Penetration Velocity. *J. Terramechanics*, 28:339-347, 1991.

Chapter 3

Discretization of Governing Equations

3.1 Control-Volume Finite Element Method

The basis of the control-volume-based finite element method (CVFEM) is the strict enforcement of the conservation laws at the discrete level by the direct discretization of the integral form of the governing conservation equations. The discretized equations so obtained possess a direct physical interpretation. Consequently, boundary conditions can be introduced with relative ease. Other advantages of the CVFEM, over conventional finite element methods, include reduced computational times and increased accuracy. The CVFEM may be conceptualized as a hybrid of the finite element method (FEM) [3, 4, 5] and finite volume method (FVM) [17].

The CVFEM has been advantageously used for the solution of a variety of fluid flow, heat transfer and convection-diffusion problems [1, 16, 2]. In this chapter, the CVFEM is applied to the solution of the dynamic wheel-soil interaction equations.

3.2 Domain Discretization

3.2.1 Finite Elements and Control Volumes

The first step in a control-volume finite-element analysis is the discretization of the computational domain into finite elements and control volumes. As in a conventional finite element method, considerable flexibility exists in the selection of element shape, size and distribution. Control volumes may be constructed around the grid nodes in an essentially arbitrary manner provided certain key conditions are met: (1) they do not overlap; (2) they completely fill the domain, i.e. there are no gaps, and (3) control-volume boundaries do not lie along inter-element boundaries.

A control-volume finite-element mesh based on 3-noded *triangular* elements is shown in Fig. 3.1 and one based on 4-noded *quadrilateral* elements is illustrated in Fig. 3.2. The algorithm for both meshes is as follows:

1. Discretize the computational domain into finite elements (triangular elements in Fig. 3.1 and quadrilateral elements in Fig. 3.2).
2. Discretize the computational domain into polygonal control volumes by joining the centroid of each element to the midpoint of its sides.

Typical interior, boundary and corner control volumes are highlighted in Figs. 3.1 and 3.2. We note that the quadrilateral-element-based mesh, in contrast with the triangular-element-based mesh, yields a control volume size distribution that varies proportionately with the element size distribution.

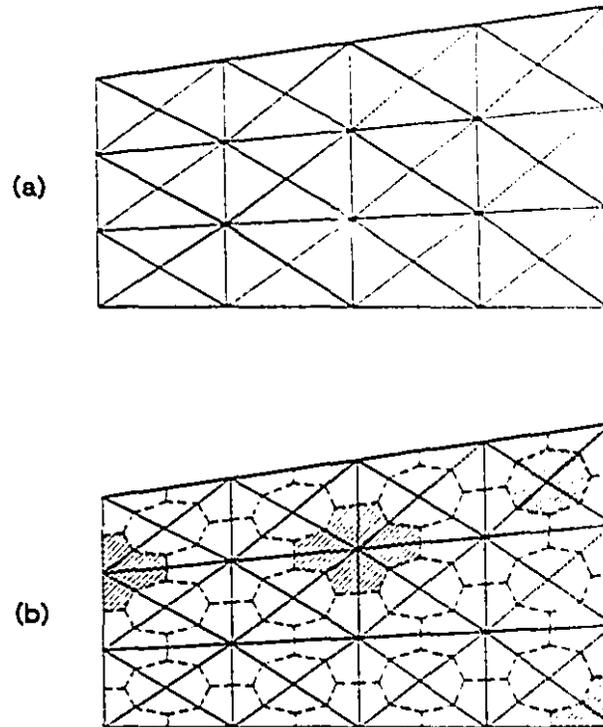


Figure 3.1: Discretization of computational domain into: (a) triangular finite elements; (b) polygonal control volumes

The discretization of an arbitrary-shaped computational domain into quadrilateral elements is considered in chapter 4. Triangular elements can be readily constructed from the quadrilateral elements.

3.3 Interpolation Functions

Displacement, velocity and acceleration are defined at the nodes. Density, strains and stresses are defined at the element level. Strains and stresses within an element depend on the assumed variation of the nodal displacements. A linear variation yields a constant strain element, which is known to be computationally very efficient. However,

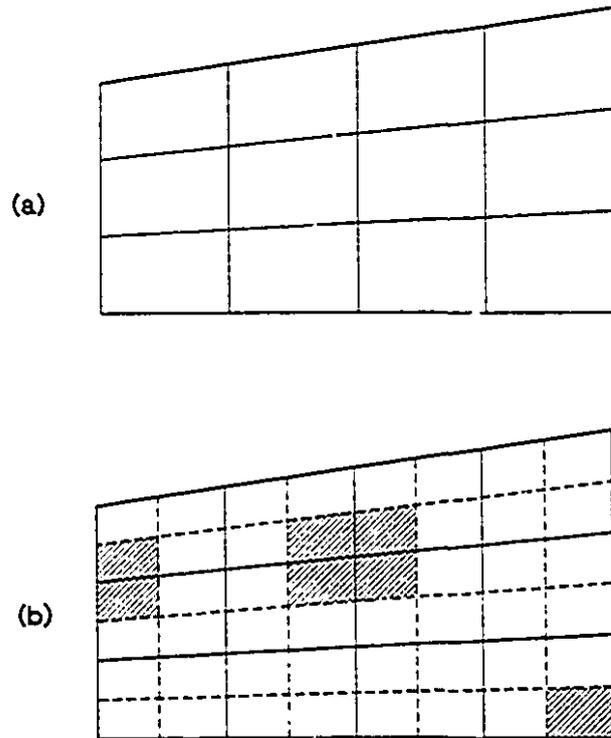


Figure 3.2: Discretization of computational domain into: (a) quadrilateral finite elements; (b) polygonal control volumes

there are some drawbacks. Constant strain triangular elements tend to be too stiff while constant strain quadrilateral elements give rise to mesh instabilities commonly referred to as **hour-glass modes**. The source of these unwanted distortions is the inability of the element forces derived from a constant strain quadrilateral element to resist all the compatible modes of deformation. There are techniques, however, for reducing or eliminating hour-glass modes (e.g. Last and Harkness [18]).

In this thesis, constant strain quadrilateral elements are used. Hour-glass control is discussed in section 3.6.6.

Continuity of stresses within elements, hence across inter-control-volume boundaries, ensures that the proposed CVFEM possesses the **conservative property**. This means, in essence, that Newton's Third Law will be enforced at the inter-control-volume boundaries. As a result, if the discretized momentum equations of an arbitrary number of control volumes are added, the inter-control-volume stresses will cancel out and the resulting momentum equation of the aggregate control-volume will be free of stresses from within, as it should. The accuracy of a solution may be considerably reduced in the absence of the conservative property.

3.4 Numbering Scheme

3.4.1 Global

Nodes are numbered (i, j) , $(i + 1, j)$, $(i, j + 1)$, $(i + 1, j + 1)$ etc., as illustrated in Fig. 3.3.

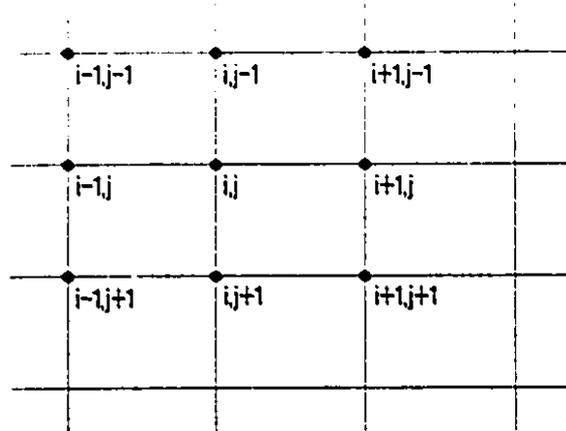


Figure 3.3: Global numbering scheme for nodes and control volumes

Elements are numbered (i, j) , $(i + 1, j)$, $(i, j + 1)$, $(i + 1, j + 1)$ etc., as illustrated in Fig. 3.4, such that an arbitrary element (i, j) is defined by nodes (i, j) , $(i + 1, j)$, $(i, j + 1)$ and $(i + 1, j + 1)$.

A control volume is identified by the node which is located on it. Thus, the numbering scheme for control volumes parallels that of the nodes. The region of the computational domain common to control volume (i, j) and element (i, j) is denoted

$$cv(i, j).e(i, j)$$

The bounding surface of control volume (i, j) is denoted $cs(i, j)$ and the intersection of $cs(i, j)$ and element (i, j) is denoted

$$cs(i, j).e(i, j)$$

	$i-1, j-1$	$i, j-1$	$i+1, j-1$
	$i-1, j$	i, j	$i+1, j$
	$i-1, j+1$	$i, j+1$	$i+1, j+1$

Figure 3.4: Global numbering scheme for quadrilateral finite elements

3.4.2 Local

The discretization of the governing equations will, for sake of computational efficiency and simplicity, be carried out element-by-element. The element-by-element discretization process is facilitated and the presentation clearer if conducted in terms of a local numbering system.

An arbitrary element (i, j) is denoted e in the local numbering system. The nodes of the element are numbered 1, 2, 3, and 4 in a clockwise fashion (Fig. 3.5) with node 1 corresponding to node (i, j) in the global system. The centroid of the element is labelled 0.

As in the global numbering scheme, a control volume is, in the local numbering system, identified by the node located on it. Thus, the control volumes associated with nodes 1, 2, 3, and 4 are denoted $cv(1)$, $cv(2)$, $cv(3)$ and $cv(4)$, respectively, and their

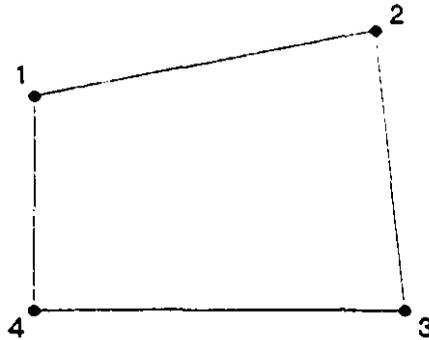


Figure 3.5: Local numbering scheme for nodes of quadrilateral element

regions of intersection with element e are denoted

$cv(1).e$

$cv(2).e$

$cv(3).e$

$cv(4).e$

respectively. This is illustrated in Fig. 3.6. The bounding surface of control volumes $cv(1)$, $cv(2)$, $cv(3)$ and $cv(4)$ are denoted $cs(1)$, $cs(2)$, $cs(3)$ and $cs(4)$, respectively, and their intersections with (the boundary of) element e are denoted

$cs(1).e$

$cs(2).e$

$cs(3).e$

$cs(4).e$

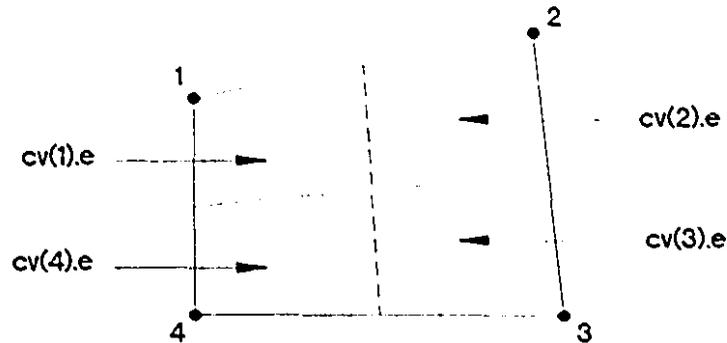


Figure 3.6: Local labelling scheme for control-volume/finite-element intersection

With reference to the inter-control-volume boundary tractions, the integration path $cs(1).e$, traversed in the clockwise direction, goes from the midpoint of side 1-2 to node 0 (centroid) to the midpoint of side 1-4. Considering the transmitting boundary tractions on side 1-2, if applicable, the integration path $cs(1).e$, traversed in the clockwise direction, goes from node 1 to the midpoint of side 1-2. The integration paths for the other control surfaces associated with element e can be readily depicted in a similar manner.

3.5 Sign convention

The continuum mechanics convention wherein normal stresses are positive in tension, and normal strains are positive in extension, is followed. (This is contrary to the usual soil mechanics usage). Shear stresses and strains are considered positive when they act in a positive coordinate direction on a surface whose outward normal is

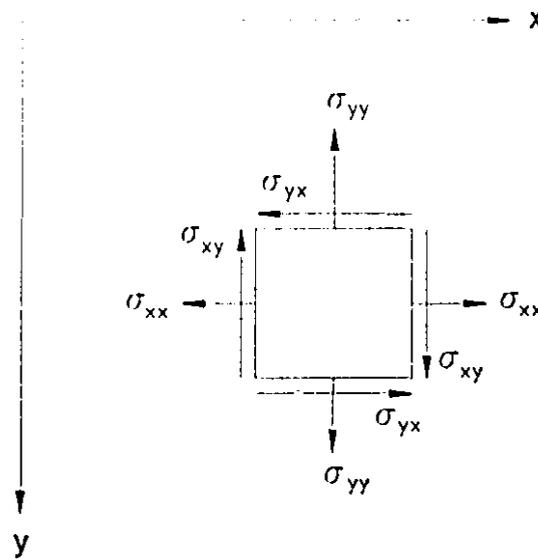


Figure 3.7: Direction of positive stresses and coordinates

in a positive direction. For example, σ_{xy} is positive if it acts in the $+y$ direction on a face whose outward normal is in the $+x$ direction, as indicated in Fig. 3.7. Note that $\sigma_{yx} = \sigma_{xy}$.

A rectangular cartesian coordinate system in which the y -axis is oriented 90 degrees clockwise from the x -axis, as depicted in Fig. 3.7, is utilized. The x -axis is parallel to the soil surface so that for a horizontal soil surface, the y -axis points in the direction of gravity. Rotational variables (e.g. angular coordinate, angular velocity, torque) increase in the clockwise direction.

3.6 Soil Subsystem

3.6.1 Momentum Conservation

The momentum conservation equations (2.10) and (2.11) are to be discretized over each of the control volumes. For an arbitrary control volume $cv(i, j)$, the momentum conservation equations are:

$$\underbrace{\int_{cv(i,j)} \rho \frac{dv_x}{dt} dV}_{\text{inertia}} = \underbrace{\int_{cv(i,j)} \rho g_x dV}_{\text{gravitational}} - \underbrace{\int_{cv(i,j)} \rho cv_x dV}_{\text{damping}} + \underbrace{\int_{cs(i,j)} \sigma_x dS}_{\text{traction}} \quad (3.1)$$

$$\underbrace{\int_{cv(i,j)} \rho \frac{dv_y}{dt} dV}_{\text{inertia}} = \underbrace{\int_{cv(i,j)} \rho g_y dV}_{\text{gravitational}} - \underbrace{\int_{cv(i,j)} \rho cv_y dV}_{\text{damping}} + \underbrace{\int_{cs(i,j)} \sigma_y dS}_{\text{traction}} \quad (3.2)$$

A term-by-term discretization of these equations follows.

Inertia Force

A "lumped mass" approximation of the inertia force yields

$$\int_{cv(i,j)} \rho \frac{dv_x}{dt} dV = \left(\frac{dv_x}{dt} \right)_{i,j} \int_{cv(i,j)} \rho dV \quad (3.3)$$

$$\int_{cv(i,j)} \rho \frac{dv_y}{dt} dV = \left(\frac{dv_y}{dt} \right)_{i,j} \int_{cv(i,j)} \rho dV \quad (3.4)$$

In words, the inertia force on control volume (i, j) is approximated by the the product of the control-volume mass and the acceleration of node (i, j) . The acceleration vector will be denoted by $d\bar{v}/dt$ and \bar{a} , interchangeably.

Damping Force

A "lumped mass" approximation of the damping term yields

$$\int_{cv(i,j)} \rho c v_x dV = (c v_x)_{i,j} \int_{cv(i,j)} \rho dV \quad (3.5)$$

$$\int_{cv(i,j)} \rho c v_y dV = (c v_y)_{i,j} \int_{cv(i,j)} \rho dV \quad (3.6)$$

In words, the damping force on control volume (i, j) is approximated by the the product of the control-volume mass, the damping coefficient and velocity of node (i, j) .

Gravitational Force

The gravitational acceleration vector \vec{g} is constant. Thus the relationship

$$\int_{cv(i,j)} \rho g_x dV = g_x \int_{cv(i,j)} \rho dV \quad (3.7)$$

$$\int_{cv(i,j)} \rho g_y dV = g_y \int_{cv(i,j)} \rho dV \quad (3.8)$$

is exact.

Mass

The integral term on the right hand sides of eqs. (3.3)–(3.8) is the control volume mass, i.e.,

$$\int_{cv(i,j)} \rho dV = m_{i,j} \quad (3.9)$$

This integral is most efficiently determined element-by-element. With reference to the local numbering system, it can be shown that

$$m_1 = \int_{cv(1)\cdot e} \rho dV = \rho\{(y_2 - y_4)(x_1 - x_0) - (x_2 - x_4)(y_1 - y_0)\}/4 \quad (3.10)$$

$$m_2 = \int_{cv(2)\cdot e} \rho dV = \rho\{(y_2 - y_0)(x_1 - x_3) - (x_2 - x_0)(y_1 - y_3)\}/4 \quad (3.11)$$

$$m_3 = \int_{cv(3)\cdot e} \rho dV = \rho\{(y_2 - y_4)(x_0 - x_3) - (x_2 - x_4)(y_0 - y_3)\}/4 \quad (3.12)$$

$$m_4 = \int_{cv(4)\cdot e} \rho dV = \rho\{(y_0 - y_4)(x_1 - x_3) - (x_0 - x_4)(y_1 - y_3)\}/4 \quad (3.13)$$

for a unit width of control volume. It can easily be verified that

$$m_1 + m_3 = m_2 + m_4 \quad (3.14)$$

Clearly then,

$$m_1 + m_3 = m/2 \quad (3.15)$$

$$m_2 + m_4 = m/2 \quad (3.16)$$

where

$$m = \int_e \rho dV \quad (3.17)$$

is the mass of element e . Element density ρ is determined from the equation of mass conservation.

Traction

A control-volume is subjected to surface force (traction) along its

1. inter-control-volume boundary
2. free-surface, if applicable
3. wheel-soil interface, if applicable, and
4. transmitting boundary, if applicable

Discretization of the traction term is considered in the section 3.8: boundary conditions.

Residual Force

The initial element stress is taken to be the centroidal stress. Consequently, the discretized momentum conservation equation of an arbitrary soil control volume $cv(i, j)$ at time $t = 0-$, just prior to the reference initial time $t = 0$, yields, in general, a nodal acceleration

$$\bar{a}_{i,j}(0-)$$

different from that which actually exists. The result is a discretization-induced inertia force given by

$$m_{i,j} \Delta \bar{a}_{i,j}(0-) \quad (3.18)$$

where $\Delta \bar{a}_{i,j}(0-)$ is the discretized nodal acceleration less the actual nodal acceleration. This force, eq. (3.18), may be conceptualized as a *residual force*. Typically, the soil domain is at rest (velocity = acceleration = 0) at time $t = 0-$. To remedy the discrepancy in the nodal accelerations, the discretization-induced inertia force, eq. (3.18), is added to the discretized control volume force at all times $t \geq 0$.

3.6.2 Time Integration

A variety of time integration schemes are available for discretizing the kinematic equations, relating acceleration to velocity and displacement. We consider three of the often used schemes.

Central Difference Scheme

The central-difference explicit scheme [7, 8] may be expressed in the form:

$$\ddot{d}^n = \frac{\dot{d}^{n+1/2} - \dot{d}^{n-1/2}}{\Delta t^{n+1/2}} \quad (3.19)$$

$$\dot{d}^{n+1/2} = \frac{d^{n+1} - d^n}{\Delta t^n} \quad (3.20)$$

where d , \dot{d} and \ddot{d} denote displacement \bar{u} , velocity \bar{v} and acceleration \bar{a} , respectively; the superscripts denote time— d^n is the displacement at time t^n , $\dot{d}^{n+1/2}$ is the velocity at time $t^{n+1/2}$, etc., and

$$t^{n+1/2} = (t^n + t^{n+1})/2 \quad (3.21)$$

$$\Delta t^{n+1/2} = (\Delta t^n + \Delta t^{n+1}) / 2 \quad (3.22)$$

$$\Delta t^n = t^{n+1} - t^n \quad (3.23)$$

Note that in eqs. (3.19) and (3.20), displacement and acceleration are defined at integral time steps while velocity is defined at half-time steps. The initial conditions d^0 and \dot{d}^0 are specified and \ddot{d}^0 is determined from the momentum conservation equation. The above difference scheme is not “self-starting”. At the initial time ($n = 0$), eq. (3.19) contains two unknowns— $\dot{d}^{1/2}$ and $\dot{d}^{-1/2}$. One cannot, therefore, solve for $\dot{d}^{1/2}$ on the basis of eq. (3.19). A so-called “starting formula”, e.g.

$$\ddot{d}^n = \frac{\dot{d}^{n+1/2} - \dot{d}^n}{\Delta t^n / 2} \quad n = 0 \quad (3.24)$$

is needed.

The momentum conservation equation relates the acceleration at a given time to the force at the same time. The damping force, transmitting boundary force and the wheel-soil contact force are velocity-dependent. Therefore, computation of these forces requires velocity at the same times as the acceleration, that is, at integral time steps. The following “extrapolation” formula

$$\ddot{d}^n = \frac{\dot{d}^{n+1} - \dot{d}^{n-1/2}}{\Delta t^n + \Delta t^{n-1} / 2} \quad (3.25)$$

may be used to determine velocity at integral time steps. The central difference scheme is conditionally stable.

Newmark's Scheme

The general equations of Newmark's time integration scheme[9] are:

$$d^{n+1} = d^n + \Delta t^n \dot{d}^n + (\Delta t^n)^2 \{(1 - 2\beta)\ddot{d}^n + 2\beta\ddot{d}^{n+1}\}/2 \quad (3.26)$$

$$\dot{d}^{n+1} = \dot{d}^n + \Delta t^n \{(1 - \gamma)\ddot{d}^n + \gamma\ddot{d}^{n+1}\} \quad (3.27)$$

where β and γ are scalar constants that determine the stability characteristics of the scheme. The values $\beta = 0.25$ and $\gamma = 0.5$ yield an implicit scheme, characterized by unconditional stability. The disadvantage of such a scheme is that the equations become spatially coupled, resulting in a more complex code. The values $\beta = 0$ and $\gamma = 0.5$ yield an explicit scheme, with conditional stability. The relative simplicity of explicit schemes is quite attractive, notwithstanding the limitation on the time step. A 'very explicit' scheme, produced by $\beta = \gamma = 0$, was presented by Grant[10]. It required, however, that a fictitious damping term be introduced to counter-balance the inherent negative numerical damping.

Predictor-corrector

The predictor-corrector algorithm [11] is an explicit scheme based on Newmark's equations. It predicts displacement and velocity at time t^{n+1} using the known acceleration at time t^n , computes the acceleration at time t^{n+1} and then corrects the predicted displacement and velocity using the acceleration at time t^{n+1} . Formally, the algorithm

is given by:

$$\sim d^{n+1} = d^n + \Delta t^n \dot{d}^n + (\Delta t^n)^2 (1 - 2\beta) \ddot{d}^n / 2 \quad (3.28)$$

$$\sim \dot{d}^{n+1} = \dot{d}^n + \Delta t^n (1 - \gamma) \ddot{d}^n \quad (3.29)$$

$$d^{n+1} = \sim d^{n+1} + (\Delta t^n)^2 \beta \ddot{d}^{n+1} \quad (3.30)$$

$$\dot{d}^{n+1} = \sim \dot{d}^{n+1} + \Delta t^n \gamma \ddot{d}^{n+1} \quad (3.31)$$

where $\sim d^{n+1}$ and $\sim \dot{d}^{n+1}$ are the 'predictor' values; d^{n+1} and \dot{d}^{n+1} are the 'corrector' values. The momentum conservation equation is explicitly solved for \ddot{d}^{n+1} using the predictor values. This algorithm is conditionally stable.

Stability

Explicit schemes are, in general, conditionally stable. The stability limit, i.e. the time step above which numerical instability occurs, may be established from physical as well as mathematical considerations. Physically, the computational time step should not exceed the time it takes the fastest travelling stress wave to completely pass through the smallest element. The element-wave transit time, for an arbitrary element, is given by

$$\Delta t \leq \left(\frac{\ell}{c} \right)_{\min} \quad (3.32)$$

where c is fastest wave speed through the element and ℓ_{\min} is the minimum length of the element . We note that

$$\left(\frac{\ell}{c}\right)_{\min} = \frac{\ell_{\min}}{c_{\max}} \quad (3.33)$$

For simplicity, one may set ℓ_{\min} equal to the minimum inter-nodal distance and then impose a factor of safety. For an arbitrary quadrilateral element, the inter-nodal distances are simply:

$$\ell_{12} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3.34)$$

$$\ell_{23} = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2} \quad (3.35)$$

$$\ell_{34} = \sqrt{(x_4 - x_3)^2 + (y_4 - y_3)^2} \quad (3.36)$$

$$\ell_{41} = \sqrt{(x_1 - x_4)^2 + (y_1 - y_4)^2} \quad (3.37)$$

The speed of propagation, c , of a stress wave through a material may be expressed in the form [13]:

$$c = \sqrt{\frac{1}{\rho} \frac{d\sigma}{d\varepsilon}} \quad (3.38)$$

where $d\sigma/d\varepsilon$ is the slope of the stress-strain curve of the material. The dilatational (normal) wave speed c_n and the distortional (shear) wave speed c_s are obtained from the normal-stress/normal-strain and the shear-stress/shear-strain curves, respectively.

For an isotropic elastic material, the slopes $d\sigma/d\varepsilon$ are given by

$$\left(\frac{d\sigma}{d\varepsilon}\right)_n = \lambda + 2G \quad (3.39)$$

$$\left(\frac{d\sigma}{d\varepsilon}\right)_s = G \quad (3.40)$$

where the subscripts n and s refer to the normal and shear curves, respectively. Thus,

$$\frac{c_n}{c_s} = \sqrt{\frac{\lambda + 2G}{G}} \quad (3.41)$$

$$= \sqrt{\frac{2 - 2\nu}{1 - 2\nu}} \quad (3.42)$$

Since $0 \leq \nu \leq 1/2$,

$$c_n > c_s \quad (3.43)$$

That is, the elastic dilatational wave travels faster than the elastic distortional wave.

A perusal of the viscoelastic constitutive equations reveals that (1) viscoelastic dilatational wave is faster than viscoelastic distortional wave speed, and (2) elastic dilatational wave is faster than viscoelastic dilatational wave speed, i.e.,

$$c_n(\text{elastic}) > c_n(\text{viscoelastic}) > c_s(\text{viscoelastic}) \quad (3.44)$$

Thus, the stability limit of an elastic material is less than that of a viscoelastic material and, therefore, constitutes a conservative estimate.

In a contact-impact problem, the stability limit may be significantly reduced depending on the speed of the impactor/penetrator. In the high-speed wheel-soil interaction problem considered herein, the reduction factor was determined by numerical experimentation.

3.6.3 Mass Conservation

The mass conservation equation, applied at the element level, yields the current element density:

$$\rho = \rho_0 \frac{A_0}{A} \quad (3.45)$$

$$= m_0/A \quad (3.46)$$

where A is the area of the element and the subscript $_0$ refers to the initial value. The initial density at the centroid of each element will be assumed to prevail throughout the entire element.

3.6.4 Constitutive Equations

The integral terms in the constitutive equations (2.60)–(2.62) are of the form

$$\begin{aligned} I(t) &= \int_{-\infty}^t K(t-\tau)\varepsilon(\tau) d\tau \\ &= \int_{-\infty}^t \frac{E_2}{\eta_1} e^{-(t-\tau)/T} \varepsilon(\tau) d\tau \end{aligned}$$

$$= \frac{E_2}{\eta_1} e^{-t/T} \int_{-\infty}^t e^{\tau/T} \varepsilon(\tau) d\tau \quad (3.47)$$

At time $t + \Delta t$,

$$\begin{aligned} I(t + \Delta t) &= \frac{E_2}{\eta_1} e^{-(t+\Delta t)/T} \int_{-\infty}^{(t+\Delta t)} e^{\tau/T} \varepsilon(\tau) d\tau \\ &= \frac{E_2}{\eta_1} e^{-(t+\Delta t)/T} \left\{ \int_{-\infty}^t e^{\tau/T} \varepsilon(\tau) d\tau + \int_t^{t+\Delta t} e^{\tau/T} \varepsilon(\tau) d\tau \right\} \\ &= \frac{E_2}{\eta_1} e^{-\Delta t/T} I(t) + \frac{E_2}{\eta_1} e^{-(t+\Delta t)/T} \int_t^{t+\Delta t} e^{\tau/T} \varepsilon(\tau) d\tau \end{aligned} \quad (3.48)$$

Using the trapezoidal rule, the integral term in eq. (3.48) is discretized as follows:

$$\int_t^{t+\Delta t} e^{\tau/T} \varepsilon(\tau) d\tau = \left[e^{(t+\Delta t)/T} \varepsilon(t + \Delta t) + e^{t/T} \varepsilon(t) \right] \Delta t / 2 \quad (3.49)$$

Substituting eq. (3.49) into eq. (3.48), one obtains

$$I(t + \Delta t) = \frac{E_2}{\eta_1} e^{-\Delta t/T} [I(t) + \varepsilon(t) \Delta t / 2] + \frac{E_2}{\eta_1} \varepsilon(t + \Delta t) \Delta t / 2 \quad (3.50)$$

Based on eq. (3.50), the constitutive equations (2.60)–(2.62) are discretized as follows:

$$\begin{aligned} \sigma_{xx}(t) &= \frac{E_2(1-\nu)}{(1+\nu)(1-2\nu)} \left\{ \varepsilon_{xx}(t + \Delta t) + \frac{\nu}{1-\nu} \varepsilon_{yy}(t + \Delta t) \right. \\ &\quad \left. - I_{xx}(t + \Delta t) + \frac{1}{2} I_{yy}(t + \Delta t) \right\} \end{aligned} \quad (3.51)$$

$$\sigma_{yy}(t) = \frac{E_2(1-\nu)}{(1+\nu)(1-2\nu)} \left\{ \varepsilon_{yy}(t + \Delta t) + \frac{\nu}{1-\nu} \varepsilon_{xx}(t + \Delta t) - I_{yy}(t + \Delta t) + \frac{1}{2} I_{xx}(t + \Delta t) \right\} \quad (3.52)$$

$$\sigma_{xy}(t) = 2G \left\{ \varepsilon_{xy}(t + \Delta t) - \frac{3(1-\nu)}{2(1-2\nu)} I_{xy}(t + \Delta t) \right\} \quad (3.53)$$

where

$$I_{xx}(t + \Delta t) = \frac{E_2}{\eta_1} e^{-\Delta t/T} [I_{xx}(t) + \varepsilon_{xx}(t) \Delta t/2] + \frac{E_2}{\eta_1} \varepsilon_{xx}(t + \Delta t) \Delta t/2 \quad (3.54)$$

$$I_{yy}(t + \Delta t) = \frac{E_2}{\eta_1} e^{-\Delta t/T} [I_{yy}(t) + \varepsilon_{yy}(t) \Delta t/2] + \frac{E_2}{\eta_1} \varepsilon_{yy}(t + \Delta t) \Delta t/2 \quad (3.55)$$

$$I_{xy}(t + \Delta t) = \frac{E_2}{\eta_1} e^{-\Delta t/T} [I_{xy}(t) + \varepsilon_{xy}(t) \Delta t/2] + \frac{E_2}{\eta_1} \varepsilon_{xy}(t + \Delta t) \Delta t/2 \quad (3.56)$$

The initial conditions

$$\varepsilon_{xx}(t) = \varepsilon_{yy}(t) = \varepsilon_{xy}(t) = 0 \quad -\infty \leq t \leq 0 \quad (3.57)$$

yield

$$I_{xx}(0) = I_{yy}(0) = I_{xy}(0) = 0 \quad (3.58)$$

Initial stresses $\sigma_{xx}(0)$, $\sigma_{yy}(0)$ and $\sigma_{xy}(0)$ are appended to the right hand sides of eqs. (3.51)–(3.53), respectively, yielding

$$\sigma_{xx}(t) = \sigma_{xx}(0) + \frac{E_2(1-\nu)}{(1+\nu)(1-2\nu)} \left\{ \varepsilon_{xx}(t + \Delta t) + \frac{\nu}{1-\nu} \varepsilon_{yy}(t + \Delta t) \right\}$$

$$- I_{xx}(t + \Delta t) + \frac{1}{2} I_{yy}(t + \Delta t) \} \quad (3.59)$$

$$\begin{aligned} \sigma_{yy}(t) = \sigma_{yy}(0) + \frac{E_2(1-\nu)}{(1+\nu)(1-2\nu)} \left\{ \varepsilon_{yy}(t + \Delta t) + \frac{\nu}{1-\nu} \varepsilon_{xx}(t + \Delta t) \right. \\ \left. - I_{yy}(t + \Delta t) + \frac{1}{2} I_{xx}(t + \Delta t) \right\} \quad (3.60) \end{aligned}$$

$$\sigma_{xy}(t) = \sigma_{xy}(0) + 2G \left\{ \varepsilon_{xy}(t + \Delta t) - \frac{3(1-\nu)}{2(1-2\nu)} I_{xy}(t + \Delta t) \right\} \quad (3.61)$$

3.6.5 Strain-Displacement Equations: Contour Integral Method

The contour integral method uses Green's theorem to express the derivative in an arbitrary-shaped region in terms of an integral around the boundary of the region.

For a region A , bounded by S , Green's theorem says:

$$\int_A \frac{\partial u}{\partial i} dA = \int_S u n_i dS \quad (3.62)$$

where u is an arbitrary function and $i \in \{x, y\}$. It can be shown that

$$\int_A \frac{\partial u}{\partial x} dA = \int_S u dy \quad (3.63)$$

$$\int_A \frac{\partial u}{\partial y} dA = - \int_S u dx \quad (3.64)$$

If the gradients $\partial u/\partial x$ and $\partial u/\partial y$ are assumed constant in the region (element), then

$$\frac{\partial u}{\partial x} = \frac{1}{A} \int_S u \, dy \quad (3.65)$$

$$\frac{\partial u}{\partial y} = -\frac{1}{A} \int_S u \, dx \quad (3.66)$$

Assuming that u varies linearly between element nodes, the integrals in eqs. (3.65) and (3.66) are readily evaluated:

$$\frac{\partial u}{\partial x} = \frac{1}{A} \{ \bar{u}_{12}(y_2 - y_1) + \bar{u}_{23}(y_3 - y_2) + \bar{u}_{34}(y_4 - y_3) + \bar{u}_{41}(y_1 - y_4) \} \quad (3.67)$$

$$\frac{\partial u}{\partial y} = -\frac{1}{A} \{ \bar{u}_{12}(x_2 - x_1) + \bar{u}_{23}(x_3 - x_2) + \bar{u}_{34}(x_4 - x_3) + \bar{u}_{41}(x_1 - x_4) \} \quad (3.68)$$

where $\bar{u}_{12} = (u_1 + u_2)/2$ etc.

Equations (3.67) and (3.68) can be cast in the form:

$$\frac{\partial u}{\partial x} = \frac{1}{2A} \{ (y_2 - y_4)(u_1 - u_3) - (y_1 - y_3)(u_2 - u_4) \} \quad (3.69)$$

$$\frac{\partial u}{\partial y} = -\frac{1}{2A} \{ (x_2 - x_4)(u_1 - u_3) - (x_1 - x_3)(u_2 - u_4) \} \quad (3.70)$$

where A is the area of the element:

$$2A = (y_2 - y_4)(x_1 - x_3) - (x_2 - x_4)(y_1 - y_3) \quad (3.71)$$

The displacement gradients are substituted into the strain-displacement eqs. (2.64)-

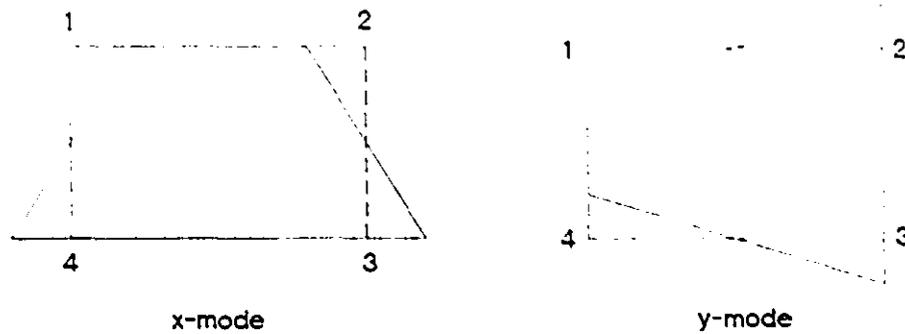


Figure 3.8: Hour-glass modes for constant-strain quadrilateral element

(2.66) to obtain the strains ε_{xx} , ε_{yy} and ε_{xy} .

3.6.6 Hour-Glass Control

Hour-glassing was introduced in section 3.3. In this section, the cause of hour-glassing is more closely examined and techniques of controlling hour-glass modes are presented.

A careful examination of the displacement-gradient equations (3.69) and (3.70) reveals that there are certain displacement fields, which are not rigid body motions, that yield zero displacement gradients, hence zero strains and stresses. For example, if δu is added to u_1 and u_3 but subtracted from u_2 and u_4 , as shown in Fig. 3.8 for $\delta u = \delta u_x$ (x-mode) and for $\delta u = \delta u_y$ (y-mode), the displacement gradients $\partial u / \partial x$ and $\partial u / \partial y$ are unchanged. Producing no stresses (and therefore no forces), the presence of such a displacement field is unresisted and can result in mesh instability. A rectangular mesh would be distorted into a pattern of *hour-glass* shapes. The absence of stresses implies zero energy. As a result, the terms *spurious zero-energy* or *kinematic modes*

are also often used.

The physical basis for hour-glassing is that prescription of a uniform strain field throughout the element results in an over-determined problem: there are eight independent displacement components per element (two at each node) but only six modes of deformation (three components of uniform strain and three rigid-body motions). The remaining two modes are realized as hour-glass modes.

A very innovative hour-glass control scheme was recently proposed by Last and Harkness[18]. It is based on the recognition that a truly uniform strain field exists within the sub-element formed by joining the mid-points of the sides of the element. Hour-glass modes are eliminated by requiring that the velocity field of the entire element be consistent with that of the constant-strain sub-element. Formally, with reference to an arbitrary point 0 within the sub-element, the velocity at any other point within the element is related to the velocity at 0, by the chain rule, as follows:

$$v = v_0 + \frac{\partial v}{\partial x}(x - x_0) + \frac{\partial v}{\partial y}(y - y_0) \quad (3.72)$$

It is computationally convenient to select the centroid of the element as the reference point. Thus,

$$x_0 = (x_1 + x_2 + x_3 + x_4)/4 \quad (3.73)$$

$$y_0 = (y_1 + y_2 + y_3 + y_4)/4 \quad (3.74)$$

$$v_0 = (v_1 + v_2 + v_3 + v_4)/4 \quad (3.75)$$

The nodal velocities, consistent with a constant-strain element, are determined from eq. (3.72). It can be shown that

$$\Delta v_1 = -\Delta v_2 = \Delta v_3 = -\Delta v_4 = \Delta h \quad (3.76)$$

where Δv_1 is the difference, at node 1, between the corrected, by eq. (3.72), and the uncorrected velocity etc. Δh represents the *amplitude* of the hour glass correction for the element. The amplitude in the x-direction will, of course, be different from that in the y-direction.

The change in momentum of the element as a result of the velocity correction is given by

$$\Delta(mv) = m_1\Delta v_1 + m_2\Delta v_2 + m_3\Delta v_3 + m_4\Delta v_4 \quad (3.77)$$

which, by virtue of eqs. (3.14) and (3.76), yields

$$\Delta(mv) = 0 \quad (3.78)$$

Thus, the momentum of the element is conserved. Non-conservation of the element momentum would introduce a fictitious body force into the momentum conservation equation of the control volumes and, therefore, produce inaccurate results.

The nodal velocity correction scheme, eq. (3.72), pertains to the nodes of a single element. Other than corner nodes, nodes are shared by more than one element, with each element requiring, in general, a different velocity correction. The global nodal

velocity correction may be obtained by equating the global change in the momentum of the node to the sum of the local (i.e. element-based) changes.

It is essential that the hour-glass velocity corrections satisfy the governing boundary conditions. Thus, at a displacement-controlled boundary, no correction should be applied. However, at a stress-controlled boundary, where nodes are free to move and may undergo hour-glass motions, hour-glass velocity correction is called for.

3.7 Wheel Subsystem

3.7.1 Momentum Conservation

Discretization of the integral terms of the momentum conservation equations (2.68), (2.69) and (2.70), i.e.,

$$\int_{S(t)} \sigma_x dS, \quad \int_{S(t)} \sigma_y dS, \quad \text{and} \quad \int_{S(t)} \sigma_t dS$$

is considered in section 3.8—Boundary Conditions.

3.7.2 Time Integration

The time integration schemes of section 3.6.2 are applicable to the linear as well as to the rotational motion of the wheel with θ , ω and α replacing d , \dot{d} and \ddot{d} , respectively.

3.7.3 Mass Conservation

The mass of the wheel is specified and remains constant, i.e.,

$$m = m_0 \tag{3.79}$$

where m_0 is the initial (specified) mass of the wheel.

3.7.4 Constitutive Equations

A zero strain-tensor prevails—spatially and temporally—for a rigid wheel. The stress state within the wheel is not needed.

3.8 Boundary Conditions

3.8.1 Inter-Control-Volume Boundary

For computational efficiency and simplicity, the inter-control-volume boundary tractions are determined element-by-element:

$$\int_{cs(1)\cdot e} \sigma_x dS = \sigma_{xx}(y_4 - y_2)/2 - \sigma_{xy}(x_4 - x_2)/2 \quad (3.80)$$

$$\int_{cs(1)\cdot e} \sigma_y dS = \sigma_{xy}(y_4 - y_2)/2 - \sigma_{yy}(x_4 - x_2)/2 \quad (3.81)$$

$$\int_{cs(2)\cdot e} \sigma_x dS = \sigma_{xx}(y_1 - y_3)/2 - \sigma_{xy}(x_1 - x_3)/2 \quad (3.82)$$

$$\int_{cs(2)\cdot e} \sigma_y dS = \sigma_{xy}(y_1 - y_3)/2 - \sigma_{yy}(x_1 - x_3)/2 \quad (3.83)$$

$$\int_{cs(3)\cdot e} \sigma_x dS = - \int_{cs(1)\cdot e} \sigma_x dS \quad (3.84)$$

$$\int_{cs(3)\cdot e} \sigma_y dS = - \int_{cs(1)\cdot e} \sigma_y dS \quad (3.85)$$

$$\int_{cs(4)\cdot e} \sigma_x dS = - \int_{cs(2)\cdot e} \sigma_x dS \quad (3.86)$$

$$\int_{cs(4)\cdot e} \sigma_y dS = - \int_{cs(2)\cdot e} \sigma_y dS \quad (3.87)$$

per unit width of control volume.

3.8.2 Transmitting Boundary

We rewrite the transmitting boundary tractions, eqs. (2.119) and (2.120), in the form

$$\sigma_n = -\rho c_d^2 \frac{\bar{v}_n}{1 - \bar{v}_n} \quad (3.88)$$

$$\sigma_t = -\rho c_s^2 \frac{\bar{v}_t}{1 - \bar{v}_t} \quad (3.89)$$

where

$$\bar{v}_n = \frac{v_n}{c_d} \quad (3.90)$$

$$\bar{v}_t = \frac{v_t}{c_s} \quad (3.91)$$

Then making use of the binomial series expansion,

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + \dots \quad (3.92)$$

We cast eqs. (3.88) and (3.89) in the form

$$\sigma_n = -\rho c_d^2 (\bar{v}_n + \bar{v}_n^2 + \bar{v}_n^3 + \dots) \quad (3.93)$$

$$\sigma_t = -\rho c_s^2 (\bar{v}_t + \bar{v}_t^2 + \bar{v}_t^3 + \dots) \quad (3.94)$$

Retaining only terms up to 2nd order, we have

$$\sigma_n = -\rho c_d^2 (\bar{v}_n + \bar{v}_n^2) \quad (3.95)$$

$$= -\rho c_d v_n - \rho v_n^2 \quad (3.96)$$

$$\sigma_t = -\rho c_s^2 (\bar{v}_t + \bar{v}_t^2) \quad (3.97)$$

$$= -\rho c_s v_t - \rho v_t^2 \quad (3.98)$$

The transmitting boundary tractions are evaluated element-by-element:

$$\int_{cs(1)\cdot\epsilon} \sigma_n ds = -\rho c_d \int_0^{\ell/2} v_n ds - \rho \int_0^{\ell/2} v_n^2 ds \quad (3.99)$$

$$\int_{cs(1)\cdot\epsilon} \sigma_t ds = -\rho c_s \int_0^{\ell/2} v_t ds - \rho \int_0^{\ell/2} v_t^2 ds \quad (3.100)$$

$$\int_{cs(2)\cdot\epsilon} \sigma_n ds = -\rho c_d \int_{\ell/2}^{\ell} v_n ds - \rho \int_{\ell/2}^{\ell} v_n^2 ds \quad (3.101)$$

$$\int_{cs(2)\cdot\epsilon} \sigma_t ds = -\rho c_s \int_{\ell/2}^{\ell} v_t ds - \rho \int_{\ell/2}^{\ell} v_t^2 ds \quad (3.102)$$

The integrals on the right hand sides of eqs. (3.99)–(3.100) are determined as follows.

A linear variation of velocity between adjacent nodes yields

$$\bar{v} = \left(1 - \frac{s}{\ell}\right) \bar{v}_1 + \frac{s}{\ell} \bar{v}_2 \quad 0 \leq s \leq \ell \quad (3.103)$$

where ℓ is the length of side 1-2 and s is the distance measured along 1-2 from 1 to 2.

Thus,

$$v_n = \left[\left(1 - \frac{s}{\ell}\right) \bar{v}_1 + \frac{s}{\ell} \bar{v}_2 \right] \cdot \bar{n} \quad (3.104)$$

$$v_t = \left[\left(1 - \frac{s}{\ell}\right) \bar{v}_1 + \frac{s}{\ell} \bar{v}_2 \right] \cdot \bar{t} \quad (3.105)$$

where \bar{n} and \bar{t} are the normal and tangential unit vectors, respectively. Integration of eqs. (3.104) and (3.105) yields

$$\int v_n ds = \left[\left(s - \frac{s^2}{2\ell}\right) \bar{v}_1 + \frac{s^2}{2\ell} \bar{v}_2 \right] \cdot \bar{n} + C \quad (3.106)$$

$$\int v_t ds = \left[\left(s - \frac{s^2}{2\ell}\right) \bar{v}_1 + \frac{s^2}{2\ell} \bar{v}_2 \right] \cdot \bar{t} + D \quad (3.107)$$

where C and D are integration constants. Evaluating the above integrals between the applicable limits, we obtain

$$\int_0^{\ell/2} v_n ds = \frac{\ell}{8} (3\bar{v}_1 + \bar{v}_2) \cdot \bar{n} \quad (3.108)$$

$$\int_0^{\ell/2} v_t ds = \frac{\ell}{8} (3\bar{v}_1 + \bar{v}_2) \cdot \bar{t} \quad (3.109)$$

$$\int_{\ell/2}^{\ell} v_n ds = \frac{\ell}{8} (3\bar{v}_2 + \bar{v}_1) \cdot \bar{n} \quad (3.110)$$

$$\int_{\ell/2}^{\ell} v_t ds = \frac{\ell}{8} (3\bar{v}_2 + \bar{v}_1) \cdot \bar{t} \quad (3.111)$$

We now consider the 2nd order terms. A simple algebraic manipulation yields

$$v_n^2 = (\bar{v}_1 \cdot \bar{n})^2 \left(1 - \frac{s}{\ell}\right)^2 + 2(\bar{v}_1 \cdot \bar{n})(\bar{v}_2 \cdot \bar{n}) \left(\frac{s}{\ell} - \frac{s^2}{\ell^2}\right) + (\bar{v}_2 \cdot \bar{n})^2 \frac{s^2}{\ell^2} \quad (3.112)$$

$$v_t^2 = (\bar{v}_1 \cdot \bar{t})^2 \left(1 - \frac{s}{\ell}\right)^2 + 2(\bar{v}_1 \cdot \bar{t})(\bar{v}_2 \cdot \bar{t}) \left(\frac{s}{\ell} - \frac{s^2}{\ell^2}\right) + (\bar{v}_2 \cdot \bar{t})^2 \frac{s^2}{\ell^2} \quad (3.113)$$

Integration yields

$$\begin{aligned} \int v_n^2 ds &= -\frac{\ell}{3} (\bar{v}_1 \cdot \bar{n})^2 \left(1 - \frac{s}{\ell}\right)^3 + 2(\bar{v}_1 \cdot \bar{n})(\bar{v}_2 \cdot \bar{n}) \left(\frac{s^2}{2\ell} - \frac{s^3}{3\ell^2}\right) \\ &\quad + (\bar{v}_2 \cdot \bar{n})^2 \frac{s^3}{3\ell^2} + C \end{aligned} \quad (3.114)$$

$$\begin{aligned} \int v_t^2 ds &= -\frac{\ell}{3} (\bar{v}_1 \cdot \bar{t})^2 \left(1 - \frac{s}{\ell}\right)^3 + 2(\bar{v}_1 \cdot \bar{t})(\bar{v}_2 \cdot \bar{t}) \left(\frac{s^2}{2\ell} - \frac{s^3}{3\ell^2}\right) \\ &\quad + (\bar{v}_2 \cdot \bar{t})^2 \frac{s^3}{3\ell^2} + D \end{aligned} \quad (3.115)$$

where C and D are integration constants. Evaluation of the 2nd order integrals between applicable limits yields

$$\int_0^{\ell/2} v_n^2 ds = \frac{\ell}{24} \left[7(\bar{v}_1 \cdot \bar{n})^2 + 4(\bar{v}_1 \cdot \bar{n})(\bar{v}_2 \cdot \bar{n}) + (\bar{v}_2 \cdot \bar{n})^2 \right] \quad (3.116)$$

$$\int_0^{\ell/2} v_t^2 ds = \frac{\ell}{24} \left[7(\bar{v}_1 \cdot \bar{t})^2 + 4(\bar{v}_1 \cdot \bar{t})(\bar{v}_2 \cdot \bar{t}) + (\bar{v}_2 \cdot \bar{t})^2 \right] \quad (3.117)$$

$$\int_{\ell/2}^{\ell} v_n^2 ds = \frac{\ell}{24} \left[7(\bar{v}_2 \cdot \bar{n})^2 + 4(\bar{v}_1 \cdot \bar{n})(\bar{v}_2 \cdot \bar{n}) + (\bar{v}_1 \cdot \bar{n})^2 \right] \quad (3.118)$$

$$\int_{\ell/2}^{\ell} v_t^2 ds = \frac{\ell}{24} \left[7(\bar{v}_2 \cdot \bar{t})^2 + 4(\bar{v}_1 \cdot \bar{t})(\bar{v}_2 \cdot \bar{t}) + (\bar{v}_1 \cdot \bar{t})^2 \right] \quad (3.119)$$

The length, ℓ , of side 1-2 is given by

$$\ell = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3.120)$$

The unit normal and tangential vectors are given by

$$\vec{n} = n_x \vec{i} + n_y \vec{j} \quad (3.121)$$

$$\vec{t} = t_x \vec{i} + t_y \vec{j} \quad (3.122)$$

where

$$t_x = \frac{1}{\ell} (x_2 - x_1) \quad (3.123)$$

$$t_y = \frac{1}{\ell} (y_2 - y_1) \quad (3.124)$$

$$n_x = t_y \quad (3.125)$$

$$n_y = -t_x \quad (3.126)$$

It can easily be shown that

$$\int \sigma_x ds = n_x \int \sigma_n ds + t_x \int \sigma_t ds \quad (3.127)$$

$$\int \sigma_y ds = n_y \int \sigma_n ds + t_y \int \sigma_t ds \quad (3.128)$$

3.8.3 Free-Surface

We consider a stress-free surface, resulting in no contribution to the control-volume traction. As indicated previously, a stressed surface can be readily accommodated. One simply integrates the given surface stresses along the appropriate control-volume boundaries.

3.8.4 Wheel-Soil Interface

In the ensuing development of the wheel-soil contact equations, the wheel is considered the ‘driver’ or ‘master’ and the contacting soil-surface control-volumes are relegated to the position of ‘passengers’ or ‘slaves’. In other words, kinematic compatibility is enforced by the wheel.

Let $cv(i)$ denote an arbitrary soil surface control volume and let the contact surface between $cv(i)$ and the wheel w be denoted

$$cs(i) \cdot w$$

The force exerted by the wheel on $cs(i)$, given by the integral of the contact stress vectors over the contact surface $cs(i) \cdot w$, is discretized as follows:

$$\int_{cs(i) \cdot w} \bar{\sigma} ds = \bar{p}_i \quad (3.129)$$

where \bar{p}_i acts at *node* i , located on $cs(i)$. The discretized momentum conservation

equation of control volume $cv(i)$, in the normal direction \vec{n}_i , is given by

$$m_i \vec{a}_i \cdot \vec{n}_i = \vec{f}_i \cdot \vec{n}_i + \vec{p}_i \cdot \vec{n}_i \quad (3.130)$$

where \vec{n}_i is the unit outward normal vector, with respect to the wheel, at node i ; \vec{f}_i is the sum of all forces, acting on control volume $cv(i)$, less the wheel contact force \vec{p}_i ; m_i is mass of control volume $cv(i)$ and \vec{a}_i is acceleration of node i .

The normal acceleration of node i equals that of the wheel by virtue of kinematic compatibility:

$$\vec{a}_i \cdot \vec{n}_i = \vec{a}_w \cdot \vec{n}_i \quad (3.131)$$

where \vec{a}_w is the acceleration of the wheel. Substituting eq. (3.131) into (3.130), we obtain

$$\vec{p}_i \cdot \vec{n}_i = m_i \vec{a}_w \cdot \vec{n}_i - \vec{f}_i \cdot \vec{n}_i \quad (3.132)$$

relating the normal contact force on node i to the wheel acceleration. Equation (3.132) is applicable to all contact nodes, regardless of their stick-slip condition.

The integral of the interface **slip** constitutive equation (2.134) over the contact surface $cs(i) \cdot w$, i.e.,

$$\int_{cs(i) \cdot w} \vec{\sigma} \cdot \vec{t} \, ds = \int_{cs(i) \cdot w} \zeta (c_a + \mu \sigma \cdot \vec{n}) \, ds \quad (3.133)$$

is discretized as follows:

$$\int_{cs(i) \cdot w} \bar{\sigma} \cdot \bar{t} \, ds = \bar{p}_i \cdot \bar{t}_i \quad (3.134)$$

$$\int_{cs(i) \cdot w} \zeta (c_a + \mu \bar{\sigma} \cdot \bar{n}) \, ds = \zeta_i (c_a \ell_i + \mu \bar{p}_i \cdot \bar{n}_i) \quad (3.135)$$

where ℓ_i is the length of $cs(i) \cdot w$; ζ_i is direction of slip (DOS) parameter at node i ; \bar{t}_i is tangential unit vector at node i . Substituting eqs. (3.134) and (3.135) into eq. (3.133), we obtain

$$\bar{p}_i \cdot \bar{t}_i = \zeta_i (c_a \ell_i + \mu \bar{p}_i \cdot \bar{n}_i) \quad (3.136)$$

A node in stick condition is subjected to the same acceleration as the wheel, i.e.,

$$\bar{a}_i \cdot \bar{t}_i = \bar{a}_w \cdot \bar{t}_i + r \alpha_w \quad (3.137)$$

where α_w is the angular acceleration of the wheel (α_w and α^w will be used interchangeably) and r is the radius of the wheel. The discretized momentum conservation equation of control volume $cv(i)$, in the tangential direction \bar{t}_i , is now written in the form:

$$\bar{p}_i \cdot \bar{t}_i = m_i (\bar{a}_w \cdot \bar{t}_i + r \alpha_w) - \bar{f}_i \cdot \bar{t}_i \quad (3.138)$$

The linear momentum conservation equation of the wheel is given by

$$m_w \bar{a}_w = \bar{f}_w + \bar{p}_w \quad (3.139)$$

where \vec{f}_w is the sum of all forces, acting on the wheel, less the soil contact force \vec{p}_w and m_w is the mass of the wheel. By Newton's Third Law (action and reaction),

$$\vec{p}_w = - \sum_i \vec{p}_i \quad (3.140)$$

We express \vec{p}_i in the form

$$\vec{p}_i = (\vec{p}_i \cdot \vec{n}_i) \vec{n}_i + (\vec{p}_i \cdot \vec{t}_i) \vec{t}_i \quad (3.141)$$

and substitute eqs. (3.140) and (3.141) into eq. (3.139) to obtain

$$m_w \vec{a}_w = \vec{f}_w - \sum_{i \in stk} [(\vec{p}_i \cdot \vec{n}_i) \vec{n}_i + (\vec{p}_i \cdot \vec{t}_i) \vec{t}_i] - \sum_{i \in slp} [(\vec{p}_i \cdot \vec{n}_i) \vec{n}_i + (\vec{p}_i \cdot \vec{t}_i) \vec{t}_i] \quad (3.142)$$

where the notation $i \in stk$ and $i \in slp$ refer to nodes in stick and slip modes, respectively.

Similarly, the equation of angular momentum of the wheel is cast in the form

$$I \alpha_w = T - r \sum_{i \in stk} \vec{p}_i \cdot \vec{t}_i - r \sum_{i \in slp} \vec{p}_i \cdot \vec{t}_i \quad (3.143)$$

where I is the moment of inertia of the wheel;

We express \vec{a}_w in component form as

$$\vec{a}_w = a_x^w \vec{i} + a_y^w \vec{j} \quad (3.144)$$

and note that

$$\vec{f}_w = (P_x + W_x)\vec{i} + (P_y + W_y)\vec{j} \quad (3.145)$$

where P_x and P_y are the applied axle wheel load in the x and y directions respectively; T is the applied wheel torque; W_x and W_y are the wheel weight in the x and y directions, respectively.

Substituting eqs. (3.132), (3.136), (3.138), (3.144) and (3.145) into eqs. (3.142) and (3.143), we obtain, after some lengthy algebra,

$$\begin{bmatrix} M_{xx} & M_{xy} & M_{xz} \\ M_{yx} & M_{yy} & M_{yz} \\ M_{zx} & M_{zy} & M_{zz} \end{bmatrix} \begin{Bmatrix} a_x^w \\ a_y^w \\ \alpha^w \end{Bmatrix} = \begin{Bmatrix} P_x + F_x + W_x \\ P_y + F_y + W_y \\ T + F_z \end{Bmatrix} \quad (3.146)$$

where

$$M_{xx} = m_w + \sum_{i \in stk} m_i(n_x^2 + t_x^2)_i + \sum_{i \in slp} m_i(n_x^2 + \mu\zeta t_x n_x)_i \quad (3.147)$$

$$M_{xy} = \sum_{i \in stk} m_i(n_x n_y + t_x t_y)_i + \sum_{i \in slp} m_i(n_x n_y + \mu\zeta t_x n_y)_i \quad (3.148)$$

$$M_{xz} = \sum_{i \in stk} r m_i(t_x)_i \quad (3.149)$$

$$M_{yx} = \sum_{i \in stk} m_i(n_y n_x + t_y t_x)_i + \sum_{i \in slp} m_i(n_y n_x + \mu\zeta t_y n_x)_i \quad (3.150)$$

$$M_{yy} = m_w + \sum_{i \in stk} m_i(n_y^2 + t_y^2)_i + \sum_{i \in slp} m_i(n_y^2 + \mu\zeta t_y n_y)_i \quad (3.151)$$

$$M_{yz} = \sum_{i \in stk} r m_i(t_y)_i \quad (3.152)$$

$$M_{zx} = \sum_{i \in stk} r m_i(t_x)_i + \sum_{i \in slp} r m_i(\mu \zeta n_x)_i \quad (3.153)$$

$$M_{zy} = \sum_{i \in stk} r m_i(t_y)_i + \sum_{i \in slp} r m_i(\mu \zeta n_y)_i \quad (3.154)$$

$$M_{zz} = I + \sum_{i \in stk} r^2 m_i \quad (3.155)$$

can be construed as mass matrix coefficients, and

$$\begin{aligned} F_x &= \sum_{i \in stk} [f_x(n_x^2 + t_x^2) + f_y(n_x n_y + t_x t_y)]_i \\ &+ \sum_{i \in slp} [f_x^i(n_x^2 + \mu \zeta n_x t_x)_i + f_y(n_x n_y + \mu \zeta t_x n_y)]_i \\ &- \sum_{i \in slp} (\zeta c_a l t_x)_i \end{aligned} \quad (3.156)$$

$$\begin{aligned} F_y &= \sum_{i \in stk} [f_x(n_x n_y + t_x t_y) + f_y(n_y^2 + t_y^2)]_i \\ &+ \sum_{i \in slp} [f_x(n_x n_y + \mu \zeta n_x t_y) + f_y(n_y^2 + \mu \zeta n_y t_y)]_i \\ &- \sum_{i \in slp} (\zeta c_a l t_y)_i \end{aligned} \quad (3.157)$$

$$F_z = \sum_{i \in stk} (r f_i)_i + \sum_{i \in slp} (r \mu \zeta f_n)_i - \sum_{i \in slp} (r \zeta c_a l)_i \quad (3.158)$$

represent the soil-imposed forces. The unit normal and tangential vectors are given by

$$n_x = (x_i - x_w)/r \quad (3.159)$$

$$n_y = (y_i - y_w)/r \quad (3.160)$$

$$t_x = -n_y \quad (3.161)$$

$$t_y = n_x \quad (3.162)$$

where x_w and y_w denote the coordinates of the center of the wheel. The direction of slip parameter ζ_i is given by

$$\zeta_i = \frac{v_s}{|v_s|} \quad (3.163)$$

where v_s is the slip velocity:

$$v_s = \bar{v}_w \cdot \bar{t} + \omega r - \bar{v}_i \cdot \bar{t} \quad (3.164)$$

Solving eq. (3.146) for a_x^w , a_y^w and α^w in terms of $\{P_x, P_y, T\}$, using the row reduction method, we obtain

$$\alpha^w = \frac{D_z N_{yy} - D_y N_{zy}}{N_{yy} N_{zz} - N_{yz} N_{zy}} \quad (3.165)$$

$$a_y^w = \frac{D_y N_{zz} - D_z N_{yz}}{N_{yy} N_{zz} - N_{yz} N_{zy}} \quad (3.166)$$

$$a_x^w = (P_x + F_x + W_x - M_{xy} a_y^w - M_{xz} \alpha^w) / M_{xx} \quad (3.167)$$

where

$$N_{yy} = M_{xy} M_{yx} - M_{xx} M_{yy} \quad (3.168)$$

$$N_{yz} = M_{xz} M_{yx} - M_{xx} M_{yz} \quad (3.169)$$

$$N_{zy} = M_{xy}M_{zx} - M_{xx}M_{zy} \quad (3.170)$$

$$N_{zz} = M_{xz}M_{zx} - M_{xx}M_{zz} \quad (3.171)$$

and

$$D_y = M_{yx}(P_x + F_x + W_x) - M_{xx}(P_y + F_y + W_y) \quad (3.172)$$

$$D_z = M_{zx}(P_x + F_x + W_x) - M_{xx}(T + F_z) \quad (3.173)$$

The wheel accelerations are, finally, substituted into the applicable equation (stick or slip) to obtain the wheel-soil interface nodal forces. We begin with the premise that all contact nodes are in stick mode and solve for the contact forces. Those nodes which fail the 'stick test', i.e. violate the maximum shear stress criterion, are re-classified as slip nodes and the solution process repeated.

We note that in the absence of wheel-soil contact, $M_{xx} = M_{yy} = m_w$, $M_{zz} = I$, $M_{xy} = M_{xz} = M_{yx} = M_{yz} = M_{zx} = M_{zy} = 0$ and eqs. (3.165), (3.166) and (3.167) reduce, respectively, to

$$\alpha^w = T/I \quad (3.174)$$

$$a_y^w = (P_y + W_y)/m_w \quad (3.175)$$

$$a_x^w = (P_x + W_x)/m_w \quad (3.176)$$

References

- [1] PATANKAR, S. V. *Numerical Heat Transfer and Fluid Flow*. McGraw-Hill Book Co., New York, 1980.
- [2] BALIGA, B. R. AND PATANKAR, S. V. A New Finite Element Formulation for Convection-Diffusion Problems. *Numerical Heat Transfer*, 3(4):393-409, 1980.
- [3] BATHE, K-J. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
- [4] ZIENKIEWICZ, O. C. *Finite Elements in Engineering Science*. McGraw-Hill Book Co., New York, 1971.
- [5] HUEBER K. H. *The Finite Element Method for Engineers*. John Wiley, New York, 1975.
- [6] THOMPSON J. F., WARSI, Z. U. A. AND MASTIN, C. W. *Numerical Grid Generation*. North-Holland, New York, 1985.
- [7] GREENSPAN, D. *Discrete Models*. Addison-Wesley, Reading, Mass, 1973.
- [8] GOULD, H. AND TOBOCHNIK, J. *An Introduction to Computer Simulation Methods: Applications to Physical Systems, Part 1*. Addison-Wesley, Reading, Mass, 1988.
- [9] NEWMARK, N. M. A Method of Computation for Structural Dynamics. *Journal of the Engineering Mechanics Division*, ASCE, Vol. 85, No. EM3, July 1959, pp. 67-94.
- [10] GRANT, J. E. Response Computation using Truncated Taylor Series. *Journal of the Engineering Mechanics Division*, ASCE, Vol. 97, No. EM2, April 1971, pp 295-304.
- [11] HUGHES, T. J. R. Analysis of Transient Algorithms with Particular Reference to Stability Behavior. In BELYTSCHKO, T. AND HUGHES, T. J. R. *Computational Methods for Transient Analysis*. Elsevier Science Publishers, Amsterdam, The Netherlands, 1983, pp 68-155.
- [12] THOMAS, G. B. *Calculus and Analytic Geometry*. Addison-Wesley, Reading, Mass, 1968.

- [13] HATA, S. AND TATEYAMA, K. Theoretical Approach to Impact Soil Compaction Through Plastic Wave Propagation. *J. Terramechanics*, 28(4):349-358, 1991.
- [14] MOBLEY, C. D. AND STEWART, R. J. On the Numerical Generation of Boundary-Fitted Orthogonal Curvilinear Coordinate Systems. *J. Computational Physics*, 34:124-135, 1980.
- [15] KANG, I. S. AND LEAL, L. G. Orthogonal Grid Generation in a 2D Domain via the Boundary Integral Technique. *J. Computational Physics*, 102:78-87, 1992.
- [16] SCHNEIDER, G. E. AND ZEDAN, M. Control-Volume-Based Finite Element Formulation of the Heat Conduction Equation. In COLLICOT, H. E. AND BAUER, P. E. *Spacecraft Thermal Control, Design, and Operation*, pp 305-327. American Institute of Aeronautics and Astronautics (AIAA), 1983.
- [17] HIRSCH, C. *Numerical Computation of Internal and External Flows*. Vol 1: Fundamentals of Numerical Discretization. 1988
- [18] LAST, N. C. AND HARKNESS, R. M. Kinematic (or Hour-Glass) Mode Control for Uniform Strain Quadrilateral by an Assumed Strain Technique. *Int. Journal for Numerical and Analytical Methods in Geomechanics*, 13:381-410, 1989.
- [19] HORNBECK R. W. *Numerical Methods*. Quantum Pub., New York, 1975.

Chapter 4

Dynamic Grid Generation

4.1 Introduction

This chapter is concerned with the discretization of the computational domain into quadrilateral elements. The construction of control volumes, as detailed in the chapter 3, is trivial once a suitable finite-element mesh has been generated. An arbitrary-shaped computational domain may be discretized into quadrilateral elements by using any one of a number of body-fitted grid generation techniques, the most common of which is based on the numerical solution of a system of coupled second-order partial differential equations (Thompson, Warsi and Mastin [1], Mobley and Stewart [2], Kang and Leal [3], Rangwalla and Munson [4]).

Many of these approaches are derived, directly or indirectly, from the concept of conformal mappings. The primary drawback with conformal mappings is the requirement that scale factors be equal in all directions, resulting in equal-sized elements. Clearly, it is desirable, for sake of accuracy, that smaller-sized elements be placed, or equivalently that grid lines be concentrated, in regions of expected rapid changes. Differences in existing grid-generation procedures are centered, for the most part, around

the manner in which grid/element concentration is achieved. It is desirable, in wheel-soil interaction, that elements be concentrated near the soil surface where, as the wheel travels, the most rapid changes are expected to occur.

The term 'dynamic grid generation' as used herein refers to the dynamic relaxation formulation and solution of the grid-generating equations.

4.2 Regular Domain

A regular domain, such as a rectangle, can be readily discretized by algebraic interpolation techniques. With reference to the node numbering scheme outlined in chapter 3, a *rectangular* computational domain is discretized into *rectangular* finite elements with nodal coordinates

$$x(i, j) = x(0, 0) + \frac{x(i_{\max}, j_{\max}) - x(0, 0)}{i_{\max}} f(i) \quad (4.1)$$

$$y(i, j) = y(0, 0) + \frac{y(i_{\max}, j_{\max}) - y(0, 0)}{j_{\max}} g(j) \quad (4.2)$$

where $f(i)$ and $g(j)$ are monotonically increasing functions of their arguments, and

$$f(0) = 0 \quad (4.3)$$

$$f(i_{\max}) = i_{\max} \quad (4.4)$$

$$g(0) = 0 \quad (4.5)$$

$$g(j_{\max}) = j_{\max} \quad (4.6)$$

The nodes $(0, 0)$ and (i_{\max}, j_{\max}) are diagonally opposite corners of the rectangular computational domain. The functions $f(i)$ and $g(j)$ control the packing of the nodes and will be referred to as *packing functions*.

To concentrate the nodes smoothly towards the soil surface ($j = 0$), the packing function $g(j)$ must satisfy the condition

$$\frac{dg}{dj}(j) < \frac{dg}{dj}(j + 1) \quad (4.7)$$

for all j . Consider the (nonlinear) hyperbolic sinh function

$$g(j) = A \sinh(Bj) + C \quad (4.8)$$

where

$$\sinh(x) = \frac{1}{2} (e^x - e^{-x}) \quad (4.9)$$

and A , B and C are constants, to be determined. Imposition of the boundary conditions (4.5) and (4.6) yields

$$C = 0 \quad (4.10)$$

$$A = \frac{j_{\max}}{\sinh(Bj_{\max})} \quad (4.11)$$

Thus, eqn. (4.8) becomes

$$g(j) = \frac{j_{\max} \sinh(Bj)}{\sinh(Bj_{\max})} \quad (4.12)$$

where B remains an arbitrary constant. The parameter B controls the intensity of the packing and will be referred to as the *intensity factor*. The intensity of the packing (towards the surface $j = 0$) increases with the (absolute) value of B .

The packing function $g(j)$ is indeterminate for $B = 0$:

$$\begin{aligned} g(j) &= \frac{j_{\max} \sinh(0)}{\sinh(0)} \\ &= \frac{0}{0} \end{aligned} \quad (4.13)$$

According to *l'Hôpital's rule* [8], if $F(a) = G(a) = 0$, and if, as t approaches a , the limit of $F'(t)/G'(t)$ exists, then

$$\lim_{t \rightarrow a} \frac{F(t)}{G(t)} = \lim_{t \rightarrow a} \frac{F'(t)}{G'(t)} \quad (4.14)$$

where prime ' denotes the derivative with respect to the argument t . Using *l'Hôpital's rule*, eq. (4.14), we obtain

$$\begin{aligned} \lim_{B \rightarrow 0} g(j) &= \lim_{B \rightarrow 0} \frac{j_{\max} \sinh(Bj)}{\sinh(Bj_{\max})} \\ &= \lim_{B \rightarrow 0} \frac{\partial [j_{\max} \sinh(Bj)] / \partial B}{\partial [\sinh(Bj_{\max})] / \partial B} \\ &= \lim_{B \rightarrow 0} \frac{j_{\max} j \cosh(Bj)}{j_{\max} \cosh(Bj_{\max})} \\ &= j \end{aligned} \quad (4.15)$$

Thus the mesh approaches uniformity, in the j -direction, as B approaches 0. To generate a uniform mesh in the j -direction, the linear function

$$g(j) = j \quad (4.16)$$

is computationally more efficient than setting $B = 0$ in the nonlinear equation (4.8).

A uniform spacing of the grid lines in the direction of travel (D.O.T) of the wheel appears logical. This is attained with the linear packing function:

$$f(i) = i \quad (4.17)$$

Notice that eqn. (4.17) satisfies the boundary conditions (4.3) and (4.4).

A schematic plot of the packing function (4.12) is shown in Fig. 4.1 for different intensity factors B : 0, 0.025, 0.05 and 0.1. ($j_{\max} = 90$).

Figures 4.2(a), 4.2(b), and 4.2(c) show rectangular meshes obtained with the packing functions

$$g(j) = j$$

$$g(j) = \frac{j_{\max} \sinh(0.2j)}{\sinh(0.2j_{\max})}$$

$$g(j) = \frac{j_{\max} \sinh(0.4j)}{\sinh(0.4j_{\max})}$$

respectively. In all cases, $f(i) = i$, $i_{\max} = 21$ and $j_{\max} = 11$.

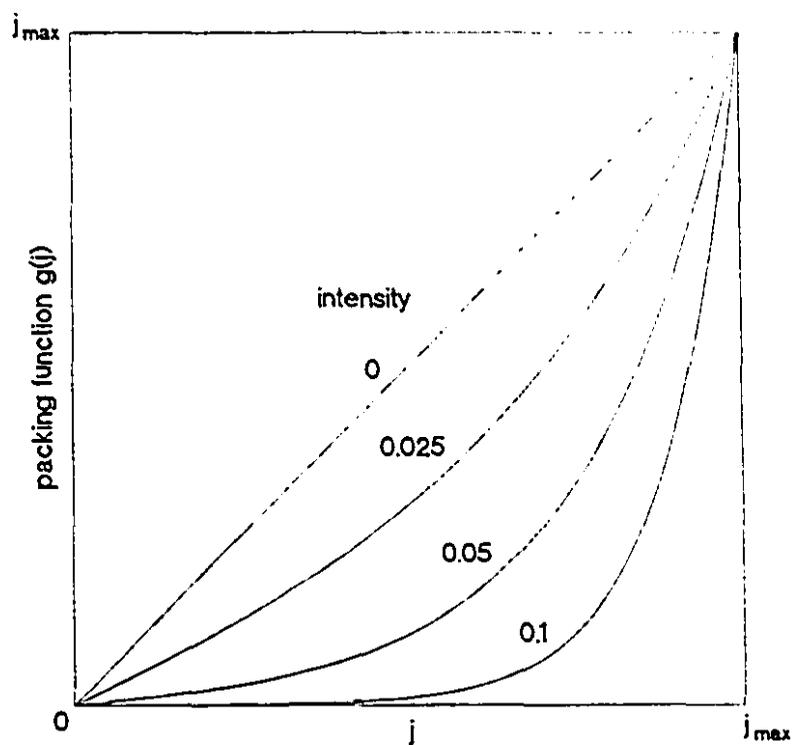


Figure 4.1: Packing function variation with intensity factor

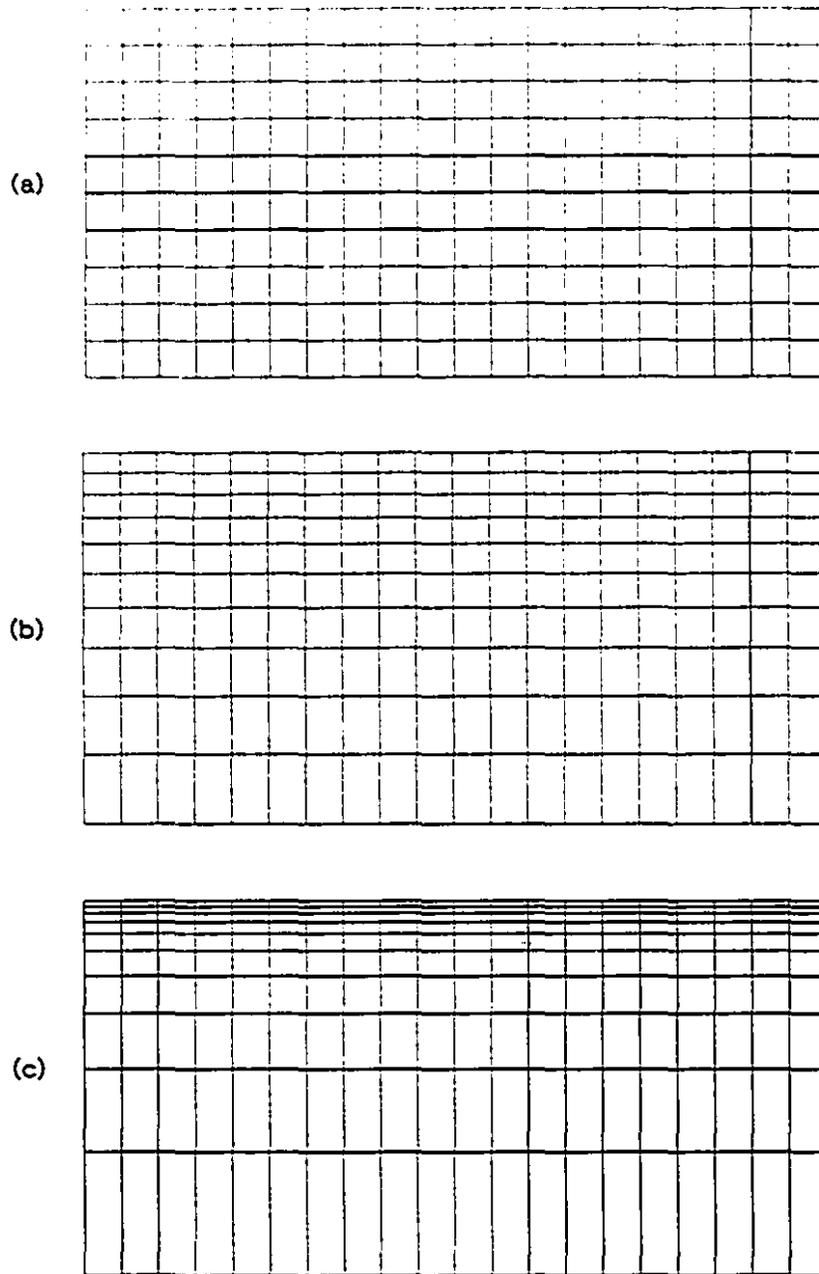


Figure 4.2: Rectangular-mesh generation with intensity factor: (a) 0; (b) .2; (c) .4

4.3 Complex Domain

4.3.1 Grid-Generation Equations

For an irregularly-shaped computational domain, we consider the orthogonal grid-generation equations of Mobley and Stewart[2]:

$$f_x = 0 \quad (4.18)$$

$$f_y = 0 \quad (4.19)$$

where

$$f_x = \frac{\partial^2 x}{\partial i^2} + \delta^2 \frac{\partial^2 x}{\partial j^2} - F \frac{\partial x}{\partial i} - \delta^2 G \frac{\partial x}{\partial j} \quad (4.20)$$

$$f_y = \frac{\partial^2 y}{\partial i^2} + \delta^2 \frac{\partial^2 y}{\partial j^2} - F \frac{\partial y}{\partial i} - \delta^2 G \frac{\partial y}{\partial j} \quad (4.21)$$

where

$$\delta = S \frac{f'(i)}{g'(j)} \quad (4.22)$$

$$F = \frac{f''(i)}{f'(i)} \quad (4.23)$$

$$G = \frac{g''(j)}{g'(j)} \quad (4.24)$$

$f(i)$ and $g(j)$ are the packing functions and the symbol prime ' denotes differentiation with respect to its argument. The constant S is known as the *scale factor* and is related to the *conformal module* \mathcal{M} of the domain [2, 4]:

$$\mathcal{M} = S \frac{i_{\max}}{j_{\max}} \quad (4.25)$$

Equation (4.25) is a necessary condition if the x and y coordinates of the four corner nodes $(0, 0)$, $(i_{\max}, 0)$, $(0, j_{\max})$ and (i_{\max}, j_{\max}) are to be specified. The conformal module of a rectangular domain is, by definition, the ratio of its length to its width:

$$\mathcal{M}_{rect} = \frac{\text{length}}{\text{width}} \quad (4.26)$$

For a simple irregular domain, the conformal module can be determined analytically [2]. In general, however, the analytical determination of the conformal module of a complex domain is very difficult. Consequently, a numerical approach is utilized (section 4.3.3).

We note that eqs. (4.20) and (4.21) may be cast in the form

$$f_x = \frac{\partial}{\partial i} \left(\frac{1}{\delta} \frac{\partial x}{\partial i} \right) + \frac{\partial}{\partial j} \left(\delta \frac{\partial x}{\partial j} \right) \quad (4.27)$$

$$f_y = \frac{\partial}{\partial i} \left(\frac{1}{\delta} \frac{\partial y}{\partial i} \right) + \frac{\partial}{\partial j} \left(\delta \frac{\partial y}{\partial j} \right) \quad (4.28)$$

of the equations governing heat conduction, seepage and torsion.

We seek to solve eqs. (4.18) and (4.19) for $x(i, j)$ and $y(i, j)$, respectively, at each

of the nodal points (i, j) .

4.3.2 Boundary Conditions

The boundary of the computational domain is discretized into four segments corresponding to $i = 0$, $i = i_{\max}$, $j = 0$ and $j = j_{\max}$. One boundary condition for solving eq. (4.18) for x and one boundary condition for solving eq. (4.19) for y are required on each of the segments. The Neumann boundary condition

$$\frac{\partial x}{\partial i} = \delta \frac{\partial y}{\partial j} \quad (4.29)$$

or

$$\frac{\partial y}{\partial i} = -\delta \frac{\partial x}{\partial j} \quad (4.30)$$

produces orthogonal grids. While orthogonal grids are not essential, they are generally desirable, particularly in curvilinear finite-difference schemes [4, 5, 6] for a number of reasons: (1) easy application of Neumann boundary conditions (2) simplifications in the finite-difference algorithm and (3) improved accuracy. Notice that eqs. (4.29) and (4.30) reduce to the *Cauchy-Riemann* equations when $\delta = 1$. A necessary second boundary condition is provided by the geometric equation of the boundary:

$$x = x(y) \quad (4.31)$$

or

$$y = y(x) \quad (4.32)$$

At the corners $(0, 0)$, $(i_{\max}, 0)$, $(0, j_{\max})$ and (i_{\max}, j_{\max}) of the computational domain, the values of x and y are specified. The Neumann condition, eq. (4.29) or (4.30), and the Dirichlet condition, eq. (4.31) or (4.32), are applied in a complementary manner at each boundary segment, as illustrated subsequently. At each boundary, a combination of one Neumann and one Dirichlet boundary condition is applied. We note that the generating equations (4.18) and (4.19) are coupled through the boundary conditions.

4.3.3 Finite-Difference Discretization

Grid-Generating Equations

The solution domain, characterized by integral values of i and j , is uniformly-spaced with $\Delta i = \Delta j$. We therefore take advantage of the relative simplicity and efficiency of the finite-difference method in handling such problems. At an arbitrary node (i, j) , the central finite-difference discretization of eqs. (4.20) and (4.21) yields

$$\begin{aligned} f_x &= (x_{i+1,j} - 2x_{i,j} + x_{i-1,j}) + \delta_{i,j}^2 (x_{i,j+1} - 2x_{i,j} + x_{i,j-1}) \\ &- F_i(x_{i+1,j} - x_{i-1,j})/2 - \delta_{i,j}^2 G_j(x_{i,j+1} - x_{i,j-1})/2 \end{aligned} \quad (4.33)$$

$$\begin{aligned} f_y &= (y_{i+1,j} - 2y_{i,j} + y_{i-1,j}) + \delta_{i,j}^2 (y_{i,j+1} - 2y_{i,j} + y_{i,j-1}) \\ &- F_i(y_{i+1,j} - y_{i-1,j})/2 - \delta_{i,j}^2 G_j(y_{i,j+1} - y_{i,j-1})/2 \end{aligned} \quad (4.34)$$

Boundary Conditions

Discretization of the boundary condition (4.30) yields

$$(y_{i+1,j} - y_{i-1,j})/2 = -\delta_{i,j}(x_{i,j+1} - x_{i,j-1})/2 \quad (4.35)$$

so that

$$y_{i-1,j} = y_{i+1,j} + \delta_{i,j}(x_{i,j+1} - x_{i,j-1}) \quad i = 0; 0 < j < j_{\max} \quad (4.36)$$

$$y_{i+1,j} = y_{i-1,j} - \delta_{i,j}(x_{i,j+1} - x_{i,j-1}) \quad i = i_{\max}; 0 < j < j_{\max} \quad (4.37)$$

$$x_{i,j-1} = x_{i,j+1} + (y_{i+1,j} - y_{i-1,j})/\delta_{i,j} \quad j = 0; 0 < i < i_{\max} \quad (4.38)$$

$$x_{i,j+1} = x_{i,j-1} - (y_{i+1,j} - y_{i-1,j})/\delta_{i,j} \quad j = j_{\max}; 0 < i < i_{\max} \quad (4.39)$$

The boundary condition (4.29) can be treated similarly. Recall that x and y are known (specified) at the four corners of the domain. Complementing the Neumann boundary conditions (4.36)–(4.39) are the Dirichlet boundary conditions:

$$x = b_3(y) \quad i = 0; 0 < j < j_{\max} \quad (4.40)$$

$$x = b_1(y) \quad i = i_{\max}; 0 < j < j_{\max} \quad (4.41)$$

$$y = b_0(x) \quad j = 0; 0 < i < i_{\max} \quad (4.42)$$

$$y = b_2(x) \quad j = j_{\max}; \quad 0 < i < i_{\max} \quad (4.43)$$

The functional forms of b_0 , b_1 , b_2 and b_3 are completely arbitrary. The system of boundary conditions applied to the solution of eqs. (4.18) and (4.19) is illustrated in Fig. 4.3. The surface $y = b_0(x)$ could represent, for example, a rough (i.e. uneven) soil surface.

Scale Factor

The scale factor S may be obtained from either of the Neumann conditions, eq. (4.29) or (4.30), by integrating along a non-boundary line of constant i or j . Integrating eq. (4.29) with respect to i , from $i = 0$ to $i = i_{\max}$ along an arbitrary non-boundary j -line:

$$\int_0^{i_{\max}} g' \frac{\partial x}{\partial i} di = S \int_0^{i_{\max}} f' \frac{\partial y}{\partial j} di \quad (4.44)$$

we obtain

$$S = g'_j(x_{i_{\max},j} - x_{0,j}) / \int_0^{i_{\max}} f' \frac{\partial y}{\partial j} di \quad (4.45)$$

Numerical integration of the denominator in eq. (4.45) by the *trapezoidal rule*[9] yields

$$\int_0^{i_{\max}} f' \frac{\partial y}{\partial j} di = \frac{\Delta i}{2} \left[\left(f' \frac{\partial y}{\partial j} \right)_{0,j} + \left(f' \frac{\partial y}{\partial j} \right)_{i_{\max},j} + 2 \sum_{i=1}^{i_{\max}-1} \left(f' \frac{\partial y}{\partial j} \right)_{i,j} \right] \quad (4.46)$$

where $\Delta i = 1$, and by the central difference approximation,

$$\frac{\partial y}{\partial j} = (y_{i,j+1} - y_{i,j-1}) / 2 \quad (4.47)$$

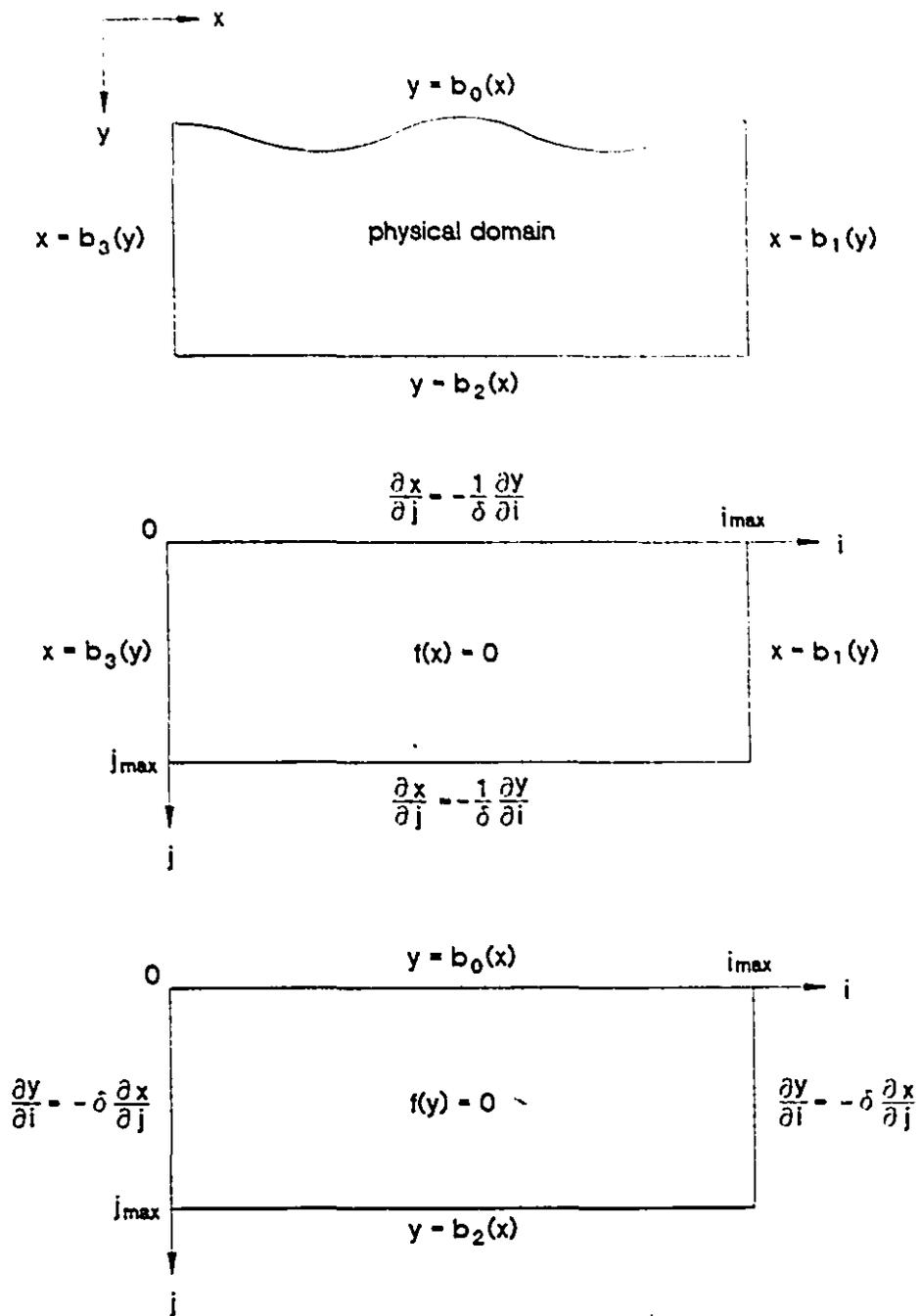


Figure 4.3: Boundary conditions for x and y

4.3.4 Solution by Dynamic Relaxation Method

Introduction

The discretized grid-generation equations $f_x = 0$ and $f_y = 0$ with f_x and f_y given by eqs. (4.33) and (4.34), respectively, each constitute a set of simultaneous equations. These simultaneous finite-difference equations may be solved iteratively by, for example, the tri-diagonal matrix algorithm (TDMA) (e.g Patankar[7], Lawal[6]) or directly by, for example, Gaussian elimination.

Herein, we consider the *dynamic relaxation method*—a technique commonly used for the static solution of structural mechanics problems. In this method, an equivalent damped dynamic problem, rather than the original static equation, is solved. The static solution corresponds to the steady-state solution of the dynamic problem. Dynamic relaxation is an explicit iterative method. Its main advantage is that it circumvents the need to solve a set of simultaneous equations. Thus, it is relatively simple and easy to program. In addition, the core storage requirement is much less than that required by 'static' finite difference algorithms.

Formulation

The dynamic relaxation method is based on the equality of the solution of the 'static' problem

$$f_x = 0$$

$$f_y = 0$$

and the steady-state solution of the 'dynamic' problem

$$m\ddot{x} + c\dot{x} = f_x \quad (4.48)$$

$$m\ddot{y} + c\dot{y} = f_y \quad (4.49)$$

The (fictitious) parameters c and m denote, respectively, the damping factor and mass density (discussed in a later section). The variables f_x and f_y may be conceptualized as forces acting in the x and y directions, respectively. Initial values of x and y , at time $t = 0$, are required. Note that initial values are required of iterative methods in general and so do not constitute an additional requirement for the dynamic relaxation method. The boundary conditions for the static problem are also applicable to the dynamic problem.

Initial Conditions

The initial coordinates x , y have a notable effect on the number of iterations required to reach the converged, static state. Obviously, the closer the initial conditions are to the solution, the fewer the number of iterations that will be required. In this thesis, we determine the initial coordinates by a packing-function-based algebraic interpolation, much like the algebraic discretization technique for a regular domain (section 4.2). Formally, the initial coordinates are given by:

for $0 < i < i_{\max}$:

$$x(i, 0) = x(0, 0) + \frac{x(i_{\max}, 0) - x(0, 0)}{i_{\max}} f(i) \quad (4.50)$$

$$y(i, 0) = b_0(x(i, 0)) \quad (4.51)$$

$$x(i, j_{\max}) = x(0, j_{\max}) + \frac{x(i_{\max}, j_{\max}) - x(0, j_{\max})}{i_{\max}} f(i) \quad (4.52)$$

$$y(i, j_{\max}) = b_2(x(i, j_{\max})) \quad (4.53)$$

for $0 < j < j_{\max}$:

$$y(0, j) = y(0, 0) + \frac{y(0, j_{\max}) - y(0, 0)}{j_{\max}} g(j) \quad (4.54)$$

$$x(0, j) = b_3(y(0, j)) \quad (4.55)$$

$$y(i_{\max}, j) = y(i_{\max}, 0) + \frac{y(i_{\max}, j_{\max}) - y(i_{\max}, 0)}{j_{\max}} g(j) \quad (4.56)$$

$$x(i_{\max}, j) = b_1(y(i_{\max}, j)) \quad (4.57)$$

for $0 < i < i_{\max}$ and $0 < j < j_{\max}$:

$$x(i, j) = x(0, j) + \frac{x(i_{\max}, j) - x(0, j)}{i_{\max}} f(i) \quad (4.58)$$

$$y(i, j) = y(i, 0) + \frac{y(i, j_{\max}) - y(i, 0)}{j_{\max}} g(j) \quad (4.59)$$

Zero initial velocities are prescribed:

$$\dot{x} = 0 \quad (4.60)$$

$$\dot{y} = 0 \quad (4.61)$$

Time Integration

We write the dynamic equations eq. (4.48) and (4.49), at an arbitrary time t^n , in the compact form

$$m\ddot{\mathbf{x}}^n + c\dot{\mathbf{x}}^n = \mathbf{f}^n \quad (4.62)$$

where

$$\mathbf{f} = \begin{Bmatrix} f_x \\ f_y \end{Bmatrix} \quad (4.63)$$

$$\mathbf{x} = \begin{Bmatrix} x \\ y \end{Bmatrix} \quad (4.64)$$

and the superscript n refers to time t^n . The central difference discretization of the temporal derivatives is expressed in the form

$$\ddot{\mathbf{x}}^n = \frac{\dot{\mathbf{x}}^{n+1/2} - \dot{\mathbf{x}}^{n-1/2}}{t^{n+1/2} - t^{n-1/2}} \quad (4.65)$$

$$\dot{\mathbf{x}}^{n+1/2} = \frac{\mathbf{x}^{n+1} - \mathbf{x}^n}{t^{n+1} - t^n} \quad (4.66)$$

where

$$t^{n+1/2} = (t^n + t^{n+1})/2 \quad (4.67)$$

$$t^{n-1/2} = (t^{n-1} + t^n)/2 \quad (4.68)$$

We consider a constant time increment Δt :

$$\Delta t = t^{n+1} - t^n \quad (4.69)$$

$$= t^{n+1/2} - t^{n-1/2} \quad (4.70)$$

by virtue of which eqs. (4.64) and (4.65) become

$$\ddot{x}^n = \frac{\dot{x}^{n+1/2} - \dot{x}^{n-1/2}}{\Delta t} \quad (4.71)$$

$$\dot{x}^{n+1/2} = \frac{x^{n+1} - x^n}{\Delta t} \quad (4.72)$$

The value of \dot{x}^n is obtained by linear interpolation:

$$\dot{x}^n = (\dot{x}^{n-1/2} + \dot{x}^{n+1/2})/2 \quad (4.73)$$

Substituting eq. (4.71) into eq. (4.62) and rearranging eq. (4.72), we obtain the recurrence formula

$$\dot{x}^{n+1/2} = \frac{m/\Delta t - c/2}{m/\Delta t + c/2} \dot{x}^{n-1/2} + \frac{f^n}{m/\Delta t + c/2} \quad (4.74)$$

$$x^{n+1} = x^n + \dot{x}^{n+1/2} \Delta t \quad (4.75)$$

The solution process simply entails the direct substitution of x at the previous time to obtain the current value. We note, however, that eq. (4.74) cannot be used to start the integration since the velocity \dot{x} is known at time t^0 , not at $t^{-1/2}$. The initial conditions are of the form

$$x^0 \neq 0 \quad (4.76)$$

$$\dot{x}^0 = 0 \quad (4.77)$$

Substituting eq. (4.77) into eq. (4.73), we obtain

$$\dot{x}^{-1/2} = -\dot{x}^{1/2} \quad (4.78)$$

which on substituting into eq. (4.74) at time t^0 yields

$$\dot{x}^{1/2} = \frac{f^0 \Delta t}{2m} \quad (4.79)$$

Observe that the damping coefficient c does not enter into the starting formula (4.79).

To summarize, the time-marching scheme is as follows:

for $n = 0$:

$$\dot{x}^{1/2} = \frac{f^0 \Delta t}{2m} \quad (4.80)$$

for $n \neq 0$:

$$\dot{x}^{n+1/2} = \frac{m/\Delta t - c/2}{m/\Delta t + c/2} \dot{x}^{n-1/2} + \frac{f^n}{m/\Delta t + c/2} \quad (4.81)$$

for $n \geq 0$:

$$x^{n+1} = x^n + \dot{x}^{n+1/2} \Delta t \quad (4.82)$$

We now seek the values of c and m which yield the fastest convergence towards the steady-state, hence static, solution and the value of Δt which ensures stability as well as accuracy of the results.

Time Step

Expressing eq. (4.62) in terms of x , we obtain, for $n \neq 0$,

$$\begin{aligned} x_{i,j}^{n+1} &= \frac{2[m - (1 + \delta_{i,j}^2)(\Delta t)^2]}{m + c\Delta t/2} x_{i,j}^n \\ &+ [\text{terms containing } x_{i,j}^{n-1}, x_{i+1,j}^n, x_{i-1,j}^n, x_{i,j-1}^n, x_{i,j+1}^n] \end{aligned} \quad (4.83)$$

From physical considerations, an increase in $x_{i,j}^n$ should result in an increase in $x_{i,j}^{n+1}$ and vice versa. Thus, the coefficient of $x_{i,j}^n$ must be positive:

$$\frac{2[m - (1 + \delta_{i,j}^2)(\Delta t)^2]}{m + c\Delta t/2} > 0 \quad (4.84)$$

That is,

$$\Delta t \leq \sqrt{m/(1 + \delta_{i,j}^2)} \quad (4.85)$$

For $n = 0$, we obtain

$$\begin{aligned} x_{i,j}^{n+1} &= \frac{m - (1 + \delta_{i,j}^2)(\Delta t)^2}{m} x_{i,j}^n \\ &+ [\text{terms containing } x_{i+1,j}^n, x_{i-1,j}^n, x_{i,j-1}^n, x_{i,j+1}^n] \end{aligned} \quad (4.86)$$

The coefficient of $x_{i,j}^n$ is positive if

$$\Delta t \leq \sqrt{m/(1 + \delta_{i,j}^2)} \quad (4.87)$$

Thus, the same time step limitation applies for all $n \geq 0$.

Damping Factor

The damping factor is the primary vehicle for reducing the motion of the dynamic problem to zero, yielding the static solution. Whilst many values of the damping factor will suffice, the value which attenuates the motion most rapidly, known as the

critical damping factor, is sought. A damping factor greater than the critical (an *over-damped* problem) yields a solution which converges to the static solution asymptotically. Thus, a small difference between successive values of \mathbf{x} does not necessarily indicate that the true solution has been reached. On the other hand, if a damping factor less than the critical is used (an *under-damped* problem), the oscillations decay with successively smaller amplitudes. This provides a useful tool for monitoring and checking the convergence of the solution to the static one.

The critical damping factor is a function of the fundamental (lowest) frequency of the undamped problem. It is shown [18, 16] that

$$c_{critical} \approx 2\omega_0 \quad (4.88)$$

where $c_{critical}$ is the critical damping factor and ω_0 is the fundamental frequency. Note that in [18], the damping factor is normalized with respect to the mass, hence the apparent difference between eq. (4.88) and the corresponding equation therein. The analytical determination of the fundamental frequency is not trivial. A numerical technique, which has been successfully used in structural mechanics [19] and seepage [16] problems, is considered. From the solution of the undamped ($c = 0$) problem, the fundamental frequency ω_0 may be obtained in one of two ways. In the first method, the oscillations of \mathbf{x} at a representative node are monitored. If the number of iterations

taken to complete one cycle is N , then

$$\omega_0 = 2\pi/N\Delta t \quad (4.89)$$

and the critical damping factor, eq. (4.88), is

$$c_{critical} = 4\pi m/N\Delta t \quad (4.90)$$

In the second method, the total kinetic energy is monitored. If the number of iterations taken to reach maximum kinetic energy is M , then

$$\omega_0 = 2\pi/4M\Delta t \quad (4.91)$$

and the critical damping factor, eq. (4.88), is

$$c_{critical} = \pi m/M\Delta t \quad (4.92)$$

From eq. (4.91), we see that the number of iterations taken to attain the maximum kinetic energy corresponds to one quarter (1/4) of a cycle of the fundamental mode.

The kinetic energy KE is given by

$$KE = \frac{1}{2} \sum m \dot{\mathbf{x}} \cdot \dot{\mathbf{x}} \quad (4.93)$$

An alternative dynamic relaxation method based on the *undamped* dynamic prob-

lem is presented in [17]. This approach makes use of the fact that a vibrating body attains maximum kinetic energy at a static equilibrium position. Implementation is relatively simple. The total kinetic energy is monitored and when it peaks, all the nodal velocities are set to zero. The process is repeated until convergence is attained. For a system vibrating in one mode, no iteration is, theoretically, necessary. In practice, however, a few iterations may be required. It is noted in [17] that in the conventional damping approach, damping forces may induce a path-change in path-dependent-processes and thus render the solution suspect. Herein lies a major advantage of the damping-independent method. There is also the advantage of by-passing the determination of a damping factor.

Mass Density

The mass density m is a fictitious parameter with no physical significance. A reduction in m to a value which does not violate the stability criterion, eq. (4.85), can result in an appreciable increase in the rate of convergence. Re-arranging eq. (4.85),

$$m \geq (\Delta t)^2(1 + \delta_{i,j}^2) \quad (4.94)$$

For a constant mass density m , the optimum time step, eq. (4.85), will vary with time because of the time-dependent parameter δ . Similarly, for a constant time step Δt , the optimum mass density, eq. (4.94), will be time-dependent. To maintain the relative simplicity of the constant time-step algorithm, we will keep the time-step fixed

and determine the corresponding optimum mass density from eq. (4.94). To provide a sufficient margin of safety against numerical instability, the mass density may be increased by a factor α , that is,

$$m = \alpha(\Delta t)^2(1 + \delta_{ij}^2) \quad (4.95)$$

where $\alpha \geq 1$.

Convergence

Convergence to the static solution is deemed to have occurred when the following quantities become vanishingly small (i.e. fall within a prescribed tolerance level):

1. difference between successive values of x
2. force f

In theory and practice, these conditions will be met almost simultaneously and so only one of the conditions needs to be monitored.

Results

Some results of computations conducted are presented in chapter 6.

References

- [1] THOMPSON J. F., WARSI, Z. U. A. AND MASTIN, C. W. *Numerical Grid Generation*. North-Holland, New York, 1985.
- [2] MOBLEY, C. D. AND STEWART, R. J. On the Numerical Generation of Boundary-Fitted Orthogonal Curvilinear Coordinate Systems. *J. Computational Physics*, 34:124-135, 1980.
- [3] KANG, I. S. AND LEAL, L. G. Orthogonal Grid Generation in a 2D Domain via the Boundary Integral Technique. *J. Computational Physics*, 102:78-87, 1992.
- [4] RANGWALLA, A. A. AND MUNSON, B. R. Numerical Solution for Viscous Flow for Two-Dimensional Domains Using Orthogonal Coordinate Systems. *J. Computational Physics*, 70:373-396, 1987.
- [5] NASSEF, R. *Laminar Flow and Heat Transfer Characteristics in Two-Dimensional Domains of Complex Geometry*. PhD Thesis. Chemical Engineering, McGill University, Montreal, Canada, 1992.
- [6] LAWAL, A. *Laminar Flow and Heat Transfer to Variable Property Power-Law Fluids in Arbitrary Cross-Section Ducts*. PhD Thesis, Chemical Engineering, McGill University, Montreal, Canada, 1985.
- [7] PATANKAR, S. V. *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publishing Co., 1980.
- [8] THOMAS, G. B. *Calculus and Analytic Geometry*. Addison-Wesley, Reading, Mass, 1968.
- [9] HORNBECK, R. W. *Numerical Methods*. Quantum Pub., New York, 1975.
- [10] DAY, A. S. An Introduction to Dynamic Relaxation. *The Engineer*, London, 219:218-221, 1965.
- [11] OTTER, J. H. R. Dynamic Relaxation. *Proc. Inst. Civ. Engrs.* 35:633-656, 1966.
- [12] OTTER, J. H. R., CASSEL, E. AND HOBBS, R. E. Dynamic Relaxation. *Proc. Inst. Civ. Engrs.* 35:723-750, 1966.
- [13] WOOD, W. L. Comparison of Dynamic Relaxation with Three Other Iterative Methods. *The Engineer*, London, 224:683-687, 1967.

- [14] PAPADRAKAKIS, M. A Method for the Automated Evaluation of Dynamic Relaxation Parameters. *Computer Methods in Applied Mechanics and Engineering*, 25:35-48, 1981.
- [15] CHOU, P. C. AND WU, L. A Dynamic Relaxation Finite-Element Method for Metal Forming Processes. *Int. J. Mech. Sci.*, 28(4):231-250, 1986.
- [16] PAREKH, C. J. Dynamic Relaxation of Seepage Problems. In DESAI, C. S. (ed.) *Numerical Methods in Geomechanics*, ASCE, New York, 1976, pp 1133-1144.
- [17] CUNDALL, P. Explicit Finite-Difference Methods in Geomechanics. In DESAI, C. S. (ed.) *Numerical Methods in Geomechanics*, ASCE, New York, 1976, pp 132-150.
- [18] UNDERWOOD, P. Dynamic Relaxation. In BELYTSCHKO, T. AND HUGHES, T. J. R. *Computational Methods for Transient Analysis*. Elsevier, Amsterdam, 1983, pp 244-265.
- [19] RUSHTON, K. R. Dynamic Relaxation Solution of Elastic Plate Problems. *J. Strain Analysis*, 3(1):23-32, 1968.

Chapter 5

Object-Oriented Program Development

5.1 Introduction

Object-oriented programming (OOP) is the latest development in software design philosophy, with seemingly limitless potential. By way of introduction, we consider two rather basic and fundamental questions:

- (a) what is object-oriented programming?
- (b) what are its benefits?

The first of these questions (what is OOP?) has no definitive answer, as evidenced by the following characterizations¹:

“My guess is that object-oriented programming will be in the 1980’s what structured programming was in the 1970’s. Everyone will be in favor of it. Every manufacturer will promote its product as supporting it. Every manager will pay lip service to it. Every programmer will practice it (differently). And no one will know just what it is.” — Reutsch [2].

¹compiled, in part, by Nelson [6]

“Object-oriented is well on its way to becoming the buzzword of the 1980’s. Suddenly everybody is using it, but with such a range of radically different meanings that no one seems to know exactly what the other is saying.” — Cox [1].

“Object-oriented has become a buzzword that implies ‘good’ programming.” — Stroustrup [3].

“I have a cat named Trash. In the current political climate, it would seem that if I were trying to sell him (at least to a computer scientist), I would not stress that he is gentle to humans and self-sufficient, living mostly on field mice. Rather, I would argue that he is object-oriented.” — King [4].

“A complete definition of what it means to be object-oriented is therefore not possible, though we can perhaps judge when one language is more object-oriented than another.” — Niestrasz [5].

This lack of a universal definition as to what makes a program object-oriented stems, in part, from the temptation to single out and herald that attribute of OOP that is most relevant to one’s need and application. Regardless of one’s definition, however, there is a general consensus that OOP represents a considerable improvement over the traditional programming approach and is well on its way to becoming the *de facto* standard.

The second question (what are the benefits of OOP?) is much less controversial. The superiority of OOP is based essentially on two powerful concepts: **encapsulation**

and **inheritance**. Encapsulation refers to the process of combining both code and data into a single entity to form a new data type called **class** while inheritance is the mechanism of deriving a new class from an existing one. Encapsulation and inheritance enable codes to be more readily reused, specialized or generalized to solve new problems. Reuse of existing code is a very efficient way to reduce the time and cost of software development. In addition, the tools of OOP allow for greater control over a program's structure and modularity.

The primary difference between the object-oriented philosophy and the traditional approach lies in the manner in which the functions and data variables are organized and implemented. In OOP, data variables and the functions that operate on them are grouped together as a unit and implemented as such while in the traditional approach, functions and variables are treated as separate entities. The OOP approach is based on the recognition that data variables and the codes that operate on them are mutually dependent and co-exist. Clearly, object-oriented program design requires a radical change in the thought process.

Object-oriented programming of the dynamic wheel-soil interaction problem is highly desirable in view of the extensions, e.g. to incorporate tire flexibility and tread-pattern, that will be required in the future. We also note that the validation of the code will greatly be facilitated by the organization of the program into self-contained mini-programs (classes) that can be independently tested.

In the rest of this chapter, the fundamental concepts of OOP are highlighted and applied to the development of an object-oriented control-volume-based finite element

program for dynamic wheel-soil interaction. The C++ programming language [7, 8, 9, 10, 11, 12, 13] is employed.

5.2 Fundamental Concepts

The fundamental concepts of object-oriented programming, namely encapsulation, friends, inheritance, virtual functions, polymorphism, operator overloading, constructors and destructors are described.

5.2.1 Encapsulation

Encapsulation, also known as **data abstraction**, is the combining of functions and data variables into a self-contained single entity to form a new data type. This new data type is called a **class** and a class variable is called an **object**. The relationship between a class and an object is analogous to that between a data type and a variable. The member functions of a class are known as **methods** and the data variables are called **instance variables**. The term **member** will be used as a generic term to denote a member function or a data variable.

Each member of a class has full access to any other member of the class. Access to a member by a non-member of the class can be unrestricted, restricted or denied, depending on the specified access privilege. Each member within a class is declared public, private or protected. A **private** member can only be accessed by members within the same class. A **protected** member can be accessed only by members within

the same class and members of classes derived from this class². A **public** member can be accessed by any member of any class. The general syntax for class declaration is:

```
class name
{
    private:
        // private members are declared here
    public:
        // public members are declared here
    protected:
        // protected members are declared here
};
```

The double slash `//` is a one-line comment symbol — everything to the right of it is ignored. Members are private by default. Thus, the access specifier `private` is optional. Usually, data members are made private while the member functions are made public, so that the data members can only be accessed by the member functions of the class. This **data-hiding** capability is a very attractive and useful feature as it guards against accidental or inadvertent corruption of data by external members.

5.2.2 Friends

Access to the private members of a class by a non-member of the class may be obtained through the **friend function** mechanism. A member that seeks access to the

²derived classes are discussed in section 5.2.3

private members of a class is declared a friend of the class with the preface `friend`, in the following manner:

```
class class1
{
    public:
        int    data1;
        void  func1();
};

class class2
{
    private:
        float data2;
        int   func2();
        friend void class1::func1();
};
```

It makes no difference whether a friend function is declared in the private or public section of the class. In the above declaration, the member function `func1` of class `class1` is given access to the private members of class `class2`, namely the data member `data2` and the member function `func2`. The friend function `func1` is merely a friend, not a member, of the `class2` family.

All the member functions of one class may be made friends of another class by declaring the entire class as a friend of the desired class as illustrated below:

```
class class2
{
    private:
        float  data2;
        int    func2();
        friend class1;
};
```

All the private members of `class2` are now accessible to any member of `class1`.

Friends functions can be conveniently used to provide privileged access to more than one class. Friend functions are particularly useful for providing quick fixes to the implementation details of a class. The concept and use of friend functions is, however, a rather controversial one since they break through the encapsulating wall that is built around the private members of a class. As a general rule, a recurrent need to use friend functions is a strong indication that the object-oriented structure of the program is weak and needs re-organization.

5.2.3 Inheritance

Inheritance is the process of creating a new class from an existing one by adding new members, overloading existing member functions, modifying access privileges, etc. The new class is known as a **derived class** while the original class is referred to as the

base class. Through inheritance, a well-tested code can be specialized or generalized to solve new problems. A derived class is declared in the form:

```
class derived: private base1, public base2
{
    // declaration of new members
};
```

In the derived class declaration, each base class is prefaced by the **access specifier** `public` or `private`, the latter being the default. The access specifier does not alter the access status, within the base class, of the base class members. The access specifier pertains to the base class members inherited by the derived class.

With a `public` access specifier, the `public` and `protected` members of the base class are `public` and `protected` members, respectively, of the derived class. The `private` members of the base class remain `private` to the base class and are not accessible to the derived class.

With a `private` access specifier, on the other hand, the `public` and `protected` members of the base class become `private` members of the derived class. The `private` members of the base class, however, remain `private` to the base class and are inaccessible to the derived class.

Note that the `private` members of the base class can only be inherited by the derived class through the mechanism of friend functions.

In some instances, it may be desirable to alter the access status of a base class member within the derived class. Suppose, for example, a base class member is trans-

mitted to a derived class as a private member. This may be converted to a public or protected member of the derived class as follows:

```
class base
{
    private:
        void  func0();

    protected:
        int   func1();

    public:
        void  func2();
        float func3();
};

class derived:private base
{
    protected:
        base::func1();

    public:
        base::func2();
};
```

In the above code, the protected and public members `func1()`, `func2()` and `func3()` of the base class `base` are transmitted to the derived class `derived` as private members

(`func0()` is not transmitted at all). Within the derived class, `func1()` is converted to a protected member while `func2()` is converted to a public member; `func3()` is not converted within the derived class and so remains a private member. In a similar manner, base class members transmitted to a derived class as public members can be converted to private or protected members. This mechanism is known as **access declaration**.

5.2.4 Virtual Functions

It is permissible to declare, in the derived class, a member function with an identical name, return value and argument list as a base class member. The mechanism of virtual functions is provided to invoke the desired member dynamically, i.e., at run-time rather than at compile-time. The syntax calls for the prefacing of the base class member with the keyword **virtual**. This declaration (**virtual**) is not required in the derived class member:

```
class base
{
    public:
        virtual void func();
};

class derived:public base
{
```

```
public:
    void func();
};
```

The desired function is selected at run-time using pointers, as illustrated below:

```
int main()
{
    base    b;
    derived d;
    base    *ptr;

    ptr = &b;
    ptr -> func();    // executes base::func();

    ptr = &d;
    ptr -> func();    // executes derived::func();
}
```

We note that the keyword `virtual` is necessary if the selection of the desired function is to be done at run-time. This process is known as **late** or **dynamic binding**. Without the `virtual` keyword, notwithstanding the use of pointers, the selection will be made at compile-time — a process known as **early** or **static binding**. Dynamic binding consumes less memory than static binding. Virtual functions are used to efficiently invoke different versions of a base class function.

5.2.5 Polymorphism

The term **polymorphism** is adapted from Greek and means 'having many shapes'. In OOP, polymorphism (also known as **function overloading**) is the use of the same name for two or more member functions of the same class. The functions must be distinguishable by the number and/or type of their arguments (calling parameters). Traditional programming languages, it will be recalled, require that a different name be used for every function. Clearly, it is preferable, from the standpoint of clarity and robustness, to use the same name for functions that perform similar actions on different types of variables. The following code segment illustrates the use of the same function name `abs` for the absolute value of an integer, a single-precision real number and a double-precision number:

```
class absolute
{
    public:
        int    abs (int);
        float  abs (float);
        double abs (double);
};
```

The appropriate function, dependent on the calling parameter, is automatically selected by the compiler. Polymorphism is often used to provide alternate constructors³ to a

³constructors are discussed in section 5.2.7

class.

5.2.6 Operator Overloading

Operator overloading is the use of the same operator (e.g. +, -, *) to conduct similar operations on different data types. For example, the use of the + operator for integer as well as real number addition constitutes operator overloading. Using operator overloading, matrix multiplication and addition, for example, can be implemented by the familiar notation:

```
A = B * C + D;    // A, B, C and D are matrix objects
```

The syntax for operator declaration is:

```
matrix operator*(matrix& A);  
matrix operator+(matrix& A);  
matrix operator=(matrix& A);
```

5.2.7 Constructors and Destructors

A constructor is a member function with the same name as the class it belongs to. Unlike normal member functions, however, a constructor is not explicitly called. Rather, it is automatically invoked when a class variable, i.e. object, is declared. Consequently, it is a convenient, and often used, method for initializing data members, as illustrated below:

```
class nodes
```

```
{  
    int M, N;  
    node (int M, int N);  
};  
  
nodes::nodes (int m, int n)  
{  
    M = m;  
    N = n;  
}
```

The constructor could, equivalently, be defined in-line, alongside its declaration, as follows:

```
nodes (int m, int n) {M=m; N=n};
```

The following object declaration

```
void main()  
{  
    nodes x(50, 30);  
}
```

assigns the values 50 and 30 to the private data members M and N, respectively, of object x of class nodes. Constructors, like other member functions, can be overloaded.

A destructor is also a member function with the same name as the class it belongs to but, unlike a constructor, it is prefaced by a tilde \sim . When an object goes out of scope, the constructor is automatically invoked. Useful and common applications of destructors include deallocation of dynamic memory and closing of files at program end. Whilst constructors and destructors have many of the characteristics of normal member functions, they do have some unique features, for example:

1. They do not have return values, not even void.
2. They cannot be inherited, though a derived class can invoke a base class' constructor and destructor.
3. Constructors cannot be virtual but destructors can.
4. Constructors and destructors are automatically invoked when objects are created and destroyed, respectively.

5.3 Program Development

5.3.1 Introduction

The journey towards an object-oriented world of computational mechanics has begun. In a recent paper entitled *Object oriented programming in scientific computations: The beginning of a new era*, Filho and Delvo [14] presented a (preliminary) sketch of the framework and basic elements of their proposed object-oriented finite element program. It consists essentially of base classes element, material and matrix which de-

fine, in somewhat general terms, the basic ingredients of a finite element analysis. The framework pertains, specifically, to the conventional finite element method, wherein matrix manipulation (assembly, solution) constitutes an integral component.

Herein, we develop an object-oriented control-volume-based finite element program for the solution of soil-structure interaction problems, in general, and dynamic wheel-soil interaction problems, in particular. A necessary and very important first step in the development of an object-oriented program is the design of a class structure for the problem. Desirable attributes of a class structure are simplicity and flexibility. Our proposed class structure consists of two sets of classes. The first set pertains to the run-time allocation of storage space for the nodal, elemental and control-volume-based variables. The second set of classes constitutes the functions and associated data for solving dynamic soil-structure interaction problems within the framework of the control-volume-based finite element method.

5.3.2 Dynamic Memory Allocation

The heap provides, relative to the stack, a huge reservoir from which large amounts of storage space can be drawn dynamically, i.e. at run-time. In a finite element analysis, where the data requirements are often considerable, dynamic memory allocation is extremely useful. Run-time memory allocation yields, relative to compile-time allocation, a smaller code size.

Dynamic memory for the nodal, elemental and control-volume-based variables is allocated by class **nodes**, class **elements** and class **cvolumes**, respectively. Nodal

variables are coordinates, displacements, velocities and accelerations. Elemental variables are stresses, strains, density and mass. Control-volume-based variables are mass, forces and hour-glass momentum.

The methods and structure of classes nodes, elements and cvolumes are very similar. Thus, we will discuss, in some detail, class nodes, listed below:

```
/* two-dimensional dynamic memory allocation for nodes */
```

```
#define type float

class nodes
{
protected:
    type** p;

    int    s1, s2, lb1, lb2, ub1, ub2, i, j;

public:
    nodes (int M, int N=1, int imin=0, int jmin=0);
    ~nodes ();

    type& node (int i, int j=0);

    void initialize (type value);
};
```

```
nodes::nodes(int M, int N, int imin, int jmin)
```

```
{

    s1 = M;

    s2 = N;

    lb1 = imin;

    lb2 = jmin;

    ub1 = lb1+s1-1;

    ub2 = lb2+s2-1;

    if (!(p = new type*[s2]))

    {

        cout << "Insufficient memory for nodes. Reduce M and/or N \n";

        exit(1);

    }

    for (j=0; j < s2; j++)

        if (!(p[j] = new type[s1]))

        {

            cout << "Insufficient memory for nodes. Reduce M and/or N \n";

            exit(1);

        }

    initialize (0.0);
```

```
}
```

```
nodes::~nodes()
```

```
{
```

```
    for (j = 0; j < s2; j++)
```

```
        delete p[j];
```

```
    delete p;
```

```
}
```

```
type& nodes::node(int i, int j)
```

```
{
```

```
    if (i < lb1 || i > ub1 || j < lb2 || j > ub2)
```

```
    {
```

```
        cout << "Illegal array index (" << i << ", " << j << ") " <<
```

```
            "sent to nodes::node(int, int) \n";
```

```
        exit(1);
```

```
    }
```

```
    return (p[j-lb2][i-lb1]);
```

```
}
```

```
void nodes::initialize (type value)
```

```
{  
    for (j = 0; j < s2; j++)  
        for (i = 0; i < s1; i++)  
            p[j][i] = value;  
}  
  
#undef type
```

Class `nodes` consists of constructor `nodes`, destructor `~nodes` and member functions `node` and `initialize`. The required array size and lower bound array indices are supplied to the constructor. Using the mechanism of **default values**, a default lower bound of zero is set. This feature provides considerable flexibility to the programmer. Memory is allocated within the constructor `nodes` and deallocated by the destructor `~nodes`. An arbitrary object `x` of class `nodes`, created by

```
nodes    x(M,N);
```

allocates dynamic memory for a two-dimensional array of size `M` by `N` for nodal variable `x`, with the array indices taking on the default lower bound `(0,0)`. (The upper bound is, of course, `(M-1,N-1)`).

The member function `node` references the values of the array elements. For example, the variable `x.node(i,j)` identifies the value of `x` at `node(i,j)`. A check is made to determine whether the indices `(i,j)` lie within the range supplied to the constructor.

Member function `initialize` is used to initialize the array elements. The following statement

```
x.initialize (2.0);
```

will assign the number 2.0 to all the array elements of **x**.

The pre-processors `#define` and `#undef` are used to allow for an easy incorporation of the desired variable type (e.g `float` for single precision, `double` for double precision, `int` for integer). If `#define type float` is replaced by `#define type double`, for example, we obtain a double-precision dynamic array.

Objects of classes `elements` and `cvolumes` are created and referenced in a similar manner:

```
elements  sigxx(M,N);
cvolumes  mass(M,N);
sigxx.elem(i,j) = 0.0;
mass.cvol(i,j) = 5.3;
```

5.3.3 Class Structure for Wheel-Soil Interaction

The framework of our proposed class structure consists of base classes `GRID`, `ELEMENT`, `INTERIOR`, `TRANSMITTING`, `BOUNDARY`, `CORNER`, `NEWMARK`, `WHEEL`, `PRINT` and `PLOT`. These classes and their functions are presented in tabular form in Table 5.1. Also shown are the memory allocation classes `nodes`, `elements` and `cvolumes`, discussed in the previous section.

The wheel-soil interaction classes are further discussed below.

1. Class `GRID` discretizes an arbitrary-shaped computational domain into quadrilateral finite elements, using the dynamic relaxation algorithm.

Table 5.1: Object Oriented Program Design: Wheel-Soil Interaction

CLASS	FUNCTION
nodes	Allocates dynamic memory for nodal variables
cvolumes	Allocates dynamic memory for control-volume-based variables
elements	Allocates dynamic memory for element-based variables
GRID	Generates finite element mesh
ELEMENT	Computes element-based variables and element contribution to associated control volumes
INTERIOR	Motion and hour-glass control of interior nodes
BOUNDARY	Boundary conditions and hour-glass control of boundary nodes
CORNER	Corner conditions and hour-glass control of corner nodes
TRANSMITTING	Transmitting boundary forces and contribution to associated control volumes
NEWMARK	Time integration scheme
WHEEL	Wheel-soil contact forces
PRINT	Prints solution to a file
PLOT	Plots solution on the screen

2. Class ELEMENT computes, the element-by-element, the element-based variables (i.e. stresses, strains, density, mass) as well as element contribution to the mass, internal traction and hour-glass momentum of the associated control volumes.
3. Class INTERIOR computes the motion and hour-glass velocity correction of the interior nodes.
4. Class TRANSMITTING computes the transmitting boundary forces exerted on the associated control volumes. It is used in association with class BOUNDARY, where applicable.
5. Class BOUNDARY consists of four classes, namely BOUNDARY1, BOUNDARY2, BOUNDARY3 and BOUNDARY4, corresponding to each of the four boundaries of the computational domain. Each of these BOUNDARY classes supplies the applicable boundary conditions and computes the forces on the associated control volumes or the motion of the boundary nodes (depending on the boundary conditions). Also computed are the hour-glass velocity corrections, if applicable, of the boundary nodes. This format, in conjunction with that of class CORNER, discussed below, allows for the easy incorporation of any desired boundary condition and makes the code readily adaptable to the solution of a wide variety of soil-structure interaction problems.
6. Class CORNER consists of four separate classes, namely CORNER1, CORNER2, CORNER3 and CORNER4, corresponding to each of the four corners of the computational domain. Each of these CORNER classes supplies the applicable

corner conditions and computes the forces on the associated control volume or the motion of the corner node (depending on the corner conditions). Corner forces computed within the BOUNDARY classes are simply passed on the CORNER class. Also computed are the hour-glass velocity correction, if applicable, of the corner node.

7. Class NEWMARK is the time integration scheme used to integrate the equations of motion.
8. Class WHEEL is provides the functions for determining the wheel-soil contact forces. It is used in association with the surface boundary.
9. Class PRINT prints specified results, e.g., soil drag, wheel sinkage, contact forces and their stick-slip status, translational velocity.
10. Class PLOT plots specified results on the screen. This is particularly useful for visualizing the deforming soil grid during wheel passage.

References

- [1] COX, B. J. *Object-Oriented Programming: An Evolutionary Approach*. Addison-Wesley Publishing Co., Reading, Mass, 1986. McGraw-Hill, New York, 1970.
- [2] REUTSCH, T. Object-Oriented Programming. *SIGPLAN Notices*, 17(9): 51-57, 1982.
- [3] REUTSCH, T. What is Object-Oriented Programming. *IEEE Software*, 5(3): 10-20, 1988.
- [4] KING, R. My Cat is Object-Oriented. In KIM, W. AND LOCHOVSKY, F. W. (eds) *Object-Oriented Concepts, Databases, and Applications*. ACM Press/Addison-Wesley, Reading, Mass, 1989.
- [5] NIESTRASZ, O. A Survey of Object-Oriented Concepts. In KIM, W. AND LOCHOVSKY, F. W. (eds) *Object-Oriented Concepts, Databases, and Applications*. ACM Press/Addison-Wesley, Reading, Mass, 1989.
- [6] NELSON, M. L. An Object-Oriented Tower of Babel. *OOPS Messenger*, 2(3): 3-11, 1991.
- [7] STROUSSTRUP, B. *The C++ Programming Language*. Addison-Wesley, Reading, Mass, 1986.
- [8] WIENER, R. S. AND PINSON, L. J. *An Introduction to Object Oriented Programming and C++*. Addison-Wesley, Reading, Mass, 1988.
- [9] HORSTMAN, C. S. *Mastering C++: An Introduction to C++ and Object-Oriented Programming for C and Pascal Programmers*. Wiley, New York, 1991.
- [10] EZZELL, B. *Turbo C++ Programming: An Object-Oriented Approach*. Addison-Wesley, Reading, Mass, 1990.
- [11] DAVIS, S. R. *Hands-on-Turbo C++*. Addison-Wesley, Reading, Mass, 1991.
- [12] HANCOCK, L., KRIEGER, M. AND ZAMIR, S. *The C Primer*. McGraw-Hill, New York, 1991.
- [13] CHIRLIAN, P. M. *Programming in C++*. Merrill Pub. Co.; Columbus, Ohio, 1990.
- [14] FILHO, J. AND DELVO, P. Object Oriented Programming in Scientific Computations: The Begining of a New Era. *Engineering Computations*. 8:81-87, 1991.

Chapter 6

Results

6.1 Introduction

The dynamic wheel-soil interaction and the grid-generation equations were coded in the C++ programming language, using object-oriented structure, in the manner detailed in chapter 5. Borland C++ compiler was used. Computations were conducted on 486-microcomputers in the Civil Engineering and Applied Mechanics Micro-Computing Facility.

The grid generation program was implemented as a stand-alone so it can be used independently of the wheel-soil interaction problem. The wheel-soil interaction code was tested, and successfully validated, in the context of a variety of transient problems, including dynamic punch indentation [1], dynamic soil compaction [2], transmitting boundary effectiveness [3] and hour-glass control [4].

In this chapter, results are presented, first for the grid generation and then for a wheel-soil interaction problem simulating aircraft landing on a soil-surface runway.

6.2 Dynamic Grid Generation

Four complex (i.e. irregularly-shaped) computational domains were used as test cases. These are:

1. inclined surface
2. sinusoidal surface
3. circular
4. triangular

The results presented herein were obtained using the undamped dynamic relaxation algorithm. The damped algorithm yielded essentially the same results. In all cases, a tolerance level of 10^{-4} , a time increment $\Delta t = 1$ and a mass density factor $\alpha = 1$ were used.

6.2.1 Inclined Surface

The inclined surface domains, with surface slopes $1/10$ and $3/10$, are shown in figs 6.1 and 6.2, respectively. The grid lines are packed near the surfaces with a packing intensity factor $B_j = 0.2$.

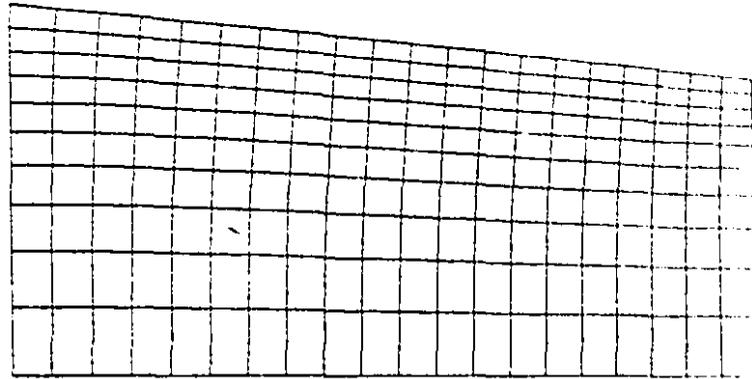


Figure 6.1: Inclined surface domain with packing of grid lines near the surface

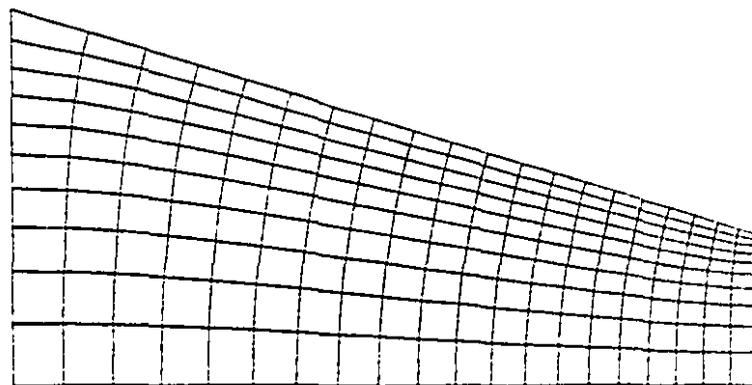


Figure 6.2: Inclined surface domain with packing of grid lines near the surface

6.2.2 Sinusoidal Surface

The sinusoidal surface domains are shown in Figs 6.3 and 6.4 with the sinusoidal surfaces defined by

$$y = R\left(1 - \cos \frac{2\pi nx}{L}\right) \quad (6.1)$$

where $R = 1$, $n = 3$ in fig 6.3 and $R = 10$, $n = 1$ in fig 6.4. The grid lines are packed near the surfaces with a packing intensity factor $B_j = 0.4$. A sinusoidal surface could be used to simulate a rough (i.e. uneven) soil terrain.

6.2.3 Circular Domain

The circular domain is shown in Figs 6.5 with no packing of the grid lines. The circular domain could be used, in future extensions, to model a deformable wheel (i.e. tire). A similar grid, generated for modelling fluid flow and heat transfer through a circular duct, can be found in Lawal[5].

6.2.4 Triangular Domain

The triangular domain is shown in Figs 6.6 with no packing of the grid lines. A similar grid was generated by Lawal[5] for modelling fluid flow and heat transfer through a triangular duct.

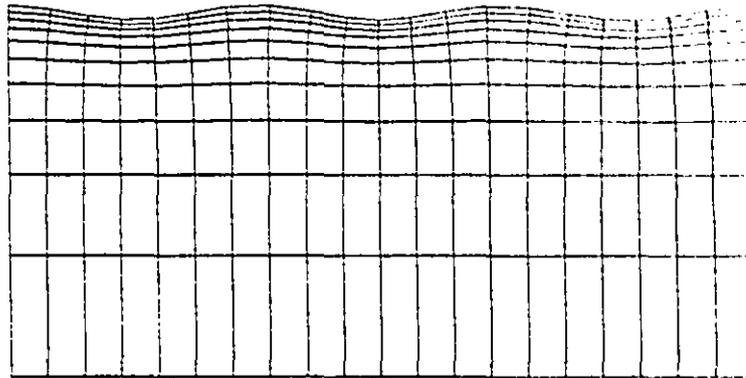


Figure 6.3: Sinusoidal surface domain with packing of grid lines near the surface

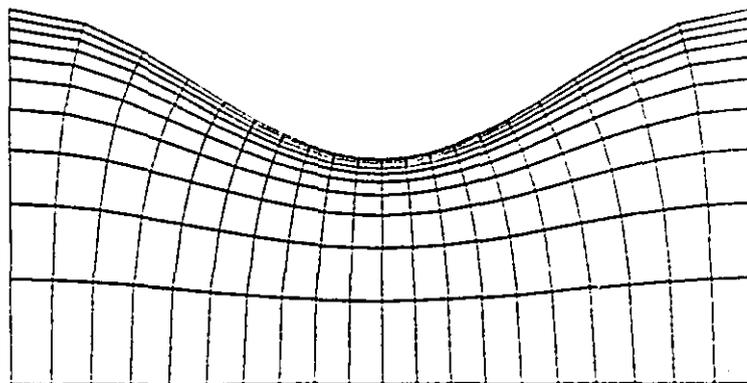


Figure 6.4: Sinusoidal surface domain with packing of grid lines near the surface

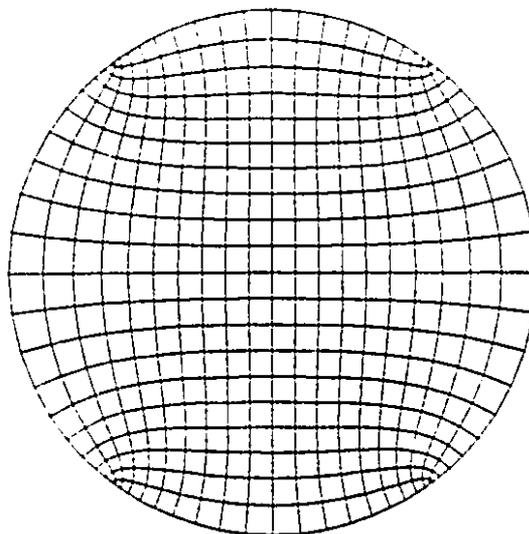


Figure 6.5: Circular domain with no packing of grid lines

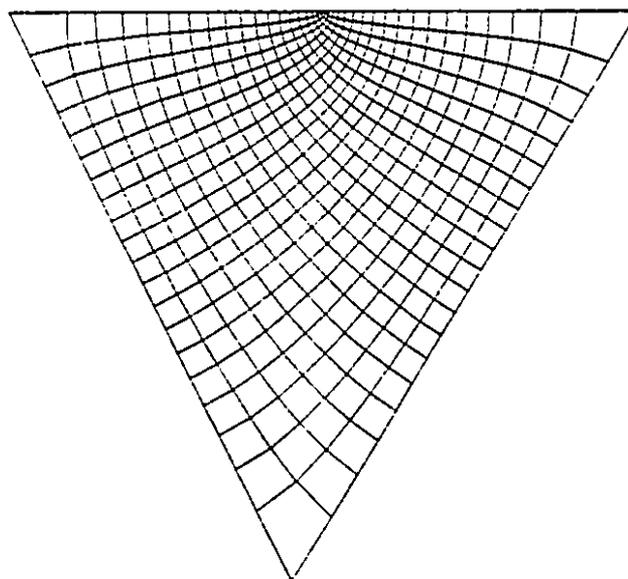


Figure 6.6: Triangular domain with no packing of grid lines

6.2.5 Discussion

Effect of time step

Numerical experimentation revealed that neither the numerical stability of the algorithm nor the accuracy of the results nor the rate of convergence is affected by the time increment Δt . This is entirely consistent with our formulation wherein the stability of the numerical scheme is embodied, not in the time step as is usually the case, but in the mass density. The reason for this, it will be recalled, was to enable the use of a time step that is independent of the evolving solution field and thus take advantage of the relative simplicity of the constant-step central-difference time integrator.

Effect of mass density

The mass density can be modified through the factor of safety α (the mass density varies linearly and proportionally with α). It was observed that the theoretical optimum α value of 1 was indeed the most computationally reliable.

Effect of tolerance level

The tolerance level had a significant effect on the rate of convergence, as depicted in Table 6.1 for the triangular domain. The sharp increase in the number of time steps taken to reach convergence as the tolerance level is decreased was characteristic of all the domains tested. As expected, the degree of orthogonality of the grid lines increases as the tolerance level is decreased. Results obtained with tolerance levels of 10^{-4} and 10^{-5} were practically the same.

Table 6.1: Effect of tolerance level on convergence rate

tolerance	time steps
10^{-1}	16
10^{-2}	132
10^{-3}	337
10^{-4}	706
10^{-5}	1311

6.3 Simulation of Aircraft Landing

6.3.1 Problem Parameters

The main gear tire of a Boeing 367-80 aircraft touches down on an unpaved soil surface, with a velocity of 42.4 m/s (95 miles per hour) at an angle of 13.9 degrees from the horizontal. The tire radius and width are 23 in (0.584 m) and 16 in (0.406 m) respectively. A vertical load of 9300 lb (41.366 kN) acts on the tire. A tire-soil frictional coefficient of 0.33 is assumed. These tire parameters were adapted from Pi[6]. The tire is considered to be highly inflated, i.e., rigid. The ground-roll operation is one of free-rolling wheel mobility, i.e., no external torque is applied to the wheel axle ($T = 0$). The soil may, and is expected to, exert an interfacial torque on the wheel causing it to rotate as it travels along the soil surface. A wheel mass of 10 lb (4.536 kg) is used.

Three different soils, adapted from Pi[6] and whose properties are listed in Ta-

ble 6.2, were considered. The soils are identified A, B and C in order of increasing strength or, equivalently, decreasing deformability. The Young's modulus, Poisson's ratio and shear modulus are considered to be three independent parameters because of the relatively low value of the shear modulus determined for the soils. The cone index CI, averaged over a 6 in (152 mm) penetration depth is also indicated. The cone index is defined, by the Waterways Experimentation Station (WES) of the U.S. Corps of Engineers, as twice the force needed to push a 30-degree cone having a base area of 0.5 in^2 (322.6 mm^2) into the soil. The cone index is, of course, a function of the depth of penetration.

The computational soil domain, shown in Fig. 6.7, is 12 ft (3.658 m) long and 4 ft (1.219 m) wide. The discretized domain consists of 450 nodes and 396 elements. There are 45 equal-spaced vertical grid lines ($i = \text{constant}$), numbered 0 through 44, along the direction of travel of the wheel and 10 horizontal grid lines ($j = \text{constant}$), numbered 0 through 9 starting from the soil surface. The horizontal grid lines are concentrated towards the soil surface with a packing intensity factor $B_j = 0.3$. The wheel touches down on soil surface node 11, that is, node (11,0).

6.3.2 Solution

Results are presented for the time evolution of (1) soil drag, (2) wheel sinkage, (3) contact nodes, (4) contact angle (length), (5) translational velocity, (6) translational distance and (7) rotational velocity of the wheel. Also presented is a snap shot, at an arbitrary time 0.02 sec., of (8) the nodal contact forces and their stick-slip status and

Table 6.2: Soil Parameters

PARAMETER	UNIT	A	B	C
CI	psi	80	120	160
	MPa	0.552	0.827	1.103
E_2	psi	1600	2500	4800
	MPa	11.03	17.23	33.10
ν		0.185	0.186	0.187
G	psi	300	300	300
	MPa	2.069	2.069	2.069
E_2/E_1		1	1	1
E_2/η_1	1/s	0.7	0.7	0.8
ρ	lb-s ² /in ⁴	0.000114	0.000127	0.000138
	Mg/m ³	1.218	1.357	1.475

(9) the velocity field within the soil domain. Finally, the effect of wheel radius on soil drag and wheel sinkage is considered. The computations were carried out until the maximum soil drag and wheel sinkage had, at least, been reached.

Figures 6.8, 6.9 and 6.10 present the time variation of soil drag ratio, defined as soil drag/wheel load, for aircraft landing on soils A, B and C respectively. In these figures, soil drag represents the resultant horizontal force exerted on the wheel by the soil and is considered positive in the direction opposite that of the wheel travel. The fluctuations in the soil drag predictions are due, in part, to the stick-slip phenomenon and, in part, to the discrete nature of the mathematical model wherein new node contact and node detachment are accompanied by finite nodal force addition or removal. Fig. 6.11 shows a third-order polynomial fit to the soil drag data. Given the increase of soil drag with soil strength, one is tempted to extrapolate and infer that in the extreme case of a paved rigid surface, a higher drag would be encountered. With no sinkage

or surface motion, a different frictional coefficient, point-contact etc., such a deduction would clearly be unjustified. What is particularly significant in these results is the high values of soil drag encountered.

Figure 6.12 shows the time variation of sinkage ratio (wheel sinkage/wheel radius) for all three soils. These are the actual numerical results, i.e., no data-smoothing was performed. The maximum wheel sinkage and the time at which it is obtained increase as soil strength decreases. The sinkage values are also high but not intolerable. Clearly, such high values can be expected in view of the high impact speed and wheel load.

Figure 6.13 is a plot of the translational velocity of the wheel. The wheel undergoes the most reduction in translational velocity on soil C until about 0.037 second when the velocities on soils C and B coincide. Between 0.037 and 0.055 second, the velocity is least on soil B. However, the deceleration at time 0.055 second is greatest on soil A, so that, subsequently, the wheel velocity becomes least on soil A.

Figure 6.14 shows that the wheel travel distance is greatest on soil A over the time range of the numerical computations. The differences are, however, not substantial.

Figure 6.15 presents the rotational velocity of the wheel. The wheel rotates fastest on soil C until about 0.02 second after which the angular velocity is greatest on soil B. The angular velocity on A is increasing faster than, and ultimately surpasses, that on B.

Figures 6.16, 6.17 and 6.18 depict, for soils A, B and C respectively, the soil surface contact nodes as a function of time. The pattern of nodal contact formation and detachment is clearly visible. At time 0.01 second, for example, nodes 13 through

19 of soil A are in contact with the wheel. By time 0.03 second, these nodes have all detached from the wheel and contact has been established with nodes 21 through 29.

The contact region, expressed in terms of the angular coordinates of the bounding contact nodes, is portrayed in Fig. 6.19, 6.20 and 6.21 for soils A, B and C respectively. Wheel-soil angular coordinate θ is measured clockwise with points directly below the center of the circle ascribed an angular coordinate of 90 degrees. Figures 6.16, 6.17 and 6.18 are complementary to Figs. 6.19, 6.20 and 6.21, respectively. For example, Figs. 6.16 and 6.19 show that the angular coordinate that corresponds to contact node 13 of soil A at time 0.01 second is 113 degrees. We note that the wheel-soil contact region decreases with soil strength.

The normal and tangential contact forces exerted by the wheel, at time 0.02 sec., on the soil-surface contact nodes are given in Figs. 6.22, 6.23 and 6.24 for soils A, B and C respectively. The contact force ratio is defined as the ratio of the contact force to the wheel load. Also indicated is the stick-slip status of each contact node. A negative value of tangential force indicates a force acting in the counter-clockwise direction with respect to the wheel center. Thus, the contact node exerts a clockwise tangential force on the wheel. As would be expected, the nodal contact force distribution peaks to the right of the vertical center-line of the wheel. We also note that in contrast with the tribology model of Yong and Foda [7] wherein it was implicitly assumed that the stick and slip regions are continuous, the present analysis reveals that the stick and slip zones can indeed be discontinuous with stick nodes occurring between slip nodes and vice versa.

Figures 6.25, 6.26 and 6.27 show the deformed soil grid, at time 0.02 sec., of soils A, B and C respectively. On close examination of these figures, one finds that the bottom transmitting boundary is displaced the most in the stiffer soil C and the least in the softest soil A. The reason for this lies in the stress wave propagation speeds within the soils. The dilatational stress wave speed is greatest in soil C and least in soil A. Accordingly, the stress waves arrive at the boundary of soil C most frequently and hence cause the greatest distortion.

Figures 6.28 through 6.36 portray, in contour mapping and three-dimensional formats, the horizontal and vertical components of the velocity field within soils A, B and C at time 0.02 second. The three-dimensional plots illustrate quite vividly the effect of the different wave propagation speeds. The soil-surface disturbances, for example, can be seen to be most extensive on soil C and least on A.

Figures 6.37 and 6.38 show the effect of wheel radius on soil drag and wheel sinkage, respectively. Clearly, soil drag as well as wheel sinkage decrease as the wheel radius increases. This would suggest, therefore, that a low-pressure tire would perform much better than a rigid wheel, in so far as minimizing soil drag and wheel sinkage are concerned.

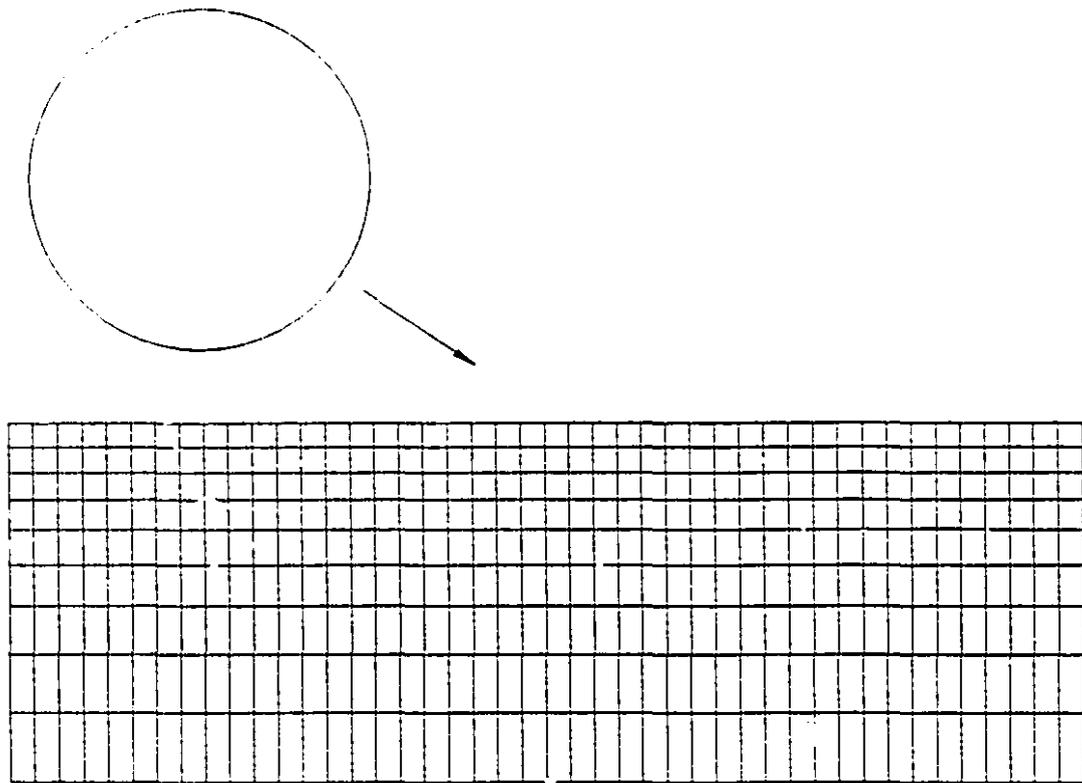


Figure 6.7: Wheel approaching landing: initial soil grid

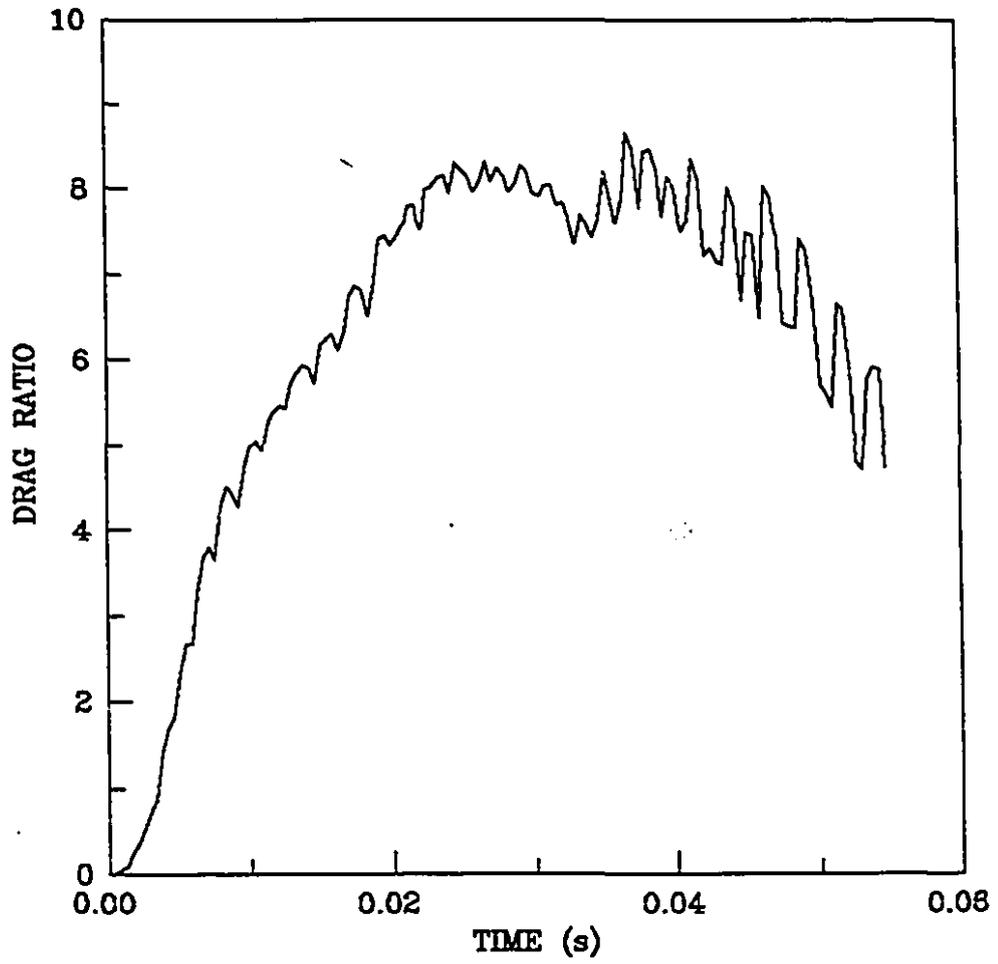


Figure 6.8: Soil drag vs time: soil A

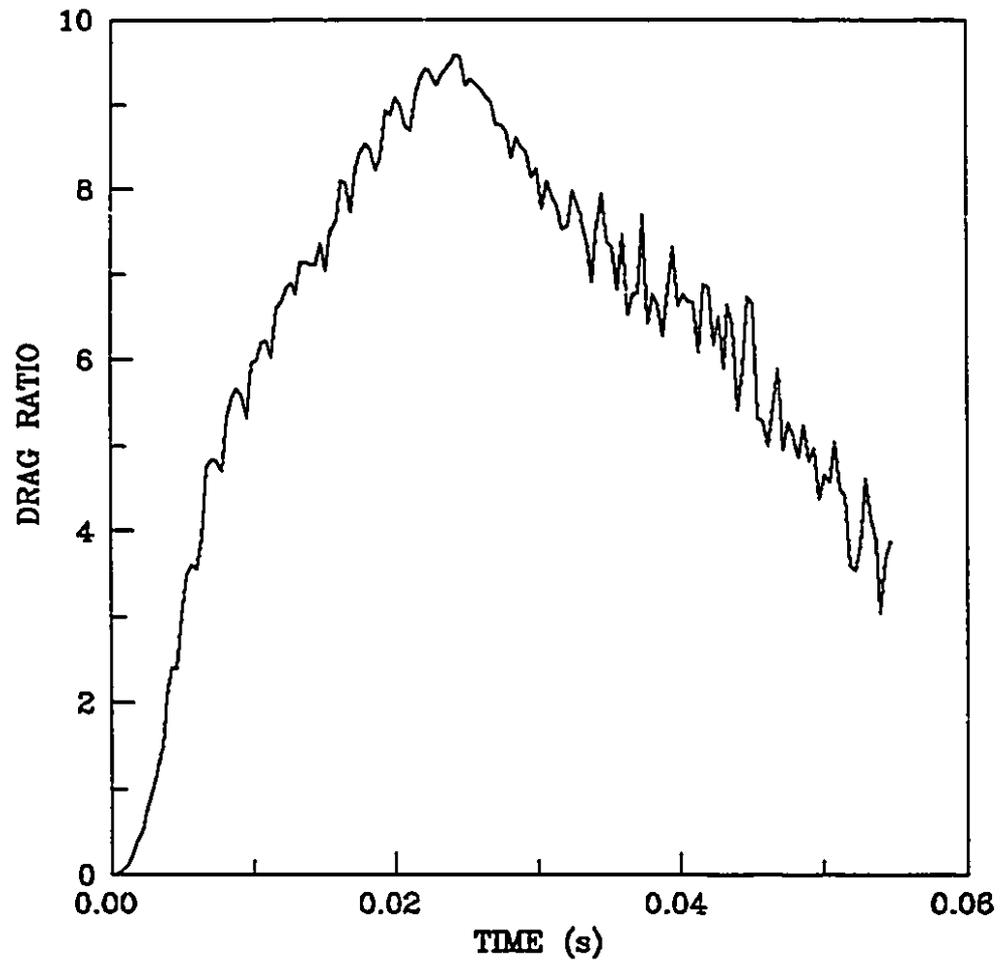


Figure 6.9: Soil drag vs time: soil B

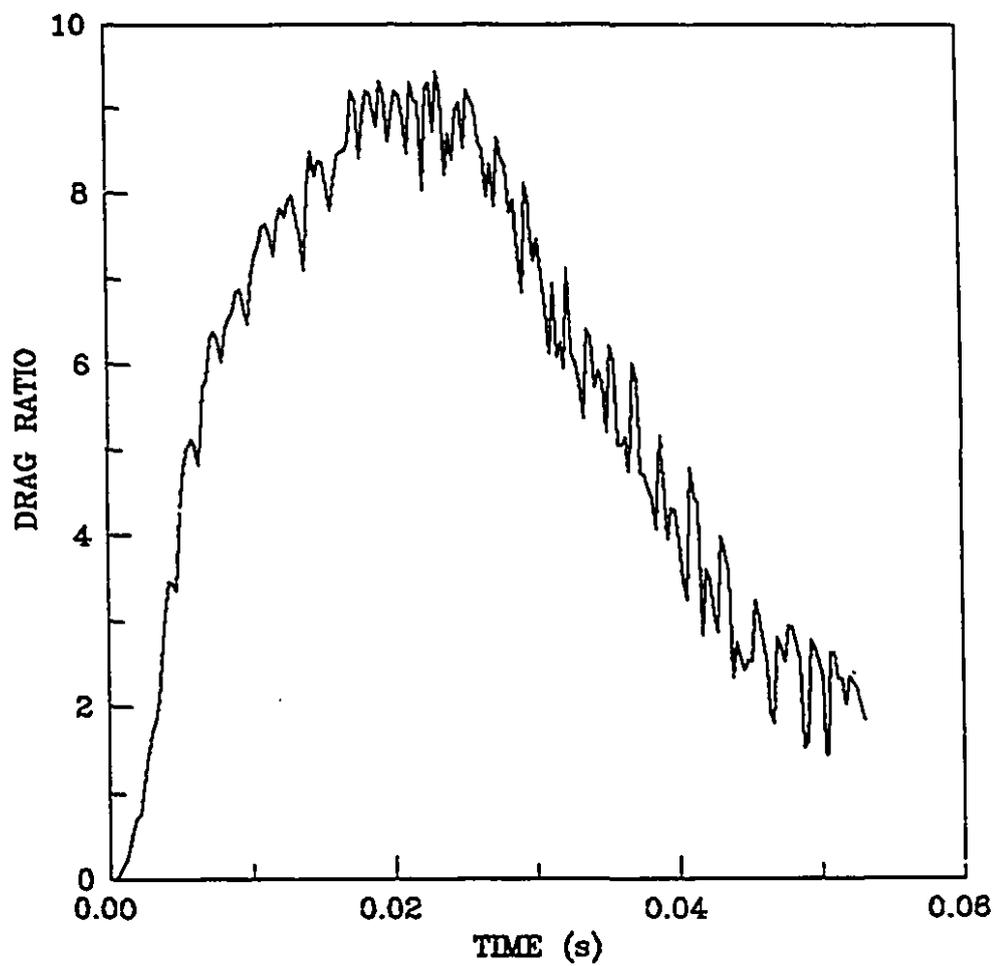


Figure 6.10: Soil drag vs time: soil C

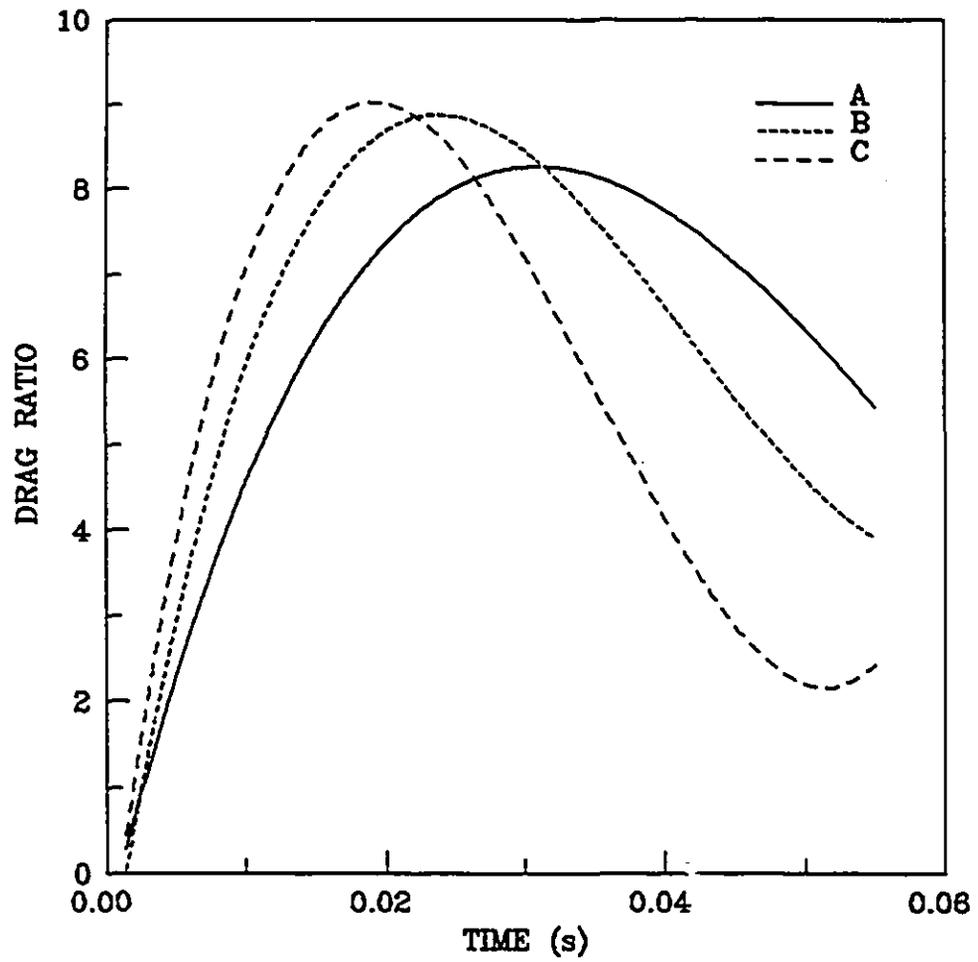


Figure 6.11: Polynomial-fitted soil drag vs time

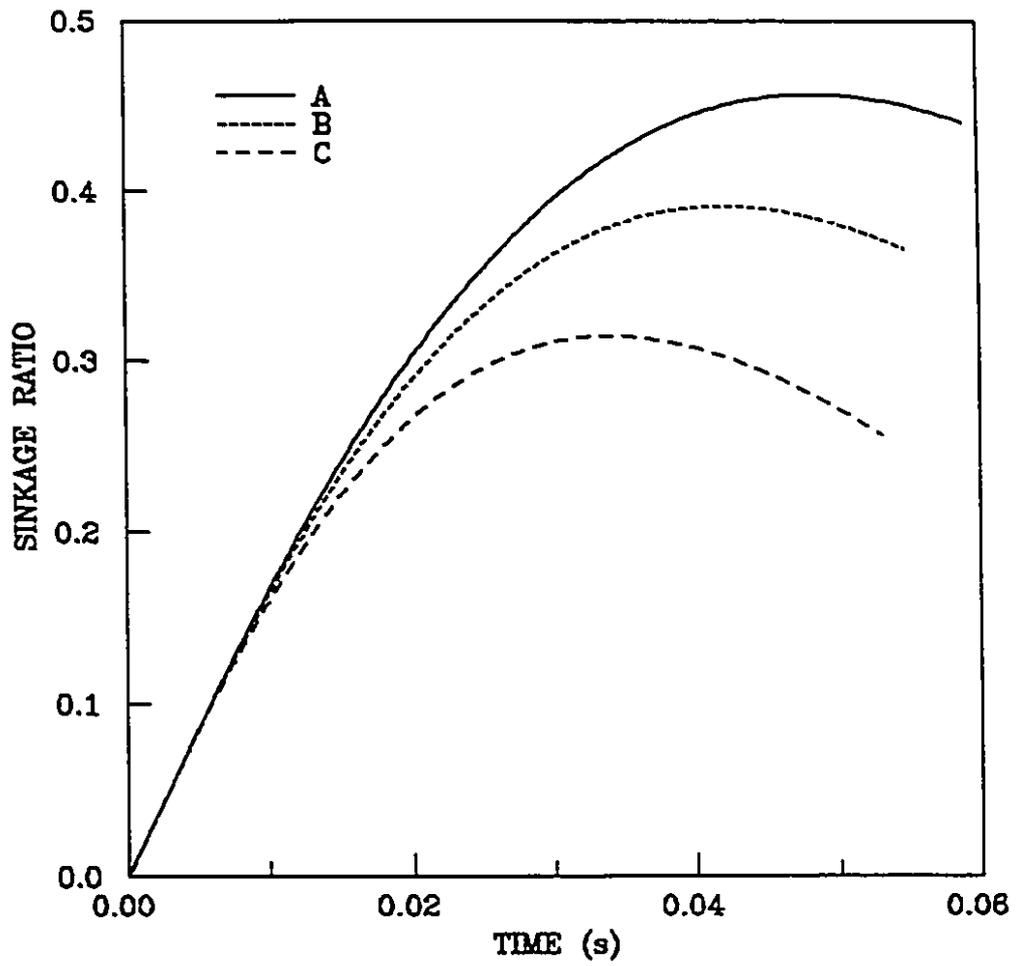


Figure 6.12: Wheel sinkage vs time

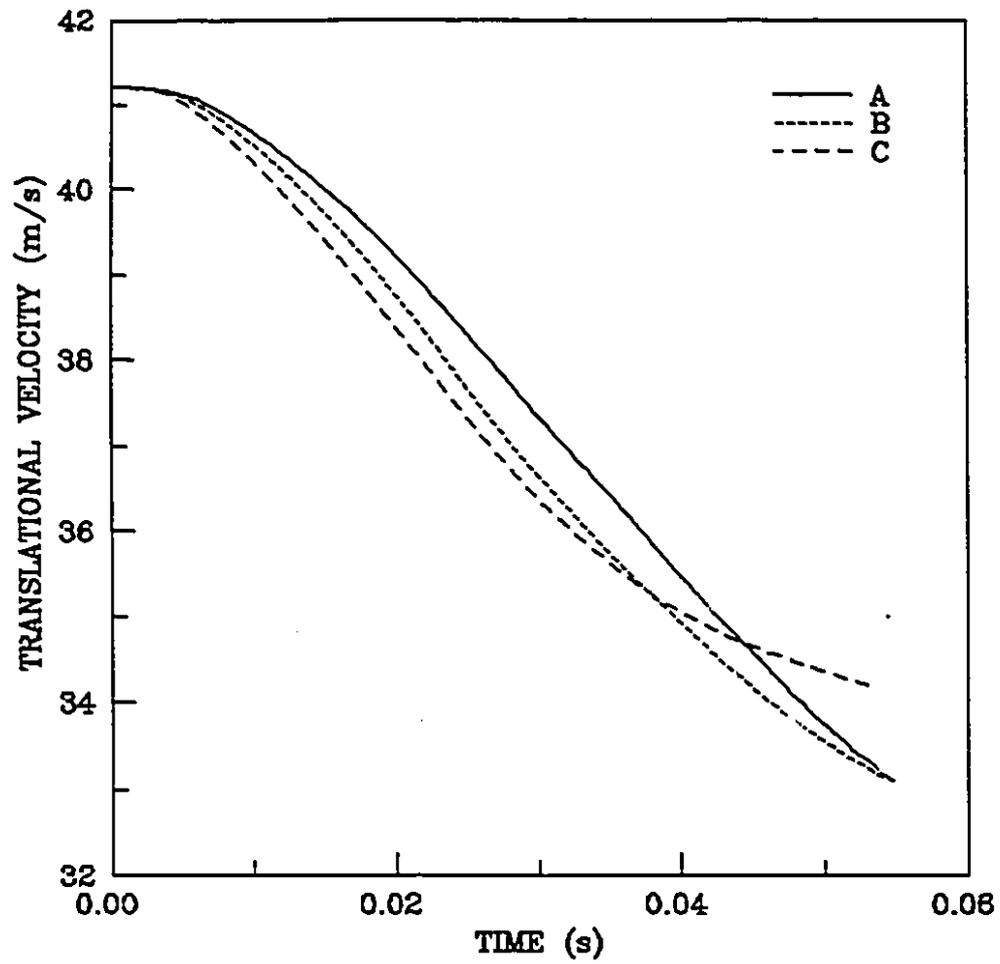


Figure 6.19: Translational velocity of wheel vs time

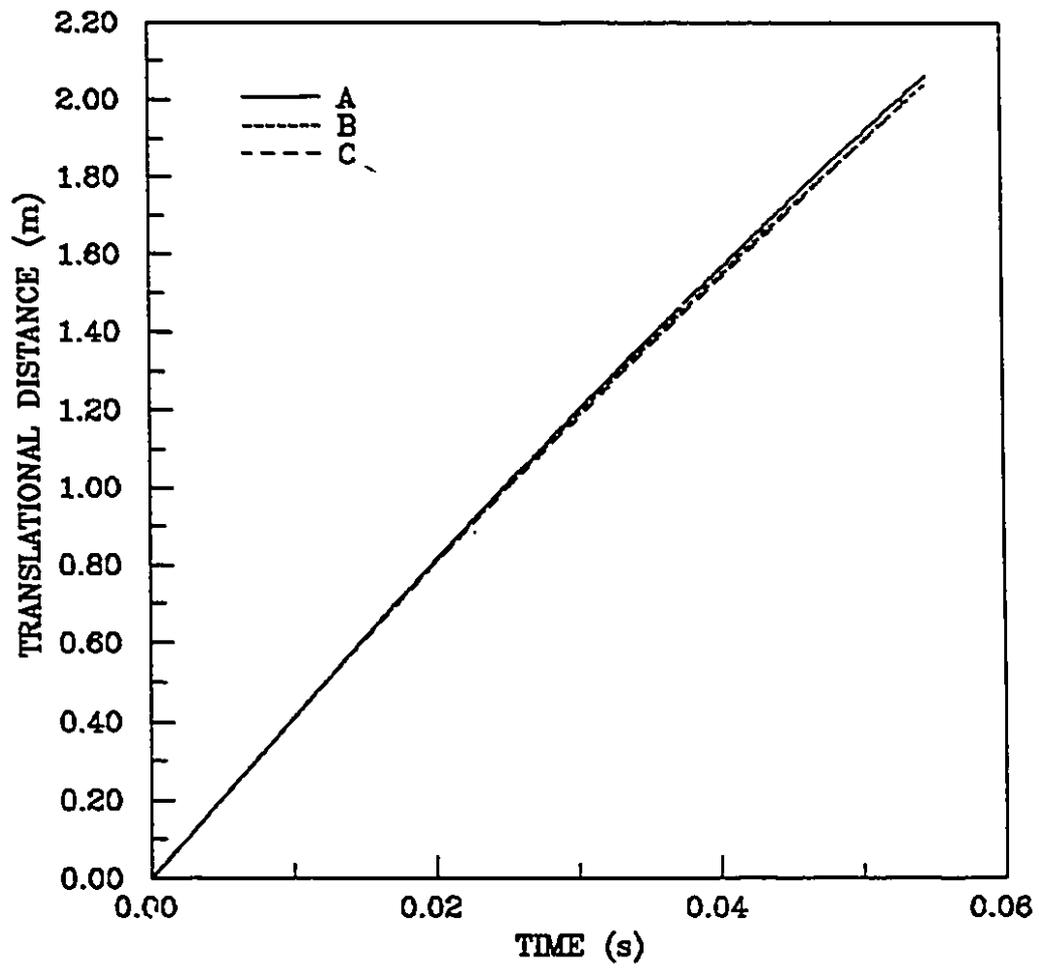


Figure 6.14: Translational distance of wheel vs time

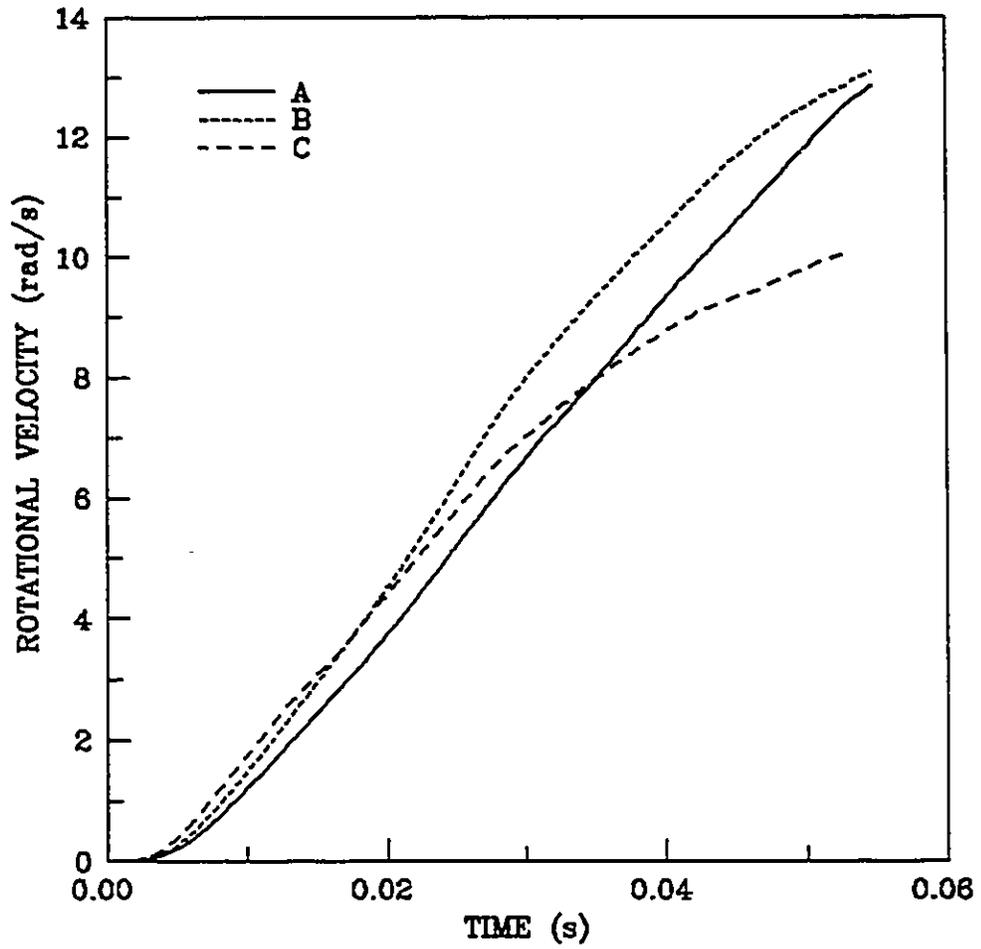


Figure 6.15: Rotational velocity of wheel vs time

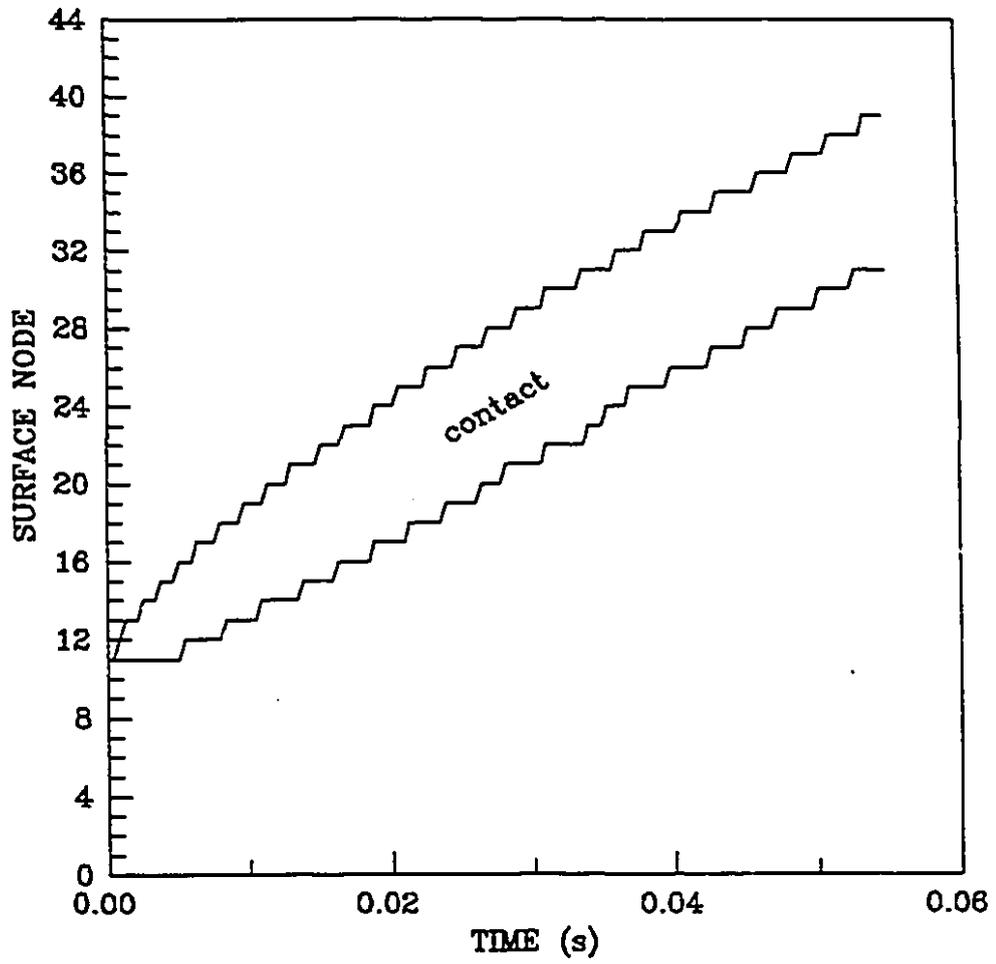


Figure 6.16: Wheel-soil contact nodes vs time: soil A

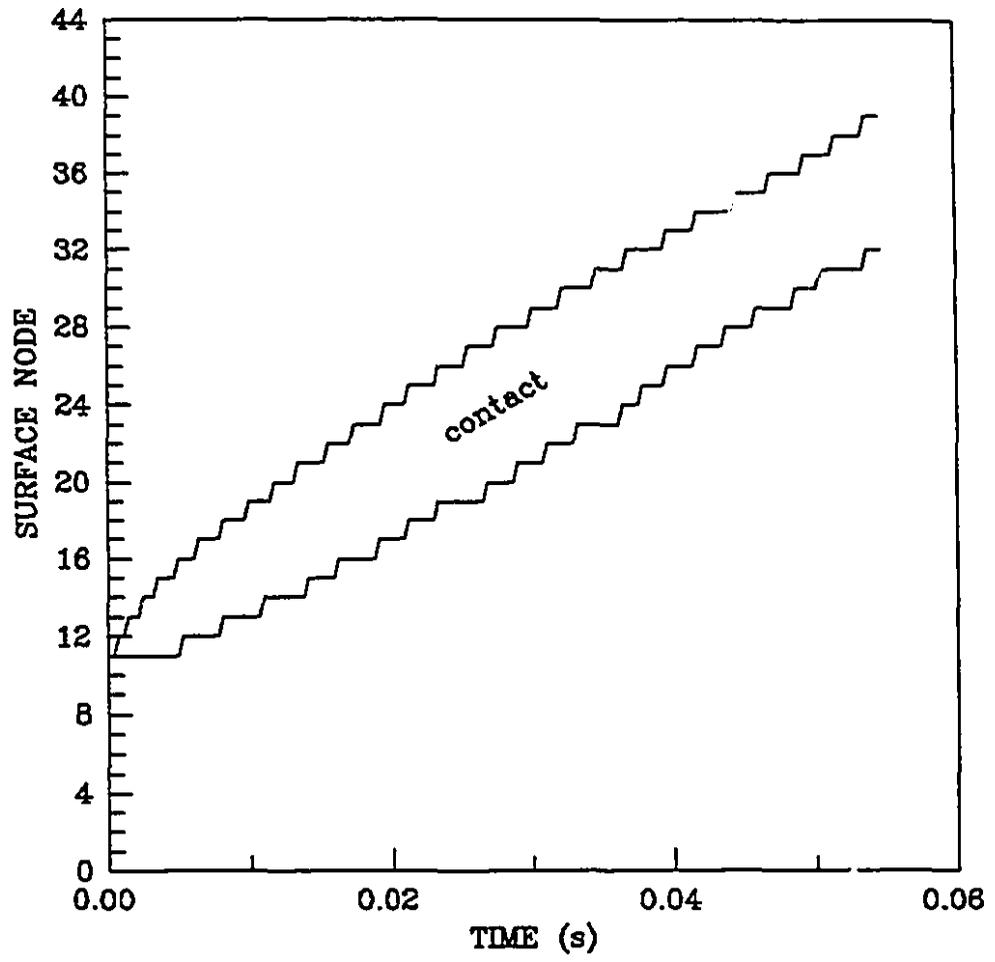


Figure 6.17: Wheel-soil contact nodes vs time: soil B

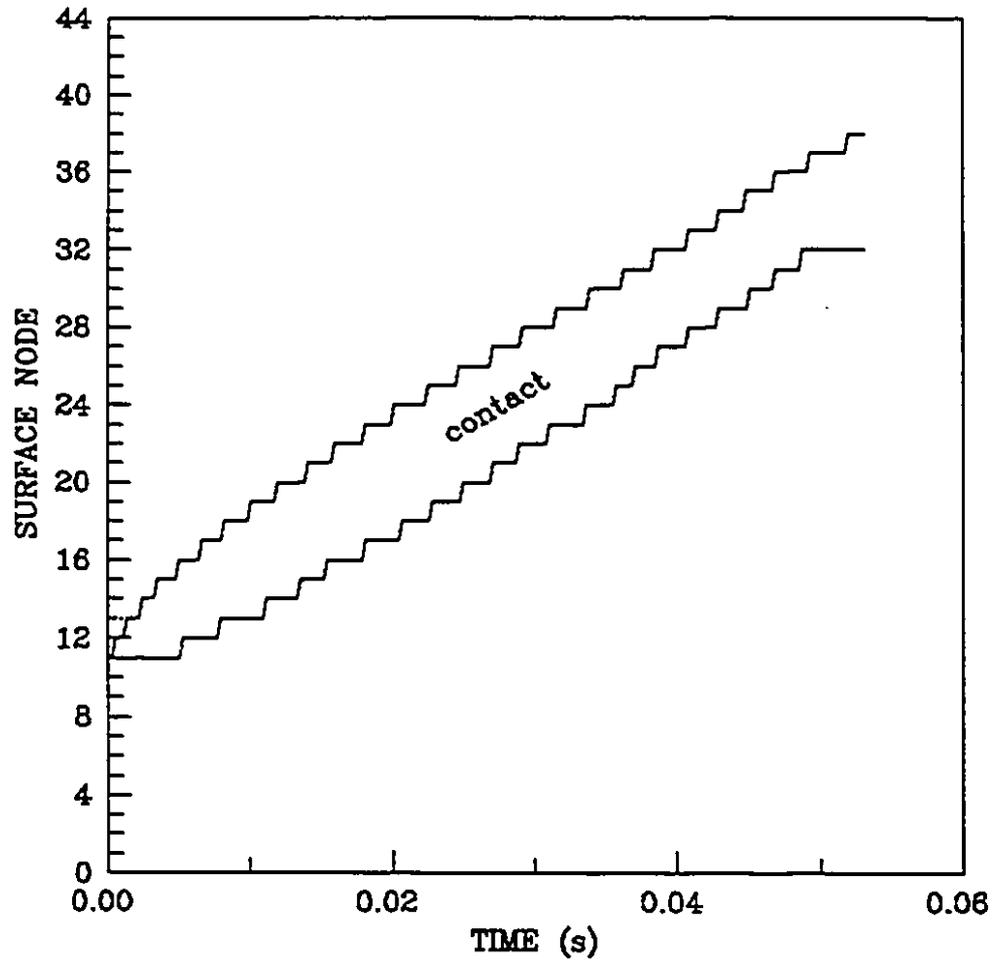


Figure 6.18: Wheel-soil contact nodes vs time: soil C

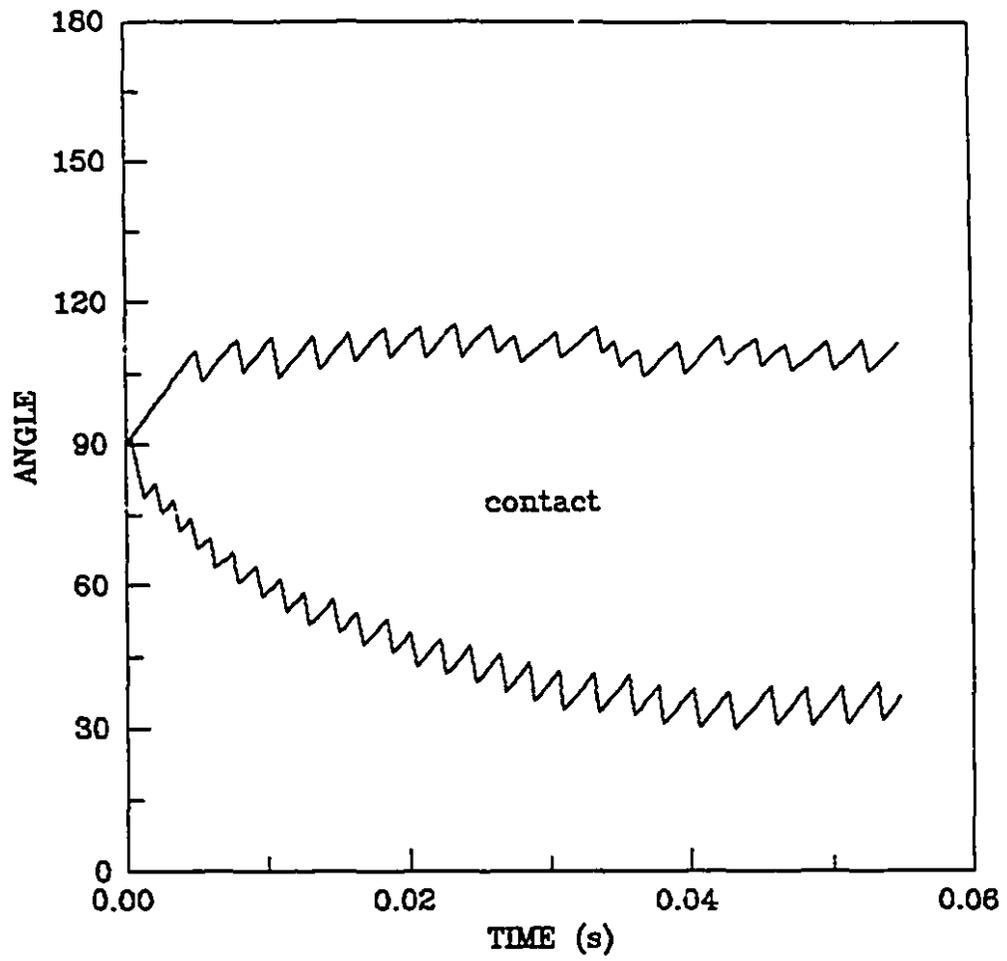


Figure 6.19: Wheel-soil contact angle vs time: soil A

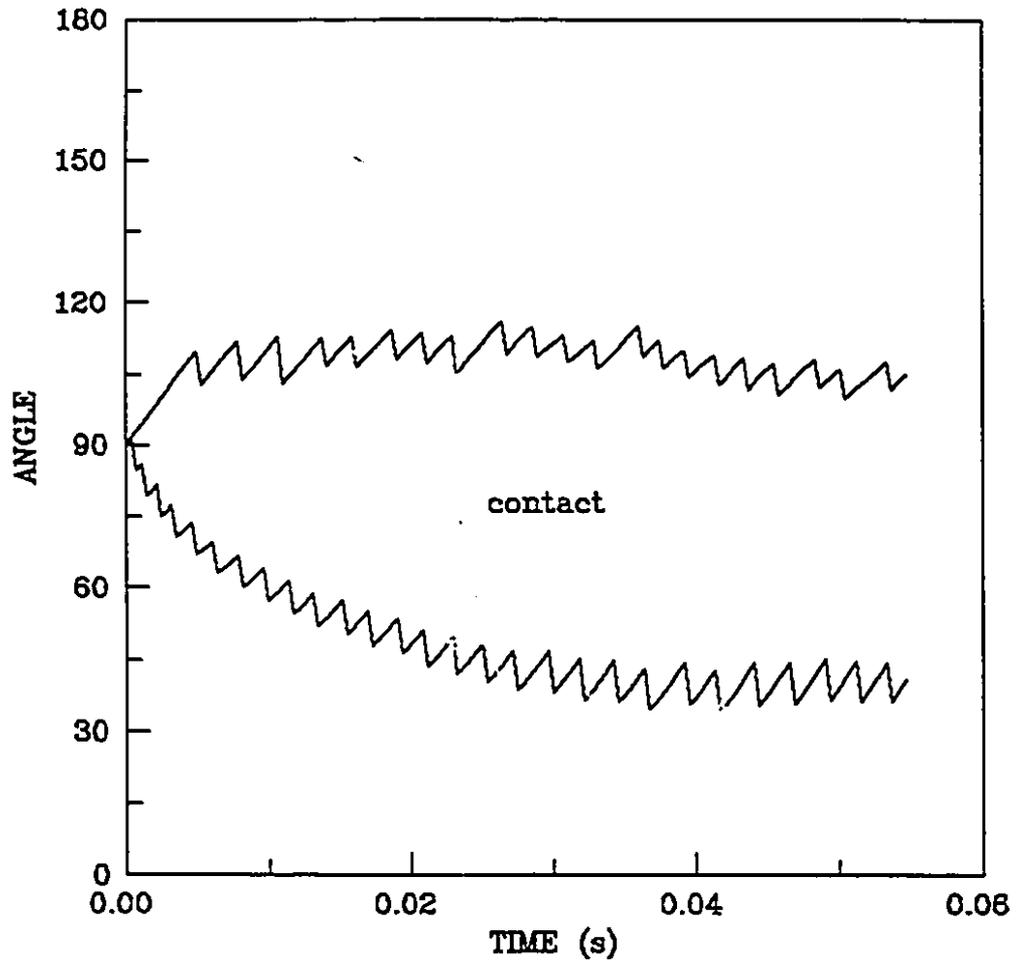


Figure 6.20: Wheel-soil contact angle vs time: soil B

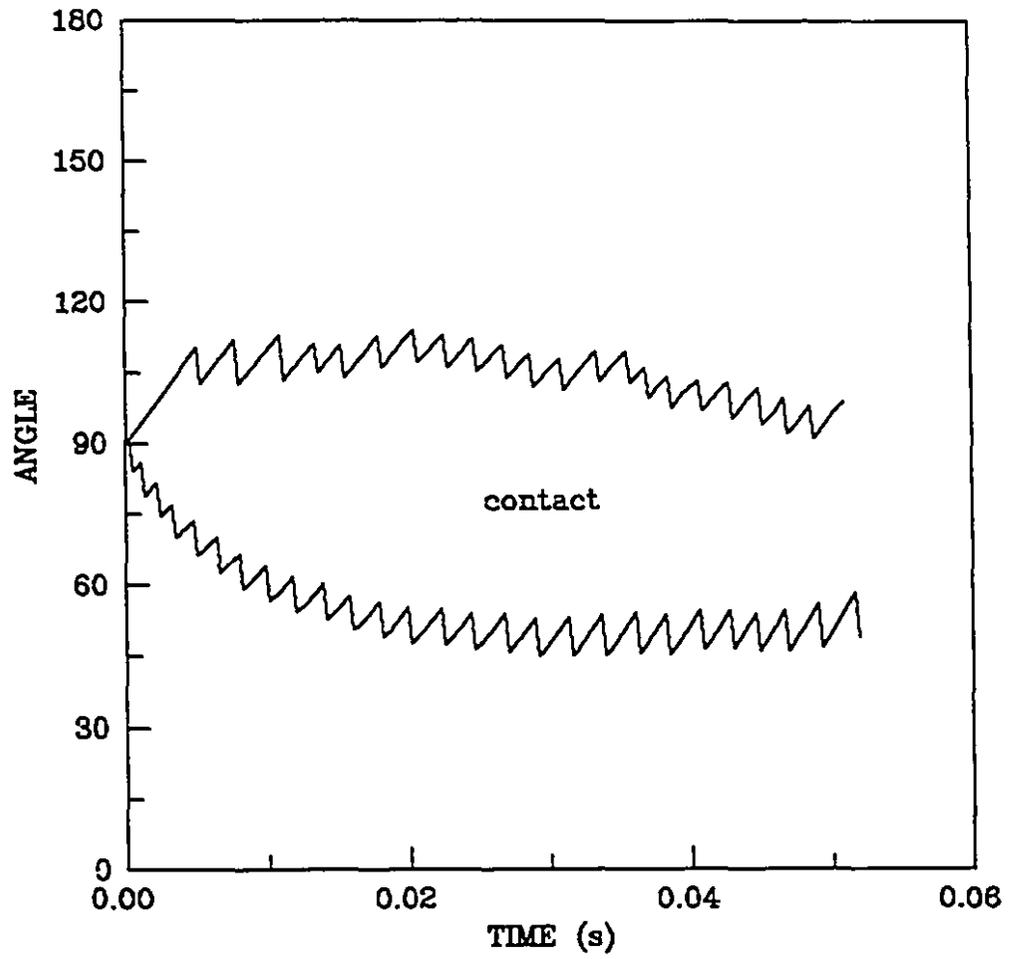


Figure 6.21: Wheel-soil contact angle vs time: soil C

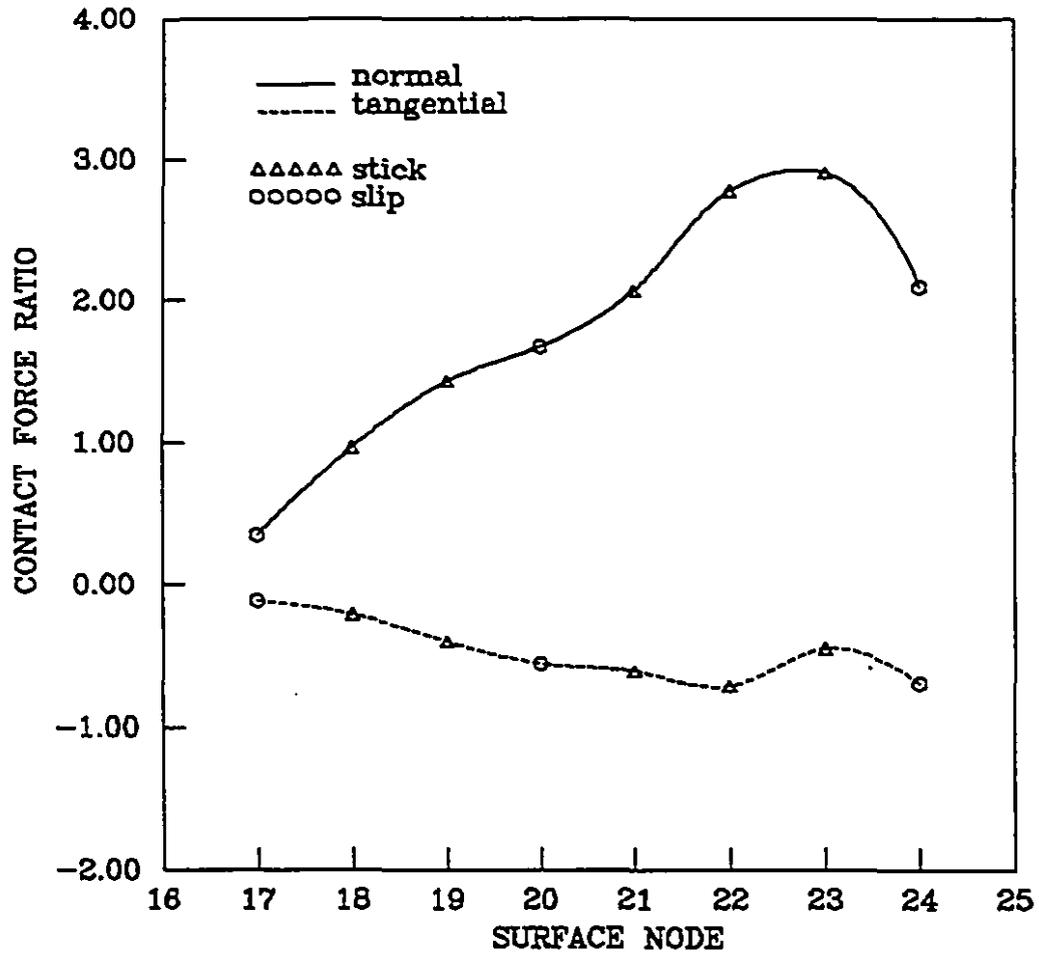


Figure 6.22: Wheel-soil contact force at time 0.020 sec: soil A

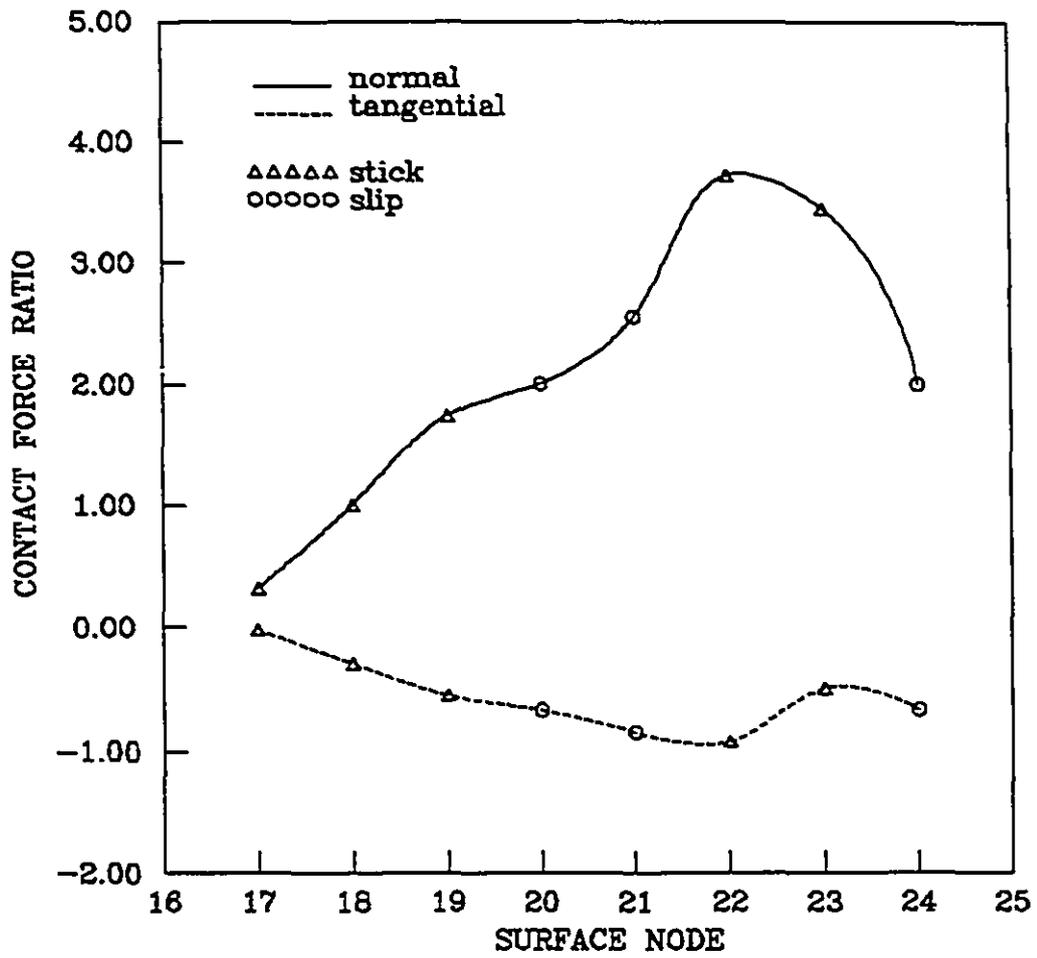


Figure 6.23: Wheel-soil contact force at time 0.020 sec: soil B

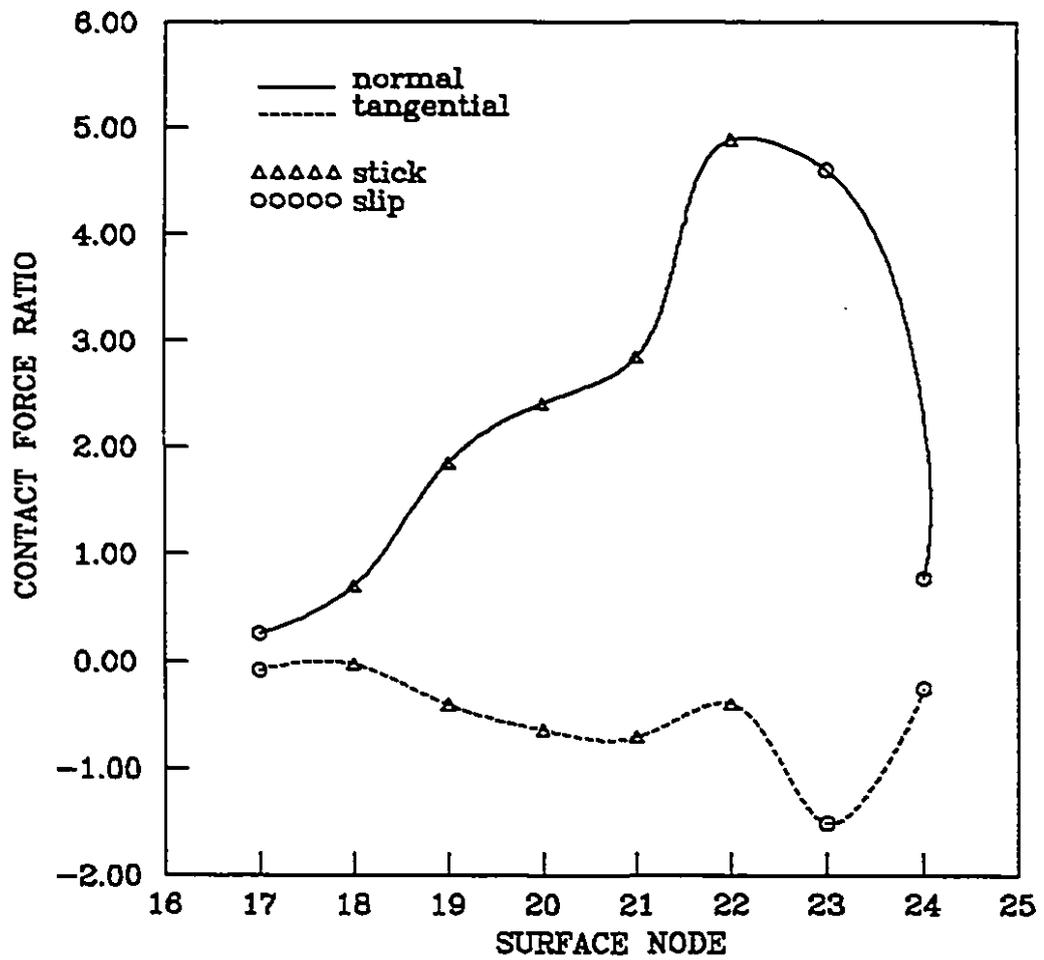


Figure 6.24: Wheel-soil contact force at time 0.020 sec: soil C

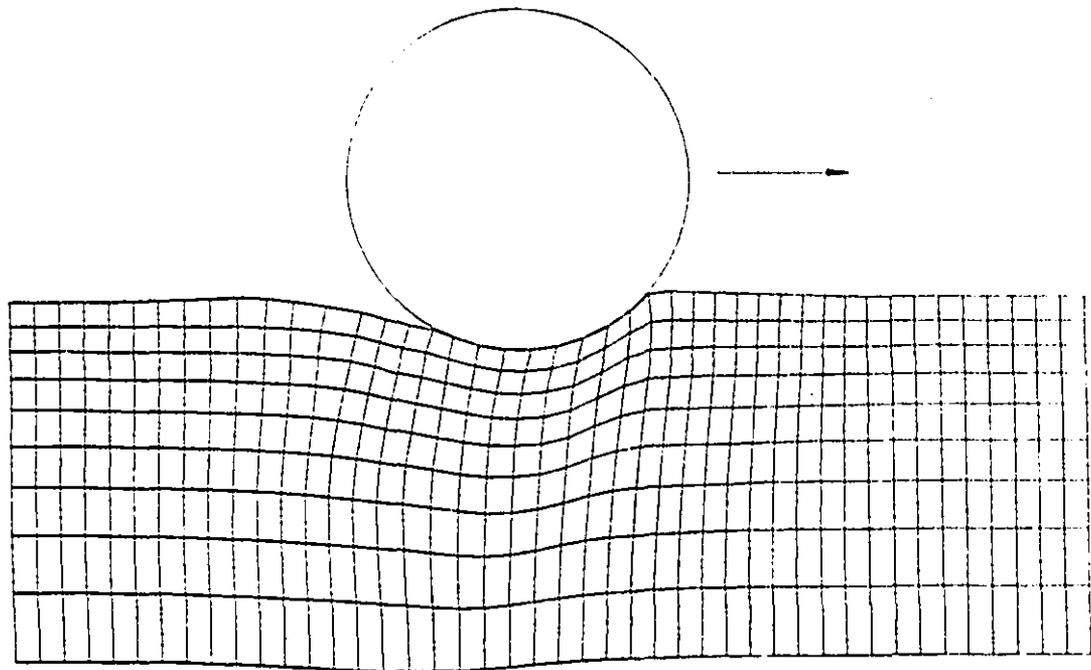


Figure 6.25: Grid deformation at time 0.020 sec: soil A

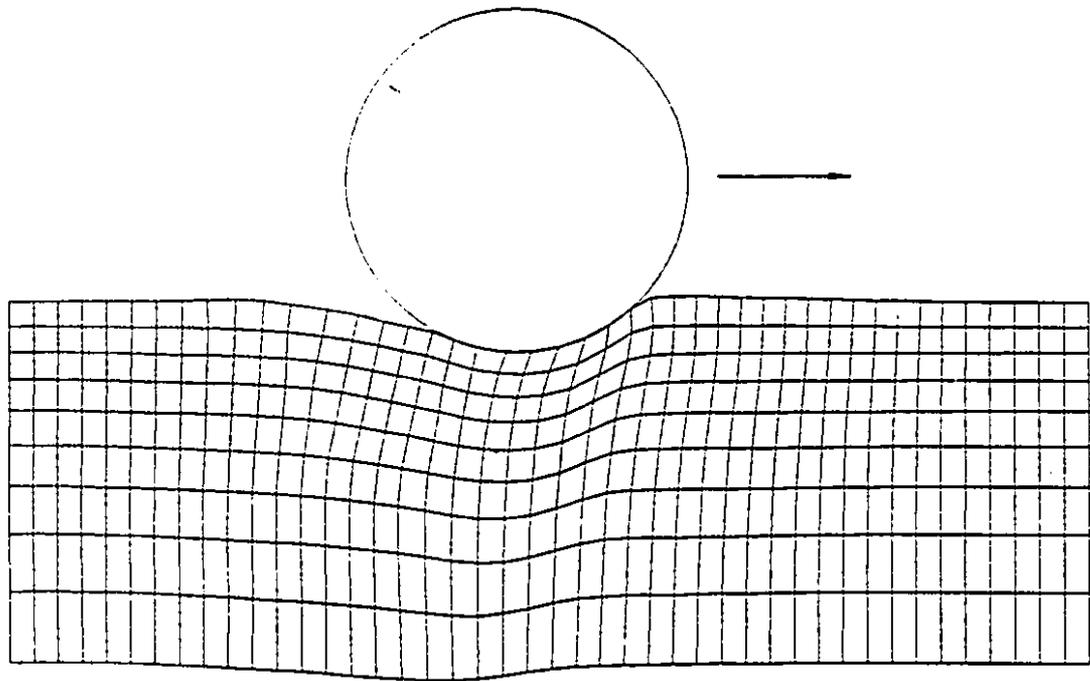


Figure 6.26: Grid deformation at time 0.020 sec: soil B

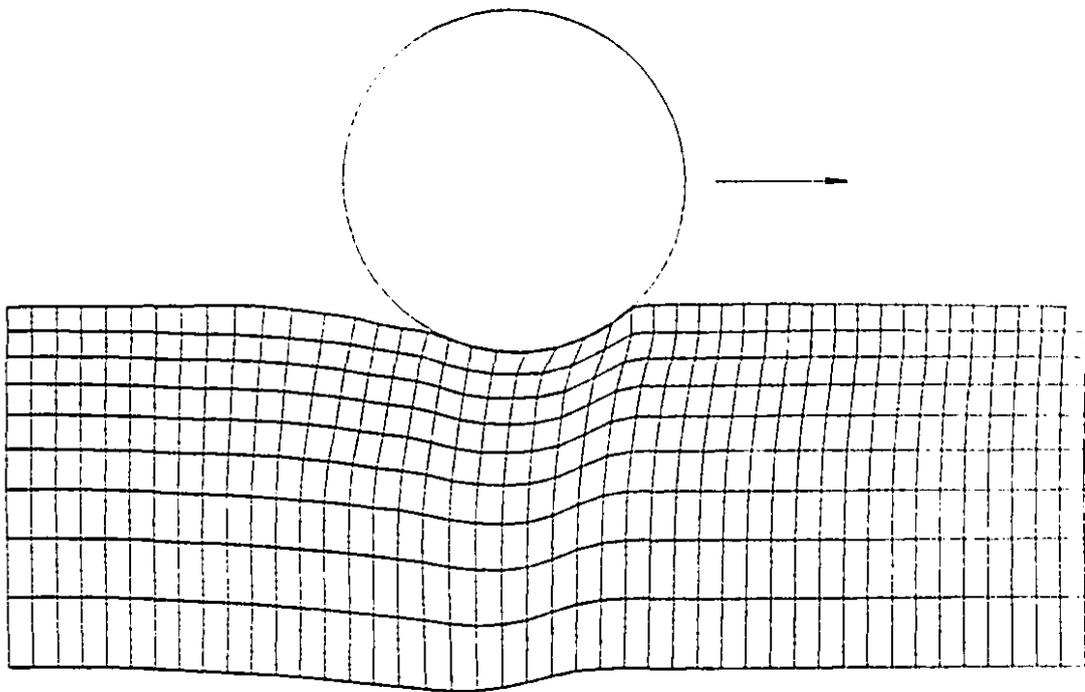


Figure 6.27: Grid deformation at time 0.020 sec: soil C

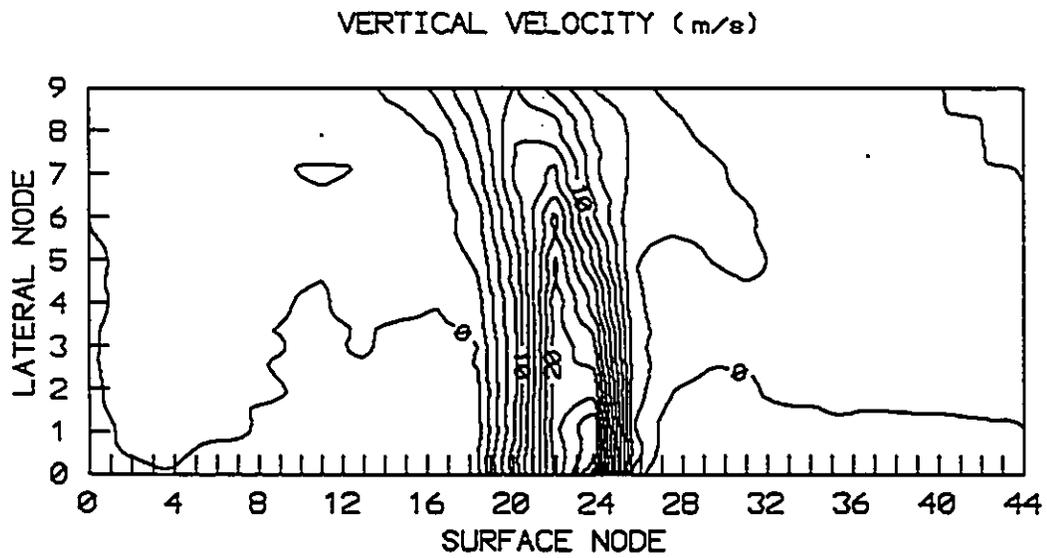
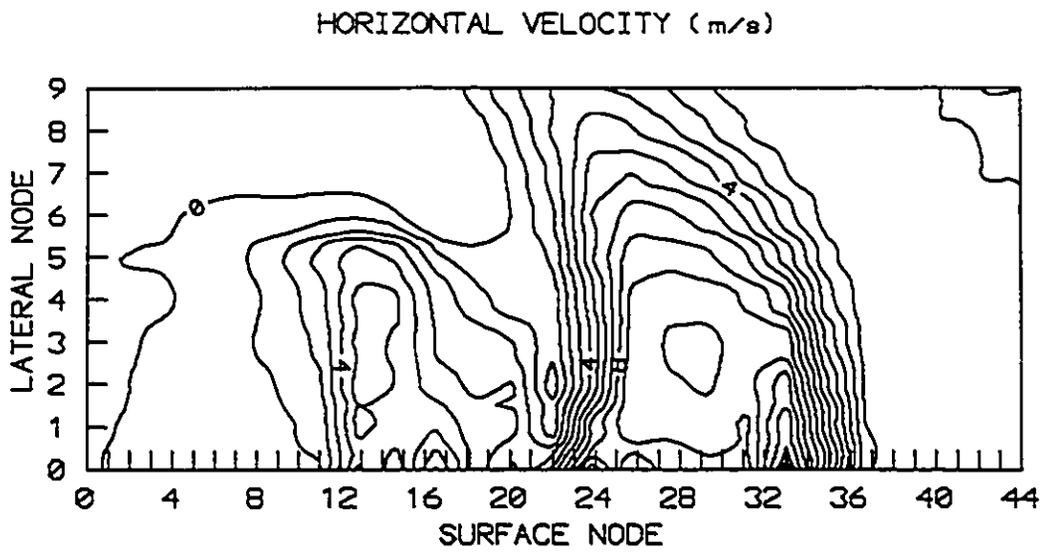


Figure 6.28: Contour map of velocity of soil A at time 0.020 sec.

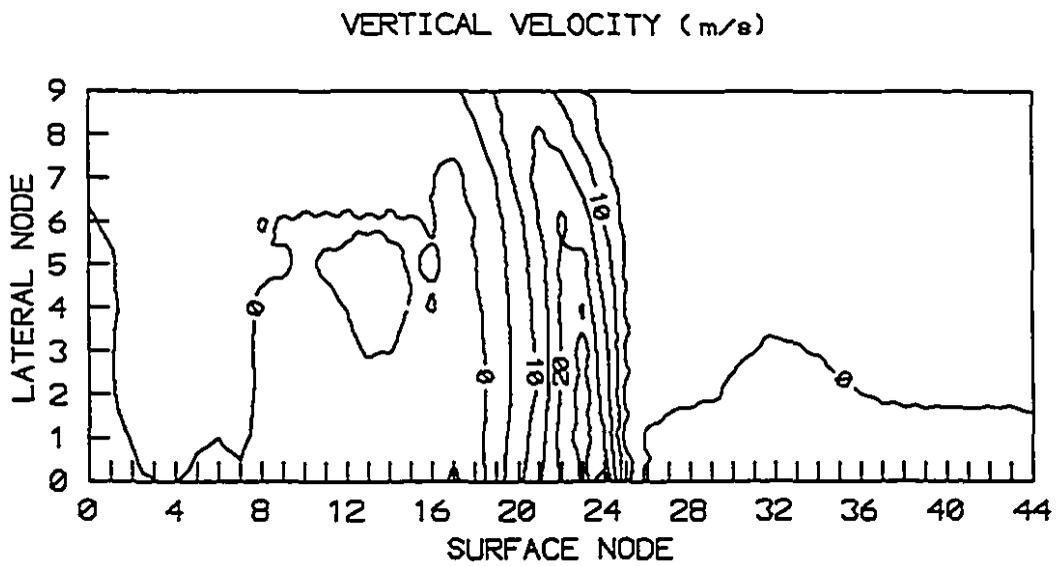
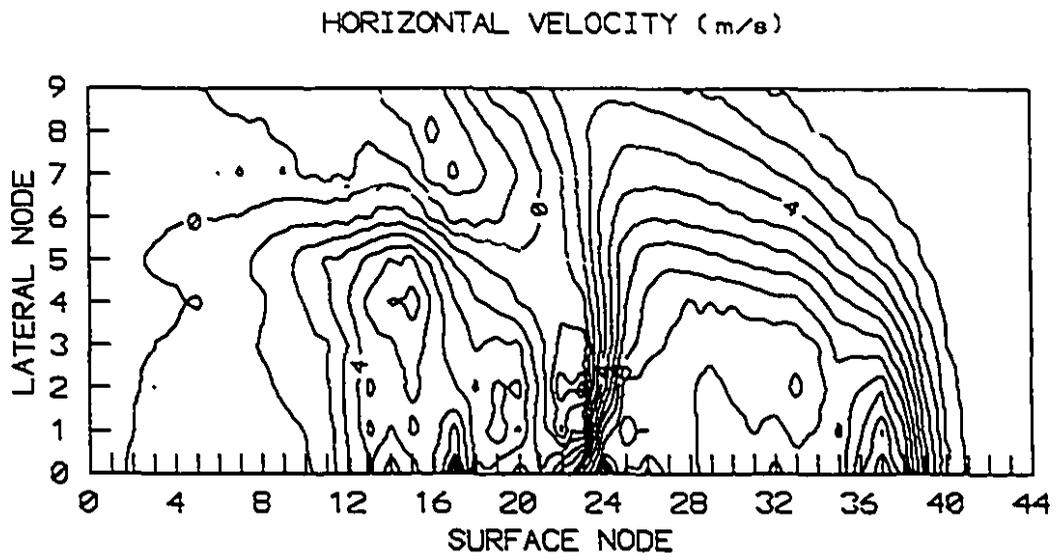


Figure 6.29: Contour map of velocity of soil B at time 0.020 sec.

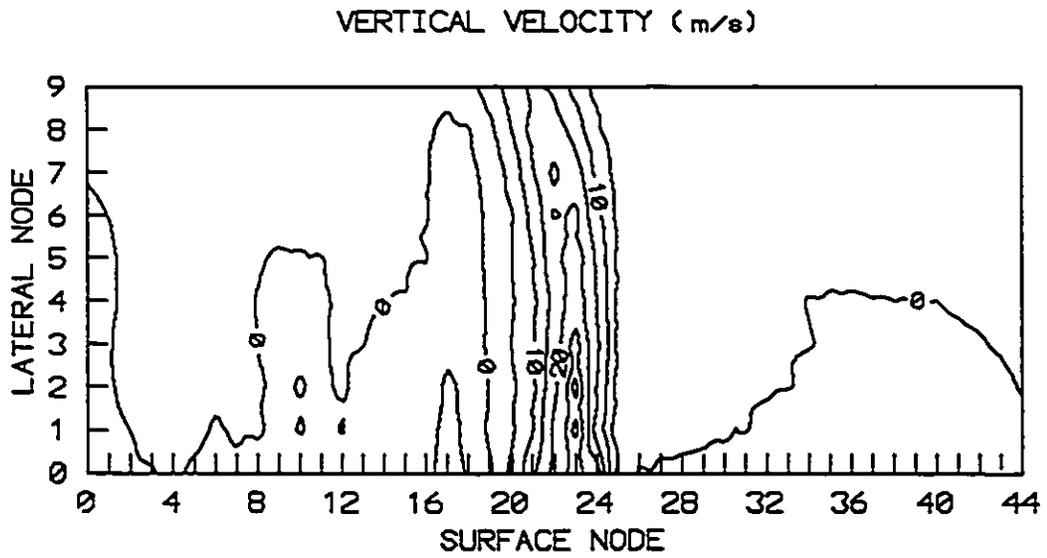
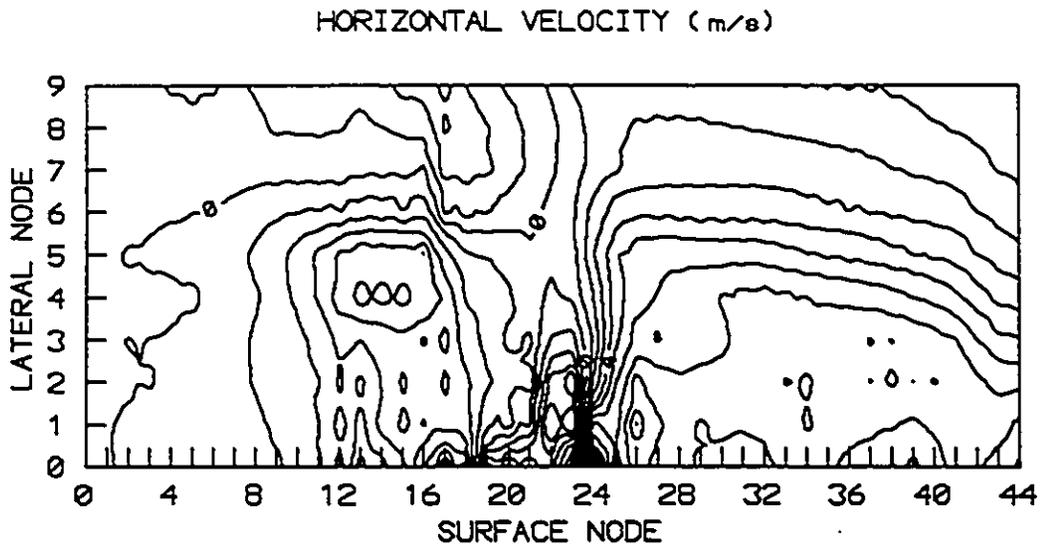


Figure 6.30: Contour map of velocity of soil C at time 0.020 sec.

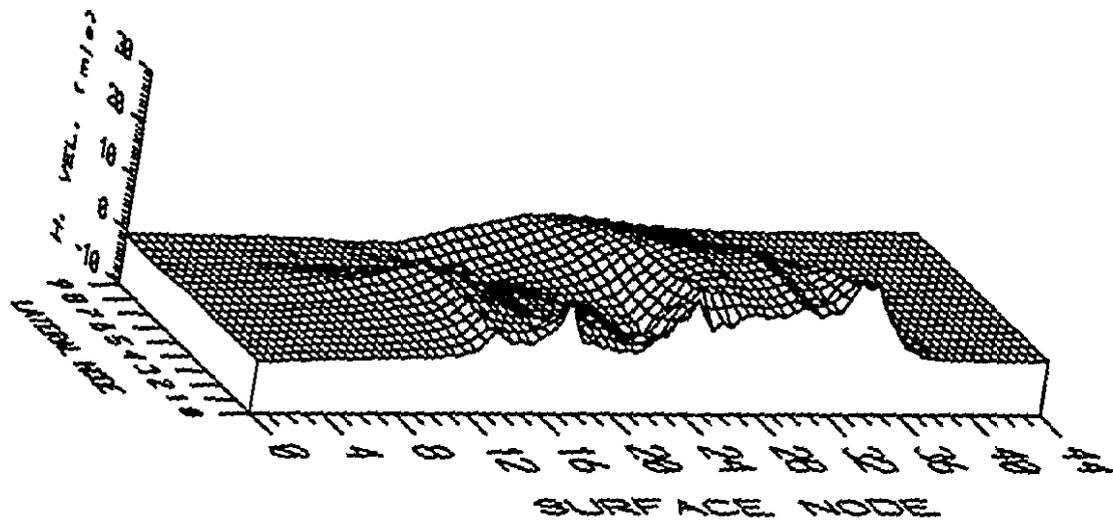


Figure 6.31: 3D plot of horizontal velocity of soil A at time 0.020 sec.

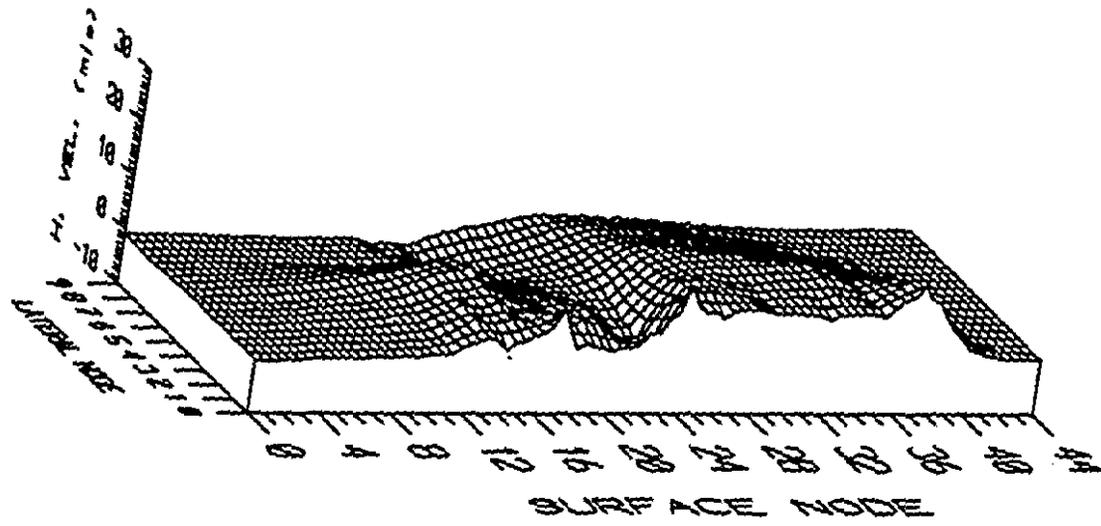


Figure 6.32: 3D plot of horizontal velocity of soil B at time 0.020 sec.

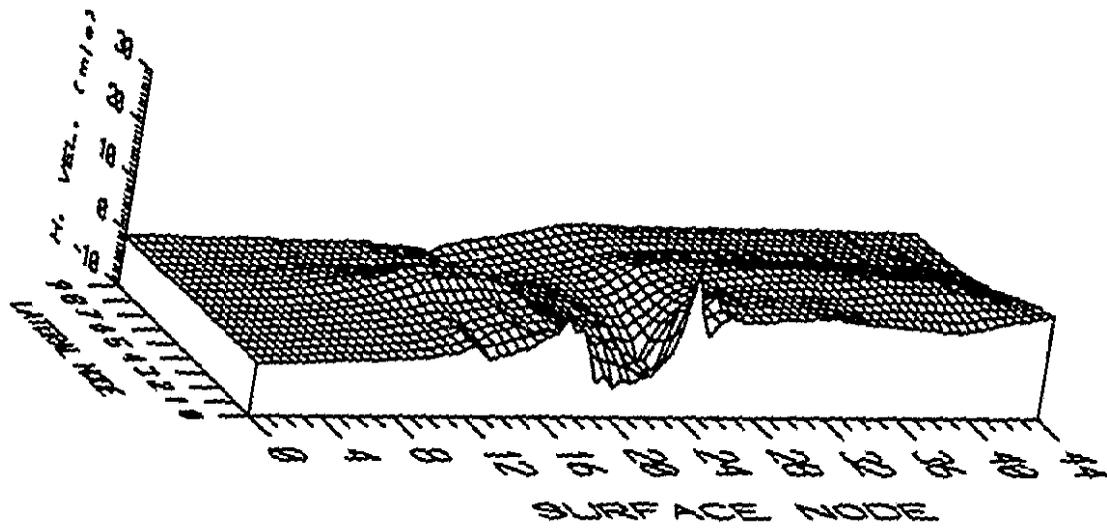


Figure 6.33: 3D plot of horizontal velocity of soil C at time 0.020 sec.

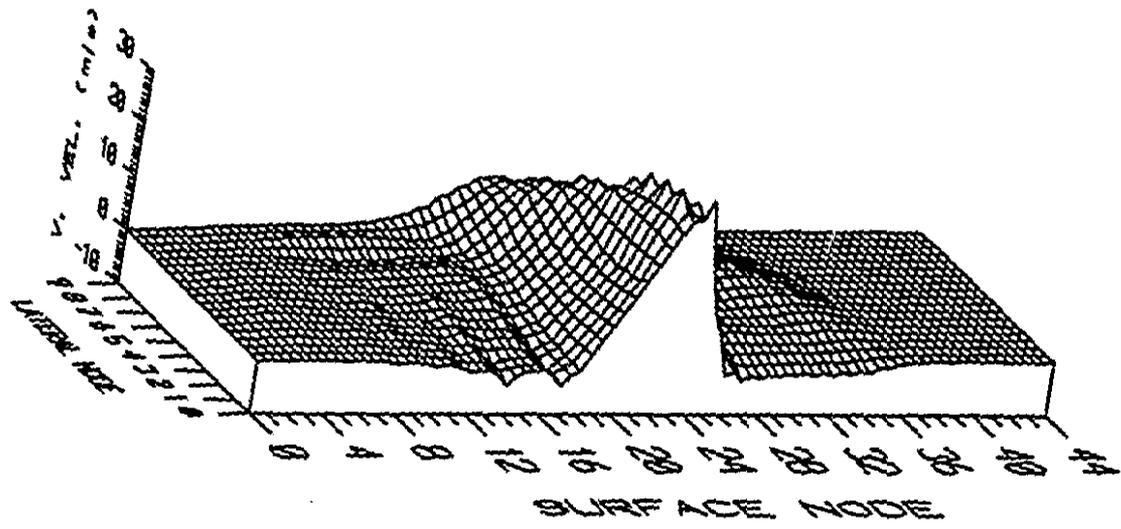


Figure 6.34: 3D plot of vertical velocity of soil A at time 0.020 sec.

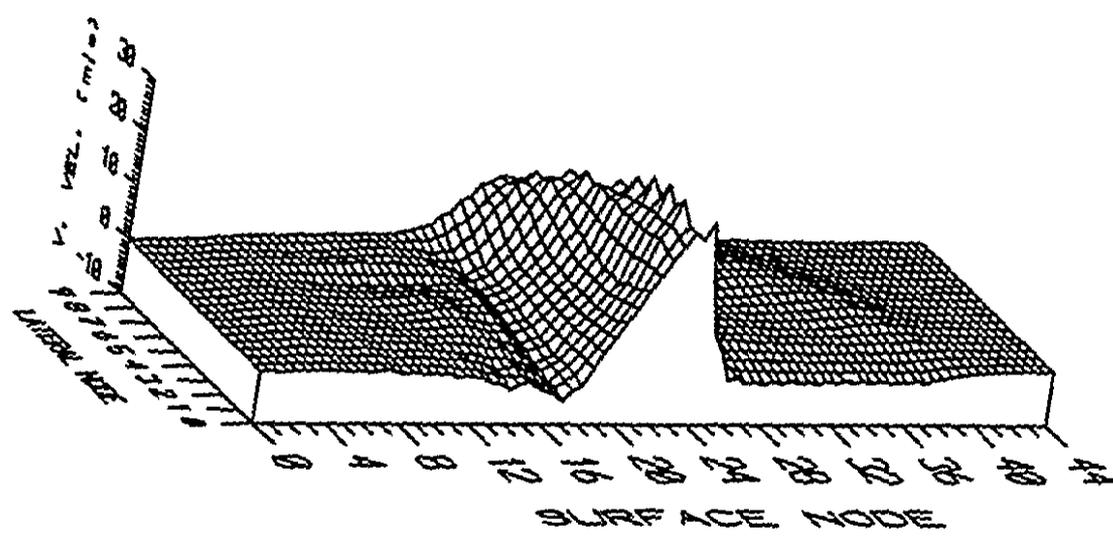


Figure 6.35: 3D plot of vertical velocity of soil B at time 0.020 sec.

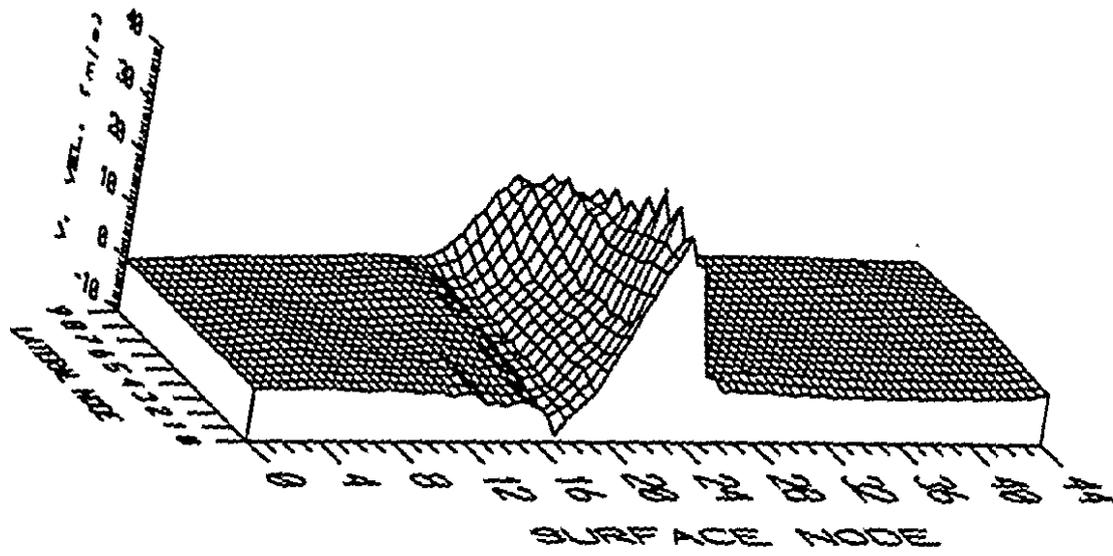


Figure 6.36: 3D plot of vertical velocity of soil C at time 0.020 sec.

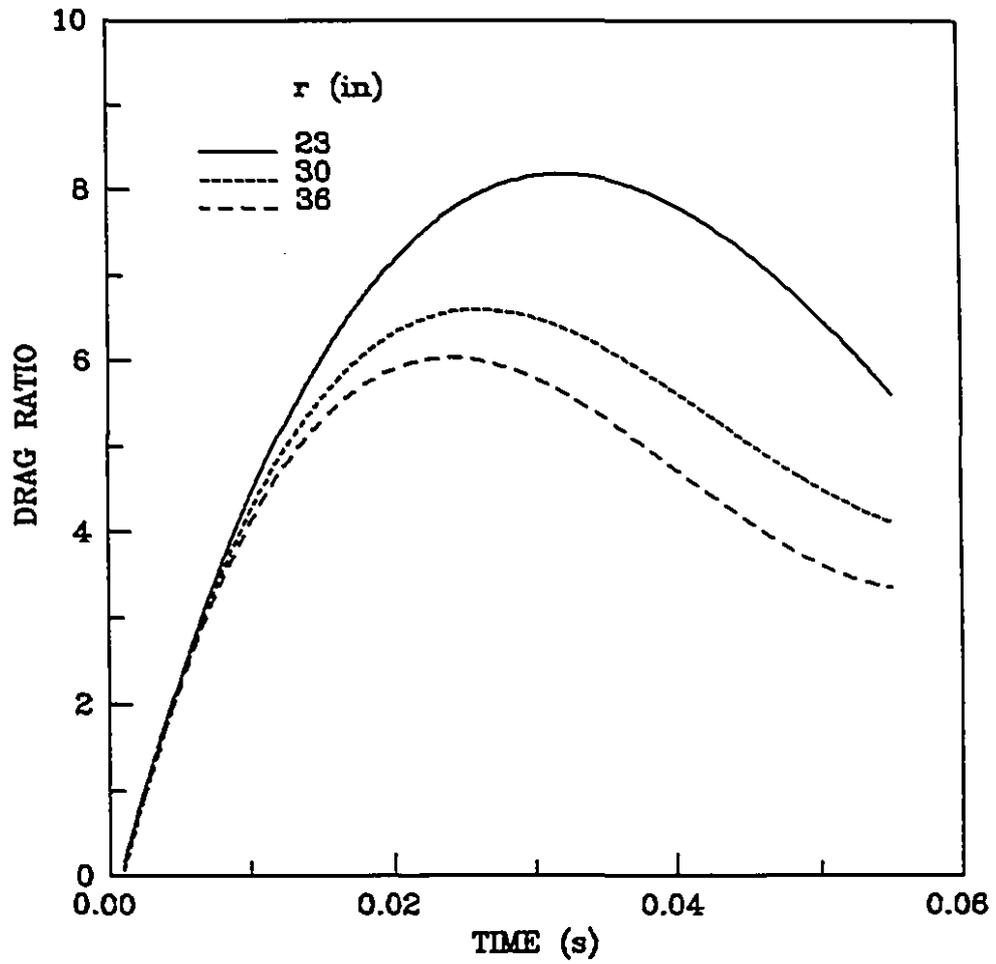


Figure 6.37: Effect of wheel radius on soil drag: soil A

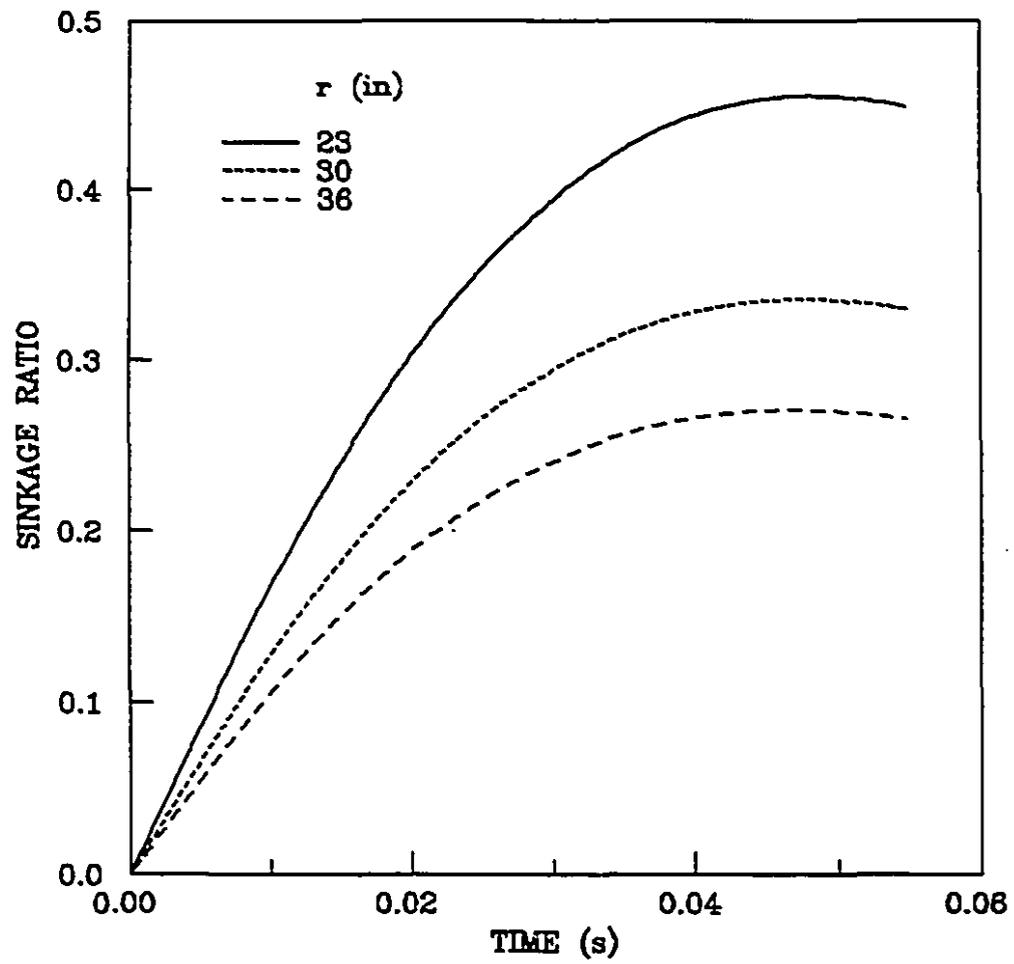


Figure 6.38: Effect of wheel radius on sinkage: soil A

References

- [1] HUGHES *et al.* A Finite Element Method for a Class of Contact-Impact Problems. *Computer Methods in Applied Mechanics and Engineering*, 8:249-276, 1976.
- [2] AGARWAL, K. B. AND RAM, B. S. Solution of equation of motion for dynamic compaction of soil. *Third International Conference on Numerical Methods in Geomechanics, Aachen, Germany, 2-6 April 1979.*
- [3] COHEN, M. AND JENNINGS, P. C. Silent Boundary Methods for Transient Analysis. In BELYTSCHKO, T. AND HUGHES, T. J. R. *Computational Methods for Transient Analysis*. Elsevier Science Publishers, Amsterdam, The Netherlands, 1983, pp 301-360.
- [4] LAST, N. C. AND HARKNESS, R. M. Kinematic (or Hour-Glass) Mode Control for Uniform Strain Quadrilateral by an Assumed Strain Technique. *Int. Journal for Numerical and Analytical Methods in Geomechanics*, 13:381-410, 1989.
- [5] LAWAL, A. *Laminar Flow and Heat Transfer to Variable Property Power-Law Fluids in Arbitrary Cross-Section Ducts*. PhD Thesis, Chemical Engineering, McGill University, Montreal, Canada, 1985.
- [6] PI, W. S. Dynamic Tire/Soil Contact Surface Interaction Model for Aircraft Ground Operations. *J. Aircraft*, 25:1038-1044, 1988.
- [7] YONG, R. N. AND FODA, M. A. Tribology Model for Determination of Shear Stress Distribution along the Tyre-Soil Interface. *J. Terramechanics*, 27:93-114, 1990.

Chapter 7

Conclusion

7.1 Summary and Conclusion

This thesis has developed a mathematical model for predicting the high-speed transient mobility of a wheel on an unpaved deformable soil surface. Motivation for this investigation was provided by the problem of aircraft/space-shuttle landing on an underlying soil terrain in an emergency.

The proposed control-volume-based finite element model incorporates the salient features of the problem, notably the wheel and soil inertia, soil strain-rate effects and stick-slip behavior of the wheel-soil interface.

Constant-strain quadrilateral elements are used, in conjunction with polygonal control volumes, in view of the resulting simplification in the discretization of the momentum conservation equations. Hour-glass or so-called kinematic or zero-energy modes, characteristic of constant-strain quadrilateral elements, are successfully controlled by ensuring that the velocity field within each element is consistent with the uniform strain-rate of the element. Although constant-strain triangular elements do not undergo hour-glass deformations and would be equally simple to implement, they

do have the disadvantage of being unduly stiff.

The discretization of an arbitrary-shaped computational domain into quadrilateral elements, using body-fitted orthogonal grid-generation techniques, is presented. The governing equations are solved by the method of dynamic relaxation.

Wave-absorbing boundary conditions are imposed along the artificial boundary of the truncated semi-infinite computational domain to prevent the reflection of incident waves back into the interior. The viscous-type boundary conditions, designed to absorb the incident waves, are modified to account for the convective inertia terms.

The wheel-soil contact forces, hence soil drag, are determined solely on the basis of the physics of the problem, i.e., no assumption regarding the distribution or magnitude of the interfacial stresses is made.

Although a viscoelastic constitutive soil model is implemented, the proposed model can incorporate any user-supplied stress-strain relationship.

An object-oriented program (OOP) of the control-volume-based finite element model is developed. An important attribute of this new programming philosophy is that code can readily be generalized or specialized to solve other problems. Thus, a variety of soil-structure interaction problems of practical importance can, with relatively little effort, be resolved within the framework of the code developed herein.

Results of numerical computations indicate that a rigid wheel, on impacting a deformable soil surface at high speed, would encounter a rather significant soil drag and undergo appreciable sinkage. Both soil drag and wheel sinkage decreased with increasing wheel radius. However, because of the obvious constraints and limitations

on the wheel size that can be used, it would appear that a rigid wheel or highly inflated tire may not be suitable for aircraft landing on soil—a low-pressure tire must be considered.

7.2 Recommendations for Further Studies

1. In view of the excessive soil drag and sinkage encountered by a rigid wheel, it is recommended that tire flexibility be incorporated into the proposed model. A low-pressure tire would no doubt encounter much less drag and sinkage. The magnitudes, of course, need to be determined so that a proper design of the aircraft landing-gear system can be effected.
2. The effect of soil surface roughness should be investigated because of the high probability that an uneven soil surface profile would be encountered in an actual field situation.
3. It is also suggested that an experimental program be initiated to verify the accuracy of the proposed model.

7.3 Contributions

The following are considered to be the primary contributions of this thesis:

1. Development of a control-volume-based finite element model of transient high-speed wheel mobility on soil for application to aircraft landing and take-off.
2. Determination of the wheel-soil contact forces on the basis of the physics of the problem, i.e., without any a priori assumptions regarding the spatial variation or magnitude of the interfacial stresses.
3. Development of a rational basis for the analysis and computer implementation of the stick-slip phenomenon along the wheel-soil interface.
4. Introduction of convective inertia terms into the viscous boundary conditions imposed on the artificial boundary of a truncated semi-infinite domain.
5. Application of the dynamic relaxation method to the generation of body-fitted orthogonal curvilinear grid over an arbitrary-shaped computational domain.
6. Development of an object-oriented control-volume-based finite element computer code (in C++) for the solution of the dynamic wheel-soil interaction and grid-generation equations. Thus, using the mechanism of inheritance, a variety of practical soil-structure interaction problems can readily be solved.
7. Presentation of previously unavailable results on transient high-speed wheel mobility on soil. This, it is envisaged, will provide future researchers the data base for validating their models.