

Discrete Element Modelling for Wheel-Soil Interaction and the Analysis of the Effect of Gravity

Eric Karpman

Department of Mechanical Engineering
McGill University, Montreal, Canada
December 2019



A thesis submitted to McGill University in partial fulfillment of the requirements
of the degree of Master of Mechanical Engineering

Supervisor: Jozsef Kövecses

© Eric Karpman 2019

“All models are wrong, but some are useful”

George E. P. Box

Abstract

The discrete element method (DEM) is widely seen as one of the more accurate, albeit more computationally demanding approaches for terramechanics modelling. Part of its appeal is its explicit consideration of gravity in the formulation, making it easily applicable to the study of soil in reduced gravity environments. The parallel particles (P^2) approach to terramechanics modelling is an alternate approach to traditional DEM that is computationally more efficient at the cost of some assumptions. Thus far, this method has mostly been applied to soil excavation maneuvers. The goal of this work is to implement and validate the P^2 approach on a single wheel driving over soil in order to evaluate the applicability of the method to the study of wheel-soil interaction. In particular, the work studies how well the method captures the effect of gravity on wheel-soil behaviour. This was done by building a model and first tuning numerical simulation parameters to determine the critical simulation frequency required for stable simulation behaviour and then tuning the physical simulation parameters to obtain physically accurate results. The former were tuned via the convergence of particle settling energy plots for various frequencies. The latter were tuned via comparison to drawbar pull and wheel sinkage data collected from experiments carried out by another research group on a single wheel testbed with a martian soil simulant in a reduced gravity environment. Sensitivity of the simulation to model parameters was also analyzed. Simulations produced promising data when compared to experiments as far as predicting experimentally observable trends, but also showed limitations in predicting the exact numerical values of the measured forces.

Résumé

La méthode par éléments discrets (DEM) est largement considérée comme l'une des approches les plus précises, bien que plus exigeantes en calcul, pour la modélisation dans le domaine de la terramécanique. Son attrait réside dans sa prise en compte explicite de la gravité dans la formulation, ce qui la rend facilement applicable à l'étude des sols dans des environnements à gravité réduite. L'approche de la modélisation terramécanique avec particules parallèles (P²) est une autre approche par éléments discrets qui est plus efficace au point de vue de temps de calcul, au détriment de certaines supposition. Jusqu'à présent, cette méthode a été principalement appliquée aux manœuvres d'excavation de sol. L'objectif de ce travail est de mettre en œuvre et de valider l'approche P² sur une seule roue traversant le sol afin d'évaluer l'applicabilité de la méthode à l'étude de l'interaction entre roue et sol. On étudie en particulier dans quelle mesure la méthode rend bien compte de l'effet de la gravité sur l'interaction roue-sol. Ceci a été réalisé en construisant un modèle et en ajustant d'abord les paramètres de simulation numériques pour déterminer la fréquence de simulation critique requise pour un comportement de simulation stable, puis en ajustant les paramètres de simulation physique pour obtenir des résultats physiquement précis. Les premiers ont été réglés avec l'analyse de la convergence des courbes d'énergie cinétique lors de la sédimentation pour différentes fréquences. Ces derniers ont été ajustés par comparaison avec les données de tirant et d'effondrement des roues tirées d'expériences réalisées par un autre groupe de chercheurs sur un banc d'essai roue unique avec un simulateur de sol martien dans un environnement à gravité réduite. L'effet des paramètres du modèle sur les résultats a également été analysé. Les simulations ont produit des résultats prometteurs par rapport aux expériences en ce qui concerne la prévision de tendances observables de manière expérimentale, mais ont également montré des limites pour la prévision des valeurs numériques exactes des forces mesurées.

Acknowledgements

I'd like to thank my supervisor, Jozsef Kövecses, for taking me on as a student and guiding this project as it evolved. Without him, I would not have had the opportunity to spend these past few years doing such interesting research. I'd also like to thank him for sharing his advice, feedback and mechanics expertise in the preparation of this thesis.

I'd also like to thank Daniel Holz, who developed the particle model described in Chapter 3 and implemented it in the *Vortex* multibody dynamics software that was used to build the models in this work. Without this amazing modelling tool, the results presented in this thesis would not have been possible. Furthermore, his guidance when I was first learning about his method was a huge help. Also, thank you to Marek Teichmann and the whole team at *CM Labs* who created and continue to improve the *Vortex* software.

Next I'd like to thank the team of researchers from Concordia, notably Parna Niksirat and Krzysztof Skonieczny, who designed, built and performed the reduced gravity experiments on the single wheel testbed that will be described in Chapter 4. The collaboration between our group and theirs was essential as many of the results presented in Chapter 6 rely on comparisons between my simulation data and the data that they collected experimentally. The model described in Chapter 5 is designed to virtually recreate their experiments. Their analysis of the experimental data, summarized in Chapter 4, was also very helpful in deciding how to analyze the simulation data in a comparable way. I'd also like to thank Krzysztof Skonieczny in particular for allowing me and some of my McGill colleagues to participate in the reduced gravity flights. Experiencing first hand what Martian gravity feels like is something that I will not soon forget.

During my time at McGill I've had the opportunity to work with the many amazing researchers of all levels that make up McGill's dynamics group. Their advice and feedback at our weekly meetings was invaluable and the work that they do is always inspiring to me. For that I'd like to thank all of them.

Last but certainly not least, I'd like to thank my parents, my brother, and my other family and friends for supporting me over these past few years. Coming back to McGill for a master's degree was always something I wanted to do, but after a few years working in industry it was not an easy decision to make. For that reason I'd like to especially thank my wonderful girlfriend for giving me the courage to finally go and do it.

Contents

Abstract	ii
Résumé	iii
Acknowledgements	iv
List of Figures	vii
List of Tables	x
Abbreviations	xi
1 Introduction	1
2 Literature Review	4
2.1 Terramechanics Models	4
2.2 Gravity Effects	12
3 Introduction to General Dynamics Modelling and the Parallel Particles Formulation	14
3.1 General Dynamics Modelling	15
3.1.1 General Dynamic Equations	16
3.1.2 Interactions Between Elements	16
3.1.3 Time-Stepping	20
3.2 Parallel Particles Formulation	22
3.2.1 Position Based Dynamics	23
3.2.2 Parallel Particles	29
3.3 Terramechanics Model Comparison	37
4 Reduced Gravity Experiments	40
4.1 Single Wheel Testbed	40
4.1.1 ExoMars Wheel	42
4.1.2 ES-2 Soil	43
4.2 Reduced Gravity Flights	43
4.3 Input Parameters	44
4.3.1 Wheel Load	45
4.3.2 Wheel Slip Ratio	45
4.3.3 Gravity	46
4.4 Outputs	47

4.4.1	Drawbar Pull	48
4.4.2	Sinkage	49
4.4.3	Images	49
4.5	Conclusions of the Experiments	49
4.5.1	Effect of gravity on DP/W	50
4.5.2	Effect of gravity on sinkage	50
4.5.3	Effect of gravity on sensitivity to loading	50
4.5.4	Effect of gravity on velocity fields	51
5	Dynamic Model and Simulation Setup	52
5.1	Mechanical Components	52
5.2	Constraints	55
5.3	Simulation Steps	59
5.3.1	Soil Preparation	59
5.3.2	Wheel Engagement	61
5.3.3	Driving	61
5.4	Simulation Parameters	62
5.4.1	Numerical Parameters	62
5.4.2	Particle Parameters	64
5.4.3	Simulation Input Parameters	67
5.5	Simulation Outputs	67
6	Results	70
6.1	Baseline Results and Discussion	71
6.2	Parametric Sensitivity	77
6.2.1	Effect of Stiffness	78
6.2.2	Effect of Damping	81
6.2.3	Effect of Adhesion	83
6.2.4	Effect of Material Density	85
6.2.5	Effect of Friction	86
6.2.6	Effect of Particle Radius	88
6.2.7	Discussion	90
6.3	Results After Tuning	92
6.3.1	Discussion	94
7	Conclusion	96
7.1	Future Work	97
A	Other Common Terramechanics Formulations	100
A.1	Bekker Model	100
A.2	Traditional Distinct Element Method (DEM)	102
B	Plots of Outputs Versus Time	104
	Bibliography	110

List of Figures

3.1	3D rendering of full dynamic system	15
3.2	Constrained rigid bodies	17
3.3	Discrete time-step	20
3.4	Distance constraint violation with weighted position correction	28
3.5	Constraint violation with weighted position corrections	31
3.6	Simplified P^2 Formulation	39
4.1	Illustration of the Single Wheel Testbed	41
4.2	3D Rendering of ExoMars Wheel	42
4.3	Compliance of ExoMars Wheel	42
4.4	Profile of a typical reduced gravity flight [1]	44
4.5	Illustration of measured outputs	47
5.1	Mechanical model (left) vs Experimental testbed (right)	53
5.2	Kinematic diagram of the mechanical system	56
5.3	Sample kinetic energy dissipation plot	64
5.4	Magnitude (top) and direction (bottom) of particle velocities in baseline simulation at Lunar-g, 164N wheel load, 70% slip	69
5.5	Outputs (Computation time, surface height, sinkage, DP) as a function of time for baseline simulation at Lunar-g, 164N wheel load, 70% slip	69
6.1	DP/W (left) and max sinkage (right) as a function of slip for baseline simulation parameters and experimental results in Earth gravity with 164N wheel load	72
6.2	Effect of gravity on max sinkage for baseline simulation (left) and experiment (right)	73
6.3	Effect of gravity on max DP/W for baseline simulation (left) and experiment (right)	73
6.4	Effect of gravity on max sinkage for baseline simulation (left) and experiment (right)	75
6.5	Effect of gravity on max DP/W for baseline simulation (left) and experiment (right)	75
6.6	Effect of reducing gravity on soil velocity field at 70% slip for baseline simulation parameters	76
6.7	DP/W (left) and max sinkage (right) as a function of slip for baseline simulation parameters and varied stiffness in Earth gravity with 164N wheel load	79

6.8	Effect of gravity on DP/W (left) and sinkage (right) for simulation with increased inter-particle stiffness	80
6.9	DP/W (left) and max sinkage (right) as a function of slip for baseline simulation parameters and varied damping in Earth gravity with 164N wheel load	81
6.10	Effect of gravity on DP/W (left) and sinkage (right) for simulation with increased inter-particle damping	82
6.11	DP/W (left) and max sinkage (right) as a function of slip for baseline simulation parameters and varied adhesion in Earth gravity with 164N wheel load	83
6.12	Effect of gravity on DP/W (left) and sinkage (right) for simulation with increased inter-particle adhesion	84
6.13	DP/W (left) and max sinkage (right) as a function of slip for baseline simulation parameters and varied material density in Earth gravity with 164N wheel load	85
6.14	Effect of gravity on DP/W (left) and sinkage (right) for simulation with increased material density	86
6.15	DP/W (left) and max sinkage (right) as a function of slip for baseline simulation parameters and varied friction in Earth gravity with 164N wheel load	87
6.16	Effect of gravity on DP/W (left) and sinkage (right) for simulation with reduced inter-particle friction	88
6.17	DP/W (left) and max sinkage (right) as a function of slip for baseline simulation parameters and varied particle radius in Earth gravity with 164N wheel load	89
6.18	Effect of gravity on DP/W (left) and sinkage (right) for simulation with reduced inter-particle friction	89
6.19	DP/W (left) and max sinkage (right) as a function of slip for baseline simulation parameters, NRC experiments, and final simulation parameters	93
A.1	Contact geometry for Bekker (left) and Wong-Reece (right) methods [2]	100
A.2	Collision detection for circular particles in DEM method [3]	102
A.3	Sample contact model for DEM	103
B.1	DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Earth-g, 164N load, 10% slip	104
B.2	DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Earth-g, 164N load, 20% slip	105
B.3	DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Earth-g, 164N load, 30% slip	105
B.4	DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Earth-g, 164N load, 40% slip	105
B.5	DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Earth-g, 164N load, 70% slip	106

B.6 DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Martian-g, 164N load, 10% slip	106
B.7 DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Martian-g, 164N load, 20% slip	106
B.8 DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Martian-g, 164N load, 30% slip	107
B.9 DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Martian-g, 164N load, 40% slip	107
B.10 DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Martian-g, 164N load, 70% slip	107
B.11 DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Lunar-g, 164N load, 10% slip	108
B.12 DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Lunar-g, 164N load, 20% slip	108
B.13 DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Lunar-g, 164N load, 30% slip	108
B.14 DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Lunar-g, 164N load, 40% slip	109
B.15 DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Lunar-g, 164N load, 70% slip	109

List of Tables

3.1	Comparison of Soil Models	38
4.1	ExoMars Wheel Dimensions	42
4.2	ES-2 Soil Parameters	43
4.3	Number of experiments run in [1] for various combinations of wheel load, wheel slip ratio and gravity	47
5.1	Initial Simulated Numerical Parameters	64
5.2	Initial Simulated Soil Parameters	65
6.1	Baseline Simulation Parameters	71
6.2	Parametric sensitivity summary	91
6.3	Final Simulation Parameters	92

Abbreviations

DEM	D iscrete E lement M ethod
PBD	P osition B ased D ynamics
P²	P arallel P articles
OGE	O n- G round E xperiment
PGE	P artial G ravity E xperiment
ESA	E uropean S pace A gency
NRC	N ational R esearch C ouncil of Canada
CIM	C enter for I ntelligent M achines
DP	D rawbar P ull
DP/W	D rawbar P ull / W eight ratio

Dedicated to Mars...

Chapter 1

Introduction

This thesis aims to take a discrete element method (DEM) framework and explore a new potential application in the field of wheel-soil modelling. Past research using a DEM particle approach to wheel-soil modelling have explicitly modelled inter-particle contact forces through springs and dampers between particles in contact. This leads to very large systems of equations that are computationally very expensive to solve. A DEM formulation where the spring-dampers are applied implicitly through position correction of contacting particles can be much more efficient since the solutions for particles can be computed in parallel. While this so-called position-based method has existed for some time - and has been implemented for modelling flexible bodies and for modelling soil excavation maneuvers for real-time simulations - this thesis aims to explore its potential for application in wheel-soil interaction modelling through comparison of simulation results with experimental data made available thanks to a research group from Concordia. Through analysis of how model input parameters affect the bulk behaviour of the model we also lay the foundation for how this method can be tuned effectively. The available experimental data includes results collected in a reduced gravity environment, and so the effect of gravity on the simulation results is also of interest. The work is presented as follows:

CHAPTER 2: A literature review is presented to provide background on the evolution and current state of the art of the field of terramechanics. This will range from the early empirical methods dating back to the early 1900s, to theoretical and semi-empirical

methods from the 1950s, to numerical methods from the 1970s - and the evolution of each of these methods right up until today. The current understanding of the effect that gravity has on soil behaviour and how this is considered in planetary rover design and testing will also be discussed.

CHAPTER 3: A more detailed description of two pre-existing but relevant mathematical formulations is provided. i) The equations governing general dynamics of finite degree of freedom systems. This is important since the wheel and all elements of the simulation other than the soil particles are governed by these dynamics equations of motion. ii) The Parallel Particle (P^2) DEM method - proposed by Holz in [4] - that governs the motion of the particles in the soil model used in this thesis.

CHAPTER 4: An overview of the reduced gravity experiments carried out by Niksirat et al [1] and the data that was gathered from them. Part of the objective is to successfully implement a wheel-soil model that can be validated by replicating experimental result gathered during Partial Gravity Experiments (PGE) and On-Ground Experiments (OGE) carried out on a single wheel testbed with a prototype of the ExoMars rover wheel mounted. This chapter describes the elements of the experiments necessary to build the simulation model, as well as the experimental results and trends that the model aims to replicate.

CHAPTER 5: A detailed look at the construction of the wheel-soil model used in simulation, constructed using the *Vortex* multibody simulation software. This includes details on how the components are defined and constrained, how the components of the model relate to their counterparts in the real-world experiments from [1], how the soil model was implemented and tuned, a description of the stages of a given simulation run, and what outputs from the simulation are recorded for post-processing and comparison to experimental results.

CHAPTER 6: Results of the simulations are presented and, where applicable, comparisons between the simulation results and the results of reduced gravity experiments from [1] are made. The section first focuses on how an initial baseline simulation captures the trends observed in the experiments and how well the simulations replicate the experimental trends that arise when reducing gravity. It then goes on to study the effect that

simulation parameters have on the outputs as well as the sensitivity of these outputs to gravity. Finally, simulation results from a tuned version of the model are presented and the discrepancies between the simulations and experiments are discussed.

CHAPTER 7: The conclusions of this thesis are summarized and suggestions for future work is detailed. Includes discussions on the challenges and assumptions that went along with working with the experimental data acquired during the reduced gravity flights, the effects of these on the results presented, additional steps for validation of the method, and the overall potential and limitations of this wheel-soil modelling method going forward.

Chapter 2

Literature Review

2.1 Terramechanics Models

Though engineers have been designing wheeled vehicles for centuries, the jump-start to progress in the field of off-road vehicles that eventually led to the birth of the field of terramechanics as it exists today can be attributed to military interest beginning in the early 1900s. Following the development of tracked vehicles for off-road use by various agriculture companies, the British War Office became interested in off-road tractor tests for their potential military application. The designs were heavily influenced by work done in the automotive industry and largely ignored any wheel-soil or track-soil relationships [5]. By 1912, the British War Office ran out of funding for the project and tractor design was continued in the private sector, which soon developed the first modern tractor [5]. As World War I began, tanks were developed mainly by adapting existing tractor designs. Since during war times there is not much room for rigorous research, modifications were made through a long and sometimes costly trial and error process. For instance, the French discovered the need to increase the track width for heavy tanks after an incident on the battlefield [5]. Between the wars, studies and developments in terramechanics shifted away from military applications and returned to their original focus on agriculture. In the 1920s, Terzaghi [6] proposed soil mechanics principles for civil engineering, but they were considered too innovative for application in wheel-soil study [5]. Still, by the

late 1920s and early 1930s researchers had begun to notice the importance of wheel-soil interactions and focused on such things as center of gravity location, sinkage, the effect of grousers and wheel dimensions in their studies [5]. While improvements and industry standards were set during this period, all studies remained empirical and therefore results were limited to the tested conditions and any generalizations could not be very accurate, and could be misleading if taken out of context [5]. With the start of World War II came renewed military interest in the field, but progress advanced in a similar fashion as during World War I, where developments were a result of trial and error. Despite this, useful observations were made over the course of the war that lead to improvements. For example, it was determined that grouser depth had an effect on performance, and the inability to further increase track width lead to investigation of whether there was an ideal distance between grousers [5].

Thus the process of trial and error continued with slow but steady progress until the middle of the 20th century, when the field of terramechanics as it is known today truly took shape as a result of M.G. Bekker's books: *Theory of land locomotion: the mechanics of vehicle mobility* [5] in 1956, *Off-the-road locomotion: research and development in terramechanics* [7] in 1960 and *Introduction to Terrain-Vehicle Systems* [8] in 1969. These works provided some of the first theoretical (or semi-empirical) models for wheel-soil behaviour. To this day they are still cited as a benchmark of the field, and are referenced often in modern work. Since their publication there have been many different kinds of models used to describe the interactions between wheel and soil. These range in their approaches from empirical, to semi-empirical, to numerical.

Empirical Models. Empirical models are ones that rely solely on experimental data to predict behaviour. This is done by recording data for a given set of conditions and wheel design and then looking for patterns that can correlate design and conditions to performance. Empirical equations can then be developed linking certain parameters of the experiments. This is a simple brute-force method that, while not very elegant, is sometimes able to predict the performance of a wheel under the same conditions that the experiments were conducted in. An early example of a purely empirical method is the Cone Index model developed by the US army during World War II to determine trafficability of a vehicle over certain terrain [9] [10]. The Waterways Experiment Station

developed a device called a cone penetrometer to provide military intelligence on the basis of a go/no-go result. It does this by measuring the force required to push a cone of known dimensions into the soil at a constant velocity and generating a Cone Index (CI) based on this force [9]. Using this index they would determine whether a vehicle would be able to drive over the terrain without getting stuck based on whether the measured CI is larger or smaller than the Vehicle Cone Index (VCI), which is a function of various vehicle parameters [10]. The bevameter technique proposed by Bekker [5] is another example of an empirical method, this time performing both normal and shear tests on the soil to determine soil properties that are then used to build a model. It performs a pressure-sinkage test by pressing a plate into the soil and measuring sinkage, as well as a shear test which measures shear displacement for various normal loads [5].

Since then, there have been many other groups who have taken an empirical approach to the study of terramechanics. In 1969, Schlegel and Morling [11] proposed calculators to optimize tractor travel speeds for maximum plowing acreage by relying in several empirically determined coefficients and equations. In 1972, Zoz [12] published a paper on his work predicting tractor field performance based on experimentally generated tire performance curves. In 1973, Wismer and Luth [13] continued the development of the empirical traction-equations that first emerged in the works of Bekker and discussed their application to wheeled off-road vehicles. In 1978, Clough et. al. [14] measured the performance of a tractor driving over different fields while varying the gear number and number of plow bodies to develop correlations that could point to combinations that produced an optimum work rate. In 2005, Gobbi et. al. [15] applied the Pacejka magic formula - developed for road tire performance prediction in 1991 [16] - to tractors driving in off-road conditions with reasonably good results. In 2010, Meirion-Griffith and Spenko [17] did an empirical study of the effect of terrain properties on planetary rover mobility, citing the fact that more sophisticated models such as the Bekker model are significantly less accurate when wheel size and loading is small. And as recently as 2013, Kane et. al. [18] employed an empirical approach to the study of multipass vehicle impacts of military vehicles driving over soft sand. These represent just a few examples of a wide body of works that have, with some degree of success, implemented empirical methods for generating terramechanics models - ranging from applications in agriculture, to military,

to planetary exploration. The fact that the approach can provide fast and reasonably accurate predictions in a narrow scope of conditions after some testing is perhaps why there are still recent examples of its use, despite the limitation that the models often lack of physical insight into the forces at play.

While all of the above mentioned variations on empirical modelling of wheel-soil dynamics can be useful, they have several significant drawbacks. First, since they largely rely on recording data from experiments and looking for correlation in the results, it is near impossible to draw any absolute conclusions or provide valid explanations for the observed behaviour. Additionally, again because they rely on experimental results, any extrapolation of the results for conditions outside of those for which there is data available cannot be accurate. Both of these drawbacks culminate in the third which is that, since new tests would need to be done to generate models for new operating conditions, the cost and time requirement of this method can become expensive if models for a large number of operating conditions are needed. This last problem can be further exacerbated if the operating condition of interest is a reduced gravity environment, as would be the case in the study of planetary rover mobility, since these conditions are very difficult and expensive to replicate on Earth.

Semi-empirical Models. Semi-empirical models have the advantage of the insight of well understood physical properties and phenomena, such as the stress-strain relationship, to create models of the soil that are tuned with real-world data. There are many types of models that can be developed in this style, but one common one is an elasto-plastic representation of the soil. In this kind of model, the deformation of the soil up to a certain failure point can be described by elastic deformation - wherein the soil returns to its original state - and by plastic deformation after this failure point. Wong uses the Mohr failure criterion in [9]. A given deformation of the soil will generate a strain that can be solved for by solving differential equations and then integrated over the surface of the wheel to obtain the interaction forces felt by the wheel. These models can vary in complexity, but many suffer from the same drawback of relying on a great deal of assumptions to be able to predict behaviour completely theoretically [19] [20] [21].

The Bekker method is currently the most widely used method for modelling wheel-soil interactions. It was first proposed in Bekker's book [5], and expanded on by himself in his subsequent books [7] [8] and by many others including Wong and Reece [20] [21] in the years that followed. In his method, Bekker combines theoretical models of stress-strain relationships and the Mohr failure criterion with experimental data to tune coefficients in the theoretical model. The coefficients are tuned using the aforementioned bevameter test. The result is a theoretical model that includes coefficients that are determined by empirical tests performed on the soil. It relies on data from experimentation to tune coefficients used in its formulation, but it uses a physics-based model to predict behaviour [8]. Certain assumptions are inevitably made. For example, Bekker simplifies the wheel to be a rigid cylinder traveling over flat soil in steady-state conditions. More recent versions of the method reduce the number of assumptions by adding more empirical coefficients that are determined experimentally [9]. Other features in newer versions of the model such as grouzers [22], multi-pass effects [23] and flexible wheels [23] also help to do away with some of the assumptions made in Bekker's original model.

The strength that allows semi-empirical models to be accurate is perhaps also its biggest weakness. Like the original empirical models, the Bekker model also requires extensive tuning of coefficients which is done through experimental tests. This allows the model to be accurate at reproducing data under the same conditions that the experiments were run in but makes it difficult to use the same model to predict behaviour in new conditions that have not first been reproduced experimentally. It might have a slightly larger scope than a pure empirical model but it still requires additional experimentation to predict behaviour in drastically different conditions. At best this can prove to be an expensive process, and at worst this can be impossible - say in the case of designing a wheel for a planetary rover destined to drive on the surface of Mars. In both cases, it is impractical to always be forced to experimentally reproduce the exact conditions one is trying to predict the behaviour for. After all, by definition, if we are to be successful at *predicting* the behaviour we must do so without using the knowledge of the expected behaviour to inform a model. Ideally, the experiments should be used as a means of validating models rather than tuning coefficients. Additionally, Bekker [5] himself pointed out that his method lacks accuracy for small wheel loads or small wheel radii. As previously

mentioned, some recent studies have even reverted back to empirical models to describe the motion of small ground vehicles to point out these flaws in the Bekker method [17]. While admittedly flawed, the Bekker method remains the industry standard and was even used in the design of the ExoMars rover [24].

Numerical Approaches. The third and final type of method for creating terramechanics models are the numerical methods. These encompass both the Finite Element Method (FEM), which treats the soil as a continuum but breaks it down into finite elements to capture the stresses generated in the soil, and the Discrete Element Method (DEM), which treats the soil as a collection of individual particles that interact with each other and the wheel. While they are computationally more expensive, improved computing power since the 1950s have made them more feasible [9]. These methods can still be time consuming, even with improved computation power, but when properly implemented they often come with several advantages over semi-empirical methods [10].

The Finite Element Method (FEM) is a common numerical method that has been used for many years in many different fields of engineering, but in the 70s and 80s its potential for application in the field of terramechanics became apparent. The essence of the method is to spatially discretize a continuum body using a piece-wise polynomial approximation. It's main limitation for modelling soil is the continuum assumption and the need for proper constitutive relations [25]. In 1971, Permupral et. al. [25] proposed a way in which the method could be adopted in the study of wheel-soil mechanics by using it to predict the stress distribution and soil deformation under the wheel of a tractor. Others such as Chung et. al. [26], Yong [27], Yong et. al. [28] and Oida [29] continued to study how the stresses in the soil could be modelled by treating the soil as a deformable continuum.

The Discrete Element Method was first proposed in 1979 by Cundall and Strack [3]. The goal was to model the soil by individually modelling the particles or groups of particles that make up the soil as distinct rigid bodies. The method uses explicit constitutive relations to describe the interactions between particles. When particles overlap, there is a force-displacement relationship developed at the point of contact. This comes in various forms, most of which are some combination of springs and dampers modelled between

interacting particles. In the original proposal, both normal and shear force increments are computed as a function of the normal and shear displacement increment, respectively. This increment is obtained by integrating the respective velocity component with respect to time and multiplying this increment by a stiffness factor. The value of the normal contact force increment is then added to the force calculated in previous time-steps. Coulomb friction is incorporated by adding the calculated shear force increment to the value obtained in previous time-steps and then checking the new shear force against the maximum allowable shear force - which is determined based on the normal contact force, the smaller of the friction angles of two colliding particles, and the smaller cohesion of the two colliding particles. Together, these represent the contact forces [3]. Thus, the original model only uses springs to model the contact, while more recent models also include damping when calculating normal and shear forces [30]. Additionally, while the original concept uses spherical particles to simplify contact detection, more recent studies have used more complex shapes such as composite geometries formed by combining multiple spheres [31] [32] or elliptical geometries [33] [34] because they claim that more complex geometries are able to more accurately capture the interlocking of soil particles. Another development in the method since its inception is the continued increase in complexity of the contact model. When originally proposed, the model consisted of a spring with a no-tension joint in the normal direction to model normal forces and a second spring with no-tension joint in the tangential direction whose force is also bounded by the laws of Coulomb friction to model shear forces [3] [9]. In later iterations, an extra torsional spring-damper was added at the contact interface [35]. Even more recently researchers began to include an additional spring without a no-tension joint in the normal direction to model the adhesive force that can exist between particles [30]. All of these models that use springs and dampers to model the interaction forces of particles in contact can be tuned to match the soil sample in question by modifying the various stiffness and damping coefficients. Many commercial and open-source software such as LIGGGHTS and EDEM exist to create soil models using various versions of DEM.

While the above mentioned papers focused more on improving or modifying DEM contact models, others have focused their study on using the method for different types of analysis, validating the accuracy of the method, and tuning the parameters to match real world

soil samples. Tuning the coefficients of these springs and dampers used for modelling the particle contact remains one of the more difficult steps in successfully using a DEM-based approach, but when tuned correctly the results can be more accurate than a semi-empirical approach [30] [36]. In 1992, Yen and Chaki [37] studied the random packing of spherical DEM particles by comparing simulation results to that of the random packing of steel balls, producing satisfactory results. In 2000, Tanaka et. al. [38] conducted a comparison study of soil deformation caused by penetration of a thin bar by comparing experimental results to those obtained using DEM. The results showed that with tuning of parameters, good simulation results could be obtained. In that paper, many important factors related to DEM are discussed. Among these is the study of how the simulation parameters, in this case the friction angle, affect simulation results - and the importance of selecting the right parameters to ensure accurate results. Also highlighted in this publication is the importance of a sufficiently small time-step to the stability and accuracy of the simulation [38]. In 2011, Briend et. al. [31] did work on the calibration of DEM parameters by simulating angle of repose and direct shear tests while varying simulation parameters in an effort to find the parameters that best match the real life results of those same tests on the desired soil sample. Others have also used similar testing to tune DEM parameters [39] [40].

Like the other discussed methods, DEM has been used in study of soil for application in the agriculture industry - for example Lia et. al. [41] who studied soil tillage using DEM. More recently, however, the method has been used in the design and simulation of wheels for planetary exploration rovers. Jiang et. al. [42] conducted reduced gravity experiments on a single wheel testbed with a lunar soil simulant and compared results to a numerical DEM simulation, Nakashima et. al. [43] studied the effect of grousers using DEM, Knuth et. al. [44] and Hopkins et. al. [45] focused on the influence of reduced gravity on granular material using DEM, and Smith et. al. [30] [46] modelled wheel-soil interaction of a rover wheel with grousers driving over rough terrain.

A variation of the classic DEM method, termed *Position Based Dynamics* (PBD), was developed by Müller et. al. in 2006 [47]. This created a simplified model where the particles have a radius but no orientation. This allows for faster computation and can

still be accurate. The original name, position-based, refers to the fact that, unlike in classic DEM, the interaction forces are applied implicitly by creating distance and velocity constraints when particles are in contact, and performing several iterations of position correction during which the positions of particles that are in violation of any constraints is changed in order to satisfy the violated constraints [47]. The magnitude of this position correction can then be transformed into an equivalent reaction force comparable to that which is applied directly by the various spring-dampers in the classic DEM method described above. The PBD method was designed as a lump parameter representation of a flexible body using particles as nodes, but the method was extended by Holz in [4] to become the *Parallel Particle* (P^2) method. This method took the formulation governing the nodes in [47] and developed a method in which the nodes behave like soil particles through design of the contact constraints. The constraints that are generated when particles are in contact are meant to implicitly have the same effect as the constitutive relations used in the traditional DEM models that were previously described [4]. The details of how the constraints are designed to achieve this will be explained in Chapter 3.

2.2 Gravity Effects

When performing Earth tests on a rover that is designed to operate in the reduced gravity environment of Mars or the Moon, the intuitive method for reproducing the reduced gravity effect is to reduce the load applied to the wheel of the rover by an equivalent amount. Thus, to test a rover on Earth that is designed to operate on the moon, one would reduce the mass of the rover to about 1/6 of its original mass while testing it on Earth so that the load applied to the wheels during tests on Earth is the same as that which will be applied when the full mass of the rover is on the moon. What this approach fails to capture is the effect that Earth's stronger gravitational field has on the soil itself. The soft soils that rovers are expected to drive over on other planets are made up of fine grains, and the effect of gravity on the interactions between these grains is overlooked when one simply reduces the mass of the rover to artificially mimic reduced gravity conditions [48]. The strategy of simulating reduced gravity by reducing the mass of the rover and maintaining the same gravity (Earth gravity) unsurprisingly yielded a

lower sinkage when tested by Kobayashi et. al. in [48] and reported by Wong in [49]. However, after Kobayashi et. al. [48] carried out reduced gravity experiments with the full rover mass in reduced gravity to see if the sinkage results in 1/6-g correspond to the sinkage results in 1-g with 1/6 of the mass, results showed that the sinkage values for the on-ground experiments did not match up with those of the reduced gravity tests. In fact, the sinkage results with the original mass in gravity fields ranging from 1/6-g to 2-g were all fairly similar. Wong [49] then compared these experimental results with what would be expected based on the standard Bekker pressure sinkage relationship:

$$\sigma = \left(\frac{k_c}{b} + k_\phi \right) \zeta^n \quad (2.1)$$

where k_c , k_ϕ and n are coefficients related to soil properties that must be tuned, σ is pressure acting on a contact patch, ζ is sinkage and b is the size of the contact patch. Through comparison of experimental results and predicted results based on simulations using Equation 2.1, Wong [49] showed that to predict sinkage on an extraterrestrial surface using identical normal loads, the relationship between sinkage on Earth and sinkage on another planet, as well as the relationship between wheel compaction resistance, depend on the ratio of the gravity fields as well as an exponential coefficient used in the Bekker model, which may not be readily available. Alternatively, predicting sinkage on an extraterrestrial surface from tests on Earth is much simpler using an identical mass, as the sinkage on Earth will be equal to the sinkage in an environment with different gravity, and the compaction resistance of the soil is only a function of the ratio of the two gravity fields [49] - thus removing the need to know any of the coefficients of Equation 2.1 in order to predict behaviour based on on-ground experiments. Kobayashi [48] came to a similar conclusion, stating that the increased flowability of the soil in reduced gravity that results from lower stress between grains combined with the reduced wheel load that results from the mass being in a lower gravity environment effectively cancel each other out and result in equal sinkage on Earth and in reduced gravity environments if equal masses are used in both sets of experiments.

Chapter 3

Introduction to General Dynamics Modelling and the Parallel Particles Formulation

In order to run a good simulation we need to create a physics-based model that can replicate real world systems. In our case, the system in question is a single wheel testbed with soft soil [1]. The mechanical system will be described in detail in Chapter 4. Modelling a system that includes wheel-soil interactions does pose a significant challenge as there are many different modelling approaches that have different strengths and weaknesses, however, other than the soil, all elements of the model used in this work are governed by well established dynamic equations that are commonly used in multibody dynamics modelling. The soil is, after all, just one component of the simulation. The first section of this chapter will focus on the basics of dynamics modelling as it is applied in the simulation tool used in this project. The second section will focus on the details of the parallel particles (P^2) method, developed in [4] by Holz, that was implemented to model the soil. Together, these two formulations are used to model the mechanical system.

3.1 General Dynamics Modelling

The model built includes some dynamic elements interacting with a terramechanics model. As such, it is important to have the proper background on how both of these elements are formulated. This section will focus on the formulation guiding the motion of the rigid body elements of the simulation that were used in this work. Figure 3.1 illustrates the model that was created. It features six main elements. Five of these can be described in a general sense as rigid bodies and are governed by the formulation described in this section. These are: the wheel - which has a composite collision geometry composed of a cylinder with boxes representing the grousers, two links which are represented as a horizontal and vertical boxes, the bin which is represented as a composite of five box geometries forming a container that is open on top, and the pressure plate which is represented as a box covering the open face of the bin. The specifics of each element will be discussed in Chapter 5. The elements interact with each other through bilateral constraints which will be defined in this section on general dynamic modelling. There are no unilateral constraints between any of these rigid bodies, which simplifies the formulation somewhat since it means there is no need for a friction model or contact constraints that can detach. The only contact between elements involves the sixth component of the model, which is the soil. The soil particle dynamics is handled in a slightly different manner, which will be described in Section 3.2.

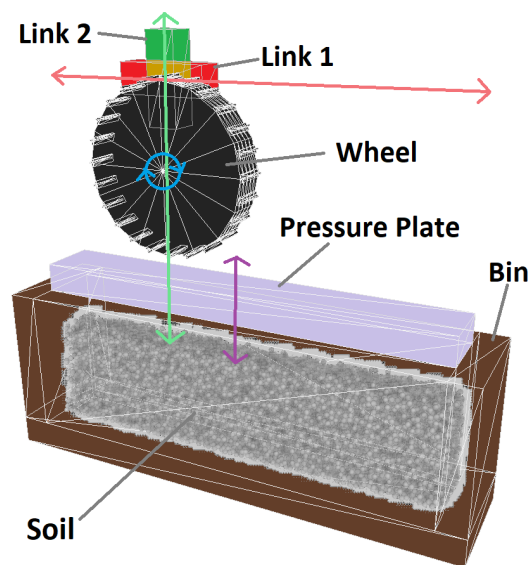


FIGURE 3.1: 3D rendering of full dynamic system

3.1.1 General Dynamic Equations

A dynamic system can be represented with a finite degree of freedom model whose state is defined by p generalized coordinates collected in $\mathbf{q} = [q_1, q_2, \dots, q_p]^T$ and n generalized velocities collected in $\mathbf{v} = [v_1, v_2, \dots, v_n]^T$. These generalized coordinates and velocities are used to define the motion of all the mass elements in the system. Often, these mass elements are grouped together to form rigid bodies. In general, the velocity components in \mathbf{v} are related to the time-derivatives of the generalized coordinates by the transformation:

$$\dot{\mathbf{q}} = \mathbf{N}\mathbf{v} \quad (3.1)$$

where \mathbf{N} is a $p \times n$ transformation matrix with $p \geq n$. The resulting system, excluding any interaction between components, can be described by the dynamic equation:

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{c} = \mathbf{f} \quad (3.2)$$

where \mathbf{M} is the $n \times n$ mass matrix, \mathbf{c} is an $n \times 1$ matrix that contains terms that are non-linear in \mathbf{v} which represent Coriolis and centrifugal components of the motion, and \mathbf{f} represents the generalized forces acting on the bodies.

3.1.2 Interactions Between Elements

In addition to the generalized forces acting on the bodies, an extra m equations are added to the system to describe the interactions between the elements through interaction velocity components $\mathbf{u} = [u_1, u_2, \dots, u_m]^T$. These are related to the velocities through the following relation:

$$\mathbf{A}\mathbf{v} = \mathbf{u} \quad (3.3)$$

where \mathbf{A} is an $m \times n$ Jacobian matrix representing the interaction kinematics. Constraints can either be of the equality type $\mathbf{u} = \mathbf{b}$, as is the case for revolute joints, prismatic joints, and spherical joints, or of the inequality type $\mathbf{u} \leq \mathbf{b}$, as would be the case for unilateral contact between bodies. For the purposes of this work we deal only with

equality constraints. Eq.(3.3) can be rewritten at the acceleration level as:

$$\mathbf{A}\dot{\mathbf{v}} + \dot{\mathbf{A}}\mathbf{v} = \dot{\mathbf{u}} \quad (3.4)$$

where $\dot{\mathbf{A}}$ is the time derivative of the constraint Jacobian and $\dot{\mathbf{v}}$ represents the generalized accelerations. Each constraint velocity component in \mathbf{u} has an interaction force component associated with it. These forces are collected in $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_m]^T$. Since the components of \mathbf{u} can represent either linear or angular velocity, the associated component of $\boldsymbol{\lambda}$ can either represent a force or a moment. Including the interaction forces in Eq.(3.2) yields:

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{c} = \mathbf{f} + \mathbf{A}^T\boldsymbol{\lambda} \quad (3.5)$$

The interaction forces are determined either by the kinematic constraints or a constitutive model - which is generally used for modelling contacts. In the case of kinematic constraints, the relative motion of interacting elements is restricted and the forces are solved for such that the constraint velocities, \mathbf{u} , are satisfied. Constitutive models are those in which the forces, $\boldsymbol{\lambda}$, are calculated as a function of stiffness and damping terms. In the case where interaction are defined through kinematic constraints, Eq.(3.5) must be solved with Eq.(3.4). They can be combined and expressed in matrix form as:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f} - \mathbf{c} \\ \dot{\mathbf{u}} - \dot{\mathbf{A}}\mathbf{v} \end{bmatrix} \quad (3.6)$$

The structure of the Jacobian matrix, \mathbf{A} , depends on the type of joint between two system elements. Two joint types used in the model illustrated in Figure 3.1 are prismatic joints and revolute joints. To illustrate how the Jacobian is structured for these joints we will consider the simple case of two arbitrary rigid bodies connected by a single joint.

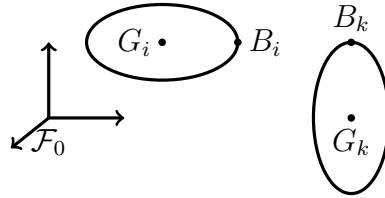


FIGURE 3.2: Constrained rigid bodies

Say there are two arbitrary three dimensional rigid bodies i and k , shown in Figure 3.2, with constraints restricting the motion between points B_i and B_k on bodies i and k . Then, the following describes the relationship between the generalized velocities of the bodies and the relative motion of the two points of interest on those bodies:

$$\underbrace{\begin{bmatrix} \mathbf{R}_i^T & \mathbf{R}_i^T r_{G_i B_i}^{\mathcal{F}_0} & \mathbf{R}_i^T & \mathbf{R}_i^T r_{G_k B_k}^{\mathcal{F}_0} \\ \mathbf{0} & \mathbf{R}_i^T & \mathbf{0} & \mathbf{R}_i^T \end{bmatrix}}_{\mathbf{J}_{6 \times 12}} \underbrace{\begin{bmatrix} \mathbf{v}_{G_i}^{\mathcal{F}_0} \\ \omega_{G_i}^{\mathcal{F}_0} \\ \mathbf{v}_{G_k}^{\mathcal{F}_0} \\ \omega_{G_k}^{\mathcal{F}_0} \end{bmatrix}}_{\mathbf{v}_{12 \times 1}} = \underbrace{\begin{bmatrix} \mathbf{v}_{B_{ik}}^{\mathcal{F}_i} \\ \omega_{B_{ik}}^{\mathcal{F}_i} \end{bmatrix}}_{\mathbf{u}_{6 \times 1}} \quad (3.7)$$

where \mathcal{F}_0 is the inertial frame, \mathcal{F}_i is a body fixed frame attached to body i , \mathbf{R}_i is the rotation matrix between the coordinate systems defined in reference frames \mathcal{F}_0 and \mathcal{F}_i , $r_{G_i B_i}^{\mathcal{F}_0}$ is the vector from G_i to B_i expressed in the inertial coordinate system, $r_{G_k B_k}^{\mathcal{F}_0}$ is the vector from G_k to B_k expressed in the inertial coordinate system, and $\mathbf{v}_{B_{ik}}^{\mathcal{F}_i}$ and $\omega_{B_{ik}}^{\mathcal{F}_i}$ are the desired relative linear and angular velocities between points B_i and B_k expressed in the coordinate system in reference frame \mathcal{F}_i . For simplicity, the coordinate systems are generally chosen so that the desired relative motion is along the axes in one of the systems. For each joint, we can form a Jacobian matrix \mathbf{A}_j by removing one or more rows from matrix \mathbf{J} which is defined in Eq.(3.7). We will examine how this is done for the two joint types of interest.

Prismatic Joint. A prismatic joint allows free or controlled relative translation between two bodies along a given axis while restricting all other relative motion between the bodies. The coordinate systems are selected such that the axis of motion is an axis of the coordinate system in the body-fixed frame of one of the two bodies. In Eq.(3.7) the relative velocity components are expressed in the coordinate system in frame \mathcal{F}_i , so we will assume that one of these three velocity components is to be left un-constrained. In the above example, this would be achieved by deleting the row of matrix \mathbf{J} that is associated with the axis of motion that is to be left free. For example, if the bodies are to be free to translate along the axis defined by the first axis of the coordinate system in

reference frame \mathcal{F}_i , then the sub-matrix of the Jacobian representing this joint would be:

$$\mathbf{A}_j = \mathbf{J}_1 \quad (3.8)$$

where the notation \mathbf{J}_1 refers to matrix \mathbf{J} with the 1st row deleted. The associated row in the \mathbf{u} vector would also be deleted, and all remaining components of \mathbf{u} would be set to zero to indicate that relative motion in these directions is not permitted. In this scenario, there would be five components of $\boldsymbol{\lambda}$ associated with the five constrained degrees of freedom. Thus, this adds five rows to the Jacobian matrix \mathbf{A} .

Revolute Joint. A revolute joint allows free or controlled relative rotation between two bodies about a given axis while restricting all other relative motion between two bodies. The coordinate systems are selected such that the axis of rotation is an axis of the coordinate system in the body-fixed frame of one of the two bodies. As before, since in Eq.(3.7) the relative velocity components are expressed in the coordinate system attached to body i we will assume that one of these three angular velocity components is to be left un-constrained. In the above example, this would be achieved by deleting the row of matrix \mathbf{J} that is associated with the axis of motion that is to be left free. For example, if the bodies are to be free to rotate along the axis defined by the first axis of the coordinate system in reference frame \mathcal{F}_i , then the sub-matrix of the Jacobian representing this joint would be:

$$\mathbf{A}_j = \mathbf{J}_4 \quad (3.9)$$

where the notation \mathbf{J}_4 refers to matrix \mathbf{J} with the 4th row deleted. The associated row in the \mathbf{u} vector would also be deleted, and all remaining components of \mathbf{u} would be set to zero to indicate that relative motion in these directions is not permitted. In this scenario, there would be five components of $\boldsymbol{\lambda}$ associated with the five constrained degrees of freedom. Thus, this too adds five rows to the Jacobian matrix \mathbf{A} .

In a general system with an arbitrary number of bodies and j constraints between bodies - each with an associated sub-matrix \mathbf{A}_j - the Jacobian matrix would look like:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_j \end{bmatrix} \quad (3.10)$$

Chapter 5 will describe how the various elements seen in Figure 3.1 are connected using the joints described above.

3.1.3 Time-Stepping

In order to run a dynamic simulation, we have to be able to update the system state from one point in time to the next. We consider a time-step from t_k to t_{k+1} with step size $h = t_{k+1} - t_k$ as illustrated in Figure 3.3. It is assumed that the state of the system is known at the start of the time-step and we would like to solve for the state at the end of the time-step.

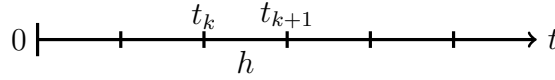


FIGURE 3.3: Discrete time-step

We use \mathbf{q}_k and \mathbf{v}_k to represent the state at the start of the time-step, and \mathbf{q}_{k+1} and \mathbf{v}_{k+1} to represent the state at the end of the time-step. A first order approximation - often called Euler approach - can be employed to step the system forward in time.

Explicit Euler Method. The explicit Euler method calculates the state of the system at time t_{k+1} based on the current time t_k by assuming that the slope of the solution is constant for the duration of the time-step, and that this slope is equal to the slope at the start of the time-step. Because this is an approximation, the assumption is more accurate when h is small. For a simple ODE the problem looks like:

$$\frac{\partial q}{\partial t} = f(t, q(t)), \quad q(t_0) = q_0 \quad (3.11)$$

With the aforementioned assumption that the slope is constant for the duration of the time-step we can write:

$$\frac{q_{k+1} - q_k}{h} = f(t_k, q_k) \quad (3.12)$$

where q_k is the approximate solution of the system at time t_k calculated during the previous time-step, $f(t_k, q_k)$ represents the slope of the function at the start of the time-step, and q_{k+1} is the unknown solution at the end of the time-step. Therefore, the approximate solution at the end of the time-step is given by:

$$q_{k+1} = q_k + hf(t_k, q_k) \quad (3.13)$$

Implicit Euler Method. The implicit Euler method, sometimes called the backward Euler method, aims to solve the same type of ODE as the previous method, seen in Eq.(3.11). The same assumption is made that the slope is constant for the duration of the time-step. The difference here is that in this version of the method it is assumed that the slope throughout is equal to the value at the end of the time-step. This leads to the following expression for the state at the end of the time-step:

$$\frac{q_{k+1} - q_k}{h} = f(t_{k+1}, q_{k+1}) \quad (3.14)$$

where $f(t_{k+1}, q_{k+1})$ represents the slope of the function at the end of the time-step. As in the previous method, this approximation is more accurate for small values of h . The backward Euler method approximates the solution at the end of the time-step as:

$$q_{k+1} = q_k + hf(t_{k+1}, q_{k+1}) \quad (3.15)$$

where the unknown q_{k+1} term is on both sides of the equation. It can then be solved for by iterating until it converges to a solution or by other solver schemes. Several methods exist to solve such a problem.

Semi-Implicit Euler Method. The semi-implicit Euler method can be used as a more accurate way of solving a pair of differential equations. Suppose we consider a case where we would like to solve the system:

$$\begin{aligned} \frac{\partial q}{\partial t} &= f(t, v(t)), \quad q(t_0) = q_0 \\ \frac{\partial v}{\partial t} &= g(t, q(t)), \quad v(t_0) = v_0 \end{aligned} \quad (3.16)$$

The values of the two functions at the end of the time-step can be approximated as:

$$\begin{aligned} v_{k+1} &= v_k + hg(t, q_k) \\ q_{k+1} &= q_k + hf(t, v_{k+1}) \end{aligned} \quad (3.17)$$

The difference between this method and the two previous methods is that v_{k+1} is calculated using the state at the start of the time-step, as in the explicit Euler method, while q_{k+1} is calculated using the state at the end of the time-step, as in the implicit Euler method. Returning to the original problem we can get the generalized positions and velocities at the end of the time-step through a semi-implicit Euler integration scheme by first updating the generalized velocities and then updating the positions using the velocity from the end of the time-step:

$$\begin{aligned} \mathbf{v}_{k+1} &= \mathbf{v}_k + h\dot{\mathbf{v}}_k \\ \mathbf{q}_{k+1} &= \mathbf{q}_k + h\mathbf{N}\mathbf{v}_{k+1} \end{aligned} \quad (3.18)$$

where the generalized acceleration terms, $\dot{\mathbf{v}}_k$, are computed by the solver based on the initial state of the system. This method is used to obtain a time-discretized version of Eq.(3.6).

3.2 Parallel Particles Formulation

Besides the elements that are governed by the dynamic equations described in the previous section, another major component of the model is the soil. The soil is governed by a formulation called the Parallel Particle (P²) method which was presented by Holz in [4] and is an evolution of the Position Based Dynamics (PBD) method presented by Müller et. al. in [47]. This method is a version of a discrete element method in which kinematic constraints implicitly apply interaction forces through position correction rather than applying them explicitly through constitutive spring-damper-type relations like traditional discrete element approaches. The P² formulation can be summarized by the pseudo-code seen in Algorithm 1 for a simulation containing n particle and integrated with a time-step of h . The initial particle position vectors, \mathbf{x}_i , velocity vectors, \mathbf{v}_i , masses, m_i , and radii, r_i

are initialized in Line 2 and then time-stepping begins. Within a time-step, the algorithm first computes a predicted position for each particle at the end of the time-step in Line 6. Then begins an iterative position correction process that corrects the initial predicted particle positions. The solver runs a collision detection algorithm in Line 9 to detect any contacts. In the case of a contact, collision constraints are generated in Line 10. Based on these constraints, position correction vectors are calculated and applied to each particle in the system in Lines 12-13, thus generating new predicted particle positions. The collision detection and position correction process in Lines 9-13 is then repeated for a fixed number of iterations before the position and velocity vectors for each particle are eventually updated at the end of a time-step in Lines 15-16. This section will provide a detailed explanation of the P² method as it is presented in [4] and [47].

Algorithm 1 Parallel Particles Algorithm

```

1: for  $i = 1 : n$  do
2:    $\mathbf{x}_i = \mathbf{x}_i^0, \mathbf{v}_i = \mathbf{v}_i^0, m_i = \mathbf{m}(i), r_i = \mathbf{r}(i)$  ▷ Initialize particle states
3:    $t = 0$ 
4: while  $t < t_f$  do
5:   for  $i=1:n$  do ▷ Beginning of time-step
6:      $\mathbf{p}_i \leftarrow \mathbf{x}_i + h\mathbf{v}_i + h^2 \frac{\mathbf{f}_e}{m_i}$  ▷ Predict particle position
7:    $c = 0$ 
8:   while  $c < n_s$  do ▷ Run  $n_s$  solver iterations
9:      $\text{CollisionDetection}(\mathbf{p}, \mathbf{r})$ 
10:     $(C_1, \dots, C_m) = \text{GenerateCollisionConstraints}(\mathbf{p}, \mathbf{r})$ 
11:    for  $i = 1 : n$  do
12:       $\Delta \mathbf{p}_i = \text{PositionCorrection}(C_1, \dots, C_m, \mathbf{p})$ 
13:       $\mathbf{p}_i \leftarrow \mathbf{p}_i + \Delta \mathbf{p}_i$  ▷ Update predicted position
14:     $c = c + 1$ 
15:   $\mathbf{v}_i \leftarrow (\mathbf{p}_i - \mathbf{x}_i)/h$  ▷ Update Velocity
16:   $\mathbf{x}_i \leftarrow \mathbf{p}_i$  ▷ Update Position
17:   $t = t + h$  ▷ End of time-step

```

3.2.1 Position Based Dynamics

In the Müller et. al. [47] framework, the system is represented by a set of point mass nodes and a series of constraints that connect the nodes. We'll consider a system that contains n nodes. The state of node i at the start of a time-step is defined by a position vector $\mathbf{x}_i = [x_{i1}, x_{i2}, x_{i3}]^T$, a velocity vector $\mathbf{v}_i = \dot{\mathbf{x}}_i = [v_{i1}, v_{i2}, v_{i3}]^T$ and a mass m_i . The

state of each node at the end of a time-step, \mathbf{x}_i^+ and \mathbf{v}_i^+ , is obtained through an iterative process that involves predicting new node coordinates and correcting the predictions if they create any constraint violations. After a fixed number of position correction iterations the final node positions and velocities are computed for the end of the time-step. The vector \mathbf{p}_i^b represents the predicted position of node/particle i after b solver iterations and $\mathbf{P}^b = [\mathbf{p}_1^b, \dots, \mathbf{p}_n^b]$ is an array containing all node positions after b solver iterations. The following procedure describes how the algorithm works for a single time-step of length h .

After initializing \mathbf{x}_i , \mathbf{v}_i and the inverse mass, $w_i = 1/m_i$, of a node, an initial predicted position $\mathbf{p}_i^0 = [p_{i1}^0, p_{i2}^0, p_{i3}^0]^T$ is calculated. This represents an initial estimate of the coordinates of the node at the end of the time-step. The initial prediction is derived through an explicit Euler integration:

$$\mathbf{p}_i^0 = \mathbf{x}_i + h\mathbf{v}_i + h^2 \frac{\mathbf{f}_e}{m_i} \quad (3.19)$$

where \mathbf{f}_e contains the external forces acting on the node, such as gravity. Once the initial predicted positions are generated an iterative position correction process begins.

At the start of each solver iteration, the solver generates a full list of constraints to represent the interactions between nodes. These constraints restrict the allowable relative position of nodes. Bilateral constraints are present in the system at every time-step and every solver iteration. In [47] the bilateral constraints fix the distance between adjacent nodes. Thus, a flexible object, for example a piece of cloth, can be modelled with such a lumped parameter model. A collision detection algorithm is also run during each iteration to generate an extra set of constraints due to contacts that exist in the system's current predicted configuration. These are unilateral constraints because they are applied only in the case of contact. When a contact is detected, constraints are added to the system. The set (C_1, \dots, C_M) contains all the constraints - bilateral and unilateral - that exist after the collision detection phase within a solver iteration. When this method is used for modelling soil particles, as in [4], the only constraints in the system are the unilateral constraints that are created during the collision detection phase because soil particles only interact with each other through contacts. The structure of the constraints developed

in [4] that are used for the modelling soil particle contact will be described in detail in Section 3.2.2.

Once a full list of constraints is generated for the current solver iteration, the solver checks which constraints are violated in the position of the nodes predicted at the end of the previous iteration - or by Eq.(3.19) for the first iteration. Because the predicted node coordinates defined by Eq.(3.19) do not consider restrictions on motion imposed by the various constraints, the predicted system state is likely to be in violation of several constraints. For each of the constraints that are violated in a given solver iteration, the solver computes position correction vectors that will move the nodes into positions that do not violate the constraint. The position correction vectors are computed based on the constraint equations and the positions and masses of the nodes involved. The process for calculating this term will be detailed later in this section. Each of the constraints also has an associated scalar stiffness factor, k_{pb} , between 0 and 1 to represent the compliance of the constraint. The stiffness factor of a constraint is denoted here by k_{pb} to differentiate it from the stiffness coefficient of a mechanical spring, denoted by k , though later a relation between the two will be defined. This means that instead of applying the full calculated position correction, the solver displaces the nodes according to the relaxed position correction vector. A stiffness factor of 1 means that there is no relaxation. A problem noted in [47] is that the effect of the stiffness factor k_{pb} becomes non-linear after multiple solver iterations. The solution proposed in [47] is to adjust stiffness factors to account for the number of iterations that the solver is set to run for. The adjusted stiffness factor is given by:

$$k'_{pb} = 1 - (1 - k_{pb})^{1/n_s} \quad (3.20)$$

where n_s is the total number of solver iterations the solver will run for. Based on the stiffness factor and position correction induced by a constraint the solver generates a new predicted position for the node. In a Gauss-Seidel-like fashion, each constraint is resolved sequentially. Therefore, the position of node i at the end of solver iteration b is given by:

$$\mathbf{p}_i^b = \mathbf{p}_i^{b-1} + \sum_{j=1}^c (k'_{pb} \Delta \mathbf{p}_i^b)_j \quad (3.21)$$

where c is the number of constraints violated by node i during the solver iteration,

$(\Delta \mathbf{p}_i^b)_j$ is the position vector applied to particle/node i due to constraint C_j during solver iteration b , $(k'_{pb})_j$ is the adjusted stiffness factor associated with constraint C_j , and \mathbf{p}_i^{b-1} is the predicted node position from the previous iteration. Once all the new predicted node positions have been calculated, the next solver iteration begins. A new list of collisions, constraints, and constraint violations is generated based on the predicted node positions from the previous iteration and another position correction is applied to each node. The coordinates and velocity of each node at the end up the time-step, \mathbf{x}_i^+ and \mathbf{v}_i^+ respectively, are given by:

$$\begin{aligned}\mathbf{v}_i^+ &= (\mathbf{p}_i^{n_s} - \mathbf{x}_i)/h \\ \mathbf{x}_i^+ &= \mathbf{p}_i^{n_s}\end{aligned}\tag{3.22}$$

where n_s is the total number of solver iterations and $\mathbf{p}_i^{n_s}$ is the predicted position of node i given by Eq.(3.21) after the final position correction iteration.

Position Correction

The following is a description of the formulation given in [47] for calculating the position correction vector for a node/particle due to a single constraint. If we consider position correction iteration b , projecting of the position based on a constraint C_j means moving a node i to a physically permissible position through a position correction vector $(\Delta \mathbf{p}_i^b)_j = [\Delta p_{i1}^b, \Delta p_{i2}^b, \Delta p_{i3}^b]_j^T$. The magnitude of position correction is weighted according to the inverse mass of the nodes. If we define an array containing the current node position vectors $\mathbf{P}^{b-1} = [\mathbf{p}_1^{b-1}, \dots, \mathbf{p}_n^{b-1}]$ calculated in the previous iteration and assume that we have a constraint of the type $C_j(\mathbf{P}^{b-1}) = 0$ that is violated, then we are trying to recover a $\Delta \mathbf{P}_j^b = [\Delta \mathbf{p}_1^b, \dots, \Delta \mathbf{p}_n^b]_j$ such that:

$$C_j(\mathbf{P}^{b-1} + \Delta \mathbf{P}_j^b) = 0,\tag{3.23}$$

where $\Delta \mathbf{P}_j^b$ is an array containing the position correction vectors applied to each particle due to constraint C_j during solver iteration b . During the constraint projection phase, in which nodes are displaced within a time-step, it is important that the linear and angular

momentum of the system be conserved so as not to introduce any artificial forces. In [47] this is achieved by ensuring that the following equations are satisfied:

$$\sum_i m_i (\Delta \mathbf{p}_i^b)_j = \mathbf{0} \quad (3.24)$$

$$\sum_i \mathbf{d}_i \times m_i (\Delta \mathbf{p}_i^b)_j = \mathbf{0} \quad (3.25)$$

where \mathbf{d}_i is the distance between node i and some center of rotation for the system, and m_i is the mass of node. The change in the total linear and angular momentum of the particles in the system caused by their position correction vectors is given by Eq.(3.24) and Eq.(3.25), respectively. To ensure that these are satisfied, the position correction is restricted to be along the gradient of C_j [47]. The gradient operator, ∇ , is used to extract the direction of greatest change of the constraint function at a given point. The effect of the operator for a system with generalized coordinates $q_1 \dots q_n$ and basis vectors $\mathbf{e}_1 \dots \mathbf{e}_n$ is given by:

$$\nabla = \sum_{i=1}^n \mathbf{e}_i \frac{\partial}{\partial q_i} \quad (3.26)$$

By taking a Taylor series expansion of Eq.(3.23) and dropping the non-linear terms, Eq.(3.23) is approximated as [47]:

$$C_j(\mathbf{P}^{b-1} + \Delta \mathbf{P}_j^b) \approx C_j(\mathbf{P}^{b-1}) + \nabla C_j(\mathbf{P}^{b-1}) \Delta \mathbf{P}_j^b = 0 \quad (3.27)$$

where $\nabla C_j(\mathbf{P}^{b-1})$ is the gradient of constraint function C_j and is given by:

$$\nabla C_j(\mathbf{P}^{b-1}) = \left[\frac{\partial C_j}{\partial \mathbf{p}_1^{b-1}}, \dots, \frac{\partial C_j}{\partial \mathbf{p}_n^{b-1}} \right] \quad (3.28)$$

If the the position correction to be along the constraint gradient, the position correction can be expressed as [4]:

$$\Delta \mathbf{P}_j^b = \beta \mathbf{W} \nabla C_j(\mathbf{P}^{b-1})^T \quad (3.29)$$

where β is a scaling factor and $\mathbf{W} = \text{diag}(w_1, \dots, w_n)$ is the diagonal matrix which contains the inverse mass, w_i , of each node [47]. Substituting Eq.(3.29) back into Eq.(3.27) yields

a position correction vector for a single node due to constraint C_j given by:

$$(\Delta \mathbf{p}_i^b)_j = -s w_i \nabla_{\mathbf{p}_i^{b-1}} C_j(\mathbf{P}^{b-1}) \quad (3.30)$$

where C_j is the violated constraint, $\nabla_{\mathbf{p}_i^{b-1}} C_j(\mathbf{P}^{b-1})$ is the gradient of C_j with respect to \mathbf{p}_i^{b-1} , and the scaling term is given by:

$$s = \frac{C_j(\mathbf{P}^{b-1})}{\sum_{k=1}^N w_k |\nabla_{\mathbf{p}_k^{b-1}} C_j(\mathbf{P}^{b-1})|^2} \quad (3.31)$$

where N is the total number of nodes implicated in the constraint equation in question. For our purposes a single constraint doesn't involve more than two nodes so we would generally have $N = 2$. Eq.(3.30-3.31) are simply calculating the total scalar distance required to satisfy the constraint, $C_j(\mathbf{P}^{b-1})$, and applying a fraction of that correction to each node in the constraint gradient direction. The fraction of the displacement that each node takes is determined according to the relative mass of the nodes. Thus, the sum of the correction applied to all nodes will equal the total required correction. All the position correction terms calculated by Eq.(3.30) are substituted into Eq.(3.21) to get the new predicted node positions at the end of a solver iteration.

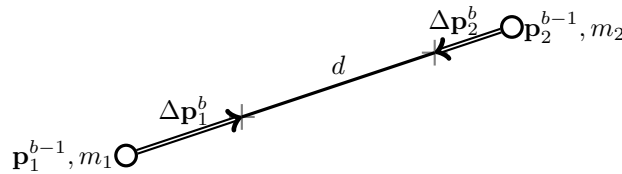


FIGURE 3.4: Distance constraint violation with weighted position correction

A simple example given in [47] is the position correction for a distance constraint that affects a pair of nodes with $m_1 < m_2$, as seen in Figure 3.4. The constraint can be expressed as

$$C_j(\mathbf{p}_1^{b-1}, \mathbf{p}_2^{b-1}) = \|\mathbf{p}_1^{b-1} - \mathbf{p}_2^{b-1}\| - d = 0 \quad (3.32)$$

where two nodes are constrained to be at a fixed distance d from one another. The gradient with respect to the first node is:

$$\nabla_{\mathbf{p}_1^{b-1}} C_j(\mathbf{p}_1^{b-1}, \mathbf{p}_2^{b-1}) = \frac{\partial (\|\mathbf{p}_1^{b-1} - \mathbf{p}_2^{b-1}\| - d)}{\partial \mathbf{p}_1^{b-1}} = \frac{\mathbf{p}_1^{b-1} - \mathbf{p}_2^{b-1}}{\|\mathbf{p}_1^{b-1} - \mathbf{p}_2^{b-1}\|} = \mathbf{n} \quad (3.33)$$

where the result defines the contact normal direction, \mathbf{n} . Similarly, they find that $\nabla_{\mathbf{p}_2^{b-1}} C_j(\mathbf{p}_1^{b-1}, \mathbf{p}_2^{b-1}) = -\mathbf{n}$. Solving Eq.(3.30) and (3.31) results in position correction vectors [47]:

$$\begin{aligned} (\Delta \mathbf{p}_1^b)_j &= -\frac{w_1}{w_1 + w_2} (\|\mathbf{p}_1^{b-1} - \mathbf{p}_2^{b-1}\| - d) \mathbf{n} \\ (\Delta \mathbf{p}_2^b)_j &= \frac{w_2}{w_1 + w_2} (\|\mathbf{p}_1^{b-1} - \mathbf{p}_2^{b-1}\| - d) \mathbf{n} \end{aligned} \quad (3.34)$$

where $w_1 > w_2$. It can be seen that the lighter node takes a greater percentage of the total displacement needed to correct the constraint violation. This example is in fact relevant for the soil simulation application of this method since, as will be explained in the next section, distance constraints similar to the one in Eq.(3.32) will be used to model part of the normal contact between particles.

3.2.2 Parallel Particles

The P² method developed by Holz [4] builds on the PBD framework and adapts it more specifically to the application of modelling soil particles. In this framework, the point mass nodes represent soil particles with spherical collision geometries. As is the case in the PBD method, these particles are defined by a position vector \mathbf{x}_i , a velocity vector \mathbf{v}_i and mass m_i , but now they have an additional property that is their radius r_i which is used for collision detection purposes. Some of the other key modifications made to the PBD method in [4] are:

- The sequential solver is replaced with a parallel solver that can process each particle individually, allowing for parallel computation of position corrections.
- A method to model a spring-damper element implicitly through constraints and position-corrections is included to represent the normal contact forces between colliding particles with mechanical stiffness and damping.
- A method to model Coulomb friction and proposed forces implicitly through constraints and position-corrections is included.

Parallel Solver

The parallelization of the position correction computation is done by independently computing the all position corrections induced by the constraints that a particle violates in any given solver iteration. Instead of applying each individual position correction to a particle, the total correction for a node, $\Delta \mathbf{p}_i^b$, is now calculated as the average of each individual position correction. Therefore, Eq.(3.21) used for calculating the predicted particle position at the end of a solver iteration is replaced with:

$$\begin{aligned}\Delta \mathbf{p}_i^b &= \frac{1}{c} \sum_{j=1}^c (k'_{pb} \Delta \mathbf{p}_i^b)_j \\ \mathbf{p}_i^b &= \mathbf{p}_i^{b-1} + \Delta \mathbf{p}_i^b\end{aligned}\tag{3.35}$$

where c is the total number of constraints that particle i is violating. Formulating the problem in this way allows for the new position of each particle to be solved for independently from the others. The parallelization is achieved by dividing the particles into separate threads that can then compute the position correction for particles at the same time. A given thread solves all its assigned particles one at a time, but the simple fact of being able to have multiple threads compute particles positions simultaneously speeds up the solver significantly [4]. After the position corrections have been applied to all particles, the solver repeats the entire position correction process for a fixed number of iterations, as was the case in the PBD framework.

Contact Constraints

In the PBD formulation, the list of constraints is made up mostly of bilateral constraints that are present at every time-step, with unilateral constraints being added to the list in the case of contacts [47]. In the P^2 formulation there are no bilateral constraints. Instead, at each time-step and during each solver iteration, a new list of constraints is generated based on which particles are in contact with each other or a rigid body. Contact detection is made simple by the spherical shape of the particles. A contact is detected if the sum of two particles' radii is larger than the distance between the two particles. If two particles

are in contact, a series of constraints representing various contact forces are generated, a position correction is computed for each constraint violated by a given particle, and the average of all the position corrections is applied to that particle - as described in the previous section. Figure 3.5 shows a simple position correction case. In this example we have detected a collision between two particles, and the solver has therefore generated a constraint $C(\mathbf{p}) = \|\mathbf{p}_1 - \mathbf{p}_2\| - (r_1 + r_2) = 0$. This constraint is violated and therefore the solver computes a position correction $\Delta \mathbf{p} = [\Delta \mathbf{p}_1, \Delta \mathbf{p}_2]$. As in PBD, the lighter particle is displaced more.

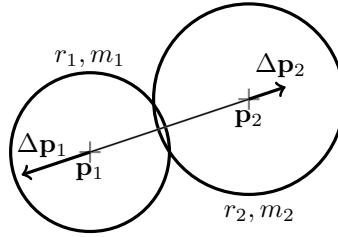


FIGURE 3.5: Constraint violation with weighted position corrections

In the P^2 framework, the magnitude of a position correction imposed by a constraint is still governed by Eq.(3.30) as in the original formulation, but the constraint equations take on a different form. The objective in designing the normal contact constraints is to define constraints whose resulting position correction will have the same effect as a spring-damper element in traditional DEM formulation. In [4] it is shown that the stiffness factor, k_{pb} , that is associated with a constraint in [47] can be correlated to the stiffness or damping coefficient in a contact model that uses a spring-damper element to model normal contact forces - such as in traditional DEM formulations [30] - with an implicit Euler integration scheme. This correlation will be important in later sections when we tune the model using the equivalent stiffness and damping coefficients that are associated with the constraint stiffness factors. To create a spring, we would want a constraint that restricts the relative position of two particles in the contact normal direction. To create a damper, we would want a constraint that restricts the relative velocities of two particles in the contact normal direction. The pair of constraints proposed in [4] to implicitly model a spring-damper element between two particles i and j during contact are formulated as:

$$C_s(\mathbf{p}_i, \mathbf{p}_j) = (\mathbf{p}_i - \mathbf{p}_j) \cdot \mathbf{n} - (r_i + r_j) = 0 \quad (3.36)$$

$$C_d(\mathbf{p}_i, \mathbf{p}_j) = ((\mathbf{p}_i - \mathbf{x}_i) - (\mathbf{p}_j - \mathbf{x}_j)) \cdot \mathbf{n} = 0 \quad (3.37)$$

where $\mathbf{n} = \frac{(\mathbf{p}_i - \mathbf{p}_j)}{|\mathbf{p}_i - \mathbf{p}_j|}$ is the unit vector in the contact normal direction. The constraint in Eq.(3.36) constraints the distance between two particles to be equal to the sum of their radii. Once Eq.(3.23) is applied to calculate the required position correction due to this constraint, a stiffness factor that we will refer to as k_{pb} is applied to the correction. Similarly, the constraint in Eq.(3.37) constraints the relative velocity in the contact normal direction between two particles to be equal to zero. Once Eq.(3.23) is applied to calculate the required position correction due to this constraint, a stiffness factor that we will refer to as c_{pb} is applied to the correction. A correlation between c_{pb} and k_{pb} and the equivalent stiffness and damping coefficients in the mechanical sense, c and k can be derived. This is done in [4] by considering a one-dimensional spring-mass-damper system and comparing the velocity at the end of a time-step if motion were governed by the P² method - in which case we have a mass subjected to two constraints described by Eq.(3.36-3.37) - to the velocity at the end of an implicit Euler integration of a spring-mass-damper system in one dimension with no gravity.

Spring-Mass-Damper System. In the form of Eq.(3.6), a one dimensional spring-mass-damper system in zero-gravity can be defined by the following equation of motion:

$$m\ddot{x} + c\dot{x} + kx = 0 \quad (3.38)$$

Holz [4] assumes that the position, x_k , and velocity, $v_k = \dot{x}_k$, of the mass element are known at time t_k and uses an implicit Euler scheme to determine the position and velocity at time-step t_{k+1} , where $h = t_{k+1} - t_k$. The resulting state at the end of the time-step is given by [4]:

$$\begin{aligned} v_{k+1} &= v_k - h \frac{c}{m} v_{k+1} - h \frac{k}{m} x_{k+1} \\ x_{k+1} &= x_k + h v_{k+1} \end{aligned} \quad (3.39)$$

P² System with Sprint-Damper Constraints. In [4], the same mass element with position x_k and velocity $v_k = \dot{x}_k$ at the start of the time-step t_k is analyzed again. The P² formulation is used to determine the state at time-step t_{k+1} . In this one dimensional

case, the initial predicted position of the particle is given by:

$$p^0 = x_k + hv_k \quad (3.40)$$

Two constraints would then be generated in the form of Eq.(3.36) and Eq.(3.37), respectively:

$$\begin{aligned} C_s(p) &= p^0 = 0 \\ C_d(p) &= (p^0 - x_k) = 0 \end{aligned} \quad (3.41)$$

Based on the same position correction formula given by Eq.(3.30) the position correction for the spring constraint will be $\Delta p_s = -p^0$ relaxed by factor k_{pb} , and the position correction for the damper constraint will be $\Delta p_d = (p^0 - x_k)$ relaxed by factor c_{pb} [4]. For a formulation with a single solver iteration this yields the following at the end of a time-step:

$$\begin{aligned} x_{k+1} &= p^0 - k_{pb}\Delta p_s - c_{pb}\Delta p_d \\ &= (1 - k_{pb} - c_{pb})p^0 + c_{pb}x_k \\ v_{k+1} &= \frac{x_{k+1} - x_k}{h} \end{aligned} \quad (3.42)$$

Solving these equations and performing a coefficient matching of the resulting equation with the result obtained in Eq.(3.39) yields the following relationship between the scaling factors and the mechanical stiffness and damping coefficients [4]:

$$\begin{aligned} k_{pb} &= \frac{h^2 \frac{k}{m}}{1 + h \frac{c}{m} h^2 \frac{k}{m}} \\ c_{pb} &= \frac{h \frac{c}{m}}{1 + h \frac{c}{m} h^2 \frac{k}{m}} \end{aligned} \quad (3.43)$$

As in [47], the constraint stiffness factors must be modified from what is defined in Eq.(3.43) in such a way that they become factors k'_{pb} and c'_{pb} that have the same effect but are independent of the number of solver iterations. In [4] this is achieved by writing

out the expression for the position correction after n_s solver iterations as:

$$\begin{aligned}
 p^{(0)} &= x_k + hv_k \\
 p^{(1)} &= (1 - k_{pb} - c_{pb})p^{(0)} + c_{pb}x_k \\
 &\vdots \\
 p^{(n_s)} &= (1 - k_{pb} - c_{pb})p^{(n-1)} + c_{pb}x_k
 \end{aligned} \tag{3.44}$$

Substituting everything into the expression for $p^{(n_s)}$, the position correction after n_s iterations, yields [4]:

$$f_{n_s}(k_{pb}, c_{pb}) = p^{(n_s)} = p^{(0)}\alpha^{n_s} + c_{pb}x_k \sum_{j=0}^{n_s-1} \alpha^j \tag{3.45}$$

where $\alpha = (1 - k_{pb} - c_{pb})$. From here the geometric series is replaced with $\frac{1-\alpha^{n_s}}{1-\alpha}$ [4]:

$$f_{n_s}(k_{pb}, c_{pb}) = p^{(0)}\alpha^{n_s} + c_{pb}x_k \frac{1 - \alpha^{n_s}}{1 - \alpha} \tag{3.46}$$

This represents the effective scaling factor generated by the k_{pb} and c_{pb} terms after n_s iterations. In order for the solution to be independent of the number of iterations, k_{pb} and c_{pb} must be replaced with terms k'_{pb} and c'_{pb} that are designed such that the overall corrected position is given by $p^{(n_s)} = (1 - k_{pb} - c_{pb})p + c_{pb}x_k$, as was the case in Eq.(3.42). In essence, the following equation must be satisfied [4]:

$$f_n(k'_{pb}, c'_{pb}) = p^{(0)}\alpha^{n_s} + c_{pb}x_k \tag{3.47}$$

It is found in [4] that this condition is satisfied if we use:

$$\begin{aligned}
 k'_{pb} &= 1 - c_{pb} \frac{1 - \alpha^{1/n_s}}{1 - \alpha} - \alpha^{1/n_s} \\
 c'_{pb} &= c_{pb} \frac{1 - \alpha^{1/n_s}}{1 - \alpha} \\
 \alpha &= (1 - k_{pb} - c_{pb})
 \end{aligned} \tag{3.48}$$

Therefore, if the collision detection determines that there is a collision between two particles then the constraint projections are applied and the resultant position correction

imposed on particle i by its collision with particle j is given by [4]:

$$\Delta \mathbf{p}_{i \leftarrow j} = k'_{pb} \Delta \mathbf{p}_{i,s} + c'_{pb} \Delta \mathbf{p}_{i,d} \quad (3.49)$$

where $\Delta \mathbf{p}_{i,s}$ and $\Delta \mathbf{p}_{i,d}$ are the position correction terms determined by plugging the respective constraint equations - Eq.(3.36) for the spring and Eq.(3.37) for the damper - into Eq.(3.30). Thus, by using a scaling factor that depends on both the desired equivalent real stiffness and damping effects and modifying it further based on the number of solver iterations being used for a given simulation, the real effect of a visco-elastic collision model between two particles is created. It is worth noting that all of the forces that are implicitly generated through the use of these coupled constraints are in the normal direction. Constraints designed to represent friction will act in the tangential direction.

Friction

The next main element in [4] is that constraints are developed to model Coulomb friction forces acting on particles that are in contact. The standard Coulomb friction law can be given by:

$$|\mathbf{F}_t| \begin{cases} \leq \mu_s |\mathbf{F}_n| & \text{if } |\mathbf{v}_\perp| = 0 \\ = \mu_k |\mathbf{F}_n| & \text{if } |\mathbf{v}_\perp| \neq 0 \end{cases} \quad (3.50)$$

where \mathbf{F}_n is the normal contact force vector, \mathbf{F}_t is the tangential friction force vector and \mathbf{v}_\perp is the relative tangential velocity vector. In this representation of friction, the friction force is related to the normal force, opposes the relative motion in the tangent direction, and is capped. To model this, the following damper constraint is proposed [4]:

$$C_f(\mathbf{p}_i, \mathbf{p}_j) = \mathbf{v} \cdot \mathbf{t} = 0 \quad (3.51)$$

$$\mathbf{v} := (\mathbf{p}_i - \mathbf{x}_i) - (\mathbf{p}_j - \mathbf{x}_j)$$

where \mathbf{t} is the unit vector in the contact tangent direction. The resultant position correction that comes from applying Eq.(3.30) to this constraint is calculated, but instead of being directly applied to the particle position or subjected to a stiffness factor like the normal contact constraints, this constraint's position correction is capped based on

the friction coefficient and the position correction in the normal direction computed by Eq.(3.49). The constraint in Eq.(3.51) requires that the relative tangential velocity between the two particles be zero. The direction of motion along the contact tangent is given by $\mathbf{t} = \frac{\mathbf{v}_\perp}{|\mathbf{v}_\perp|}$ where the tangential component of the velocity is given by $\mathbf{v}_\perp = \mathbf{v} - (\mathbf{v} \cdot \mathbf{n})\mathbf{n}$ - with \mathbf{n} being the unit vector in the contact normal direction. If this relative tangential velocity is zero, then there is no position correction induced by the friction constraint. If, however, the relative tangential velocity is non-zero, then the position correction imposed on particle i by its contact with particle j is [4]:

$$\Delta \mathbf{p}_{i \leftarrow j} = \frac{w_i}{w_i + w_j} \epsilon \mathbf{t} \quad (3.52)$$

where ϵ is the total position correction required to fully satisfy the constraint in Eq.(3.51) and the previously defined unit tangent vector \mathbf{t} is also the constraint gradient direction, $\nabla_{\mathbf{p}_i} C_f$. Since we want $C_f(\mathbf{p}_i, \mathbf{p}_j) = 0$ to satisfy the constraint, if the actual value of $C_f(\mathbf{p}_i, \mathbf{p}_j)$ is non zero we can say that $\epsilon = C_f(\mathbf{p}_i, \mathbf{p}_j)$. In the the same way that standard Coulomb friction models caps the tangential friction force as a function of the normal force using the friction cone, the tangential position correction is applied as a capped displacement ϵ_f based on the normal position correction ϵ_n which is determined by the normal contact constraint applied in Eq.(3.49). This yields an overall position correction caused by the contact between two particles that combines friction and normal contact forces which is expressed as [4]:

$$\Delta \mathbf{p}_{i \leftarrow j} = \frac{w_i}{w_i + w_j} (\epsilon_f \mathbf{t} + \epsilon_n \mathbf{n}) \quad (3.53)$$

where

$$\epsilon_f = \begin{cases} \epsilon & \text{if } |\epsilon| \leq \mu_s |\epsilon_n| \\ \text{sgn}(\epsilon) \mu_k |\epsilon_n| & \text{otherwise} \end{cases} \quad (3.54)$$

Adhesion

In [4] the change in velocity resulting from a position correction is analyzed to develop a relation between the position correction ϵ applied to a particle i with mass m_i and the equivalent constraint force which is implicitly induced by the position correction. This

relation can be written as [4]:

$$\lambda = \frac{m}{h^2} \epsilon \quad (3.55)$$

where λ is the equivalent constraint force induced by a position correction of ϵ , and h is the length of the time-step. In order to apply adhesion to the particle interaction, the adhesive force is expressed as [4]:

$$F_a = aA_c$$

$$A_c = \pi r_{min}^2 \left(1 - \left(\frac{\zeta}{r_{min}} - 1 \right)^2 \right) \quad (3.56)$$

where a is the adhesion coefficient, A_c is contact surface area of the two particles, ζ is the penetration, r_{min} and the smaller of the two colliding particle radii, and F_a is the adhesive force. The adhesive force is then converted to an equivalent position correction ϵ_a using Eq.(3.55). The adhesion gets applied differently in both the normal and tangential directions. In the normal direction it is applied by creating an un-relaxed damper constraint in the form of Eq.(3.37) and capping the displacement allowed by the constraint at ϵ_a . In the tangential direction, ϵ_a is added to the cap of the friction force in Eq.(3.54) [4].

The relation given in Eq.(3.55) is used both for calculating the position correction induced by adhesive forces, and for calculating the force that must be applied to other rigid bodies in the simulation that interact with the P² particles. When a rigid body in the system collides with a particle, constraints are generated and the position correction terms are computed as usual. The difference being that while the particle will feel the contact force implicitly as a result of position correction, the calculated position correction that would be applied to the body is instead converted to an external force to be applied to the body through Eq.(3.55) and added as an external force acting on the body in Eq.(3.6).

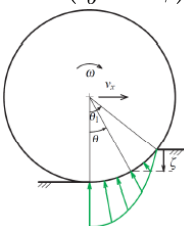
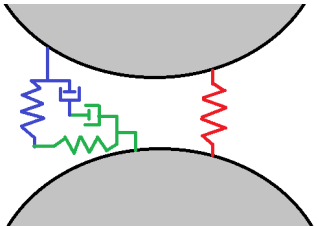
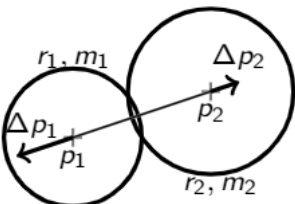
3.3 Terramechanics Model Comparison

The methods outlined in the previous sections all come with their own advantages and drawbacks. What was ultimately used to build the wheel-soil model for this project was, as previously stated, an implementation of the P² method. The software used is

Vortex and it is a product developed by *CM Labs* for the purpose of creating simulation environments. The two main alternatives to the P^2 method that is implemented in this work are variations of the semi-empirical Bekker model [5] [20] [21] and variations of a traditional DEM method [3] [30] where particles are treated as rigid bodies with orientation. A more detailed description of the formulations for the later two methods can be found in Appendix A, but a survey of Table 3.1 will help to compare and contrast these methods.

The Bekker method and its variants, with all the assumptions they carry, are very fast. Because they use a simple equation to relate pressure to sinkage, they have a relatively small computational cost associated with them. This is of particular importance when simulations need to be able to run in real time. With proper tuning of the coefficients in the pressure sinkage equation, these methods can give good results within the limited scope for which the coefficients are were calculated and the equations have been proven to be reliable. As previously mentioned though, the results are only reliable in the scope of the experiments that were used to tune the parameters of the formulation, which makes it hard to account for something like behaviour in reduced gravity without doing expensive additional testing.

TABLE 3.1: Comparison of Soil Models

Bekker Method	Traditional Discrete Element Method	Parallel Particles
Bekker, 1956	Cundall and Strack, 1979	Holz, 2014
Semi-Empirical	Numerical	Numerical
Physics-based equations relate soil pressure to wheel sinkage. Equations include coefficients that are determined experimentally. $\sigma = \left(\frac{k_c}{b} + k_\phi \right) \zeta^n$	Soil modelled as discrete bodies. Contact modelled using combinations of springs, dampers and friction models	Soil modelled as discrete particles. Contact forces modelled implicitly using constraints and position correction
		

The traditional DEM methods are on the other side of the spectrum. They can provide more details but simulations can take days or even weeks to run. The fact that this method models individual particles makes the study of soil velocity fields straightforward. This method also makes it easy to study the effect of gravity because it is easy to directly change the external force that gravity applies to each discrete element.

The P^2 approach - summarized in Figure 3.6 for a simple case of two colliding particles - is a version of DEM that retains many of the advantages of the traditional DEM. Gravity can be directly modified and soil velocity field is directly observable. However, there are key changes to the model. The model implicitly models contact forces through position correction and uses an iterative solver to account for coupling of the interactions. The illustrations of the DEM and P^2 contact models in Table 3.1 illustrate this difference between the methods. Additionally, particle orientation is not considered in this scheme. These features allow for reduced computational cost to the point where it has been used for real-time simulation of soil-excavation [4]. As of now, however, there has not been much research into how applicable this model is to the study of wheel-soil interaction - either in real-time or otherwise. This work explores non-real-time applications for using the P^2 method to model wheel-soil interaction. If successful, the P^2 method has the potential to be a much faster discrete element method for studying the general effect of gravity on wheel-soil interactions.

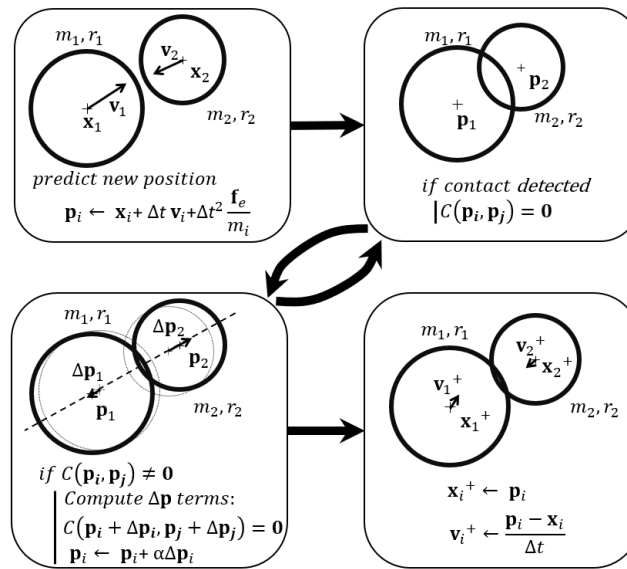


FIGURE 3.6: Simplified P^2 Formulation

Chapter 4

Reduced Gravity Experiments

The main goal of this work is to attempt to build a model that can reproduce results gathered during a series of reduced gravity experiments that were conducted by a Concordia research team [1]. A full description of the experiments, as well as other work done to create a semi-empirical wheel-soil model to reproduce the results, can be found in [50] and [1]. Some of the important aspects of the setup that are necessary for the model constructed in Chapter 5 are outlined here.

4.1 Single Wheel Testbed

Let us begin with the mechanism as a whole. The testbed is an automated robotic mechanism and can be seen in Figure 4.1 with its key components highlighted. Three motors and actuators control the motion of the wheel during the experiments. First, a linear actuator controls the horizontal motion of the wheel. The control of this actuator is paired with that of the wheel motor, which controls the angular velocity of the wheel and records motor current and torque via a torque sensor. A vertical load is applied to the wheel by pneumatic actuators while allowing for the wheel to translate freely in the vertical direction. The motor current of the linear actuator is recorded and in post-processing is converted to an applied force and the vertical position of the wheel is

also recorded to measure sinkage [1]. The testbed allows for mounting of different wheels. Details of the wheel that was used in these experiments will be discussed in Section 4.1.1.

The sandbox itself contains 80kg of the ES-2 martian soil simulant, the properties of which will be discussed in Section 4.1.2 [1]. As seen in Figure 4.1, the sandbox has one wall that is made of glass so that the - with the help of a properly positioned mirror - a high-speed camera can capture video images that are later processed to analyze the soil velocity field through a visualization process described in [51].

Another important feature of the testbed is its soil preparation mechanism. The soil preparation process was designed in [52] and was shown to create similar soil conditions for each run through repeatability of the experiments during the reduced gravity flights [1] [52]. The process involves covering the sandbox, blasting the soil compressed air to loosen the soil and vibrating the box to level the soil [52].

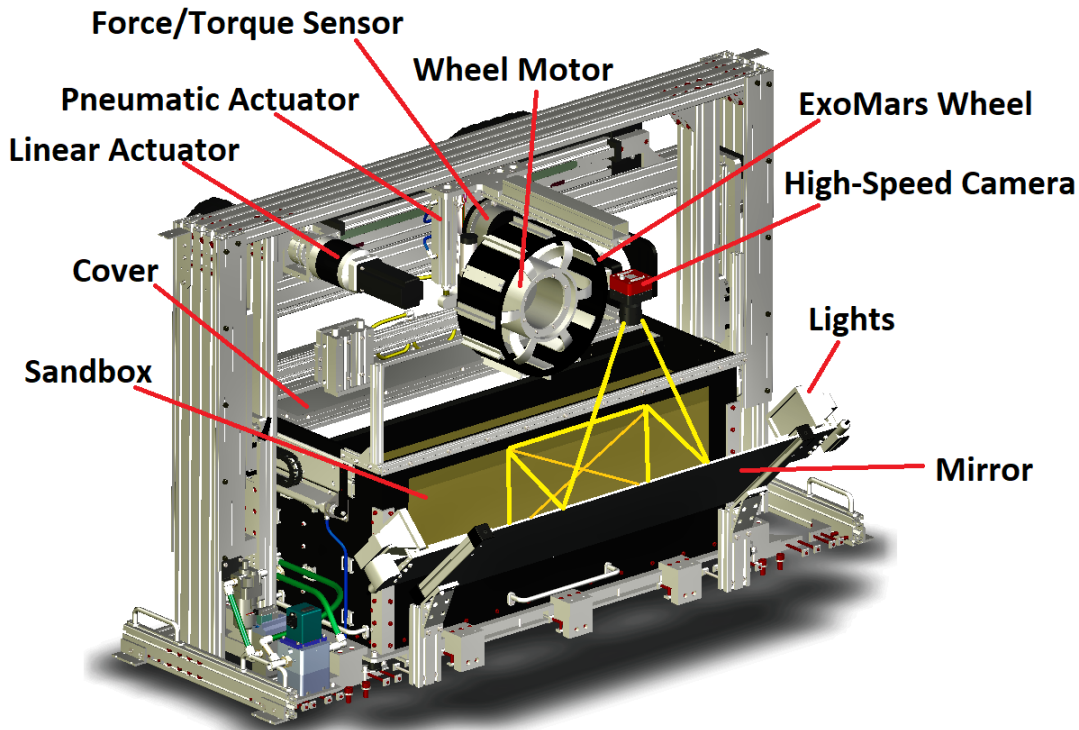


FIGURE 4.1: Illustration of the Single Wheel Testbed

4.1.1 ExoMars Wheel

The wheel that was mounted in the testbed is a prototype of the ExoMars rover wheel, designed by ESA. All known design parameters of the wheel are listed in Table 4.1. As can be seen in Figure 4.2, the wheel has a very flexible outer ring, with a stiffer inner portion. This outer portion of the wheel compresses almost completely when the load of the rover is applied to it - resulting in a much larger contact patch with the soil that can be seen in Figure 4.3. The flexibility of the wheel and the shape of the contact patch as determined in [50] will play an important role in the modelling of the wheel that will be discussed in Chapter 5. As mentioned, the wheel is equipped with a motor to drive it and a force/torque sensor to measure the wheel outputs. In addition, a potentiometer tracks the vertical position of the wheel hub [1]. Thus, the total mass of the wheel assembly is significantly larger than the mass of the wheel on its own.

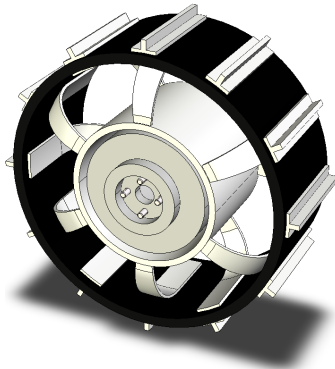


FIGURE 4.2: 3D Rendering of ExoMars Wheel



FIGURE 4.3: Compliance of ExoMars Wheel

TABLE 4.1: ExoMars Wheel Dimensions

Parameter	Units	Value
Diameter	[mm]	285
Mass	[kg]	1.743
Mass (with sensors)	[kg]	9.6
Width	[mm]	120.8
Grousers	[#]	12
Grouser size	[mm]	$25L_g \times 5t_g \times 13H_g \times 110W_g$

4.1.2 ES-2 Soil

A few ES-2 soil parameters were available to help guide the tuning of particle parameters, these are summarized in Table 4.2. However, soil parameters are very sensitive to things that are very hard to measure or predict, such as humidity, temperature, moisture, etc. [53] [54] [55] The exact temperature and moisture content at the time of the experiments is not fully known. This is one reason that the theoretical soil parameters provided are not enough to select the appropriate simulation parameters. Another reason for not being able to accurately set the particle parameters directly without a tuning process is that the particle parameters available for tuning do not all directly correlate to the physical parameters given in Table 4.2.

TABLE 4.2: ES-2 Soil Parameters

Parameter	Units	Value
Dry Bulk Density	$[kg/m^3]$	1450 ± 25
Cohesion	$[kPa]$	0.75 ± 0.75
Internal Friction Angle	$[^\circ]$	37 ± 5
Grain Sizes	$[\mu m]$	30-100

4.2 Reduced Gravity Flights

In order to study soil behaviour in reduced gravity, the entire testbed described in Section 4.1 was loaded onto a Falcon-20 aircraft at the National Research Council of Canada (NRC) which followed a flight profile similar to the one illustrated in Figure 4.4 [1]. This flight pattern is a common method to achieve a reduced or zero-g effect. The key part of the flight is the parabolic part of the trajectory at the top of which the reduced-g effect can be felt. The aircraft, after flying in steady flight for long enough to prepare the next run of the experiment, enters a 2-g pull-up maneuver to enter the parabola and begin its ascent. At the peak of the parabola - as the aircraft turns down towards the Earth - a period between 20 and 30 seconds of reduced or zero gravity is experienced. The degree to which gravity is effectively reduced is determined by the pitch and speed of the aircraft [1]. During the period of reduced gravity, one experiment is carried out. The aircraft would then begin its descent and pull out of the parabolic maneuver by

way of a 2-g pull-out. The aircraft would then return to level flight, during which time a repeatable soil-preparation mechanism was activated to reset the wheel and soil in order to be ready for the next parabola with the same initial conditions [52]. Each flight consisted of between 5 and 10 parabolas, after which maintenance was required on the testbed before the next flight - for a total of 16 parabolas [1].

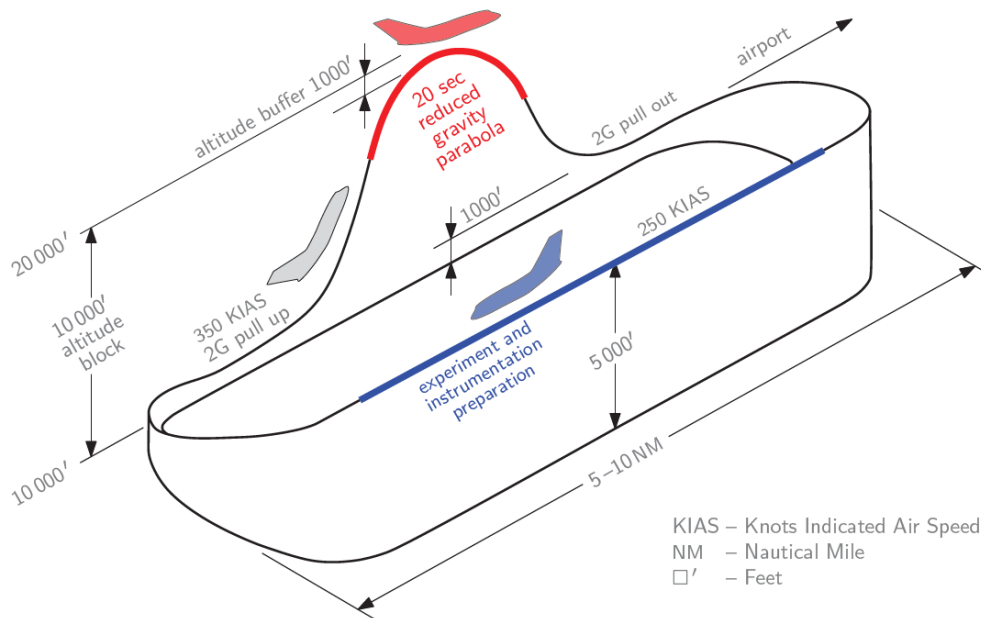


FIGURE 4.4: Profile of a typical reduced gravity flight [1]

4.3 Input Parameters

Various runs of the experiment used different experimental parameters in order to gather a wide range of data on the effect of gravity on a wheel driving over soft soil. Two actuators control the motion of the wheel in the testbed - a horizontal linear actuator and a wheel motor. By keeping the rotational speed of the wheel such that there is a constant rim speed of $r\omega = 20\text{mm/s}$ and varying the translation speed, different wheel slip values can be targeted. A pneumatic cylinder applies a vertical load on the wheel. This load is calibrated in different gravity levels such that the total load (including the mass of the wheel) was at a constant value for all gravity levels [1].

4.3.1 Wheel Load

As described in Section 4.1, a wheel load, W , was applied to the wheel during the experiments to reproduce the normal load that a wheel might feel due to the mass of the rover that it is attached to. The total wheel load accounts for both the weight of the wheel apparatus, given in Table 4.1, and the vertical load applied by the pneumatic cylinders. The load was set to be either $164N$ or $225N$, depending on the experiment run. The cylinder was programmed to keep the wheel load constant in all gravity conditions by taking into consideration the effect of reduced gravity on the wheel and applying increased load in reduced gravity conditions such that the total load, W , is consistent across all tests in Earth, martian and lunar gravity. The reduction in gravitational force acting on the wheel is directly compensated for by increasing the force that is applied by the cylinder so that the overall load remains the same. Thus, the load applied by the cylinder can be expressed as [1]:

$$W_{pn} = W - m_w g \quad (4.1)$$

where W_{pn} is force that is applied by the cylinder, W is the desired overall load, m_w is the total mass of the wheel apparatus (including instrumentation) and g is the acceleration due to gravity. It is pointed out in [1] that the load applied by the cylinder was not in reality perfectly constant during testing, but rather periodic with an average applied force that is consistent with the desired total loads of either $164N$ or $225N$.

4.3.2 Wheel Slip Ratio

Wheel slip is an important measure of wheel performance because of its close relation to the amount of traction the wheel is able to develop under the given conditions [5]. Good performance can be measured by how little slip is required given the amount of traction. Conversely, maximizing traction for a preset slip-controlled test is an equivalent measure of performance - as was the case in these experiments. The wheel slip ratio is defined as:

$$s = \frac{r\omega - v_x}{r\omega} \times 100\% \quad (4.2)$$

where v_x is the translational velocity of the wheel hub, r is the undeformed radius of the wheel, ω is the angular velocity of the wheel, and together $r\omega$ is the rim speed of the wheel (kept at a constant 20mm/s). The tests carried out in [1] were slip-controlled and measured traction for various slip conditions that were varied between 10% and 70% in increments of 10%. As previously mentioned, and as seen in Figure 4.3, the wheel is deformable. It could therefore be argued that the wheel slip ratio might be not be trivial to accurately determine. However, since the rim of the wheel is made of steel it is assumed in [1] that the overall perimeter, and thus the rim speed given by $r\omega$, does not change significantly when the wheel is deformed. Therefore, it is assumed that the slip definition given by Eq.(4.2) still holds for the deformable wheel. This is supported by Wong in [9] where they say the value of r used for Eq.(4.2) for a deformable wheel should be the rolling radius which is taken as $r_r = \frac{p}{2\pi}$ where p is the distance that the wheel hub would travel in one wheel revolution with zero wheel slip. Therefore, assuming that the perimeter of the steel wheel is relatively unchanged when the wheel deforms, the definition for wheel slip ratio given in Eq.(4.2), taking r to be the radius of the undeformed wheel, holds true.

4.3.3 Gravity

Since the reduction of gravity was also something that could be controlled by altering the profile of the flight, the runs were divided between Martian and Lunar gravity - approximately a third and a sixth of Earth gravity, respectively. There were also tests run on the ground at full Earth gravity. Table 4.3 illustrates the number of experiments that were carried out in [1] for each combination of the wheel load, wheel slip, and gravity input parameters. Due to the limited flight time available to conduct partial gravity experiments (PGEs), the only experiments that were repeated were the 164N load and 20% slip cases for both Lunar and Martian gravity.

TABLE 4.3: Number of experiments run in [1] for various combinations of wheel load, wheel slip ratio and gravity

Slip (%)	Wheel Load (N)	Number of Tests Cases		
		Earth-g	Martian-g	Lunar-g
10	225	1	1	0
20	225	1	1	0
30	225	1	1	0
10	164	1	1	1
20	164	3	3	2
30	164	1	1	1
40	164	1	1	1
70	164	1	1	1

4.4 Outputs

As discussed, the testbed was equipped with instrumentation to measure various outputs that are of interest. These were described in more detail the Section 4.1. A high-speed camera captured images of the soil that were later used to analyze the differences in the velocity field of the soil as the wheel drives over it in both reduced gravity and Earth gravity. The force applied by the linear actuator to maintain the constant horizontal velocity is also recorded by converting the motor current into an applied force. Finally, the position of the wheel hub is tracked via potentiometer in order to measure wheel sinkage.

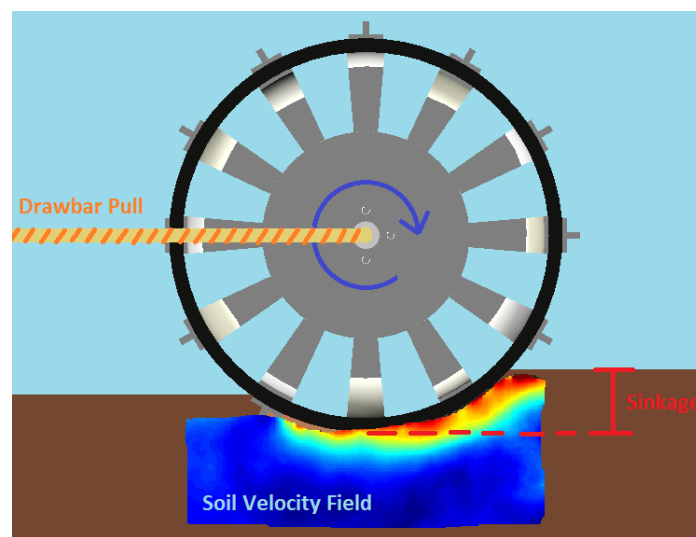


FIGURE 4.5: Illustration of measured outputs

4.4.1 Drawbar Pull

Drawbar pull (DP) is a common indicator of the traction being generated by the wheel under given conditions. It is defined as:

$$DP = T - \sum R \quad (4.3)$$

where T is the total thrust generated and $\sum R$ is the sum of resistive forces from the soil acting on the wheel. In the slip-controlled experiments the DP is equal to the force that the linear actuator applies to maintain the constant horizontal velocity of the wheel. All other conditions being the same, increased sinkage often has the effect of decreased DP because the extra soil surface in contact with the wheel results in more resistive force which in turn results in lower DP as per Eq. (4.3). Similarly, increased soil strength with all other conditions the same will result in lower DP for the same reason. It is of course possible for both sinkage and/or soil strength to increase along with DP, but this would usually be the result of multiple conditions changing. For example, the experiments in [1] show that we will see an increase in both sinkage and DP as the wheel slip ratio is increase from 10% to 70%. When comparing the results of different experiments, drawbar pull weight ratio, DP/W , is often used for comparison in place of directly comparing the DP values. This quantity how much a vehicle can tow relative to its own weight. In the context of the experiments conducted in [1], it is equal to the DP divided by the total wheel load. Due to the previously discussed periodicity of the wheel load, W , this was calculated as the average DP divided by the average measured W for each experiment [1]. Since it has been shown in [56] that DP/W is approximately constant within the operating conditions of a wheel, comparing the DP/W ratio rather than directly comparing DP between tests can lead to a more meaningful comparison. An assumption in the simulation model that will be presented in Chapter 5 is that applying a constant W in line with the average experimental W rather than trying to replicate the time-varying W that was applied in experiments would be acceptable. The finding in [56] supports this assumption as we can assume that the DP/W at the peaks and valleys of the real applied load would still result in a relatively constant DP/W . Thus, analysis of DP/W rather than DP also makes sense for the purposes of comparison between the experiments and simulations.

4.4.2 Sinkage

The sinkage is another measure that affects wheel performance. It represents how much the wheel sinks into the soil. Sinkage increases with slip and can result in higher resistive forces from the soil, but also higher traction. Higher sinkage at the same slip value is likely to result in reduced DP for the reasons discussed in Section 4.4.1. However, when we have increased sinkage caused by increased slip, the result tends to be higher DP - as will be seen in the results section. The sinkage was measured as the vertical displacement of the wheel hub relative to the soil surface by a potentiometer mounted on the wheel apparatus. Since the wheel was flexible, the wheel hub displaces vertically both due to sinkage and deformation of the wheel. However, for a given loading condition, the deflection of the wheel was assumed to be constant and the amount of displacement attributed to the deformation of the wheel is subtracted from the overall measurement [1].

4.4.3 Images

As discussed, a high-speed camera captures video footage of every experiment and processes them through a procedure presented in [51]. This process outputs a detailed map of the magnitude and direction of the individual soil particles under the wheel as it drives. This allowed for direct comparison of soil flow at equivalent points in the trials for various test conditions. Following in the footsteps of the experiments carried out by Nakashima [43], being able to qualitatively observe the increase flow in the soil in reduced gravity is an important part of the experiments.

4.5 Conclusions of the Experiments

Niksirat et. al. [1] draw many conclusions about the effect that varying gravity has on the various outputs. These trends will form the basis for the initial validation of the applicability of the P² simulation model in Chapter 6.

4.5.1 Effect of gravity on DP/W

Niksirat et. al [1] showed that the DP/W value was, on average, 20% lower in lunar gravity when compared to Earth gravity. For Martian gravity, the overall DP/W reduction was slightly smaller - 6% for the 164N loading case, 14% for the 225N loading case and 8% if all cases are combined. While these changes between Earth and Martian gravity were quantifiable, they were deemed not statistically significant for Martian/Earth comparison. However, the fact that reduction in DP/W was observed both in Martian gravity and, to a larger degree, in Lunar gravity suggests that DP and gravity are related. The study concludes that reduction in DP/W indicates reduced soil strength in lower gravity [1].

4.5.2 Effect of gravity on sinkage

The results found in [1] for the effect of gravity on sinkage were far more conclusive in all gravity levels. They found that, on average, there was a 38% increase in sinkage in Lunar gravity compared to Earth gravity, an increase of 27% when comparing the Martian and Earth gravity tests for 164N load and a 47% increase when comparing the Martian and Earth gravity tests for 225N load.

4.5.3 Effect of gravity on sensitivity to loading

Another observation made in [1] is the fact that changing gravity has an effect on how much certain outputs are affected by increased wheel loads. This was especially apparent for the sinkage output. When load was increased from 164N to 225N in Earth gravity, the sinkage increased by an average of 14%. Meanwhile when the same load increase was tested in Martian gravity it lead to an average sinkage increase of 31%, suggesting that reduced gravity leads to an increased sensitivity of sinkage to wheel load [1].

4.5.4 Effect of gravity on velocity fields

Unlike DP and sinkage, the velocity field of the soil is less simple to quantify for each experiment. However, a simple qualitative analysis and comparison of the soil velocity field images captured in [\[1\]](#) can help to characterize the effect that gravity has on the velocity and flowability of the soil. The comparison shows beyond a doubt that there is increased flow in the soil in reduced gravity, consistent with the theory put forth in [\[43\]](#) that soil strength is reduced in lower gravity.

Chapter 5

Dynamic Model and Simulation Setup

Vortex is used to model both the rigid bodies and particles in this work. The particle formulation described in Chapter 3 is fully implemented in the latest version of *Vortex* and is used for real-time simulation of rigid body/particle interaction in the context of soil excavation maneuvers. In that context it is acceptable to use particles and time-steps that are large enough to simulate realistic-looking soil behaviour in real-time. In the set-up described in this section we will detail how this particle system was pushed beyond the scope of what it was originally implemented for to run simulations at much smaller time-steps, with much smaller particles, and with the ultimate goal of creating a model that is validated not by how realistic the behaviour appears to be visually but by how well it can match experimental data and trends.

5.1 Mechanical Components

While the behaviour of the particles is the main focus of study, on their own they cannot be used in any useful simulation without the ability to interface with other types of elements - in this case mechanical system models generated using rigid bodies. The physical maneuver being simulated is actually fairly simple: a single rover wheel driving

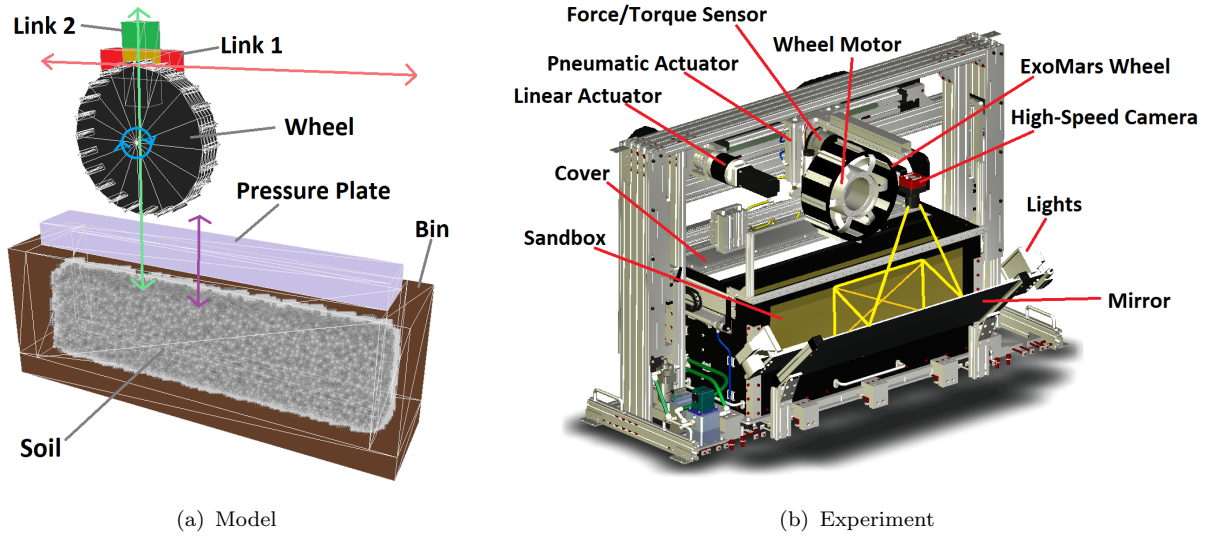


FIGURE 5.1: Mechanical model (left) vs Experimental testbed (right)

in a straight line over soft soil. However, this still requires a mechanism to mount it onto the testbed. Figure 5.1(b) illustrates the mechanical system being modelled and Figure 5.1(a) illustrates the dynamic model that was created to represent the mechanical system. This section will discuss the geometry and purpose of each element of the model and Section 5.2 will go into detail on how the elements interact through constraints like those described in Section 3.1 in order adequately capture the motion of the mechanical system.

Wheel. The wheel itself is perhaps the most complex element to model. This is because the real ExoMars rover wheel that was used in the experiments that are used to validate the simulation is flexible. Were it not for the flexibility of the wheel, most of the parameters listed in Table 4.1 would be directly applied to create a collision geometry for the rigid body wheel used in simulation. Because it can be very difficult to model the flexibility of the wheel directly, a technique often used in the literature is the determination of an effective wheel radius. This effective radius, which is larger than the undeformed wheel radius, is then used in calculations in place of the real radius because it allows the model to more accurately capture the true shape of the contact patch. This technique was originally suggested by Bekker [5] and continues to be used today so that terramechanics models designed for rigid wheels can be used to model the motion of flexible wheels. Many other works have developed other methods for determining more complex shapes for the

contact patch such as elliptical [57], parabolic [58] and exponential [59] approximations. In [50], still images captured during the experiments described in [1] are analyzed using Taubin's method [60] to determine an appropriate effective radius. This effective radius was used in [50] to model the experiments using a semi-empirical terramechanics model that assumes rigidity of the wheel. The analysis in [50] found that the radius of the contact patch is approximately double the radius of the uncompressed wheel. In order to use a rigid body to represent the wheel in the model in this work, the same effective radius technique was used to more accurately capture the shape of the contact patch. With the underlying assumption that a rigid wheel would have to be made twice as large to have a contact patch that is similar to that of the wheel in the experiments, the other parameters given in Table 4.1 had to be altered accordingly. The radius, as mentioned, was doubled - and so it followed that the number of grousers should also be doubled so as to maintain the same frequency of grousers around the perimeter of the wheel. What remained constant were the overall mass of the wheel, the width of the wheel, as well the dimensions of the grousers. The collision geometry for the wheel is a compound geometry made up of standard cylinder and rectangular shapes positioned to capture the shape of the wheel and grousers. While this collision geometry is active, in the context of the simulation the only elements that the wheel will come into contact with are the particles.

Link 1. This link, depicted as a red box in Figure 5.1(a), is a body with no active collision geometry. It is essentially used as a moving frame of reference. It is constrained such that it can only move at a fixed velocity in the horizontal direction relative to the inertial reference frame, or the "ground" as it can be referred to. This axis of movements is shown as a red line in Figure 5.1(a). It is necessary for simulation purposes and does not directly represent any real part of the experimental testbed. As such, its mass is set to be negligible so as not to have an impact on the simulation outputs. It is simply necessary so that the vertical axis constraining the motion of the load moves laterally as the wheel moves forward. Physically, it represents the slider of the linear actuator shown in Figure 5.1(b).

Link 2. Link 2 is depicted as a green box in Figure 5.1(a). Like link 1, it has no active collision geometry but unlike link 1, it does represent real components in the system. Physically, it represents the instrumentation attached to the wheel in the assembly shown

in Figure 5.1(b). It has a mass of $7.857kg$ which, when combined with the mass of the wheel itself, results in a wheel assembly with the same total mass as the instrumented wheel described in Chapter 4. Through a prismatic constraint designed to mimic the motion induced by the pneumatic actuators in the real system seen in Figure 5.1(b), the motion of link is restricted to be in the vertical direction relative to link 1. This axis of motion is shown as a green line in Figure 5.1(a). Link 2 is also connected to the wheel through a revolute joint that allows the wheel to rotate about an axis in the body-fixed frame of link 2 that is orthogonal to the axis of motion of the aforementioned prismatic joints.

Bin. The main function of the bin is to act as the domain in which the simulation is carried out. The bin is simply a container for the particles to settle in. It is made up of five rectangular collision geometries combining to form a single rigid body, each enforcing one of the limits of the domain with the top remaining open. Unlike the two links, this element's contact geometry is active, however, in the context of this model the only elements it will come into contact with are the particles. It is shown in brown in Figure 5.1(a). The bin is constrained to be static at all times and so there are no constraints applied to it.

Pressure Plate. The final component of the mechanical system is a plate that is used to compress the particles before the wheel is engaged. It has a rectangular collision geometry, illustrated in purple on Figure 5.1(a) and has the same length and width as the opening at the top of the bin. A prismatic joint allows only for motion in the vertical direction. The axis of this motion is shown in dark purple in Figure 5.1(a). The purpose of this component is to apply a compaction pressure to the particles before the wheel is engaged just as the soil in the experiments is compacted before the experiments are carried out.

5.2 Constraints

A subset of the mechanical model is the set of constraints that are used to move the wheel during simulation. There are only a few constraints needed to produce the desired motion. Three constraints control the motion of the wheel, another is used to apply pressure to

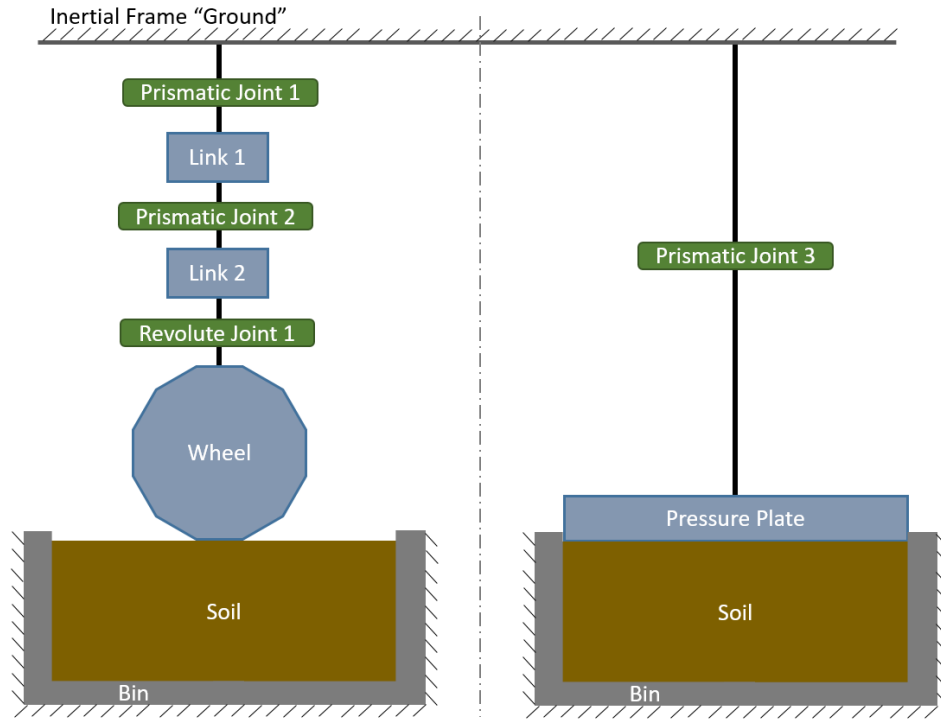


FIGURE 5.2: Kinematic diagram of the mechanical system

the soil in the soil preparation phase of the simulation - which will be discussed in Section 5.3.1 - and another set fix the bin to remain static.

The schematic shown in Figure 5.2 illustrates how the five bodies in the mechanical model interact with each other through constraints. In Section 3.1 we discussed how bilateral constraints are added to a dynamic system to control to relative motion of the various elements of the simulation. It was shown that the relation $\mathbf{M}\dot{\mathbf{v}} + \mathbf{c} = \mathbf{f} + \mathbf{A}^T \boldsymbol{\lambda}$ combined with the constraint equations $\mathbf{A}\mathbf{v} = \mathbf{u}$ could be used to calculate the unknown constraint forces $\boldsymbol{\lambda}$ if the constraint velocities \mathbf{u} are specified. There are some constraints in this model that will have capped constraint forces, meaning that the equalities shown above will become inequalities and form a Linear Complimentary Problem (LCP). The *Vortex* solver handles these constraints by asking for a desired constraint velocity as well as a maximum/minimum constraint force, λ_{max} , that is permissible for the constraint in question. In the cases where we would like the developed constraint force to be capped at a specific value, we still specify the constraint velocities, \mathbf{u} , but we also indicate what the maximum acceptable constraint forces are. If the calculated constraint force for a specific constraint exceeds its limit, λ_{max} is taken as the solution for the associated element of $\boldsymbol{\lambda}$.

Translation. The first of the three constraints that controls the overall motion of the wheel is the translation constraint. This constraint is part of the prismatic joint that connects link 1 to the ground and is labeled as *Prismatic Joint 1* in Figure 5.2. In the experimental setup shown in Figure 5.1(b), a linear actuator moves the wheel assembly forward at a constant velocity. The force that this actuator applies to maintain this constant velocity is recorded as the DP. Likewise, the prismatic joint between link 1 and the ground is designed such that the constraint velocities, \mathbf{u} , are equal to the velocity that the linear actuator moves the wheel with in the experiments. The associated component of the developed constraint forces, $\boldsymbol{\lambda}$, is interpreted as the DP output for the model. In this case, there is no restriction on the permissible value for the developed constraint force.

Loading. The second of the three constraints that controls the overall motion of the wheel is the loading constraint. This constraint is part of the prismatic joint that connects link 1 to link 2 and is labeled as *Prismatic Joint 2* in Figure 5.2. In the experimental setup shown in Figure 5.1(b), a pneumatic actuator applies a constant load to the wheel assembly while allowing the wheel to move freely in the vertical direction. To represent this component in the model, the prismatic joint was defined to control the constraint force that is developed. As in the previous case, the constraint velocities, \mathbf{u} , are defined. In this case, however, a maximum constraint force is enforced because we would like the developed constraint force to be equal to the force applied by the pneumatic actuator. The maximum value of the constraint force is set to be equal to the load applied by the pneumatic actuator - defined by Eq.(4.1). To ensure that the developed force is always equal to the maximum force, the constraint velocity is set such that the computed component of $\boldsymbol{\lambda}$ will always exceed the limit, and thus the limit constraint force, λ_{max} , is always developed and taken as the solution for the associated component of $\boldsymbol{\lambda}$. Thus, the load applied to the wheel by the pneumatic actuator in the experiments is reflected in the constraint forces developed by the prismatic joint between links 1 and 2.

Rotation. The final of the three wheel constraints is the rotation constraint. This constraint is part of the revolute joint that connects link 2 to the wheel and is labeled as *Revolute Joint 1* in Figure 5.2. In the experimental setup shown in Figure 5.1(b), the

wheel motor applies a torque to the wheel to maintain a constant angular velocity. Likewise, the revolute joint between link 2 and the wheel is designed such that the constraint velocities, \mathbf{u} , are such that the the 20mm/s rim speed that was used in the experiments described in Chapter 4 is reproduced. Because the effective radius used for the modelled wheel is larger than the radius of the real wheel, in order to achieve the same rim speed in the model, the component of \mathbf{u} corresponding to angular velocity about the wheel's axis is set according to:

$$\omega = \frac{v_{rim}}{r} \quad (5.1)$$

where ω is the component of \mathbf{u} corresponding to angular velocity about the wheel's axis, v_{rim} is the desired rim speed and r is the radius of the wheel in the model - which is equal to the effective radius of the contact patch of the deformable wheel used in experiments.

Compaction. The last notable constraint in the system is the one controlling the compaction pressure applied to the particles before the wheel is engaged. This constraint works in much the same way as the loading constraint between links 1 and 2. This constraint is part of the prismatic joint that connects the pressure plate to the ground and is labeled as *Prismatic Joint 3* in Figure 5.2. In the experimental setup shown in Figure 5.1(b), before the wheel is allowed to drive, there is a soil preparation phase. As discussed in Chapter 4, this phase involves compacting the soil with compressed air. To apply a similar compaction in the model, a pressure plate is used to compress the particles. As was the case for the loading joint, the constraint velocities, \mathbf{u} , are defined as well as maximum allowable constraint forces. In this case, the maximum constraint force component associated with the vertical axis is set to be equal to the desired compaction force - which is derived based on the desired compaction pressure and the total surface area of the pressure plate. As was done previously, the constraint velocity associated with the vertical direction is set such that the computed component of $\boldsymbol{\lambda}$ will always exceed the limit, and thus the limit constraint force value is always developed.

5.3 Simulation Steps

The various elements of the simulation are divided into phases. During each phase, the constraint velocities of the mechanism are set to the appropriate values in order to achieve the desired motion. The following is a chronological list of the various simulation phases, what occurs during that phase, and what triggers the simulation to move on to the next phase. The three phases are: soil preparation, wheel engagement, and driving.

5.3.1 Soil Preparation

An important aspect of any DEM type soil simulation is the method in which the soil itself is initialized. It has been shown that the configuration of the soil particles can have a large impact on the bulk response of the system [61] [62]. For this reason, it is important that there always be a consistent soil preparation phase for each simulation. The method for preparing the soil has four stages: particle generation, settling, compaction, and resettling.

Before beginning the description of the soil preparation process, it is important to define what is meant when we say that particles are settled. In real life this definition is trivial. A pile of soil is settled when it comes to rest. In DEM simulation this definition is less trivial since there will be small instabilities causing extraneous particles to have instantaneous jumps in velocity, particularly at lower frequencies. Thus, it is important to define what a settled pile is in the context of a simulation. In order to determine if the particles are settled, the total kinetic energy of the pile is calculated. If the total kinetic energy of all the particles in the system is below a certain cut-off energy value, the pile is said to have settled. The exact value of the cut-off energy is varied depending on the stage of the simulation, as will be discussed.

In the first stage of the soil preparation phase of the simulation, the desired number of particles with the specified radii and properties are generated at random positions within the confines of a bin representing the testbed. This is done by randomly generating a position within the domain of the bin, then randomly selecting one of the allowed radii,

and finally making sure that this new particle does not interfere with any of the already generated particles. The process is repeated until the specified number of particles is added to the model. Once all the particles are generated, we move to stage two.

In the second stage of the soil preparation phase, the particles are allowed to settle under the influence of whatever gravity the simulation is being run under (Martian, Lunar, or Earth). Once the particles have settled in the bin according to the aforementioned definition of the word "settled" in the context of a DEM simulation, we move to stage three.

In the third stage of the soil preparation phase, the pressure plate - which covers the whole surface of the bin - is lowered from the top to compress the particles. This step serves three purposes. First, as the sinkage of the wheel will later be measured as the distance between the wheel hub and the surface of the soil, a clearly defined soil surface level is necessary to accurately calculate sinkage. Since the particles are generated at random, the height of the pile in the rectangular bin can be uneven at first. This makes accurately determining the height of the pile difficult. The compaction makes the surface level more clearly defined. Second, as previously discussed, the soil compaction has an impact on soil behaviour [61] [62] and this stage ensures that the particle compaction for all simulation runs is the same. Third, as discussed in Chapter 4, the soil samples underwent similar compaction in the experimental soil preparation phase. Thus, including this phase as part of the model helps it more accurately match the experiments. The constraint that controls the motion of the pressure plate has the plate move down towards the soil at a constant velocity while developing a maximum constraint force equivalent to a pressure of $5000Pa$ evenly applied to the soil surface. Once the force applied to the plate is no longer sufficient to move it any further towards the soil, the plate is removed from the simulation - triggering the final stage of the soil preparation phase.

The final stage of soil preparation is the re-settling of the particles. After compaction under the pressure of the plate, the particles are allowed to settle again as the compaction relaxes. Once the particles resettled, the soil preparation phase is complete and the next phase of the simulation can begin.

5.3.2 Wheel Engagement

Once the various stages of soil preparation have been completed, the wheel maneuver can begin. The first phase of this process involves engaging the wheel by lowering it into the soil. Initially, the wheel is suspended above the bin where it will not interfere with the soil preparation phase. Once the soil is set, the desired velocity for the vertical direction is set to $-0.1m/s$. This is so that the constraint will lower the wheel until it contacts the soil, and then keep lowering it as much as it can until the constraint force reaches its maximum value. Recall that the constraint force developed in the vertical direction is not allowed to exceed the desired wheel load. Once the max constraint force is reached, the wheel will eventually come to rest. Once the wheel comes to rest, the particles are once again allowed to resettle while in contact with the wheel. After the particles have resettled, we move on to the third and final phase of the simulation.

5.3.3 Driving

The final phase of the simulation is the actual driving motion of the wheel. Until this phase of the simulation, the constraint velocities associated with the horizontal and angular velocity of the wheel are set to zero. Once the wheel is lowered into the soil and the particles have settled, the wheel can begin to move forward. At this point, the constraint velocities associated with these two degrees of freedom are set to their non-zero values determined based on the desired wheel slip ratio and rim speed. As discussed, the constraint forces associated with these constraints are not capped, therefore the desired motion in these directions is always enforced. The constraint forces developed are recorded as outputs of the simulation for later analysis. The simulation comes to an end after 25 seconds of simulation time has elapsed. This duration was chosen because it is approximately the same duration as the experiments.

5.4 Simulation Parameters

Tuning the multitude of parameters involved in the simulation is crucial to getting accurate results. In order to streamline this process, the parameters available for tuning are broken down into three distinct but equally important categories. These are the numerical parameters, the particle parameters, and the simulation input parameters. The methods used to tune each set of parameters will be discussed in this section.

5.4.1 Numerical Parameters

Unlike explicit DEM, for the P² soil model there are two numerical parameters that can be tuned. These are the simulation frequency (or time-step size) and the number of solver iterations for the particle solver. The estimation of the maximum allowable time-step for DEM simulation is a key issue in the field. Too large a time-step can lead to large particle overlap from one time-step to the next, causing instability in the system, but too small a time-step can lead to unnecessarily long computation time or even numerical rounding error [63]. Cundall and Strack's [3] original proposal of the method included a recommended critical time-step that was related to the natural frequency of a spring-mass system consisting of the mass of the smallest particle in the system and the stiffness coefficient used for modelling the contact. This led to a critical time-step of $2\sqrt{m/k}$. Tsuji et. al. [63] used a method based on Cundall and Strack's proposal in which they began with the oscillation period of the spring-mass system, $2\pi\sqrt{m/k}$, and divided half of this period by a factor N to determine the appropriate time-step. They did this by measuring the dissipation of kinetic energy in the system as particles in a tube came to rest for various values of n . It was shown that for larger time-steps the kinetic energy dissipated more slowly, and that for $N \leq 4$ the simulation was unstable. Tanaka et. al. [38] used a slightly different approach to determine critical step size - calculating the energy consumed by the dashpot when a particle collides with the wall and claiming that the energy in the system post-collision must be less than before - but arrive to a similar critical time-step of $0.75\sqrt{m/k}$.

Kinetic Energy Dissipation Convergence. The method used to select the appropriate step size is based on the method proposed by Tsuji et. al. [63]. Their method involved calculating the dissipation of kinetic energy of the particles in an explicit DEM model as they settle under the influence of gravity. The total kinetic energy of the system is defined as the sum of the kinetic energy of all the particles in the model and is given by:

$$KE = \sum_{i=1}^n \frac{1}{2} m_i v_i^2 = \sum_{i=1}^n \frac{2}{3} \rho \pi r_i^3 v_i^2 \quad (5.2)$$

where m_i , v_i , r_i are the mass, velocity, and radius of the i th particle, respectively, and ρ is the density of the soil material. Tsuji et. al. [63] hypothesized that the critical time-step for convergence of the energy dissipation plot was some fraction of half of the natural frequency of the system, given by $\Delta t = \pi \sqrt{m/k}$. They plotted the energy dissipation for different frequencies:

$$f = \frac{1}{\Delta t} = \frac{N}{\pi \sqrt{m/k}}, \quad \{N \in \mathbb{R}, N > 0\} \quad (5.3)$$

where Δt is the time-step length, m is the mass of the lightest particle in the system, k is the stiffness coefficient used for the particle contact model, and N is a coefficient indicating that a fraction, $1/N$, of half of the natural frequency is being tested. They found that when they used $N = 5$, or a simulation frequency equal to one tenth of the natural frequency, the energy dissipation plots converged. This process is applied in the model described in this chapter to determine the appropriate simulation frequency. Particles are initially generated at random in the bin and are then allowed to fall and settle at the bottom of the bin under the force of gravity. The total kinetic energy given by Eq.(5.2) is calculated at each time-step of the simulation, and the dissipation plots for different simulation frequencies are analyzed for convergence. For any given convergence test, the particle parameters and number of simulation iterations are kept constant. The only variable is the simulation frequency. Figure 5.3 is a sample of a convergence test plot with 20 particle solver iterations per step.

Results from various kinetic energy dissipation convergence tests show a clear convergence of the plots as the frequency is increased. The initial particle parameters are different from the ones used to generate the energy dissipation plots in Figure 5.3, but the same procedure was used to determine the appropriate numerical simulation parameters - listed

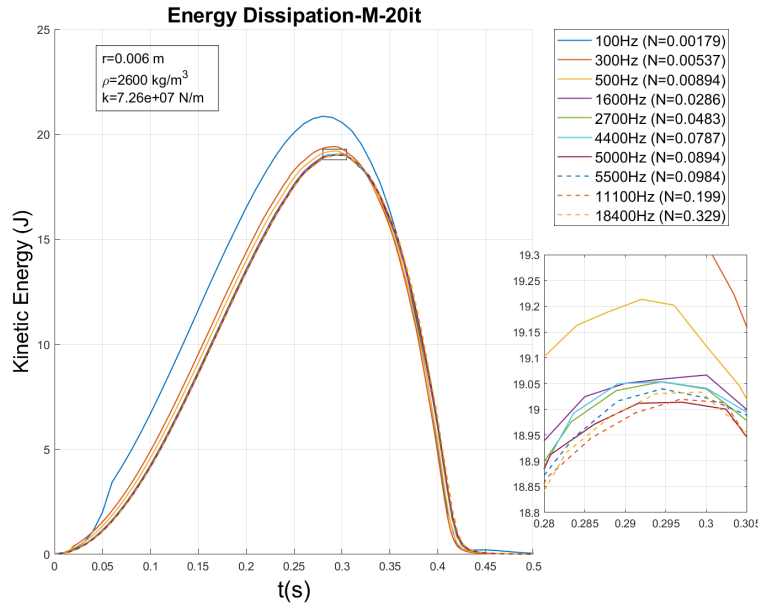


FIGURE 5.3: Sample kinetic energy dissipation plot

in Table 5.1 - for the initial set of physical particle parameters. For the particle parameters used in this work, and with the solver iteration count fixed at 20, it was generally found that the energy dissipation plots converged for frequencies corresponding to $N \approx 0.09$ in Eq.(5.3).

TABLE 5.1: Initial Simulated Numerical Parameters

Parameter	Units	Value
Simulation Frequency, f	[Hz]	5000
PB Solver Iterations		20

5.4.2 Particle Parameters

The particle parameters refer to many quantities, mostly affecting to the formulation of particle contact constraints described in Section 3.2. Particle parameters are difficult to accurately tune because the resulting behaviour of the particles must reflect the behaviour of real-world soil but the parameters available for tuning do not all correlate directly to any measurable soil properties. Lack of understanding into how DEM parameters affect the bulk system behaviour has in itself be the focus of studies such as in [36]. The parameters in this work were selected through a parametric sensitivity analysis that will

be described in detail in Chapter 6. To begin this process a few key properties were selected for tuning. These were adhesion (a), contact stiffness (k), contact damping (c), particle/particle friction coefficient (μ), material density (ρ), and particle radii (r_i). The number of particles in the system is also a parameter that can be varied, however in this case we treat it as a function of the particle radii - where given the average size of the particles, the number of particles is adjusted so that there is adequate soil depth. The outputs resulting from different combinations of parameters are compared to equivalent experimental outputs to determine which parameters are the closest match for the soil that was used.

TABLE 5.2: Initial Simulated Soil Parameters

Parameter	Units	Value
Particle Radii, r_i	$[mm]$	6, 7, 8
Number of Particles		17500
Material Density, ρ	$[kg/m^3]$	2600 [44]
Stiffness, k	$[N/m]$	7.257×10^7
Damping Ratio, γ	$[1/ms]$	2500
Damping, c	$[Ns/m]$	2.06×10^6
Adhesion, a	$[Pa]$	750
Kinetic Friction Coefficient, μ		0.75

The initial values for a , μ and ρ were determined either directly or indirectly by the ES-2 soil parameters provided in Table 4.2 and information from the literature. Adhesion (a) is taken directly as the cohesion value listed in the ES-2 soil properties. Material density (ρ) is used to determine the mass of each particle. In the P² formulation particles are treated as point masses, but the mass of a particle is determined as the product of the volume of a sphere of the associated radius and the material density, which is the same for all particles in the simulation. It is set to a value significantly larger than the bulk density listed in the ES-2 properties because the voids in between particles mean that the overall bulk density is lower than the material density of individual grains. The exact value for material density was chosen based on values used in other works for similar soil simulants [30] [44]. The kinetic friction coefficient (μ) is set as the tangent of the internal friction angle of the soil [43]. In *Vortex*, the static friction coefficient is determined by a static friction scale, which was left at the default value of 1.3 for all simulations. For simplicity, particle-bin stiffness, damping and friction coefficients were considered to

be the same as the equivalent values for inter-particle contact. Particle-wheel contact had the same friction coefficient, but the stiffness and damping were set to lower values to fit the default wheel stiffness properties of *Vortex*. To ensure that the interaction properties for the wheel-soil contact model was not having too large an impact on the results, simulations were run with the particle-wheel stiffness and damping set to be the same as the particle-particle values and it was shown to have little to no impact on the outputs. Since simulating with the actual grain size of soil is not computationally feasible, it is widely accepted in DEM simulation that much larger particle radii can be used for simulation. This is because simulating the actual size of the grains would be far too computationally demanding. The initial radii were selected based on values used in other DEM wheel-soil publications such as [30] in which Martian soil simulants are modelled. Based on the size of the particles, the total number of particles used was selected such that the soil depth was enough to allow for a reasonably thick buffer of particles that are not affected by the wheel's motion between the wheel and the bottom of the bin. The initial contact stiffness was set according to:

$$k = \frac{4G}{3(1 - \nu)} \quad (5.4)$$

where G is the shear modulus of the soil - estimated to be $0.0381 \times 10^9 Pa$ in [50] - and ν is Poisson's ratio - set to 0.3 according to the value used by in DEM to modelling of Martian soil simulants in the literature [30]. Damping was initially set to correspond to a damping ratio of about 2500 because it is the order of magnitude that is used as a default value in the *Vortex* implementation of P² particles. It is then tuned to find the value that best matched experimental results. Damping is given by:

$$c = \zeta c_c \quad (5.5)$$

where c is the damping coefficient, ζ is the damping ratio, and c_c is the critical damping value given by $c_c = 2\sqrt{km}$. As previously mentioned, in addition to finding the set of parameters that best matches the experimental outputs, Chapter 6 will comment on the effect that these parameters have on the resulting simulation output.

5.4.3 Simulation Input Parameters

Besides all the parameters that need to be tuned to run the simulation, certain parameters are just seen as inputs required to simulate the different tests that were carried out during experimentation on the single wheel testbed. These are the magnitude of gravity, the desired wheel slip value, and the load applied to the wheel. The input parameters were mostly selected to correspond to the inputs for which there is experimental data for comparison, listed in Table 4.3.

5.5 Simulation Outputs

There were three main outputs of interest from the simulations of the wheel driving over soil. These were the drawbar pull, sinkage and soil velocity field. The drawbar pull is taken to be the constraint force developed by the constraint controlling the lateral motion of the wheel, as described in section 5.2. As is done in the experiment analysis in Chapter 4, the average DP/W ratio for each slip value is analyzed in the results presented in Chapter 6. This is calculated by taking the average DP recorded during a run of the simulation and dividing it by the wheel load for that simulation - which in the case of the simulations is constant throughout.

The sinkage is defined as:

$$\Delta z = z_{surf} - (z_w - r_w) \quad (5.6)$$

where z_{surf} is the calculated soil surface level, z_w is the z -coordinate of the center of the wheel, and r_w is the radius of the wheel. Since the soil is modelled as a granular medium, the surface level is not an absolute number, and can vary locally. At each step, the average surface level of the particles in the bin is computed according to the procedure described in Algorithm 2. The experiments described in Chapter 4 used the maximum sinkage during an experiment as a way of comparing trials rather than comparing the plots of sinkage as a function of time. Therefore, the maximum sinkage that each simulation achieves during its 25 seconds of simulation time is the output of interest.

Algorithm 2 Determining DEM particle surface level

```

1:  $\Delta x = \text{interval}$  ▷ Fixed sample interval for surface level calculation
2:  $x \text{ range} = [x_-, x_+]$  ▷ Limits of the domain containing the undisturbed particles
3:  $POS_{3 \times n}$  ▷ Global  $(x, y, z)$  coordinates of all  $n$  particles
4:  $j = 1$  ▷ Interval counter
5: for  $x = x_- : \Delta x : x_+ - \Delta x$  do
6:    $\text{max} = 0$ 
7:   for  $i = 1 : n$  do
8:     if  $POS(1, i)$  in range  $[x, x + \Delta x]$  then
9:       if  $POS(3, i) > \text{max}$  then
10:         $\text{max} = POS(3, i)$ 
11:    $\text{height}(j) = \text{max}$  ▷ Find the max particle height in each interval
12:    $j = j + 1$ 
13:  $\text{surface level} = \text{mean}(\text{height})$  ▷ Take the average height of intervals

```

As previously mentioned, one of the main advantages of a DEM soil model is that the velocity field is easy to extract from the model. Unlike semi-empirical models in which there is no clear way to extract the velocity field below the surface, in DEM we can generate this velocity field by plotting the magnitude and direction of the velocity vectors of the particles in the model. In the simulations for this work, this was done in order to qualitatively compare the flow of the particles for different simulation inputs in an effort to assess the effect that gravity has on soil flowability. An example of the visualization of the particle velocity field can be seen in Figure 5.4.

The animation snapshot in Figure 5.4 was run using the baseline simulation parameters described in the previous sections. It is worth noting that Figure 5.4 is a 2D side-view visualization of 3D elements. Figure 5.5 shows the measured outputs, sinkage and DP, as a function of time during this simulation. Also plotted is the surface height of the undisturbed soil - as calculated using Algorithm 2 - as a function of time. The constant surface height is used as an indicator of simulation stability. Additionally, the computation time for each time-step using the baseline parameters is plotted as a function of both the simulation time and the time-step. Simulations were run on an AlienWare Aurora desktop computer.

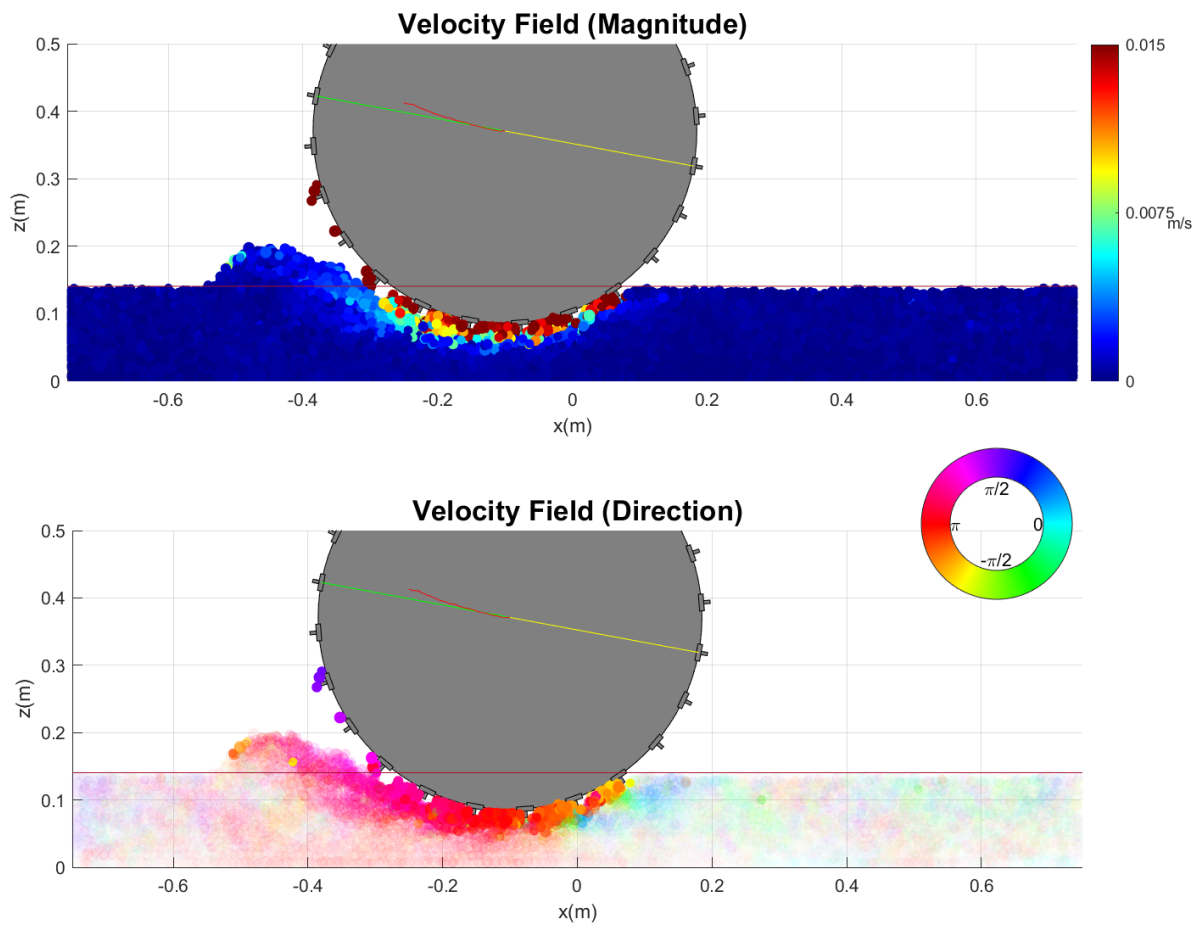


FIGURE 5.4: Magnitude (top) and direction (bottom) of particle velocities in baseline simulation at Lunar-g, 164N wheel load, 70% slip

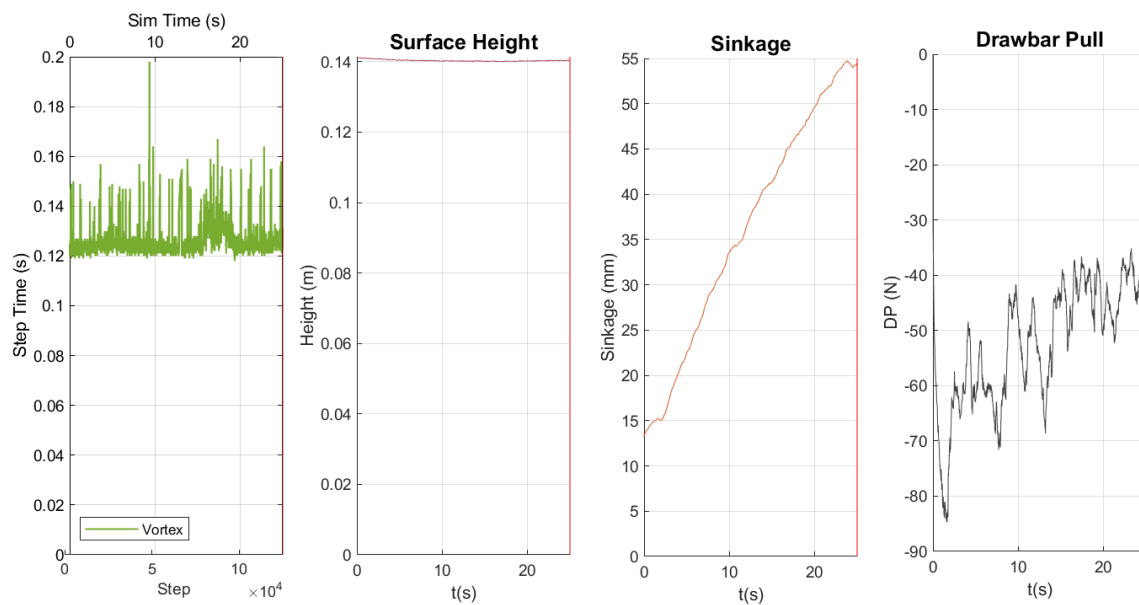


FIGURE 5.5: Outputs (Computation time, surface height, sinkage, DP) as a function of time for baseline simulation at Lunar-g, 164N wheel load, 70% slip

Chapter 6

Results

The goal of this work is to produce simulation results that are in reasonable agreement with the experimental results collected in the tests described in Chapter 4. To this end, there are two levels of agreement that can be defined. The first level of agreement between simulations and experiments is a qualitative matching of trends that were observed in the experiments, especially trends related to how the system behaves in reduced gravity. This was done through detailed analysis of the simulation results obtained using a baseline set of simulation input parameters that were chosen for the reasons described in 5. The second level of agreement between simulations and experiments is the quantitative matching of the simulation and experiment outputs. This result is much more challenging to achieve and it was not expected that we would get a perfect match. This is due in part to the fact that quantitative matching of the outputs requires accurate tuning of the model parameters, which can be challenging, but also in large part due to the potential inaccuracy of the experimental reduced gravity results. The potential source of error in the experimental results will be discussed in detail in this chapter. The challenge of accurately tuning the model is also addressed in this chapter through a parametric sensitivity analysis of the simulation outputs. This analysis involves changing one parameter at a time from the baseline parameter set and commenting on how the change affects the simulation outputs and the sensitivity of those outputs to changes in gravity. Through this analysis we gain a deeper understanding of the relationship between the model parameters and the physical behaviour of the system.

6.1 Baseline Results and Discussion

With data available from both on-ground experiments (OGE) and partial gravity experiments (PGE), one source of validation for the P² particle method is its ability to replicate some of the general trends observed experimentally when running experiments with the same loading and wheel slip in varying gravity conditions. The results presented in this section are recorded prior to tuning of parameters based on experimental data. The initial set of parameters, selected for the reasons described in Section 5.4, is again summarized in Table 6.1. With these parameters, every combinations of load, slip and gravity that was tested experimentally was simulated. This required a total of 21 simulations that span five slip cases, two wheel loading cases, and three gravity levels.

TABLE 6.1: Baseline Simulation Parameters

Parameter	Units	Value
Simulation Frequency, f	[Hz]	5000
PB Solver Iterations		20
Particle Radii, r_i	[mm]	6, 7, 8
Number of Particles		17500
Material Density, ρ	[kg/m ³]	2600
Adhesion, a	[Pa]	750
Stiffness, k	[N/m]	7.257×10^7
Kinetic Friction Coefficient, μ		0.75
Damping, c	[Ns/m]	2.06×10^6

When compared to the experimental data, this data set produced drawbar pull weight ratios (DP/W) that were significantly lower than the experiments as well as max sinkage values that were slightly below what was observed in experiments, as shown in Figure 6.1. Plots of each of the outputs as a function of time for each gravity and slip case for the experiments and baseline simulation parameters can be found in Appendix B. It should be noted that, despite disparities between the simulation and experimental outputs, the results in Figure 6.1 are still quite promising as a proof of concept for the P² particle modelling method, as it did give reasonable results prior to any tuning of parameters. The average error in sinkage for this first set of results was a reasonable 4.4% while the average DP/W error was a larger 45%. This suggests that the model is capturing the motion reasonably well, and the forces less so. At the qualitative level the results are even more encouraging. Figure 6.1 shows that the model captures the shape of both curves

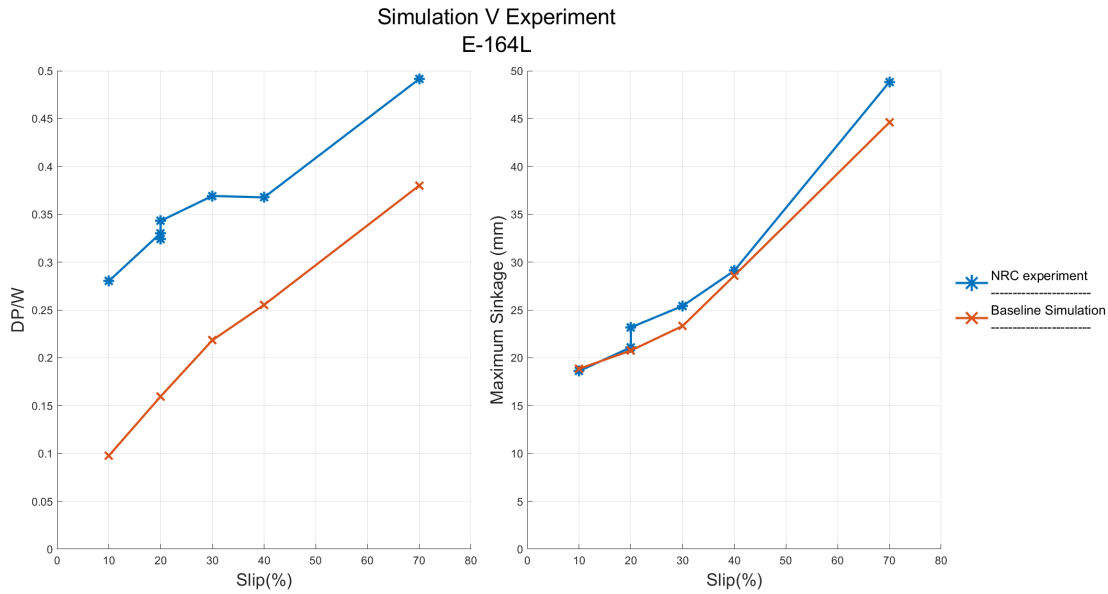


FIGURE 6.1: DP/W (left) and max sinkage (right) as a function of slip for baseline simulation parameters and experimental results in Earth gravity with 164N wheel load

fairly well. The parametric sensitivity analysis that will follow later in this chapter will also show that for different parameter combinations, the qualitative agreement of the shape of these two curves is still present even as the values of the outputs change.

At a quantitative level, the experimental results are difficult to work with because they were collected in less than ideal conditions. This is due to the fact that the gravity level was not always perfectly consistent for the duration of an experiment and the fact that testbed is affected by the vibration of the aircraft. For these reasons, the quantitative results obtained may not be reliable and would therefore be even more difficult to replicate through simulation. Despite this, when looking at the effect that gravity has on the outputs there are a few qualitative and quantitative observations that can be made. The comparison of simulation results in Earth and Lunar gravity - illustrated in Figures 6.2 and 6.3 shows that reducing gravity has the expected effect of increasing sinkage and decreasing DP/W, however, the effect of gravity is not quite as significant in the simulations. In the experiments, the average increase in max sinkage between Earth and Lunar gravity was 38% [1]. The same comparison in the baseline simulation showed a 14% increase in max sinkage. The average drop in DP/W between Earth and Lunar gravity was 20% in the experiments [1] and a slightly higher 24% in the baseline simulation. This seemingly larger sensitivity to gravity for the simulation DP/W output is probably due to the fact

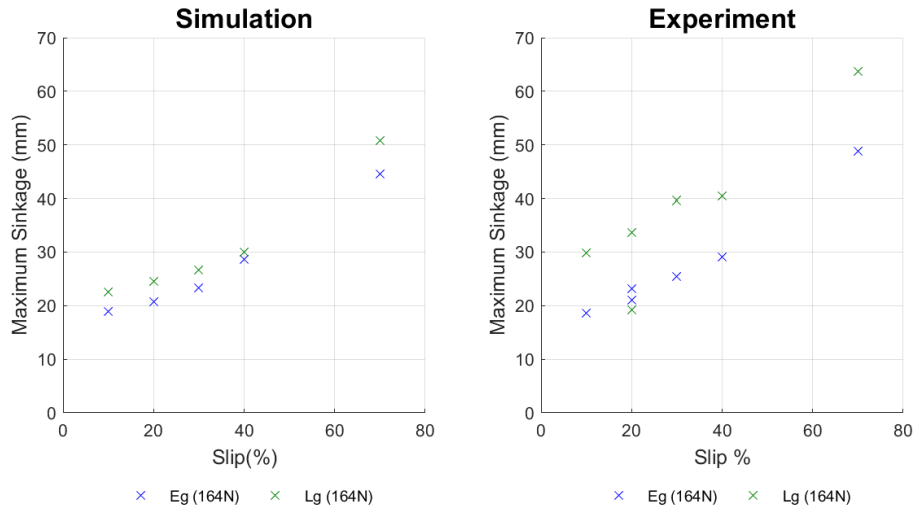


FIGURE 6.2: Effect of gravity on max sinkage for baseline simulation (left) and experiment (right)

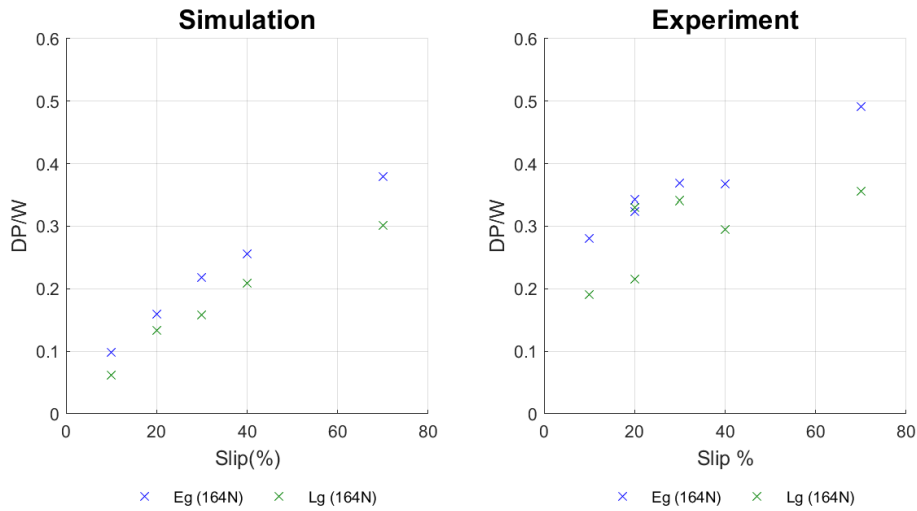


FIGURE 6.3: Effect of gravity on max DP/W for baseline simulation (left) and experiment (right)

that the actual values of DP/W in the simulations were smaller than the experimental results. These two results seem to contradict each other as it seems that depending which output we use as an indicator, we would come to a different conclusion about whether it is the baseline simulations or the experiments that are more sensitive to gravity. Because the change in maximum sinkage due to gravity was more drastic and also more consistent in the simulations, it was decided that the change in the maximum sinkage output would be used as the indicator for sensitivity of the simulation to gravity. By this metric, the

baseline simulation is less sensitive to variations in gravity than the experiments. The smaller sensitivity to gravity of the simulation compared to the experiments is even more clear when comparing the Earth and Martian gravity simulations seen in Figures 6.4 and 6.5. Experimentally there was an average 27% increase in sinkage between Earth and Martian gravity for 164N wheel load, meanwhile the simulations only showed a 4.5% increase in sinkage. As was the case for the Lunar-g comparison, the sensitivity of DP/W to gravity was similar in both experiments, where the experiments showed a 9.9% drop in DP/W and the simulation showed a 9.6% drop.

Figure 6.5 also highlights one experimental trend that is not captured by the simulation, and that is the effect that increasing the wheel load has on DP/W. In the experiments, increasing wheel load with all other inputs remaining the same had the effect of slightly increasing the DP/W while also increasing sinkage. In the baseline simulation, the sinkage increases with wheel load but the DP/W decreases. However, Niksirat et. al. [1] comment primarily in the effect that increased loading has on sinkage, since they deem this effect to be statistically significant while the effect on DP/W may not be. In particular, they look at how gravity affects the sensitivity of the output to increased loading. In Earth gravity, they saw an average maximum sinkage increase of 14% when the load was increased from 164N to 225N. The same load increase in Martian gravity led to a 31% increase in maximum sinkage. The same comparison in the baseline simulations shows a 24% increase in sinkage at Earth gravity and a slightly larger 26% increase at Martian gravity. Thus, while the percent change is not the same, the baseline simulation does replicate the experimental conclusion that decreasing gravity increases the output sensitivity to wheel loading. The effect of each parameter on the sensitivity of the simulation to gravity will be explored in the next section in order to get better results.

One of the other key points that the reduced gravity experiments set out to prove is the effect that reduced gravity has, not just on the load felt by the soil but on the flow of the soil itself. In [1], the authors show velocity field data for several equivalent test runs under varying gravity conditions to qualitatively illustrate the higher degree of motion of soil particles in reduced gravity. Figure 6.6 shows a similar comparison of a 70% slip, 164N load simulation test case in which all simulation parameters except for gravity are the same. Plots of the soil motion under the wheel are shown for each gravity level at

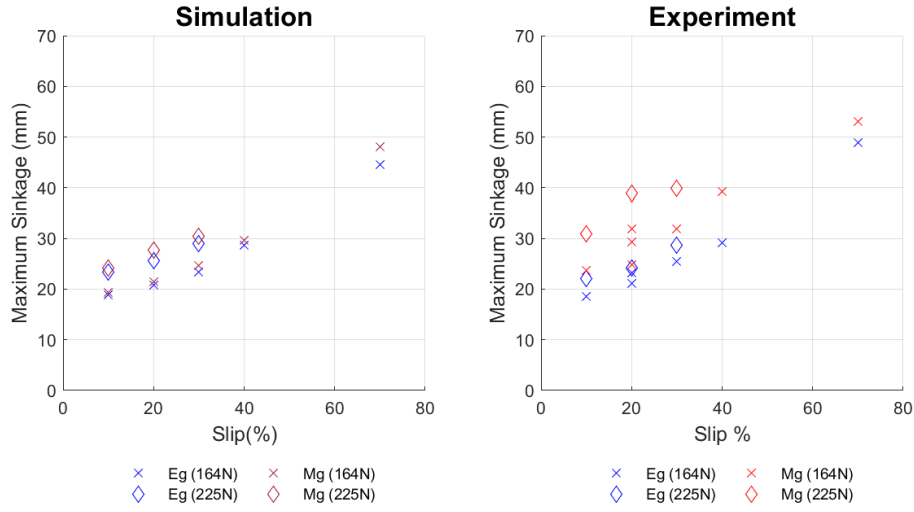


FIGURE 6.4: Effect of gravity on max sinkage for baseline simulation (left) and experiment (right)

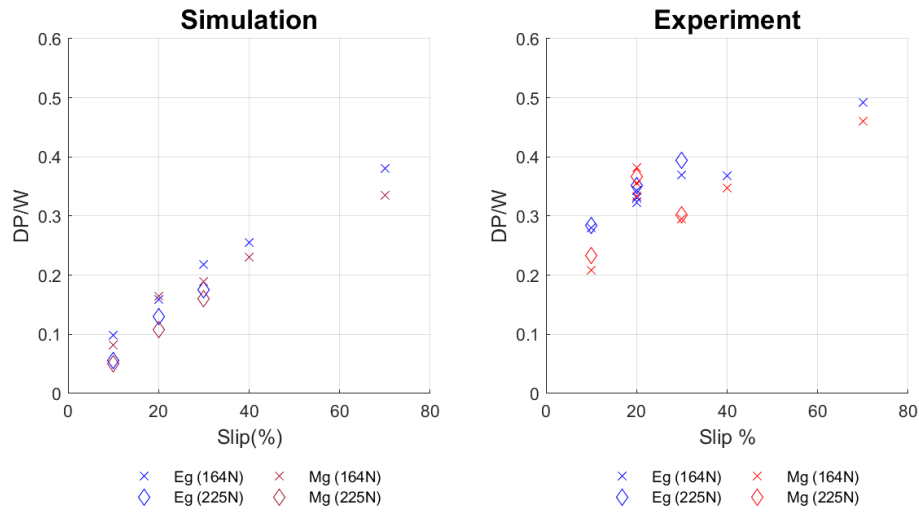


FIGURE 6.5: Effect of gravity on max DP/W for baseline simulation (left) and experiment (right)

5 second intervals. From these plots we can qualitatively confirm that the P^2 soil model is able to replicate the trend observed experimentally in [1]. This is particularly evident in the soil near the rear of the wheel (left) where we can visually see that the band of particles that are in motion is generally larger in the Lunar gravity cases than in the Earth and Martian gravity cases.

While the outputs are not perfectly matched with the experiments in this first set of results, there are important conclusions and observations that can be made. Firstly, the

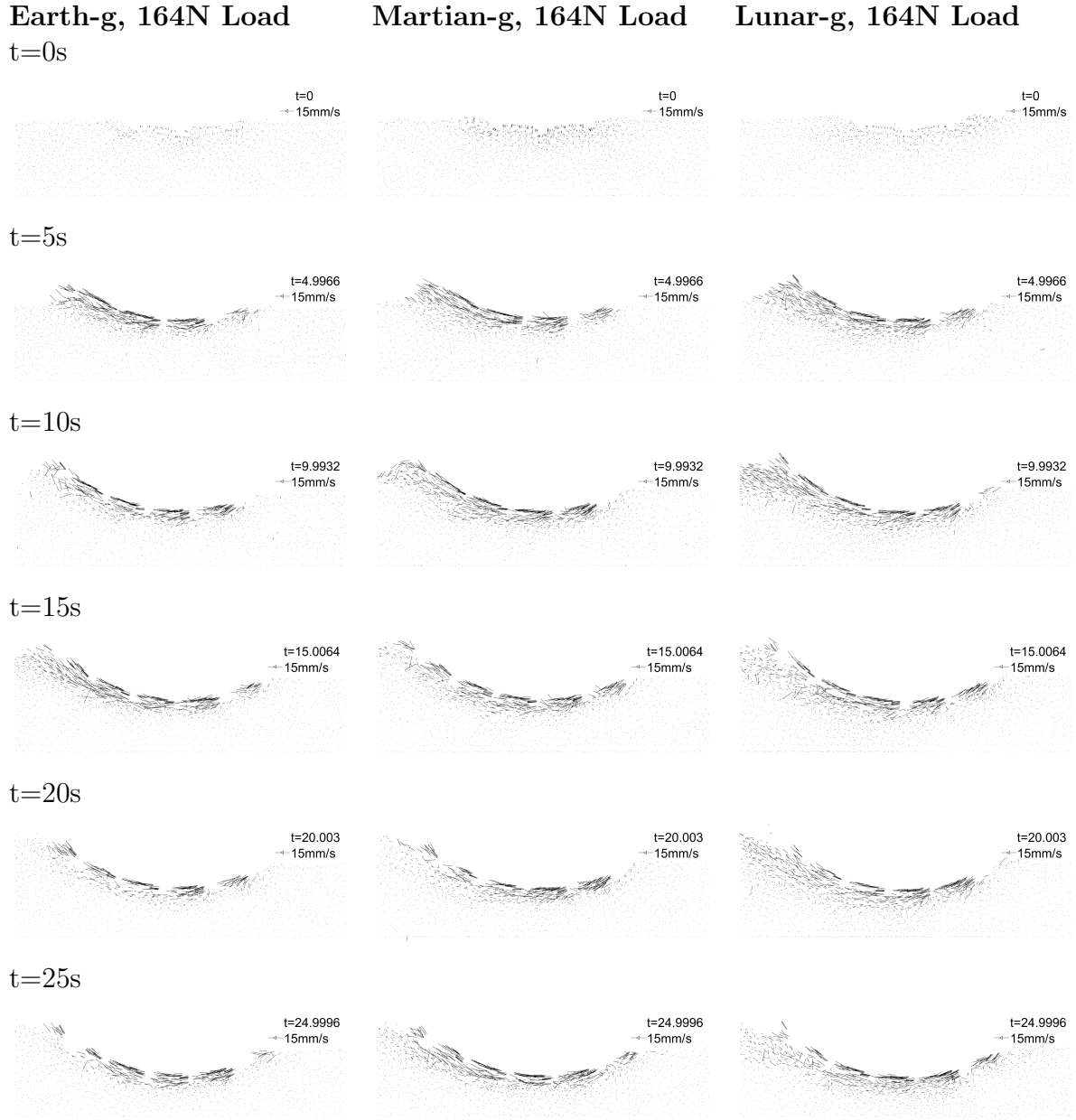


FIGURE 6.6: Effect of reducing gravity on soil velocity field at 70% slip for baseline simulation parameters

effect of gravity on the soil behaviour is what the reduced gravity flights set out to prove in the first place. With the objective of disproving the intuitive method of testing planetary rovers on Earth by simply testing them on Earth and applying the same load they would experience on the Moon or Mars, Niksirat et. al. [1] performed Earth-g and reduced-g tests, applying the same load in each case. If the aforementioned intuitive approach worked, there would not have been any significant difference between the Earth-g and reduced-g outputs - this was not the case. Our simulations were able to confirm that the

same load in different gravity levels does not result in the same outputs. The simulation results presented in this section also showed that our modelling approach was able to predict the qualitative effects that gravity had on the outputs of interest - even if the actual numerical values of those outputs did not match perfectly. The simulations showed that reduced gravity leads to increased sinkage, decreased DP/W, and a qualitatively observable increase in soil flowability. Secondly, setting the effect of gravity aside, the general shape of the DP/W vs Slip and Max Sinkage vs Slip curves match the shape that is expected based on the experimental results. At least at the qualitative level, this modelling method is reasonably successful at predicting some wheel-soil behaviour. All of these results combined suggest that, with some tuning of the parameters, the method would be able to reproduce the experimental results and, by extension, predict outputs for untested input parameters.

6.2 Parametric Sensitivity

As has been noted by many researchers since DEM was first proposed, one of the main drawbacks of the method is the difficulty related to selecting the appropriate parameters, and the lack of understanding of how these parameters affect the system response. Smith et. al. [30] select values for every parameter except damping by correlating the parameters with measurable soil sample properties through theoretical equations. They proceed to tune the one remaining parameter by trial and error. This process can be long and does not help to understand the physical effect of each parameter. Yan et. al. [36] go a step further and analyze the effect of the variation of various parameters on system behaviour by varying the Youngs modulus, coefficient of restitution, static friction coefficient and rolling friction coefficient and analyzing how changes to these parameters affected bulk system responses such as the angle of repose and discharge time for the angle of repose test. In our case, the effect of the parameters on the DP/W and on maximum sinkage will be analyzed.

This section will focus on the effect that each individual parameter in Table 6.1 has on the DP/W and maximum sinkage outputs. This is done by starting from the baseline

simulation just described and modifying the value of a single parameter while keeping all other parameters the same as described in Table 6.1. The objective here is not just to find the right combination of parameters, but to understand how each parameter affects the outputs so that this information can be used to tune the parameters to match experiments in a more systematic way.

To save computation time, most parameter combinations were only simulated for three to six of the 21 test cases. These simulations were selected so as to sufficiently well capture all the effects we wish to study. The 10%, 30% and 70% slip cases with 164N wheel load at Earth and Lunar gravity were simulated for new variations of the parameters. In some cases only the Earth simulations were carried out for the purposes of analyzing the effect of the parameter on sinkage and DP/W without looking at the effect it has on sensitivity to gravity. This allows us to not only observe how each parameter affects the sinkage and DP/W outputs, but also to determine which parameters, if any, affect the sensitivity of the model to changes in gravity. This is an important output to monitor because, as was illustrated in Figures 6.2 - 6.5, the baseline simulation was less sensitive to gravity than the experiments. The results gathered from each new set of simulations help to gain an understanding of how each parameter affects the simulation outputs and provides insights into how the model can be more efficiently tuned. The parameters that will be varied are the stiffness coefficient (k), the damping coefficient (c), the adhesion constant (a), the material density (ρ), the inter-particle kinetic friction coefficient (μ) and particle radius r . These are not the only parameters that could have an effect on the outputs. Parameters such as wheel-particle friction, bin-particle friction, particle size, and packing pressure could also affect the results. In order to make for a reasonable amount of simulations required given the long computation time required for each simulation, this work focuses on the effect of the six parameters mentioned above.

6.2.1 Effect of Stiffness

Figure 6.7 illustrates the effect that the stiffness coefficient, k , has on the simulation outputs that are relevant in this study. Because stiffness also has an effect on the necessary time-step - given the relationship between time-step size and natural frequency defined

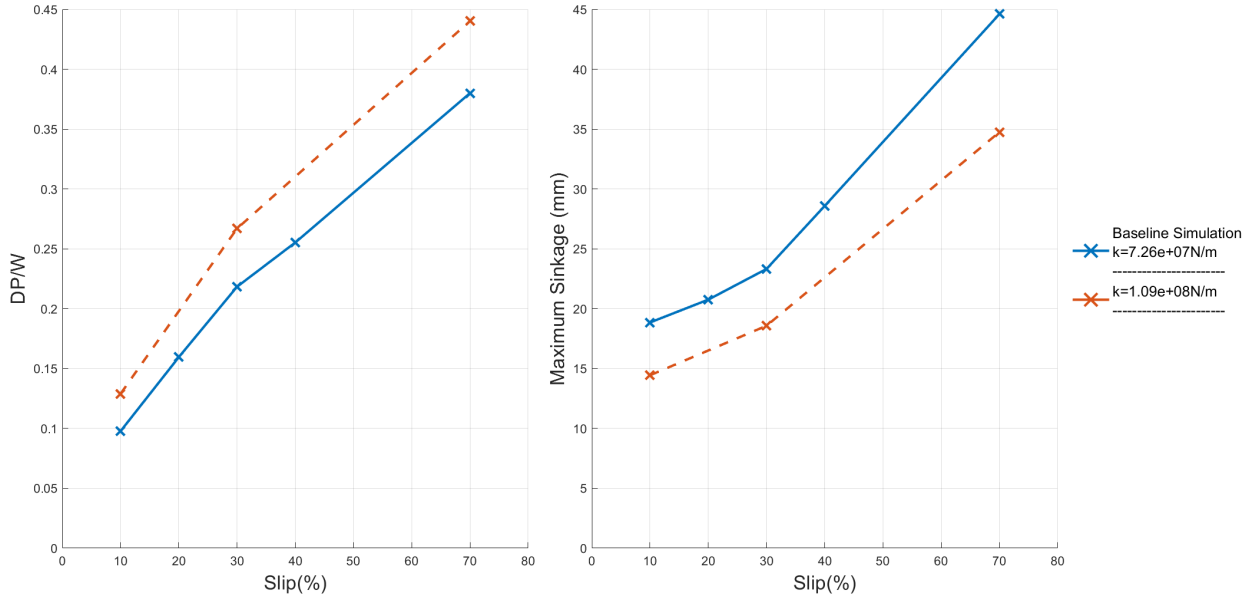


FIGURE 6.7: DP/W (left) and max sinkage (right) as a function of slip for baseline simulation parameters and varied stiffness in Earth gravity with 164N wheel load

in Eq.(5.3) - as the stiffness was varied from the baseline simulation parameter value the simulation frequency was adjusted accordingly.

In Figure 6.7 we see what happens when the stiffness is increased from its baseline value by 50%. This has the effect of decreasing the maximum sinkage. Additionally - as is usually the case when sinkage decreases - the drawbar pull increases. This inverse relationship between sinkage and drawbar pull is due to the fact that there is more resisting force applied to the wheel by the particles when the wheel sinks more since there are more particles in contact with the wheel [1]. Drawbar pull would be increased further if there was a way to decrease soil strength (i.e. resistive force per particle) without increasing sinkage too much. Given the results from Section 6.1, this is precisely the kind of effect we are looking for, since we need to greatly increase the drawbar pull without decreasing sinkage in order to match the experiments. In this case, an increase in stiffness resulted in an average maximum sinkage drop of 22% and an average DP/W increase of 23%. These results suggest that there is a inverse relationship between stiffness and sinkage and a direct relationship between stiffness and DP/W. Because only one one variation of the parameter was tested it is possible and even likely that this relationship is non-linear. The results also suggests that neither output is significantly more or less sensitive to variation in stiffness. Figure 6.7 also illustrates how the general shape of the curves is similar as

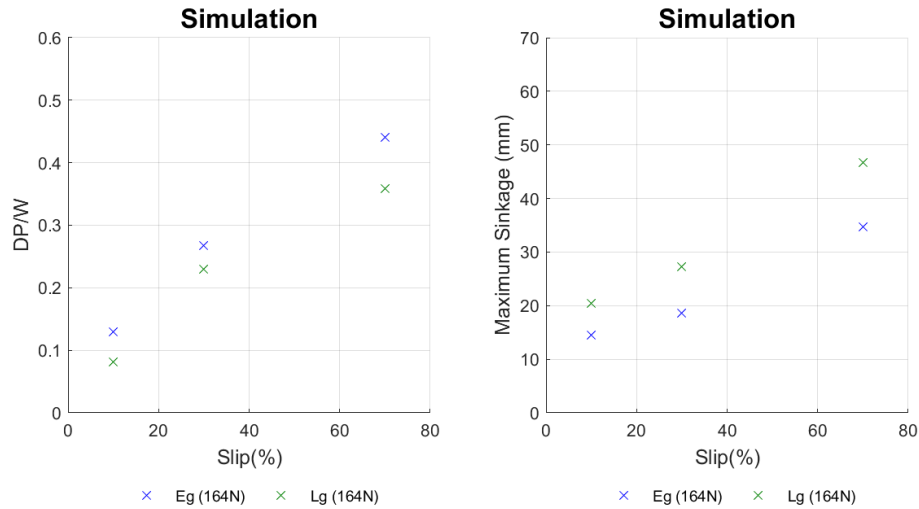


FIGURE 6.8: Effect of gravity on DP/W (left) and sinkage (right) for simulation with increased inter-particle stiffness

stiffness is varied, with the curves largely being shifted higher or lower on the Y-axis as the parameter is varied.

Figure 6.8 shows some interesting behaviour with respect to how the system behaves in reduced gravity with the increased stiffness parameter. On a qualitative level, one of the key differences between the baseline simulations and the experiments is that in the experiments the sinkage output was much more sensitive to changes in gravity than in the simulation. This was evidenced when comparing the 38% increase in max sinkage between Earth-g and Lunar-g measured experimentally to the 14% increase in max sinkage in the baseline simulation. When stiffness was increased by a factor of 1.5, the average increase in max sinkage when gravity was reduced from Earth-g to Lunar-g was 41%. This suggests that increasing stiffness could help to better match the simulation results to the experiments. Because the baseline simulation matched the Earth gravity sinkage results so closely, an increase in stiffness will result in a worse match for the maximum sinkage output. A possible solution for this would be to pair increased stiffness with a change in one or more other parameters to offset the decrease in sinkage that is associated with increasing stiffness.

6.2.2 Effect of Damping

Figure 6.9 illustrates the effect that the damping coefficient, c , has on the simulation outputs that are relevant in this study. The baseline damping coefficient is equivalent to a damping ratio of around 2500, a value that is close to the default damping used for real-time particle simulation in *Vortex*. In this plot, damping was reduced from its original value to a damping ratio of 0.3, which is a damping value used in other DEM works [30]. The average effect across all slip values at Earth gravity with 164N wheel load was a 40% increase in DP/W and a 47% decrease in sinkage. This suggests that there is a direct relationship between damping and sinkage and an inverse relationship between damping and DP/W. As before, these relationships may be non-linear. Determining linearity of the relationship would require further simulations. The results also suggests that sinkage is more sensitive to damping than DP/W.

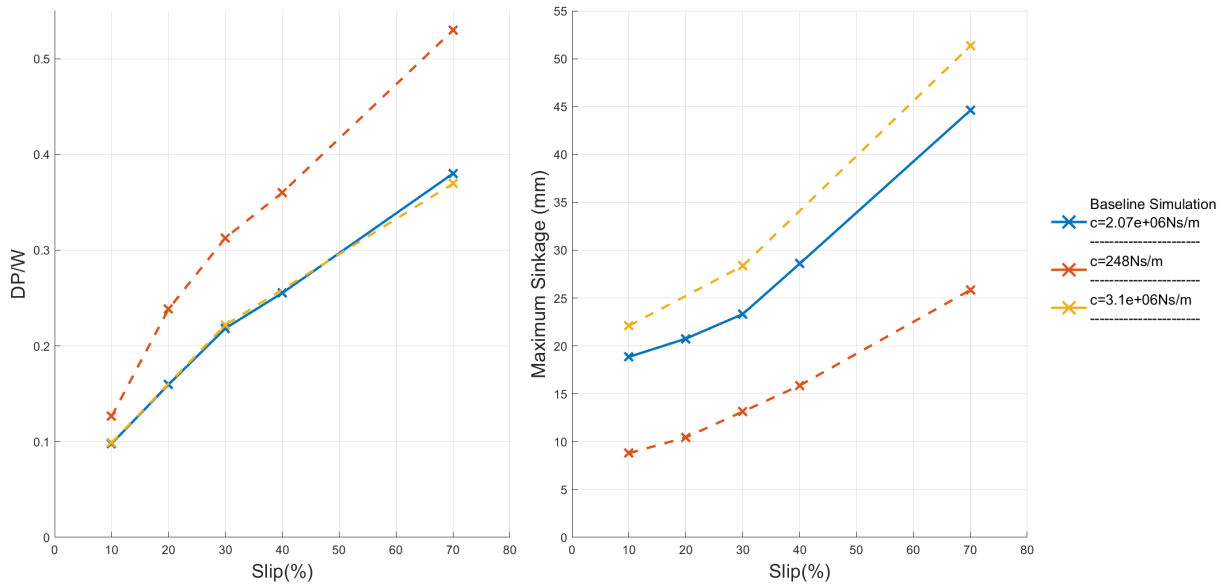


FIGURE 6.9: DP/W (left) and max sinkage (right) as a function of slip for baseline simulation parameters and varied damping in Earth gravity with 164N wheel load

A second variation of damping that is seen in Figure 6.9 is the effect of increasing damping by the same 50% relative to its baseline value, as was done for stiffness. Since decreasing damping decreased sinkage, the result that increased damping increased maximum sinkage was expected. The average increase in maximum sinkage was 18%. This result is significant because a relatively small increase in damping has a significant impact on

the sinkage and almost no effect on DP/W. The sinkage increases noticeably in this case while the DP/W is almost unchanged in comparison - decreasing by a mere 0.05%. This is in start contrast to the much more drastic change to damping that occurred when the damping ratio was set to 0.3 in the first variation of the damping parameter, resulting in noticeable changes to both outputs. We can conclude that any change in damping will have an effect on the sinkage, but only a significant change (i.e. by one or several orders of magnitude) will affect the DP/W in a significant way. The ability of damping to be adjusted so as to impact one output more than the other makes it very useful for model tuning and supports the DEM tuning method used in other works where damping is the only parameter that is changed [30].

Next, as was done for stiffness, the effect of damping on sensitivity to gravity was tested by running the same simulations with varied damping coefficients at Lunar-g. The results showed that increased damping reduced the sensitivity of the maximum sinkage output to changes in gravity. As seen in Figure 6.10, the result of comparing Earth-g and Lunar-g results after increasing damping by a factor of 1.5 is a 11% increase in max sinkage and a 36% drop in DP/W. Therefore we can say that for higher damping the sinkage less sensitive to gravity in than baseline simulation while the DP/W appears to be more sensitive.

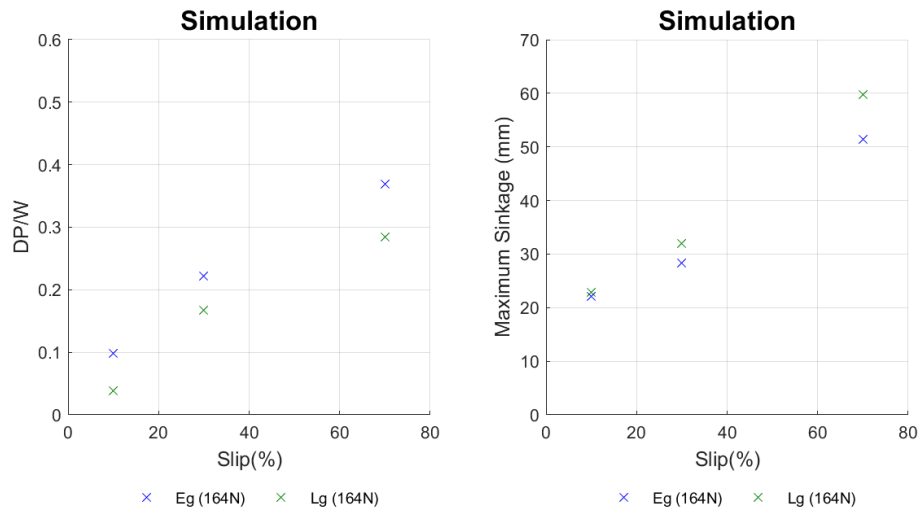


FIGURE 6.10: Effect of gravity on DP/W (left) and sinkage (right) for simulation with increased inter-particle damping

6.2.3 Effect of Adhesion

Figure 6.11 illustrates the effect of the adhesion coefficient, a , on the simulation outputs that are relevant in this study. This is analyzed by running simulations with two new variations of the adhesion coefficient. The first is a 50% percent increase of this parameter, as was done for stiffness and damping, and the second is with no adhesion. Both variations have a relatively minimal effect on the output. This suggests that, for the values tested, the adhesive force is overshadowed by other more dominant forces such as the ones associated with stiffness and damping. For the zero-adhesion case, the DP/W increased by 7% and the maximum sinkage decreased by 2%. For the case where adhesion was increased to 150% of its baseline value (to 1125 Pa) the result was a DP/W increase of 4% and a maximum sinkage increase of 3%. Increasing the adhesion even further, to the 5000 Pa that has been used by other researchers modelling Martian soil simulants [30], still showed minimal changes in the outputs. The result was a drop in maximum sinkage of 5% and an increase in DP/W of 5% relative to the baseline simulation. This result is interesting because makes adhesion the only parameter whose effect on the outputs does not seem to be consistent as it continues to increase or decrease. Looking at the sinkage of the 70% slip case it can be seen that at 1125 Pa there is more sinkage than at 0 Pa, but the sinkage drops again as the adhesion is raised to 5000 Pa.

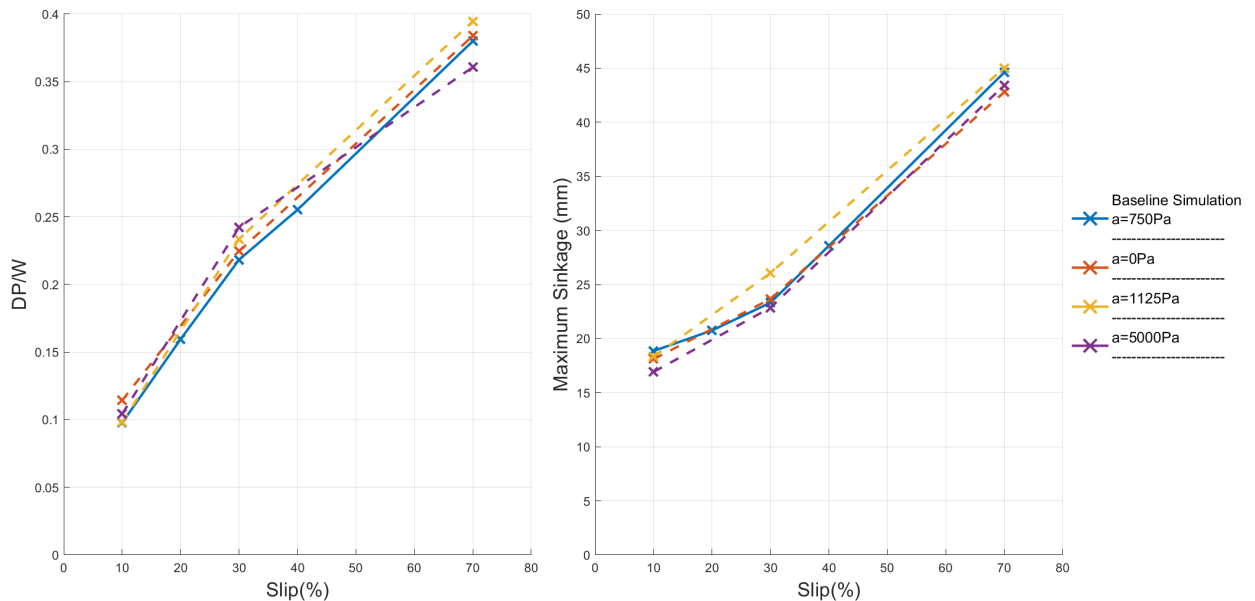


FIGURE 6.11: DP/W (left) and max sinkage (right) as a function of slip for baseline simulation parameters and varied adhesion in Earth gravity with 164N wheel load

The effect of adhesion is thus much more complex than the more straightforward effects of some of the other parameters that it can be said without a doubt that a certain change in a certain direction will universally have the effect of either raising or lowering a given output. Despite all this, given the relatively small effect that adhesion has on the outputs in all cases suggests that for the outputs being monitored there is little evidence that tuning adhesion will help much with the goal of matching the experiments quantitatively. It is worth noting, however, that Holz [4] showed, using the P^2 framework, that adhesion does have an effect on the angle of repose of the particles. Meaning that adhesion can have an impact in certain types of simulations but it is simply not a dominant force in the case of wheel-soil interaction unless it is set to be well above what was initially calculated based on the real internal friction angle of the soil.

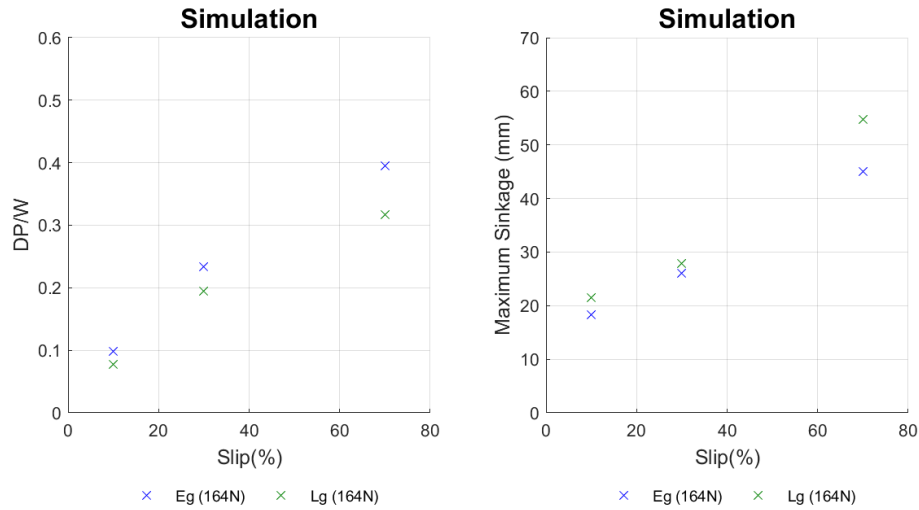


FIGURE 6.12: Effect of gravity on DP/W (left) and sinkage (right) for simulation with increased inter-particle adhesion

Figure 6.12 illustrates the effect that increased adhesion has on the sensitivity of the outputs to changes in gravity. Unsurprisingly, the sensitivity to gravity for the 1125Pa adhesion case demonstrates a sensitivity to gravity that is very similar to the baseline case. The average sinkage increase between Earth-g and Lunar-g is of 15% and the average DP/W drop is of 19%. Thus, the sinkage output is slightly more sensitive to gravity while the DP/W output is slightly less affected by gravity than in the baseline cases. This result reinforces the fact that adhesion is not the most useful parameter to tune since its overall effect is small in comparison to the effect of other parameters.

6.2.4 Effect of Material Density

Within a simulation, the particles generated come in a range of sizes. For the baseline simulation, the size of particles ranged from $6mm$ to $8mm$. While in the P^2 formulation the particles are treated as point masses, as has been discussed, the mass of a particle is determined as the product of the volume of a sphere of the associated radius and the material density, which is the same for all particles in the simulation. Figure 6.13 illustrates the effect that the material density, ρ , has on the simulation outputs. As was the case when stiffness was varied, when material density changes it is also necessary to alter the time-step accordingly. This is again because the particle mass, like the stiffness coefficient, has an impact on the natural frequency of the system - as per Eq.(5.3). Besides simulation frequency and material density all simulation parameters are the same as the ones from the baseline simulation summarized in Table 6.1. The results show that the relationship between material density the outputs is similar to the relationship the outputs have to changes in stiffness. There is a direct relationship between density and DP/W and an inverse relationship between density and sinkage. More specifically, increasing material density from $2600kg/m^3$ to $3500kg/m^3$ led to a 34% increase in DP/W and a 17% decrease in max sinkage while a decrease of material density to $2000kg/m^3$ led to a 12% decrease in DP/W and a 10% increase in sinkage.

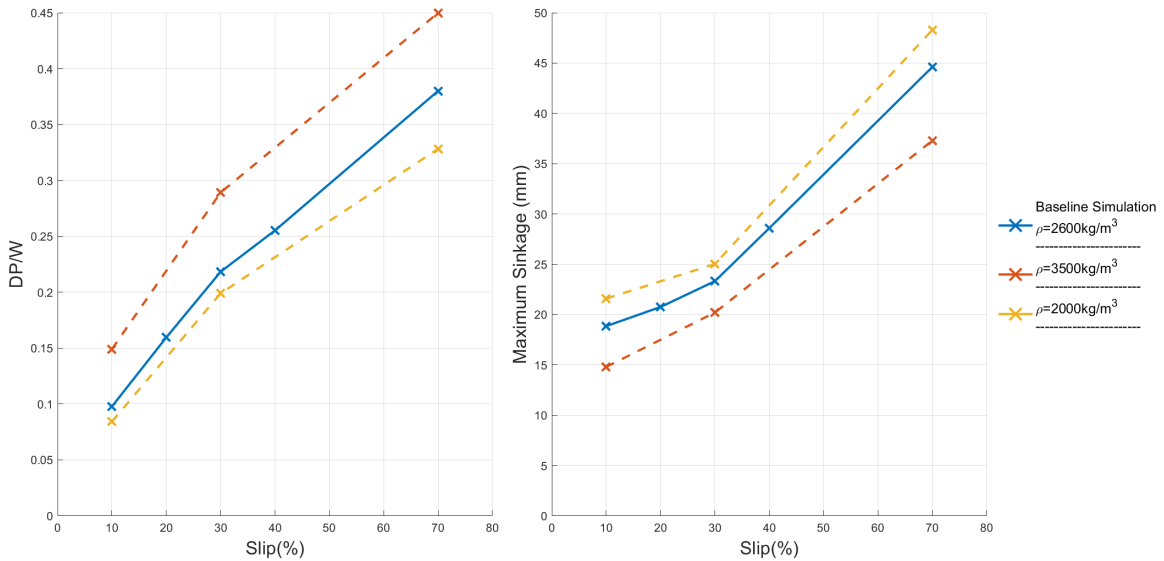


FIGURE 6.13: DP/W (left) and max sinkage (right) as a function of slip for baseline simulation parameters and varied material density in Earth gravity with 164N wheel load

Figure 6.14 shows the effect that material density has on the sensitivity of the model to changes in gravity. In the baseline simulation, the average sinkage increase when gravity was reduced from Earth-g to Lunar-g was 14%. In the $3500\text{kg}/\text{m}^3$ density case the percentage increase in sinkage is up to 23%. The percentage decrease in DP/W for the same density case is 22%, which is similar to the 24% decrease in the baseline simulation. The main takeaway from this comparison is that material density does have a significant impact on the sensitivity of the model outputs to changes in gravity.

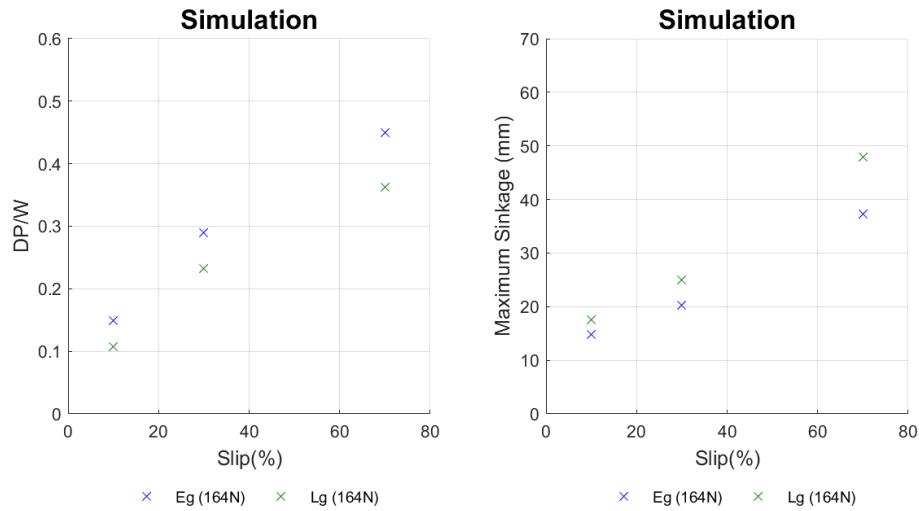


FIGURE 6.14: Effect of gravity on DP/W (left) and sinkage (right) for simulation with increased material density

6.2.5 Effect of Friction

Figure 6.15 illustrates the effect that the inter-particle kinetic friction coefficient, μ , has on the simulation outputs. Because particle orientation is not considered in the P^2 formulation. This means that static and sliding friction exist in the system but the only possible relative motion between particles is sticking or sliding, not rolling. Yan et. al. [36] found that decreasing the inter-particle friction coefficients has the effect of increasing the soil flowability. Increased flowability is associated with reduced soil strength [1] and thus, the increased sinkage seen in Figure 6.15 as friction is gradually decreased from the baseline value is in line with the parametric analysis of friction in traditional DEM methods. For

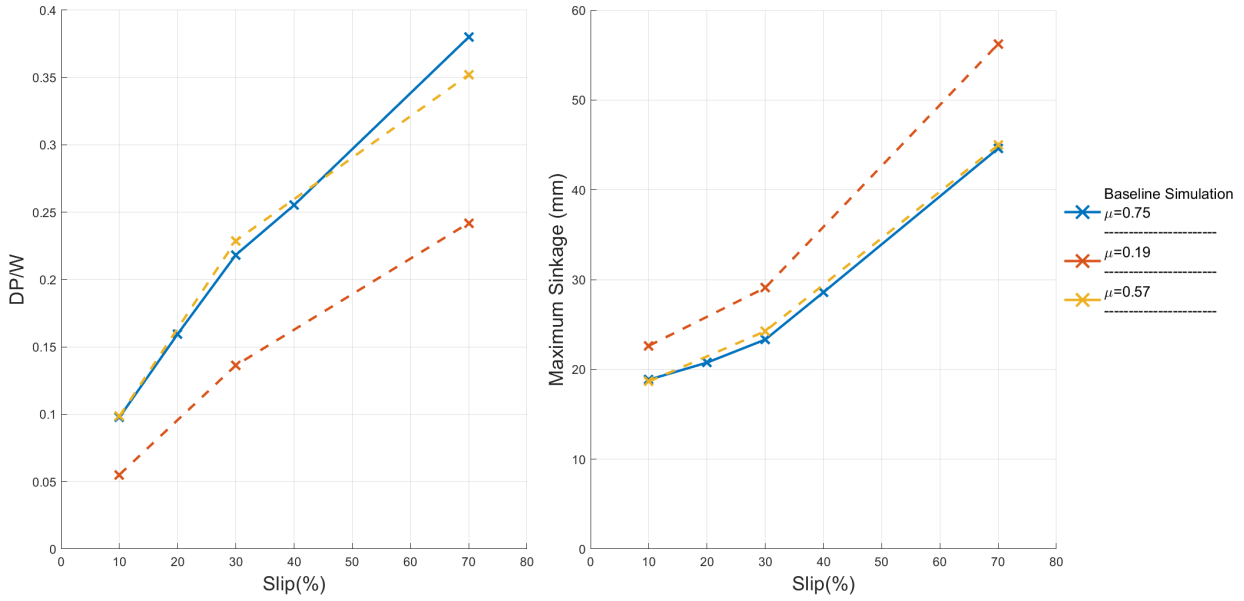


FIGURE 6.15: DP/W (left) and max sinkage (right) as a function of slip for baseline simulation parameters and varied friction in Earth gravity with 164N wheel load

the larger decrease in friction coefficient, to the extreme low value of $\mu = 0.19$, the max sinkage increased by 24% from the baseline simulation while DP/W dropped by 39%.

Figure 6.16 shows how reduced kinetic friction affects the sensitivity of the outputs to changes in gravity. Simulations were run in Earth-g and Lunar-g for the $\mu = 0.19$ friction case and the results showed that, while the DP/W drop was not drastically different between Earth-g and Lunar-g, there was a huge spike in the sensitivity of the sinkage output to gravity. On average, the sinkage increase between Earth and Lunar gravity was of 150%. The DP/W drop was by an average of 28%. This result conclusively proves that lower friction greatly increases the sensitivity of the model to gravity. For tuning purposes however, the $\mu = 0.19$ is an extreme case that corresponds to an internal friction angle of 11° which is a far cry from the $37 \pm 5^\circ$ value taken from the known soil properties in Table 4.2. The $\mu = 0.57$ corresponds to a friction angle of 30° which is closer to being in the range of the internal friction angle of the ES-2 soil sample used in the experiments. For this value Figure 6.15 showed that there was not significant deviation from the baseline. Therefore, while we can say that the model can be very sensitive to drastic changes in friction, if we limit ourselves to the range of permissible values dictated by Table 4.2, the effect of tuning friction is less significant.

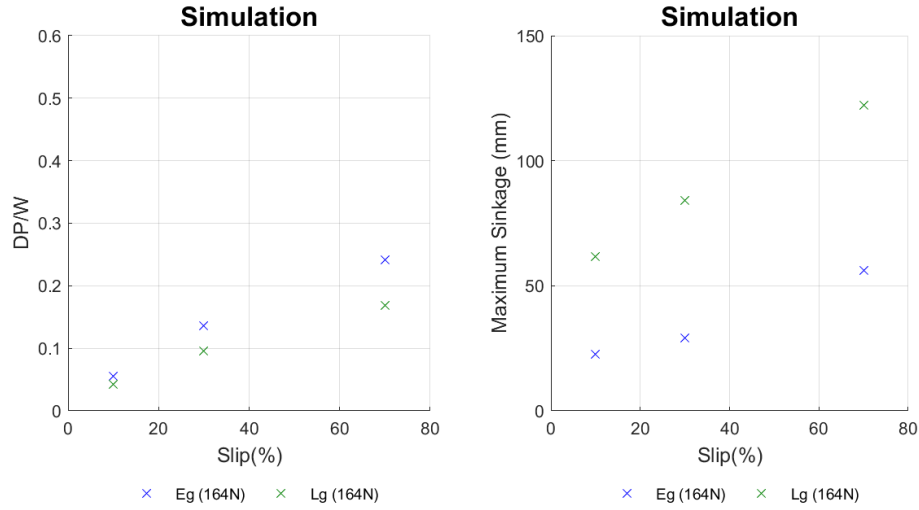


FIGURE 6.16: Effect of gravity on DP/W (left) and sinkage (right) for simulation with reduced inter-particle friction

6.2.6 Effect of Particle Radius

Figure 6.17 illustrates the effect that the particle radius, r , has on the simulation outputs. As was the case when stiffness was varied, when particle radius changes it is also necessary to alter the time-step accordingly. This is again because the particle mass, like the stiffness coefficient, has an impact on the natural frequency of the system - as per Eq.(5.3). Besides simulation frequency and particle radius, all simulation parameters are the same as the ones from the baseline simulation summarized in Table 6.1. The results show that there is a direct relationship between radius and both DP/W and sinkage. More specifically, decreasing the particle radius range from 6 – 8mm to 4 – 6mm led to a 13% decrease in DP/W and a 9% decrease in max sinkage. While at first glance these results seem to create a worse match with the experimental results, the particle radius parameter is particular when compared to other parameters in that its relationship with both recorded outputs is in the same direction (i.e. direct or indirect). Since, based on the baseline results, getting a better match with the experiments will necessarily require tuning of at least one parameter that has this quality, this result suggests that the particle radius is an important tuning parameter.

Figure 6.18 shows the effect that material density has on the sensitivity of the model to changes in gravity. In the baseline simulation, the average sinkage increase when gravity

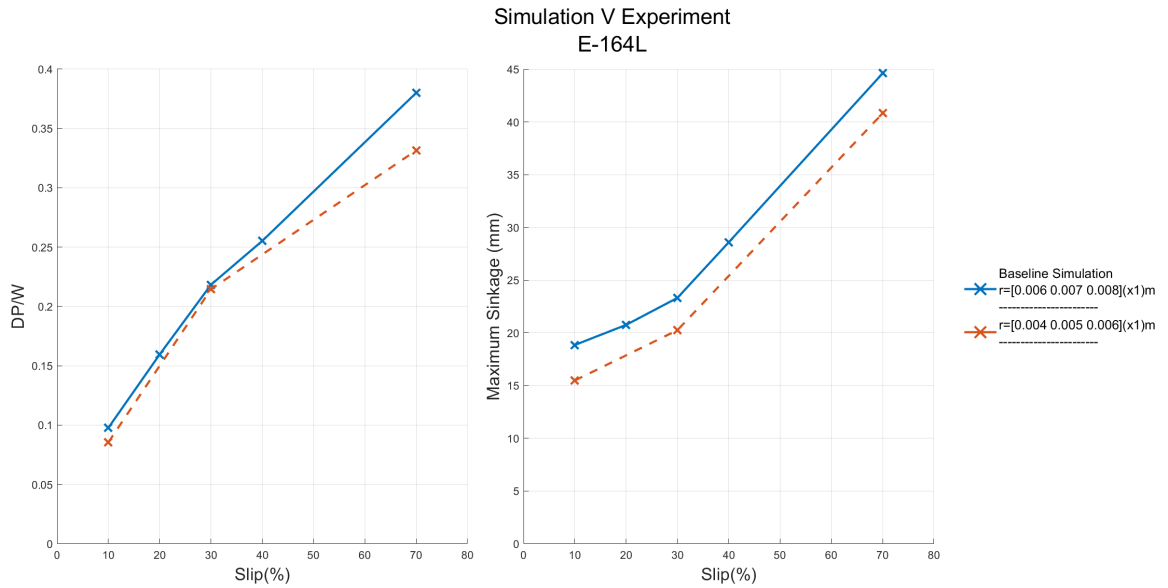


FIGURE 6.17: DP/W (left) and max sinkage (right) as a function of slip for baseline simulation parameters and varied particle radius in Earth gravity with 164N wheel load

was reduced from Earth-g to Lunar-g was 14%. In the 4 – 6mm particle radius case, the percentage increase in sinkage is up to 38%. The percentage decrease in DP/W for the 4–6mm particle radius is 32%, which also significantly larger than the baseline simulation. The main takeaway from this comparison is that particle radius has a significant impact on the sensitivity of the model outputs to changes in gravity - with smaller particles being more sensitive to reduced gravity.

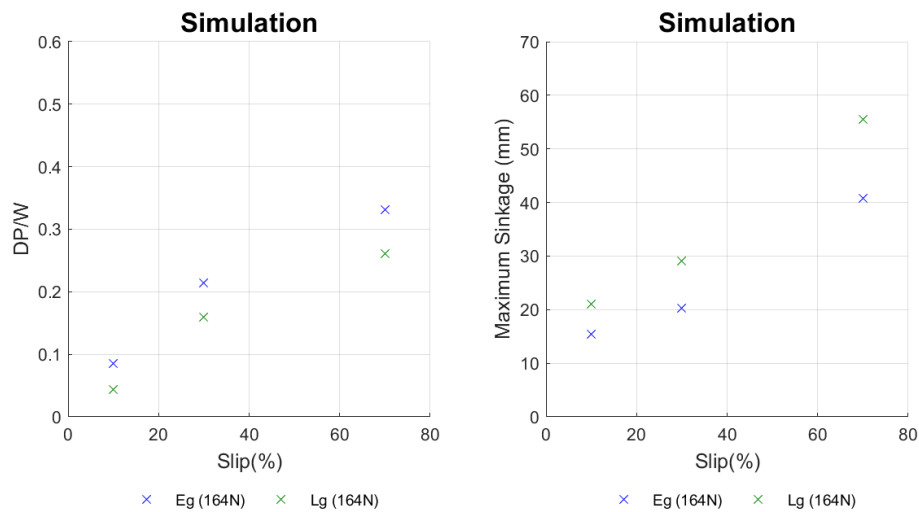


FIGURE 6.18: Effect of gravity on DP/W (left) and sinkage (right) for simulation with reduced inter-particle friction

6.2.7 Discussion

Based on the results presented in this section, it is clear that choosing the correct parameters to model a given soil sample is no simple task. For the purposes of this project only six parameters were analyzed, but the degree to which each affects the outputs make it clear that coming up with a systematic way to select the parameters is crucial to effectively implementing this type of method. Consider the fact that there are more parameters still that could have been analyzed. Particle size and compaction pressure come to mind immediately. It has been mentioned that the particle size used in this simulation, which is known to be much larger than the size of the grains, was selected out of necessity to allow for reasonable computation time. Some analysis was done on simulations with smaller particles to study the effect of particle size, and it was shown that smaller particles result in less sinkage - but the full effect of particle radius is left to be explored in future research due to the computation time required for extensive simulations using smaller particles. Future work could determine whether the extra computation time that would be required to simulate a larger amount of smaller particles is necessary. The degree of accuracy required from the model would depend on the application - specifically whether speed or accuracy is more important. Additionally, while this work used DP/W and sinkage as the outputs of interest for characterizing the effect of each parameter, other indicators could also be used. As previously mentioned, in [36] they look at how the parameters affected outputs such as the angle of repose of particles, discharge time of particles through an opening, and the inter-particle contact forces that are developed. DP/W and sinkage can be considered bulk system responses, while an analysis of the magnitude of position corrections applied - which can be equated to the contact forces developed - could be considered an intrinsic system response [36]. The number of ways to do a parametric analysis is virtually endless, here we presented one while leaving the door open for others to be explored.

Of the parameters that were tested it was found that the outputs were both quite sensitive to stiffness and material density. Both parameters were shown to have a direct relationship with DP/W and an inverse relationship with maximum sinkage, though these relationships may be non-linear. Outputs were also determined to be highly sensitive

TABLE 6.2: Parametric sensitivity summary

Change from baseline	Max Sinkage at Earth-g	DP/W at Earth-g	Change in Sensitivity of Sinkage to Gravity (from 14% baseline)
k ↑ 50%	↓ 22%	↑ 23%	+24%
c ↑ 50%	↑ 18%	↓ 0.05%	−3%
c ↓ 99%	↓ 47%	↑ 40%	<i>un-tested</i>
a ↑ 50%	↑ 3%	↑ 4%	+1%
a ↑ 667%	↓ 5%	↑ 4%	<i>un-tested</i>
a ↓ 100%	↓ 2%	↑ 7%	<i>un-tested</i>
ρ ↑ 35%	↓ 17%	↑ 35%	+9%
ρ ↓ 23%	↑ 10%	↓ 12%	<i>un-tested</i>
μ ↓ 24%	↑ 1%	↓ 0.6%	<i>un-tested</i>
μ ↓ 75%	↑ 24%	↓ 39%	+136%
r ↓ 29%	↓ 9%	↓ 13%	+24%

to damping, but for this parameter there was an inverse relationship with DP/W and a direct relationship with maximum sinkage. It was also shown that the maximum sinkage output was significantly more sensitive than DP/W to small changes in damping. The outputs displayed much smaller sensitivity to adhesion and friction when variations in these parameters is limited to the range of values that is in line with the known soil parameters. Table 6.2 summarizes all the variations that were tested for each variable and how they affected the model outputs.

It was also shown that changing the parameters has a quantifiable effect on the sensitivity of the model to variations in gravity. This is an important finding, as it was shown that the baseline simulation was less sensitive to gravity than the experiments that are being used for validation. Stiffness, friction and particle radius are the parameters whose effect on gravity was the most significant, but the effect of friction was only significant when its value was well below the value expected based on the known soil properties - meaning that tuning stiffness or particle size are best methods for improving the agreement between simulation and experimental sensitivity to gravity. Table 6.2 lists the degree to which tuning each parameter made the percent increase in maximum sinkage between Earth-g and Lunar-g go up or down.

6.3 Results After Tuning

Given the results of the parametric sensitivity simulations, it is logical that stiffness and damping be the focus for the parameter tuning. This is because the other parameters - friction, adhesion, and material density - can be directly correlated with the ES-2 soil properties given in Table 4.2. Contact stiffness and damping, however, are more difficult to relate to measurable soil properties. While equations like Eq.(5.4) attempt to relate stiffness to real measurable soil properties, the use of this equation for determining the baseline simulation stiffness and damping does represent one of the biggest assumptions that was made - particularly since the shear modulus and Poisson's ratio used in the calculation are not exactly known for the soil sample in question.

TABLE 6.3: Final Simulation Parameters

Parameter	Units	Value
Simulation Frequency, f	[Hz]	6700
PB Solver Iterations		20
Particle Radii, r_i	[mm]	6, 7, 8
Number of Particles		17500
Material Density, ρ	[kg/m ³]	2600
Adhesion, a	[Pa]	750
Stiffness, k	[N/m]	1.31×10^8
Kinetic Friction Coefficient, μ		0.75
Damping, c	[Ns/m]	3.31×10^6

The goal of the parametric sensitivity analysis is to ultimately use the knowledge gained to assess the parameters and ultimately get a better agreement between simulation and experimental outputs. The baseline simulation parameters can be thought of as an educated first guess, the results of which were promising as a pre-tuning output. Many of the general trends observed in the experiments were also present in the simulation results, but the simulation was noticeably less sensitive to changes in gravity and the DP/W values were lower than expected. In order to achieve a sensitivity to gravity that is closer to what was obtained experimentally, the stiffness was increased. This has the added effect of also increasing DP/W, thus fixing two of the main conflicts between the baseline simulations and experimental outputs. One undesirable side-effect of increasing stiffness is that it also results decreased sinkage. The original sinkage results were very close, and even a bit below what was determined experimentally. In order to offset this

undesired sinkage drop, damping was also increased by a factor comparable to the factor of 1.5 that was done in the parametric sensitivity test. This value was chosen because it was shown that this results in an increase in sinkage and has a negligible effect on the DP/W output.

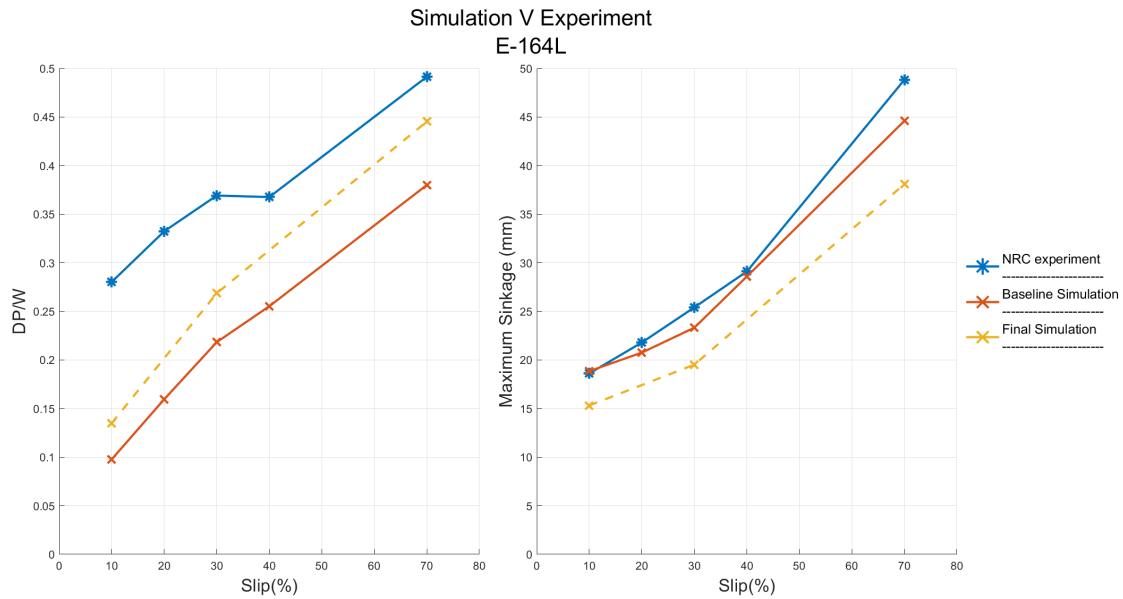


FIGURE 6.19: DP/W (left) and max sinkage (right) as a function of slip for baseline simulation parameters, NRC experiments, and final simulation parameters

The results of this final simulation are not perfect, the comparison in Figure 6.19 makes this clear, but it represents an improvement in some senses from the baseline. Though the sinkage is now further from the experiments than the baseline, the match for both DP/W and sensitivity to gravity are improved in this iteration of the model. With more study of parametric effects, it can be possible to get a better match. We stop here in this work because it illustrates the overall point that with tuning the model can be made to better match the experiments - though a perfect match may not be possible for reasons which will be discussed in the next section. The particle radius was not changed in this final simulation because the effect of particle radius on the simulation results is unclear from the simulation results available

6.3.1 Discussion

After many attempts at replicating experimental results with various combinations of simulation parameters, it became clear that it was unlikely that it would be possible to exactly match the experimental results. There are several factors that could cause this. First and foremost, the reliability of the experimental data, particularly for the reduced gravity, must be considered. The conditions under which the reduced gravity experiments were carried out introduce many possible sources of uncertainty. The unintended periodic loading that was applied to the wheel in experiments instead of the constant wheel load that was planned for is one issue that was only addressed in simulation by assuming that applying the average of this periodic load and analyzing the DP/W ratio rather than the DP directly would be sufficient corrective actions. It is not clear what kind of effect these peaks and valleys of higher and lower loading would have on the outputs. This mechanical fault was discovered very late in the post-processing phase of the experimental data after many simulations had already been carried out. It was been shown in works such as [56] that within the operating range of a wheel the DP/W does not vary much with increased or decreased loading, and so we were comfortable carrying on with the constant wheel load assumption for the purposes of simulations. Another source of error in the experiments described in [1] is the variance in soil properties that could exist between trials. Finally, vibrations of the single-wheel device during operation, vibrations of the plane in which the device was loaded for the reduced gravity flights, and the not perfectly consistent gravity level felt during the reduced gravity portion of the flight could also introduce some error which is hard to account for in simulations. For all of these reasons, replicating general trends rather than numerical results was always expected to be the more likely outcome of this work. Section 6.1 details how many of the general trends observed experimentally were captured from the first set of simulations that were done.

Besides experimental error, a major cause for the discrepancy between experimental results and the multiple iterations of simulation results using different parameters could be the formulation itself. One of the main challenges that prevented a perfect match for both the maximum sinkage and DP/W output was the fact that they are directly linked to each other, and from the start they were off by much different degrees. The

initial maximum sinkage data was very close to what was obtained experimentally, but with a much lower DP/W . Through the parametric sensitivity tests, it was shown that in almost every case when sinkage increased, drawbar pull would decrease, and vice-versa. Because of this it was not possible to find parameters that corrected the DP/W - which was 45% lower than the experiments - without worsening the match of the sinkage data - which was less than 5% below the experimental value. This begs the question as to why, for approximately same sinkage, the average drawbar pull in the simulations is so much lower than in the experiments. The answer to this may lie in the very definition of what drawbar pull is. As Eq.(4.3) indicates, it is the difference between the thrust that the wheel is generating and the total resistive force from the soil acting on the wheel. Smaller than expected DP/W values mean larger than expected resistive forces in the soil. There could be many reasons for this, but the most logical is the omission of particle rotation from the particle formulation. It has been previously suggested that lack of particle rotation in a DEM-type simulation might result in unrealistically high shear forces [35]. This would suggest that if particle rolling were considered in the formulation, the resistive forces acting on the wheel would decrease resulting in higher drawbar pull - which would be more in line with the experimental results.

Despite these challenges and uncertainties, there are many positive takeaways from the results presented in this work. Even the baseline results were quite encouraging as a proof of concept of the P^2 soil model. The model was able to capture many important trends related to the behaviour of the system as the input conditions are varied. Specifically, the qualitative effects of wheel slip ratio and gravity on the outputs are reflected in these results. At the quantitative level, the results were not in perfect agreement with the experiments, but the parametric sensitivity analysis combined with an understanding of the shortcomings of both the particle formulation and experimental setup give reason to believe that a better agreement is possible if these issues are properly addressed.

Chapter 7

Conclusion

In this thesis, an implicit position-correction based variation of the traditional DEM method for soil modelling is implemented for use in wheel-soil interaction modelling. The soil settling under the effect of gravity, the soil being compressed by a constant pressure, and a wheel driving over the compressed soil is simulated using the framework of the *Vortex* multibody dynamics simulation software. The drawbar pull and wheel sinkage outputs are compared as individual parameters are varied in order to qualitatively and quantitatively describe the sensitivity of the outputs to each of the studied parameters.

In theory, after correct tuning of the parameters, the resulting soil model is applicable to a wider scope of situations for which experimental results do not exist than would be the case for a semi-empirical model. In a semi-empirical approach, a set of coefficients is valid for a unique soil sample paired with a specific wheel in a specific gravity environment [5]. If any of these factors are varied, a new set of experiments is required for tuning. With a DEM approach such as the one proposed in this thesis, the coefficients are specific only to the soil sample - meaning that once correctly tuned, one can reliably model the same soil interacting with a different wheel or in a different gravity environment.

The overall trends observed in the experiments do seem to be reflected in the simulation results. All other parameters being constant, increasing the wheel slip has the desired effect of increasing both sinkage and drawbar pull in a qualitatively similar way. Additionally, this method is able to capture the experimentally observed effect that decreasing

gravity increases motion in the soil, increases sinkage and decreases drawbar pull. The simulation parameters do have an effect on the degree to which gravity affects the outputs, but in all cases the change is in the right direction to some degree.

The work presented has many potential applications. Wheel-soil models have many uses that range from real-time applications in simulators to design optimization simulations. The reduced gravity element of the work applies specifically to rovers or other mechanisms that are designed to interact with soil in reduced gravity. This research could aid in the design of the rovers themselves or in the mission planning for existing rovers to avoid situations in which a rover attempts a maneuver that it cannot complete. A prime example of why this kind of work is important is NASA's Spirit rover which got stuck in a sand dune on Mars in 2009. The rover was unable to free itself which resulted in the termination of that mission in 2010 [64] [65]. In a similar fashion, this work could aid in the design of excavation tools for mechanisms that may be sent to Mars or other planets in the years to come to aid with construction. There are also applications for this work here on Earth in the design and mission planning of vehicles and mechanisms intended for operation on unstructured terrain such as soil or sand. The P^2 wheel-soil model could also have potential uses in real-time simulators. This application would likely require much larger particles to allow for larger time-steps, as is done for P^2 soil excavation simulations in [4]. The effects of these larger particles on the accuracy of the results which would have to be analyzed to determine the viability of this method for real-time wheel-soil simulations. Another potential application for this work would be in the field of fluid dynamic modelling. DEM can also be used for fluid simulations in which a liquid is modelled as discrete particles. Thus, the P^2 method could also have potential applications in this field of study.

7.1 Future Work

The work detailed in this thesis leaves many possibilities open for further exploration of applications of the P^2 method for modelling of wheel-soil interactions. These range

from improvements to the model, to new types of experimental validation, to additional experimental setup suggestions that would help to further validate or refine the model.

Model Improvement. In terms of model improvements, there are a few options that can be explored. The first relates to the particles themselves. The P^2 method achieves its improved efficiency both through the omission of particle orientation as a factor and the use of a parallelizable iterative solver scheme to couple particle interactions. Modifying the method to account for orientation of the particles using a similar iterative angular position correction scheme would be an interesting addition to improve results while still maintaining the parallel solver scheme that makes the method significantly faster than traditional DEM. The addition of particle orientation to the P^2 method would also allow for the possibility of non-spherical particles. This would also be an interesting avenue to explore as many studies have suggested that more complex particle shapes are better at capturing the interlocking of particles observed in real soil samples [31] [32] [33] [34] [45]. Another improvement to the model that could be made would be the representation of the wheel itself. As mentioned in Section 5.1, it was assumed that increasing the radius of the wheel from its actual size to the effective radius of the deformed flexible wheel would allow for the wheel to be modelled as a rigid body for the mechanics analysis purposes. This assumption was supported by a previous study using the same data [50], however, a future iteration of this model could improve upon it by doing away with this assumption and instead tackling the complex task of modelling the flexible wheel. All DEM methods have the advantage that the soil model is completely independent of the wheel model. Thus, implementing the same particle model with an improved wheel model could be seamless. Alternatively, rather than completely modelling the flexible wheel, other assumptions could be tested and compared to the one that was made for this study.

Simulation Types. Another category of work that could be done to expand on this research is the study of different simulation configurations. The scope of this work focused almost exclusively on the simulation described in Section 5.3, where soil was poured into a bin, the soil was compacted under a constant vertical pressure and the wheel was lowered into the soil and allowed to drive. Different sections of this simulation process were analyzed in order to tune parameters but in the end this process was the main focus.

This was mainly due to the fact that experimental data was available for this case. There are, however, many other types of experiments that are routinely used to characterize soil and these would be worth modelling. These include pressure-sinkage tests [46], soil shear tests [46], wheel digging tests [30] [44] [45], cone penetrometer tests [66] and angle of repose tests [39] [36]. For parametric sensitivity analysis in particular, analyzing the effect of each parameter on the outputs of these different simulations would help gain extra insight into the effect that each parameter has on bulk soil properties.

Experiment Design. The goal of the work presented in [1] was to study the effect of gravity on wheel soil interaction. In the design of the experiment, it was not considered that the data would be used to tune or validate a model. Therefore there are several suggestions for improvement that could be used in the design of future experiments in order to make the data collected more useful for modelling purposes. One such improvement would be to collect experimental data using a wheel that is less complex to model. This could entail using a rigid wheel rather than a flexible one and/or using a wheel with no grousers. An example of such an experiment can be seen in the experiments described in [48] and [46]. Since the main goal of a future experiment would be to see if the soil model behaves in a way that is representative of real life, it would be beneficial that all other elements of the simulation be simple to model more exactly. If the wheel used were not flexible, the rigid body assumption would in fact be quite valid and the effective radius assumptions would be unnecessary.

P² v DEM. Another avenue that could be explored in future work is a direct comparison between the outputs of a P² wheel-soil model and a traditional DEM wheel soil model. This comparison would help to easily understand the effects that the assumptions made in the P² formulation have on the behaviour of the system.

Computer-aided parameter tuning. The parametric sensitivity analysis allowed for a systematic tuning of parameters through an understanding of the effect of each parameter on the outputs. Future work might address the tuning process further and perhaps develop a computer-aided tuning method that involves an optimization algorithm to determine the ideal parameter combination more quickly and efficiently.

Appendix A

Other Common Terramechanics Formulations

A.1 Bekker Model

The Bekker method is currently one of the most widely used method for modeling wheel-soil interactions. It is a semi-empirical method that was proposed by Bekker [5], and expanded on by Wong [9] [20]. In his method, that was first proposed in 1956, Bekker simplifies the wheel to be a rigid cylinder traveling over flat soil in steady state conditions. Figure A.1 illustrates the model used in the Bekker method.

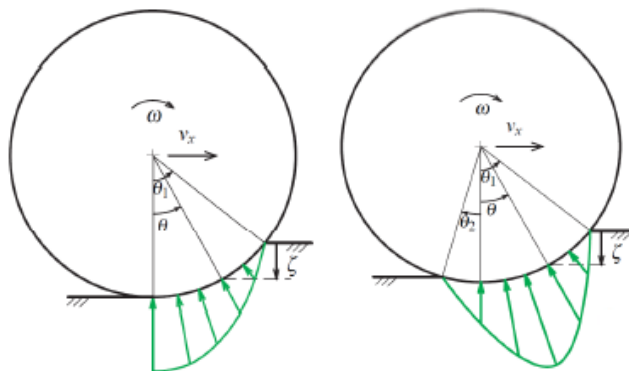


FIGURE A.1: Contact geometry for Bekker (left) and Wong-Reece (right) methods [2]

The Bekker method relies on knowing how much the wheel is initially penetrating the soil, and how much the soil recovers after the wheel has passed over [2]. In his formulation,

Bekker [5] defines the slip ratio as:

$$s = \begin{cases} (r\omega - v_x)/(r\omega) & \text{if driving} \\ (r\omega - v_x)/(v_x) & \text{if breaking} \end{cases} \quad (\text{A.1})$$

where v_x is the horizontal velocity of the center of the wheel, r is the radius of the wheel, and ω is the angular velocity of the wheel. The Bekker method and its recent modifications first rely on physical experimentation to characterize the properties of the soil [5]. The required tests are a penetration test to measure normal stress-displacement relationships, and a shear test to measure shear stress-displacement relationships. The pressure-sinkage relationship that Bekker proposes in [8] is given as:

$$\sigma = \left(\frac{k_c}{b} + k_\phi \right) \zeta^n \quad (\text{A.2})$$

where k_c , k_ϕ and n are determined experimentally, b depends on the test setup, σ is the pressure on the wheel and ζ is the sinkage [2]. Once the soil properties have been determined through experimentation, equations are developed to calculate the normal and shear stresses. In Wong and Reece's [20] [21] version of the model, the equations developed to compute the shear stress and pressure along the contact patch between the wheel and the soil. There are many variations of the method that have been proposed over the years [20] [9], but one expression for the stresses as a function of the angle about the wheel is [2]:

$$\begin{aligned} \sigma &= \left(\frac{k_c}{b} + k_\phi \right) R^n (\cos \theta_w - \cos \theta_1) \\ \tau &= (c + \sigma \tan \phi) \left(1 - e^{-\frac{j_d}{K_d}} \right) \end{aligned} \quad (\text{A.3})$$

where θ_1 is the entry angle, c is the cohesion term, j_d is shear displacement as a function of the angle about the wheel, K_d is shear deformation, and θ_w is a function of the slip and exit angle, θ_2 . These are then integrated over the contact patch to get the forces and torques acting on the wheel [2]:

$$\begin{aligned}
F_n &= \int_S (\sigma_z + \tau_z) dS \\
F_t &= \int_S \tau_x dS \\
F_c &= \int_S \sigma_x dS \\
T_{rr} &= \int_S r(\tau - \tau_x) dS
\end{aligned} \tag{A.4}$$

A.2 Traditional Distinct Element Method (DEM)

The distinct (or discrete) element method was first proposed by Cundall and Strack [3] in an effort to model a granular medium by considering the distinct soil particles that make up the soil. The following is a brief description of the formulation of the method as proposed in [3].

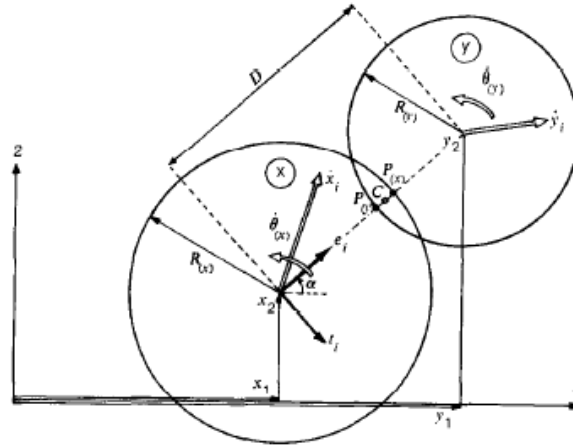


FIGURE A.2: Collision detection for circular particles in DEM method [3]

As shown in Figure A.2, the particles in this model are allowed to overlap. When the distance D between two particles is smaller than the sum of the radii of the particles, a contact is detected. The model then generates force that is directly proportional to the overlap distance and applies it to both particles. Thus, the normal and shear contact forces take the form [3]:

$$\begin{aligned}\Delta F_n &= k_n \Delta n = k_n \{(\dot{x} - \dot{y})e_i\} \Delta t \\ \Delta F_s &= k_s \Delta s = k_s \{(\dot{x} - \dot{y})t_i - (\dot{\theta}_x R_x + \dot{\theta}_y R_y)\} \Delta t\end{aligned}\tag{A.5}$$

where the resultant ΔF_n and ΔF_s are added to any existing contact forces from previous time-steps. What is essentially being modeled in Eq.(A.5) is a spring connecting any two particles that are in contact. Similarly, contact damping forces at time step N are added to the formulation as follows [3]:

$$\begin{aligned}(D_n)_N &= c_n \dot{n} = c_n [\dot{x} - \dot{y}]_{N-\frac{1}{2}} e_i \\ (D_s)_N &= c_s \dot{s} = c_s \left[(\dot{x} - \dot{y})_{N-\frac{1}{2}} t_i - (\dot{\theta}_x R_x + \dot{\theta}_y R_y)_{N-\frac{1}{2}} \right]\end{aligned}\tag{A.6}$$

where c_n and c_s are directly proportional to the stiffness factors - related by a constant β . Once all contact forces are determined for a time-step, accelerations are updated according to Newton's second law. One final inclusion that Cundall and Strack [3] adds to their model is the option to have global damping during the solving of Newton's equations for the updated accelerations:

$$\begin{aligned}m_x \ddot{x}_i &= \sum [(F_x)_i + (D_x)_i] - C \dot{x}_i \\ I_x \ddot{\theta}_x &= \sum M_x - C^* \dot{\theta}_x\end{aligned}\tag{A.7}$$

where C and C^* are damping coefficients. Since it's inception by Cundall and Strack [3], others have adapted this method specifically for terramechanics applications and modeling of wheel-soil contact forces [9] [38]. Iwashita et. al. [35] add an extra torsional spring and damper to create two more parameters available for tuning, and Smith et. al. [30] include an additional constitutive force to represent a cohesive force between particles.

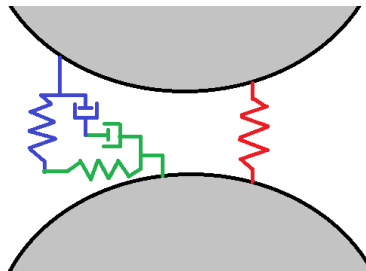


FIGURE A.3: Sample contact model for DEM

Appendix B

Plots of Outputs Versus Time

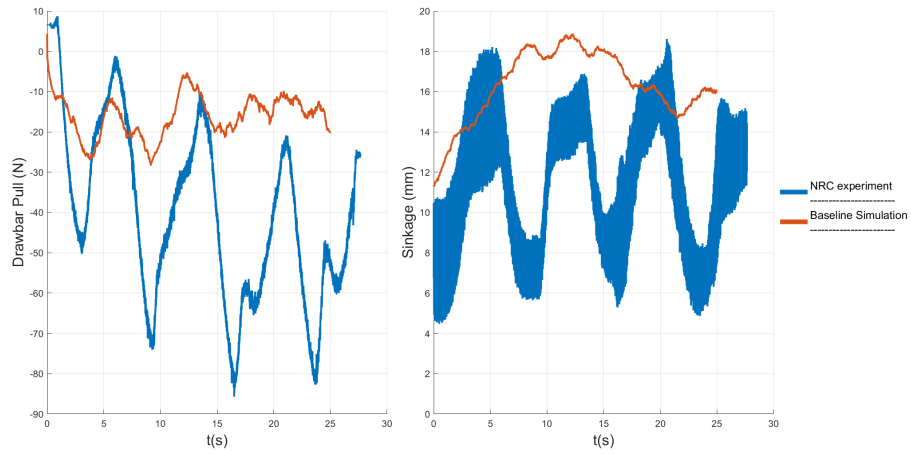


FIGURE B.1: DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Earth-g, 164N load, 10% slip

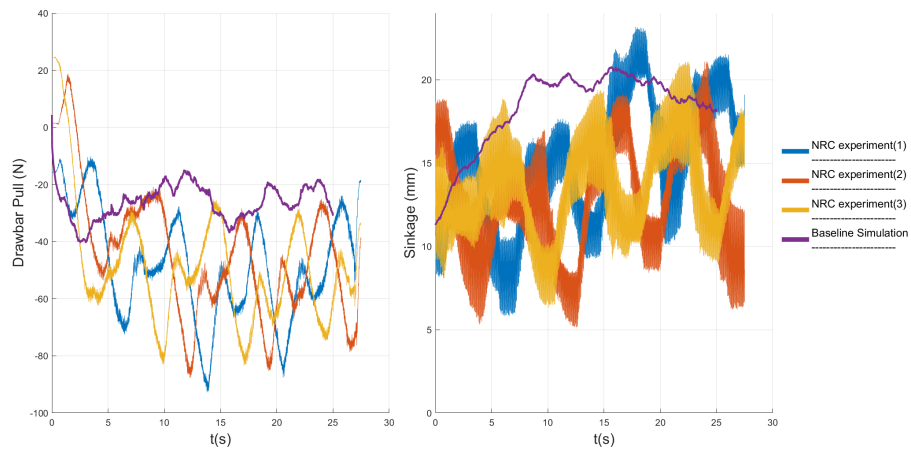


FIGURE B.2: DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Earth-g, 164N load, 20% slip

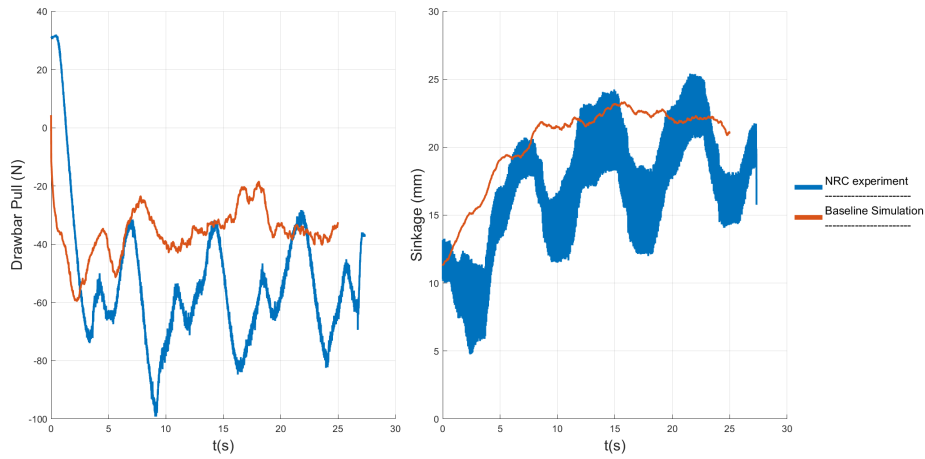


FIGURE B.3: DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Earth-g, 164N load, 30% slip

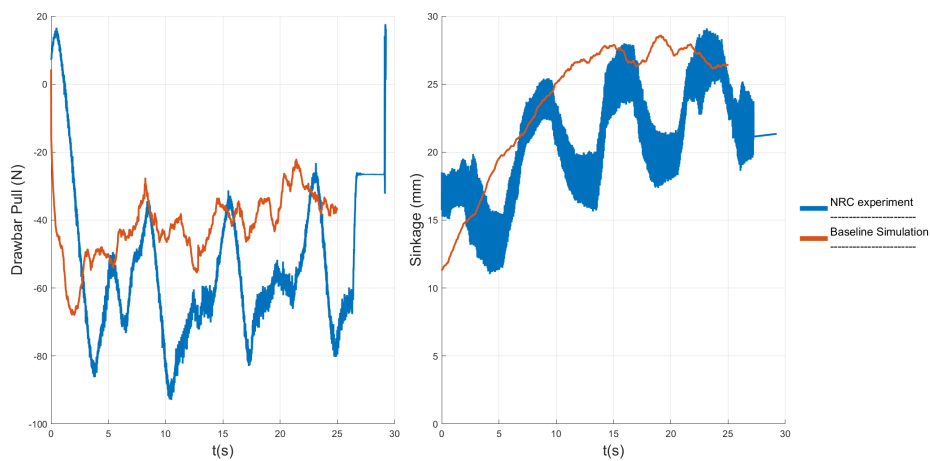


FIGURE B.4: DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Earth-g, 164N load, 40% slip

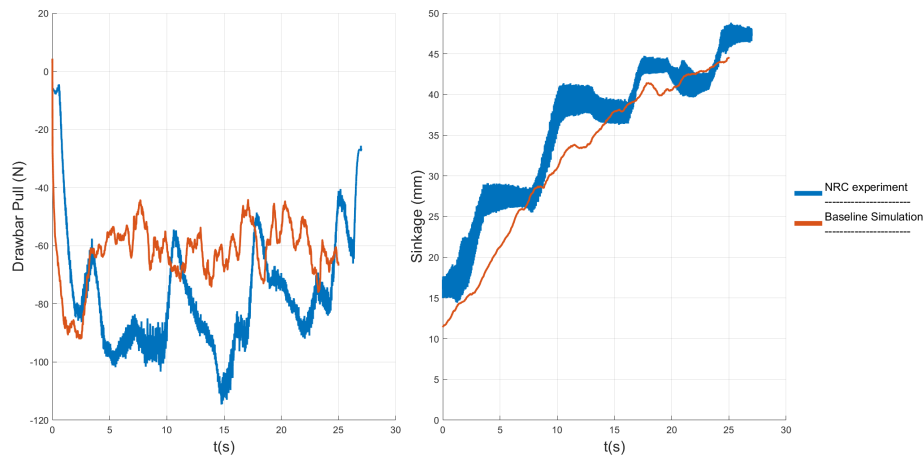


FIGURE B.5: DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Earth-g, 164N load, 70% slip

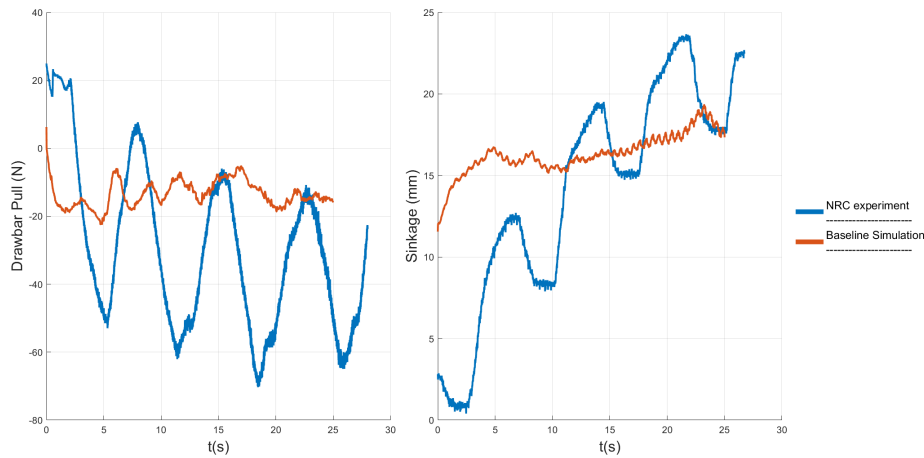


FIGURE B.6: DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Martian-g, 164N load, 10% slip

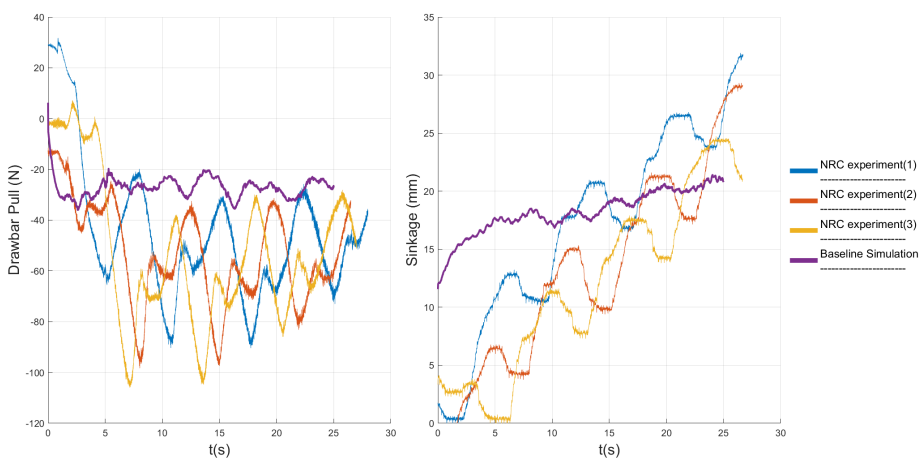


FIGURE B.7: DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Martian-g, 164N load, 20% slip

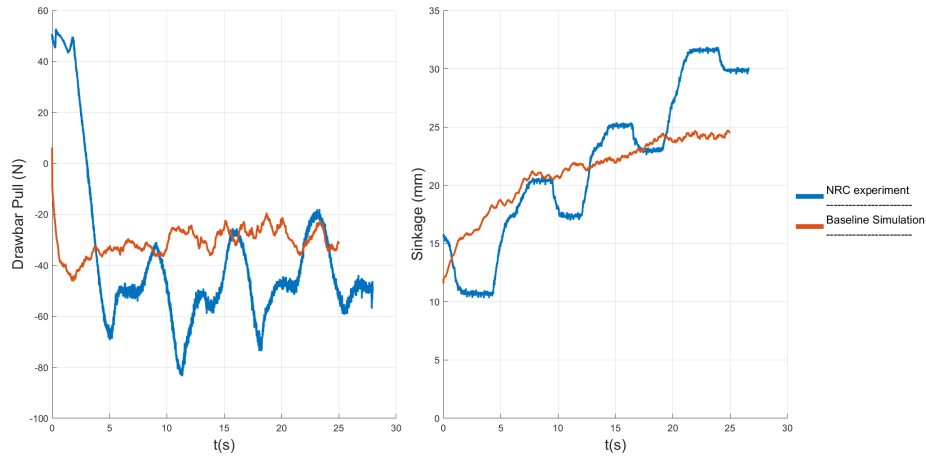


FIGURE B.8: DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Martian-g, 164N load, 30% slip

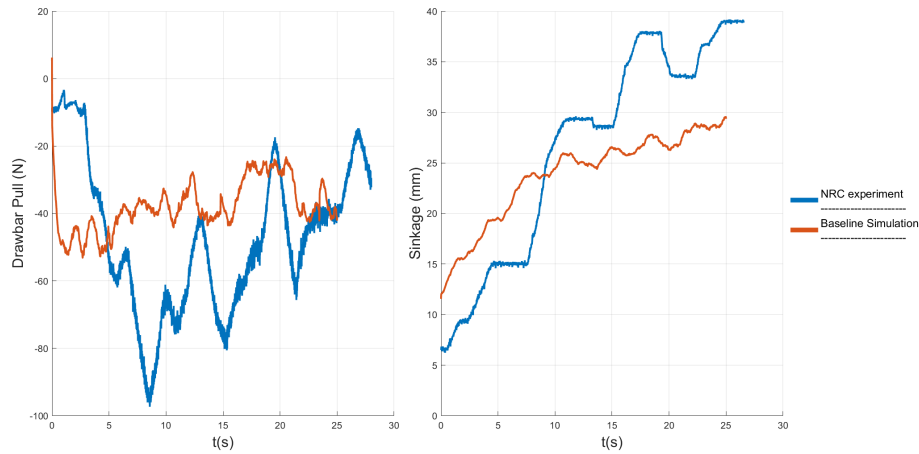


FIGURE B.9: DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Martian-g, 164N load, 40% slip

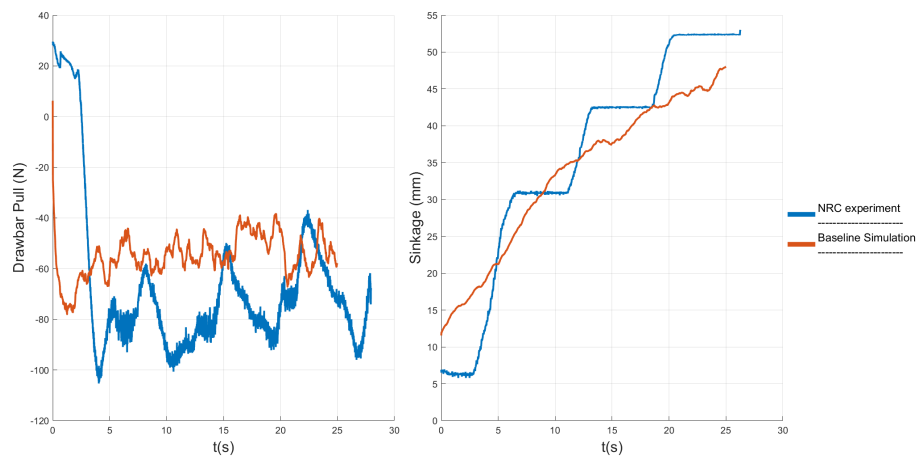


FIGURE B.10: DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Martian-g, 164N load, 70% slip

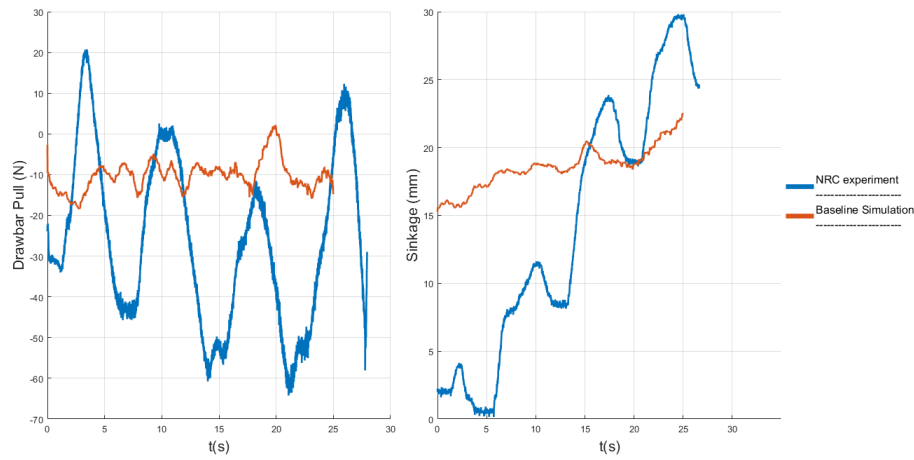


FIGURE B.11: DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Lunar-g, 164N load, 10% slip

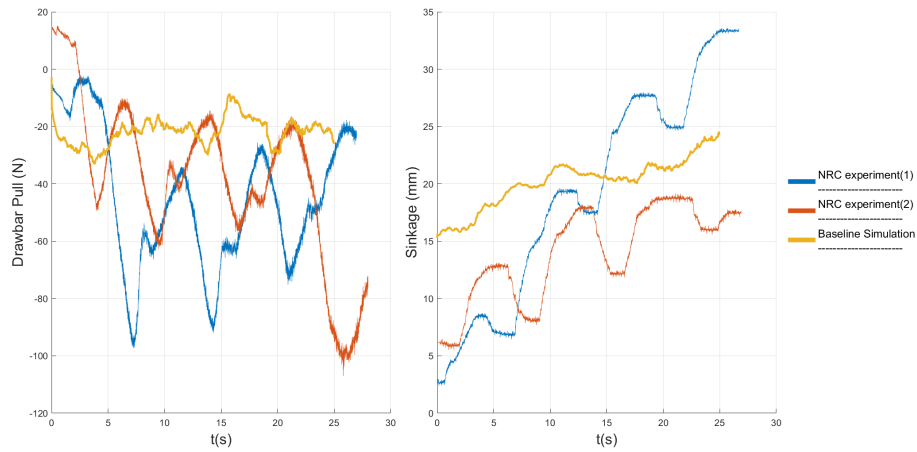


FIGURE B.12: DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Lunar-g, 164N load, 20% slip

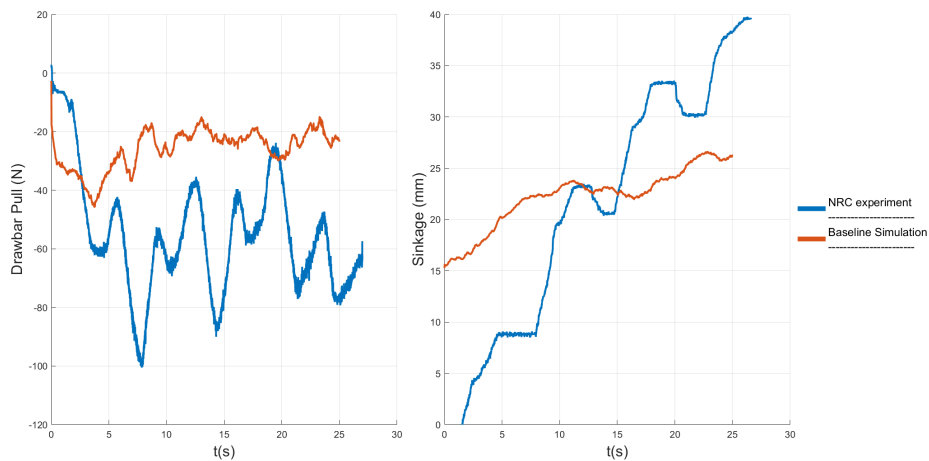


FIGURE B.13: DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Lunar-g, 164N load, 30% slip

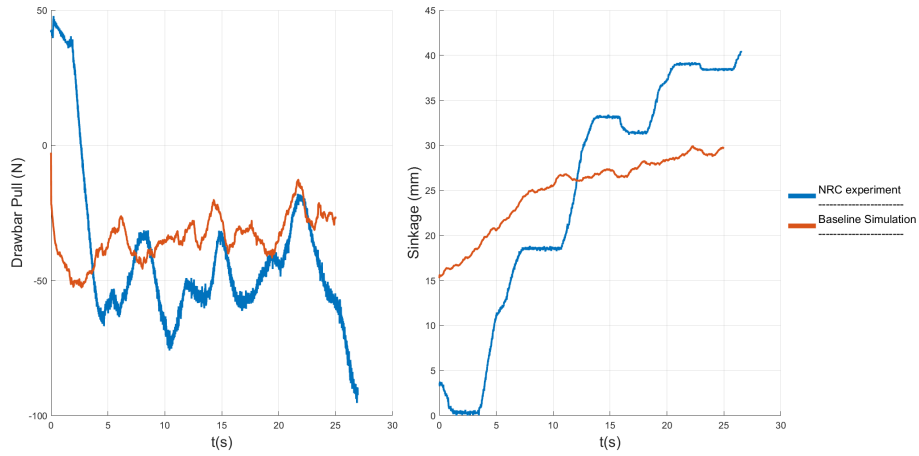


FIGURE B.14: DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Lunar-g, 164N load, 40% slip

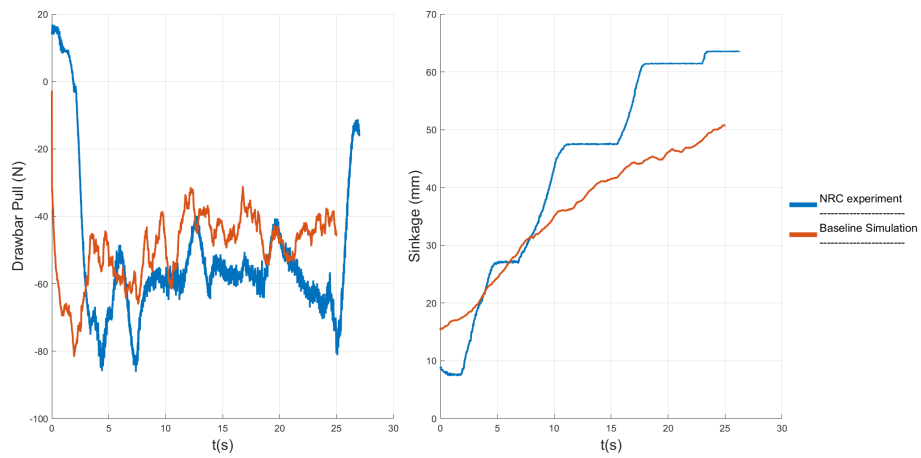


FIGURE B.15: DP/W (left) and max sinkage (right) as a function of time for baseline simulation parameters and NRC experiments: Lunar-g, 164N load, 70% slip

Bibliography

- [1] P. Niksirat, A. Daca, and K. Skonieczny. The effects of reduced-gravity on planetary rover mobility. *International Journal of Robotics Research*, 2019. (in review).
- [2] A. Gimenez, L. Kovacs, D. Holz, M Teichmann, and J. Kovecses. Dynamic simulation of wheeled vehicles: Models and algorithms. *Proceedings of the 19th International & 14th European-African Regional Conference of the ISTVS*, 2017.
- [3] P. A. Cundall and O. D. L. Strack. A discrete numerical model for granular assemblies. *Géotechnique*, 29(1):47–65, 1979.
- [4] D. Holz. Parallel particles (p2): A parallel position based approach for fast and stable simulation of granular materials. *Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS*, 2014.
- [5] M. G. Bekker. *Theory of land locomotion: the mechanics of vehicle mobility*. Ann Arbor: University of Michigan Press, 1956.
- [6] N. Patel, R. Slade, and J. Clemmet. Old earth pressure theories and new test results. *Engineering News Record*, 85:632, 1920.
- [7] M. G. Bekker. *Off-the-road locomotion: research and development in terramechanics*. Ann Arbor: University of Michigan Press, 1960.
- [8] M. G. Bekker. *Introduction to Terrain-Vehicle Systems*. Ann Arbor: University of Michigan Press, 1969.
- [9] J.Y. Wong. *Terramechanics and Off-Road Vehicle Engineering*. Butterworth-Heinemann, Reading, Massachusetts, 2009.

-
- [10] H. Wang, H.-J. Zhang, K. Wang, C. Zhang, L. Wang, and F. He. Application of terramechanics in off-road vehicle performance prediction. *Proceedings 5th International Conference on Mechanics and Mechatronics (ICMM)*, 2017.
 - [11] J. E. Schlegel and R. W. Morling. Optimum travel speed for maximum ploughing acreage. *Transaction of the ASAE*, pages 690–694, 1969.
 - [12] F. M. Zoz. Predicting tractor field performance. *Transaction of the ASAE*, pages 249–255, 1972.
 - [13] R. D. Wismer and H. J. Luth. Off-road traction prediction for wheeled vehicles. *Journal of Terramechanics*, 10(2):49–61, 1973.
 - [14] D. Gee-Clough, M. Mcallister, G. Pearson, and D. W. Everden. The empirical prediction of tractor-implement field performance. *Journal of Terramechanics*, 15(2):81–94, 1978.
 - [15] M. Gobbi, M. Aiolfi, M. Pennati, G. Previati, F. Levi, M. Ribaldone, and G. Mastinu. Measurement of the forces and moments acting on farm tractor pneumatic tyres. *Journal of Vehicle System Dynamics*, 43:412–433, 2005.
 - [16] H.B. Pacejka and E. Bakker. The magic formula tyre model. *Proceedings of the 1st Tyre Colloquium*, 1991.
 - [17] G. Meirion-Griffith and M. Spenko. An empirical study of the terramechanics of small unmanned ground vehicles. *Proceedings IEEE Aerospace Conference*, 2010.
 - [18] J. R. Kane, P. Ayers, H. Howard, A. Anderson, and D. Koch. Dynamics of viscoelastoplastic soil under a moving wheel. *Journal of Terramechanics*, 50(3):175–183, 2013.
 - [19] G. Dagan and M. Tulin. A study of the steady flow of a rigid-plastic clay beneath a driven wheel. *Journal of Terramechanics*, 6(2):9–27, 1969.
 - [20] J. Y. Wong and A. R. Reece. Prediction of rigid wheels performance based on analysis of soil-wheel stresses, part i. performance of driven rigid wheels. *Journal of Terramechanics*, 4(1):81–98, 1967.

-
- [21] J. Y. Wong and A. R. Reece. Prediction of rigid wheels performance based on analysis of soil-wheel stresses, part ii. performance of towed rigid wheels. *Journal of Terramechanics*, 4(2):7–25, 1967.
- [22] R. A. Irani, R. J. Bauer, and A. Warkentin. A dynamic terramechanic model for small lightweight vehicles with rigid wheels and grousers operating in sandy soil. *Journal of Terramechanics*, 48:307–318, 2011.
- [23] C. Senatore and C. Sandu. Off-road tire modeling and the multi-pass effect for vehicle dynamics simulation. *Journal of Terramechanics*, 48:265–276, 2011.
- [24] N. Patel, R. Slade, and J. Clemmet. The exomars rover locomotion subsystem. *Journal of Terramechanics*, 47:227–242, 2009.
- [25] J.V. Perumpral, J. B. Liljedahl, and W. H. Perloff. A numerical method for predicting the stress distribution under a tractor wheel. *Journal of Terramechanics*, 8(1):9–22, 1971.
- [26] T.J. Chung and J. K. Lee. Dynamics of viscoelastoplastic soil under a moving wheel. *Journal of Terramechanics*, 12(1):15–31, 1975.
- [27] R.N. Yong. Analytical predictive performance for physical performance of mobility. *Journal of Terramechanics*, 10(4):47–60, 1973.
- [28] R.N. Yong and E.A. Fattah. Prediction of wheel-soil interaction and performance using the finite element method. *Journal of Terramechanics*, 13(4):227–240, 1976.
- [29] A. Oida. Analysis of rheological deformation of soil by means of finite element method. *Journal of Terramechanics*, 21(3):237–251, 1984.
- [30] W. Smith and H. Peng. Modeling of wheel soil interaction over rough terrain using the discrete element method. *Journal of Terramechanics*, 50, 2013.
- [31] R. Briend, P. Radziszewski, and D. Pasini. Virtual soil calibration for wheel-soil interaction simulations using the discrete-element method. *Canadian Aeronautics and Space Journal*, 57(1):59–64, 2011.

-
- [32] Y.P. Cheng, Y. Nakata, and M.D. Bolton. Discrete element simulation of crushable soil. *Géotechnique*, 53(7):633–641, 2003.
- [33] J. M. Ting, M. Khwaja, L.R. Meachum, and J.D. Rowel. An ellipse-based discrete element model for granular materials. *International Journal for Numerical and Analytical Methods in Geomechanics*, 17:603–623, 1993.
- [34] J. M. Ting and J.D. Rowel. Ellipse-based discrete element model for granular materials: Validation testing. *Proceedings of the 10th Conference on Engineering Mechanics*, 2:1288–1291, 1995.
- [35] K. Iwashita and M. Odaz. Rolling resistance at contacts in simulation of shear band development by dem. *3rd Workshop in Virtual Reality Interactions and Physical Simulation VRIPHYS*, 124(3):285–292, 1998.
- [36] Z. Yan, S. K. Wilkinson, E. H. Stitt, and M. Marigo. Discrete element modelling (dem) input parameters: understanding their impact on model predictions using statistical analysis. *Comp. Part. Mech.*, 2:283–299, 2015.
- [37] K. Z. Y. Yen and T. K. Chaki. A dynamic simulation of particle rearrangement in powder packings with realistic interactions. *Journal of Applied Physics*, 71(7):3164–3173, 1992.
- [38] H. Tanaka, M. Momozu, A. Oida, and M. Yamazaki. Simulation of soil deformation and resistance at bar penetration by the distinct element method. *Journal of Terramechanics*, 37:41–56, 2000.
- [39] A. P. Grima and P. W. Wypych. Discrete element simulations of granular pile formation: Method for calibrating discrete element models. *International Journal for Computer-Aided Engineering and Software*, 28(3):314–339, 2011.
- [40] Z. Asaf, D. Rubinstein, and I. Shmulevich. Determination of discrete element model parameters required for soil tillage. *Soil and Tillage Research*, 92(1–2):227–242, 2007.
- [41] B. Lia, Y. Chen, and J. Chen. Modeling of soilclaw interaction using the discrete element method (dem). *Soil and Tillage Research*, 156:177–185, 2016.

-
- [42] M. Jiang, Y. Dai, L. Cui, and B. Xi. Experimental and dem analyses on wheel-soil interaction. *Journal of Terramechanics*, 76:15–28, 2018.
- [43] H. Nakashima, H. Fujii, A. Oida, M. Momozu, Y. Kawase, H. Kanamori, S. Aoki, and T. Yokoyama. Parametric analysis of lugged wheel performance for a lunar microrover by means of dem. *Journal of Terramechanics*, 50:153–162, 2007.
- [44] M.A. Knuth, J.B. Johnson, M.A. Hopkins, R.J. Sullivan, and J.M. Moore. Discrete element modeling of a mars exploration rover wheel in granular material. *Journal of Terramechanics*, 49(1):27–36, 2012.
- [45] M.A. Hopkins, J.B. Johnson, and R. Sullivan. Discrete element modeling of a rover wheel in granular material under the influence of earth, mars, and lunar gravity. *Earth and Space Conference 2008: Proceedings of the 11th Aerospace Division International Conference on Engineering, Science, Construction, and Operations in Challenging Environments*, 323, 2008.
- [46] W. Smith, D. Melanz, C. Senatore, K. Iagnemma, and H. Peng. Comparison of dem and traditional modeling methods for simulating steady-state wheel-terrain interaction for small vehicles. *7th Americas Regional Conference of the ISTVS, Tampa, Florida*, 2013.
- [47] M. Muller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position based dynamics. *Journal of Engineering Mechanics*, 2006.
- [48] T. Kobayashi, Y. Fujiwara, J. Yamakawa, Y., and K. Omine. Mobility performance of a rigid wheel in low gravity environments. *Journal of Terramechanics*, 47:261–274, 2010.
- [49] J.Y. Wong. Predicting the performances of rigid rover wheels on extraterrestrial surfaces based on test results obtained on earth. *Journal of Terramechanics*, 49: 49–61, 2012.
- [50] L. L. Kovacs, B. Ghotbi, F. Gonzalez, P. Niksirat, K. Skonieczny, and J. Kovecses. Effect of gravity in wheel/terrain interaction models. *Journal of Field Robotics*, 2019. (in review).

- [51] K. Skonieczny, S.J. Moreland, V.M. Asnani, C.M. Creager, Inotsume H, and D.S. Wettergreen. Visualizing and analyzing machine-soil interactions using computer vision. *Journal of Field Robotics*, 31:753–769, 2014.
- [52] K.Skonieczny, P. Niksirat, and A. Nassiraei. Rapid automated soil preparation for testing planetary rover-soil interactions aboard reduced-gravity aircraft. *Journal of Terramechanics*, 86:35–44, 2019.
- [53] C. Brunskilla, N. Patel, T. P. Gouache, G. P. Scott, C. M. Saaja, M. Matthews, and L. Cui. Soil simulant sourcing for the exomars rover testbed. *Journal of Terramechanics*, 48(6):419–438, 2011.
- [54] T. P. Gouache, N. Patek, C. Brunskill, G. P. Scott, C. M. Saaj, M. Matthews, and L. Cui. Soil simulant sourcing for the exomars rover testbed. *Planetary and Space Science*, 59(8):779–787, 2011.
- [55] B. Al-Hashemi and B. Al-Amoudi. A review on the angle of repose of granular materials. *Powder Technology*, 330:397–417, 2018.
- [56] Freitag D, A. Green, and K. Melzer. Performance evaluation of wheel for lunar vehicles. *Marshall Space Flight Center. National Aeronautics and Space Administration*, pages 1–53, 1970.
- [57] Y. Favaedi, A. Pechev, M. Scharringhausen, and L. Richter. Prediction of tractive response for flexible wheels with application to planetary rovers. *Journal of Terramechanics*, 48(3):199–213, 2011.
- [58] L. Karafiath. Soil-tire model for the analysis of off-road tire performance. Technical report, Grumman Aerospace Corporation, 05 1972.
- [59] E.C. Schmid. Interaction of vehicle and terrain results from 10 years research at ikk. *Journal of Terramechanics*, 32(1):2–26, 1995.
- [60] G. Taubin. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1115–1138, 1991.

- [61] H. P. Zhu, Z. Y. Zhou, R. Y. Yang, and A. B. Yu. Discrete particle simulation of particulate systems: A review of major applications and findings. *Chemical Engineering Science*, 63(23):5728–5770, 2008.
- [62] A. Gibbesch and B. Schafer. Multibody system modelling and simulation of planetary rover mobility on soft terrain. *Proc. of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space - iSAIRAS*, 2005.
- [63] Y. Tsuji, T. Kawaguchi, and T. Tanaka. Discrete particle simulation of two-dimensional fluidized bed. *Journal of Terramechanics*, 77:79–87, 1993.
- [64] R. Arvidson, J. Bell, P. Bellutta, N. Cabrol, J. Catalano, J. Cohen, L. Crumpler, D. DesMarais, T. Estlin, W. Farrand, R. Gellert, J. Grant, R. Greenberger, E. Guinness, K. Herkenhoff, J. Herman, K. Iagnemma, J. Johnson, G. Klingelhofer, R. Li, K. Lichtenberg, S. Maxwell, D. Ming, R. Morriss, M. Rice, S. Ruff, A. Shaw, K. Siebach, P. DeSouza, A. Stroup, S. Squyres, R. Sullivan, K. Talley, J. Townsend, A. Wang, J. Wright, and A. Yen. Spirit mars rover mission: Overview and selected results from the northern home plate winter haven to the side of scamander crater. *Journal of Geophysical Research: Planets*, 115(E7), 2010.
- [65] M.S. Rice, M. Reynolds, G. Studer-Ellis, J.F. Bell, J.R. Johnson, K.E. Herkenhoff, D. Wellington, and K.M. Kinch. The albedo of mars: Six mars years of observations from pancam on the mars exploration rovers and comparisons to moc, ctx and hirise. *Icarus*, 314(1):159–174, 2018.
- [66] S. Miyai, M. Kobayakawa, T. Tsuji, M. Sato, K. Imamura, and T. Tanaka. Dem study on vertical plate penetration into dry granular materials (influence of particle size). *Proc. of the joint 10th Asia-Pacific Conference of ISTVS and 39th Annual Meeting of Japanese Society for Terramechanics on Artificial Intelligence, Robotics and Automation in Space - iSAIRAS*, 2018.