

Accurate and Real-Time Granular Flow Modeling of
Robot-Terrain Interactions

Amin Haeri

A THESIS
IN
THE DEPARTMENT
OF
ELECTRICAL AND COMPUTER ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
CONCORDIA UNIVERSITY
MONTREAL, CANADA

SEPTEMBER 2021

© Amin Haeri, 2021

**CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES**

This is to certify that the thesis prepared

By: _____

Entitled: _____

and submitted in partial fulfillment of the requirements for the degree of

Doctor Of Philosophy

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

	Chair
	External Examiner
	External to Program
	Examiner
	Examiner
	Thesis Supervisor (s)

Approved by _____
Chair of Department or Graduate Program Director

Date of Defence _____
Dean,

Abstract

Accurate and Real-Time Granular Flow Modeling of Robot-Terrain Interactions

Amin Haeri, Ph.D.

Concordia University, 2021

An important challenge in robotics is understanding the interactions between robots and deformable terrains that consist of granular material. Granular flows and their interactions with rigid bodies still pose several open questions. A promising direction for accurate, yet efficient, modeling is using continuum methods. Also, a new direction for real-time physics modeling is the use of geometric machine learning. This research advances continuum and machine learning methods for modeling rigid body-driven granular flows, for application to space robotics (where the effect of gravity is an important factor) as well as terrestrial industrial machines.

For accurate and efficient design applications, this research develops a continuum method comprising a modern constitutive model, nonlocal granular fluidity (NGF), and a state-of-the-art numerical solver, material point method (MPM). We design a numerical approach, within a hyperelasticity framework, to implement the dynamical form of the viscoplastic NGF constitutive model in three-dimensional MPM. This approach is thermodynamically consistent, and the dynamical form includes the non-local effect of flow cessation. This is verified by both quantitative measurements and qualitative visualization of our excavation and wheel experiments. Furthermore, this research explores the gravity sensitivity of continuum numerical solvers. It explains why MPM is an appropriate continuum solver to model granular flows under different gravity.

For real-time control applications, this research considers the development of a subspace machine learning simulation approach. To generate training datasets, we utilize our high-fidelity MPM-NGF method. Principal component analysis (PCA) is used to reduce the dimensionality of data. This research shows that the first few principal components of our high-dimensional data keep almost the entire variance in

data. A graph network simulator (GNS) is trained to learn the underlying subspace dynamics. The learned GNS is then able to predict particle positions and interaction forces with good accuracy. More importantly, PCA significantly enhances the time and memory efficiency of GNS in both training and rollout. This enables GNS to be trained using a single desktop Graphics Processing Unit (GPU) with moderate Video-RAM. This also makes the GNS real-time on large-scale 3D physics configurations (and 700x faster than our MPM-NGF).

Acknowledgments

I sincerely thank my advisor, Prof. Krzysztof (Chris) Skonieczny. I have been very fortunate to have an advisor who gave me the freedom to explore my ideas, and at the same time guide me when I exactly needed it. I truly learned the art of decision-making and task prioritization from his insightful guidance.

Many thanks go to my collaborators Dominique Tremblay and Adriana Daca at our research group, and Daniel Holz and Marek Teichmann at CM Labs Simulations Inc. who contributed in many ways to the research work in this thesis. It was a pleasure to work with them.

The members of the Aerospace Robotics Laboratory, both past and present, have been wonderful colleagues, and I am especially thankful to Meysam Effati, Amir Nasiraei, Pierre-Lucas Aubin-Fournier, Jean-Sébastien Fiset, Shawn McGrory, Braden Stefanuk, Tyson Boer, and Parna Niksirat for their helpful opinions during our group meetings.

I would also like to thank Yuanming Hu (MIT), Alvaro Sanchez-Gonzalez (DeepMind), and David Henann (MIT) for making their source code available and for always answering my questions, which accelerated my research.

Special thanks to my parents and brother for their encouraging advice throughout my life. And last but certainly not least, I would like to thank my wife for her love and support during my study.

Contents

List of Figures	viii
List of Tables	xiii
1 Introduction	1
1.1 Motivation	1
1.2 Problem Statement	3
1.3 Related Work	3
1.3.1 Continuum Modeling of Granular Flows	3
1.3.2 Real-Time Physics Modeling	8
1.4 Contributions	10
1.5 Thesis Outline	11
1.6 Publications	11
2 Continuum Modeling of Granular Flows	13
2.1 Introduction and Background	13
2.1.1 Nonlocal Granular Fluidity	13
2.1.2 Material Point Method	16
2.2 Numerical Approach of Nonlocal Constitutive Model	23
2.2.1 Implementation	23
2.2.2 Validation	29
2.3 Gravity Sensitivity of Continuum Numerical Solvers	38
2.3.1 Reduced-gravity TC flow experiment	39
2.3.2 Analytical Method for reduced-gravity TC flow	41
2.3.3 Finite Element Method for reduced-gravity TC flow	43

CONTENTS

2.3.4	Material Point Method for reduced-gravity TC flow	44
2.3.5	Discussion	49
3	Real-Time Physics Modeling	52
3.1	Introduction and Background	53
3.1.1	Graph Neural Networks	53
3.1.2	Principal Component Analysis	59
3.2	Subspace Graph Network Simulator	61
3.2.1	Data	63
3.2.2	Reduced Data	66
3.2.3	Learning	68
3.2.4	Rollout	73
4	Conclusion	83
4.1	Limitations and Future Work	85
	List of Notations	104

List of Figures

1.1	Flow regimes survey of Martian and Lunar rovers with different grain diameters, gravities and driving speed (qs and inter stand for quasi-static and intermediate regimes).	4
1.2	Comparison of constitutive models for intermediate regime. The *-symbol indicates that the feature is an addition to the base model. . .	6
1.3	Comparison of methods for granular flow modeling, indicating which feature(s) each method has.	8
2.1	Simulation procedure (i.e. algorithm) overview for material point method (MPM) with nonlocal granular fluidity (NGF) model.	18
2.2	Left: Near-rigid-boundary particle (blue square) with compatible (black circles) and incompatible (red circles) grid nodes in its kernel support in 2D. Red squares and red unfilled square are rigid body's particles and COM. Blue area represents area of particles with the same kernel. Right: Transferring particle information to grid nodes inside particle's kernel (9 nodes for quadratic B-Spline interpolation in 2D)	21
2.3	Possible states for a particle. Red (1): under compression, light blue (2): newly stress-free, blue (3): stress-free, light red (4): newly under compression. Gray area represents granular material.	24
2.4	Overview (left) and pseudo code (right) of our nonlocal granular fluidity numerical approach for MPM.	26
2.5	Top: Trial timestep illustration while plastic flow is frozen. Bottom: Particle state illustration with logarithmic Hencky elastic strain in 1D.	27
2.6	Left: Robotic equipment and experiment setup. Right: Wheel-soil model in MPM (right).	31

LIST OF FIGURES

2.7	Average normalized subsurface soil velocity magnitude between 5 and 6 second marks from experiment (left) and MPM-NGF (right) with 20% (top) and 70% (bottom) slips.	32
2.8	Left: Interaction forces (drawbar pull and wheel load) and sinkage from experiment and MPM-NGF with 20% slip. Right: Average drawbar pull to wheel load ratio and maximum sinkage at different slips from experiment and MPM-NGF.	33
2.9	Robotic equipment and experiment setup.	34
2.10	Geometries of soil deformations in MPM-NGF and experiment for 5-cm depth.	34
2.11	Interaction forces from MPM-NGF and experiment for 2-cm (left) and 5-cm (right) depths.	35
2.12	Excavation modeling with MPM-NGF (left) and MPM-DP (right). Soil modeled with DP experiences unexpected fluid-like behaviors in untouched areas with no accumulating pile.	36
2.13	Industrial excavator modeled via MPM-NGF.	36
2.14	Top: Velocity field in 3D silo. Bottom: Plastic shear strain rate field shown in a thin layer of 3D silo.	37
2.15	Taylor Couette cell, with chosen top (blue) and bottom (red) reference surfaces. The inner cylinder is moving and the outer cylinder is fixed.	39
2.16	Comparison of normalized angular velocity profiles for the top and bottom surfaces in One-G (left) and Micro-G (right). The velocity profiles shown only extend up to 10 particle diameters from the inner cylinder as further away there is very little particle motion [Murdoch et al., 2013a].	40
2.17	Normalized angular velocity versus normalized distance from inner cylinder in One-G and Micro-G from AM and experiment for top and bottom surfaces.	42
2.18	Normalized angular velocity profiles in One-G and Micro-G from FEM and experiment for top and bottom surfaces	43
2.19	Tangential velocity ($v_\theta = r\omega$) in r-z plane in both One-G (left) and Micro-G (right) conditions from FEM	44

LIST OF FIGURES

2.20	Normalized angular velocity profiles in One-G and Micro-G from MPM with NGF model and experiment for top and bottom surfaces.	45
2.21	Normalized angular velocity profiles in One-G (left) and Micro-G (right) from MPM versus experiment for top and bottom surfaces.	46
2.22	Comparison of normalized radial velocity in One-G from MPM with experiment for top surface. No data is available from microgravity experiment.	46
2.23	Shear band illustration in One-G (left pair) and Micro-G (right pair), based on particle normalized velocity magnitude at top (left) and bottom (right) at the 22 second mark from MPM versus experimental images [Murdoch et al., 2013a].	47
2.24	Moving areas in vertical plane, in One-G (left) and Micro-G (right) from MPM.	47
2.25	Normalized angular velocity profiles in One-G and Micro-G from MPM with DP model and experiment for top and bottom surfaces.	48
2.26	Shear reversal results for top surface in One-G (top) and Micro-G (bottom).	48
2.27	Normalized mean pressure field in vertical (i.e. r-z) plane in MPM and FEM.	50
2.28	Mean inertial number field in vertical (i.e. r-z) plane in MPM and FEM.	50
3.1	A sample graph with nodes, directed edges, and features.	54
3.2	Illustration of one-step computational graph of node A in graph shown in Figure 3.1.	56
3.3	Algorithm overview of full graph network [Battaglia et al., 2018].	58
3.4	Training in our subspace GNS. Data X includes time series of flow states S , rigid states R , and interaction forces F . Also, S' and Z are reduced flow states and reduced system states. Here, states are particle positions.	62

LIST OF FIGURES

3.5 Rollout in our subspace GNS. Initial Data $X_{(0)}$ includes initial flow states S_0 and (initial) rigid states $R_{(0)}$. Also, S'_0 and $Z_{(0)}$ are reduced initial flow states and reduced (initial) system states. In the end, results X include time series of flow states S , rigid states R , and interaction forces F . Also, S' and Z are reduced flow states and reduced system states. Here, states are particle positions. 63

3.6 Side view (cross-section) of some Excavation and Wheel training examples with various initial and boundary conditions generated via MPM. 65

3.7 **Outset:** Normalized cumulative sum of the transformation matrix eigenvalues (modes) representing their energy. **Inset:** Position MSE between original full data and mapped-back full data. Errors bar represent one standard deviation. 67

3.8 Training loss convergence in ~ 15 M steps (left) and steady-state rollout error of positions for both Excavation and Wheel. 72

3.9 Distribution of PCA and GNS position MSEs in Excavation and Wheel configurations. Note that the varying width of bars is only due to logarithmic scaling of the x-axis. 75

3.11 Ground truth and predicted Excavation and Wheel interaction forces. They are plotted in forward (drawbar pull), vertical (wheel load), and lateral directions. 76

3.10 Side views of the Excavation and Wheel (cross-section) ground truth and predicted simulations in time (at 1, 3, and 5 second marks). Red to black colors indicate the particle velocity magnitudes. Arrows with blue to green colors represent directions and magnitudes of interaction forces. 77

3.12 Underlying subspace dynamics predicted by GNS in time for both the Excavation and Wheel configurations (vs ground truth). Subspace particle velocity magnitudes and fullspace interaction forces are visualized. 78

3.13 Underlying fullspace dynamics predicted by GNS at 3 second mark for both the Excavation and Wheel configurations (vs ground truth). Fullspace particle velocity magnitudes and interaction forces are visualized. 78

LIST OF FIGURES

3.14	Side views of 6 Excavation test examples at 3 second mark. Red to black colors indicate the particle velocity magnitudes. Arrows with blue to green colors represent directions and magnitudes of interaction forces.	81
3.15	Side views of 6 Wheel test examples at 3 second mark. Red to black colors indicate the particle velocity magnitudes. Arrows with blue to green colors represent directions and magnitudes of interaction forces.	81
3.16	Interactions forces in 9 Excavation test examples. Each plot contains forces in forward, vertical and lateral directions.	82
3.17	Interactions forces in 9 Wheel test examples. Each plot contains forces in forward, vertical and lateral directions.	82

List of Tables

2.1	Material parameters for NGF model. First set members are for excavation, industrial excavator, wheel and silo models (unless otherwise noted), and second ones are for Taylor-Couette flow model. Also, $A = 0.48$ and $I_0 = 0.278$	30
2.2	Specifications for simulations of 20-sec 5-cm depth excavation, 6-sec industrial excavator, 20-sec 20% slip wheel, 12-sec silo, and 22-sec One-G Taylor-Couette (TC) flow. Mean percentage errors (MPE) in excavation forward force, wheel load, and TC flow angular velocity are specified relative to experiments. * indicates excavation model with relaxed Young’s modulus for significant computational efficiency yet acceptable accuracy.	30
2.3	Mean absolute percentage error (MAPE) of velocity field in TC flow from DP and NGF models relative to experiment in One-G and Micro-G for top and bottom surfaces.	48
3.1	Datasets specifications. Motion types include (1) a discrete ramp-horizontal-ramp motion, and (2) a continuous curved motion. Also, Grav., Fric., and Dia. stand for gravity, soil internal friction angle, and wheel diameter.	64
3.2	MPM (Physics), subspace GNS (Ours) and fullspace GNS (Ref) runtime comparison. GPU ¹ , GPU ² , CPU ¹ , and CPU ² are NVIDIA GeForce RTX 3080 10GB, NVIDIA Tesla P6 16GB, Intel Core i7-10700 2.9GHz 8-Core, and Intel Core i7-6700 3.4GHz Quad-Core, respectively. One CPU ¹ core and four CPU ² cores are used.	74

LIST OF TABLES

3.3	Test example specifications. Motion type 2 indicates blade has continuous curved motion. Also, Grav., Fric., and Dia. stand for gravity, soil internal friction angle, and wheel diameter.	76
3.4	Position MSE and force MPE of test examples in subspace GNS rollout. MSEs (mean squared error) and MPEs (mean percentage errors) are specified relative to MPM. Also, MPM force MPEs are relative to experiment. Forces are in forward and vertical directions in Excavation and Wheel respectively.	80

Chapter 1

Introduction

1.1 Motivation

Planetary rovers are space exploration robots designed to move over the surface of extraterrestrial bodies. Some rovers have been driven by astronauts while others have been partially autonomous robots. Rovers are made to collect scientific data and samples; they are extremely helpful for exploring the solar system. The Global Exploration Roadmap identifies Mars, the Moon, and asteroids as priority destinations [ISEC, 2013]. The terrains of Mars and the Moon consist of fine granular regolith with embedded rocks, as do those of a recently discovered class of “rubble-pile” asteroids [Fujiwara et al., 2006, Rozitis et al., 2014]. Understanding the nature of interactions with granular terrains is, thus, crucial to exploring these high priority destinations. Although rovers are advanced technologies, they have still encountered setbacks. The entrapment of the Mars Exploration Rover Spirit in soft regolith and the tears and punctures in the Mars Science Laboratory Curiosity rover’s wheels demonstrate some of the current challenges of such granular terrains.

On Earth, also, construction vehicles are extensively in contact with granular materials, both with their wheels or tracks and their digging implements. Modeling these interactions can be very useful for training operators or for eventually automating aspects of these operations. This work include a collaboration with CM Labs Simulations, who develops training simulators for heavy equipment operation, which

1.1. MOTIVATION

includes construction equipment and vehicles of all types. Their simulators are deployed at numerous sites and training facilities throughout Canada and the world. Accurate and real-time modeling of machine-terrain interactions is crucial to the fidelity of such training simulators.

A deeper understanding of robot-granular flow interactions can contribute to the design and control of robots. An important factor to study issues encountered in robot design and control is the effect of gravity. Recent work [Kovács et al., 2020] shows this can have significant impacts on rover traction. Repeated empirical experiments to study various aspects of each new robot-terrain interaction might be difficult and/or expensive, so computer modeling can contribute to this end. However, the accurate and efficient modeling of granular flows is still an area of open research.

Granular flows and their interactions with rigid bodies still pose several open questions. In fact, their modeling is complex as they can experience various solid-like, fluid-like and gas-like deformations in time. Current robot-terrain interaction models have insufficient predictive power for robot wheel design and mobility control, depend on many difficult to measure parameters, and/or have very high computational complexity. In terms of accuracy, one current direction of research is the discrete element method (DEM), which simulates contact mechanics for millions of individual granular particles. This state-of-the-art approach demonstrates promise in modeling planetary rover interactions. But it is so computationally intensive as to be infeasible to implement on a rover for online mobility control [Zhu et al., 2005], and for large physical domains can be untenably expensive even in offline design applications [Dunatunga and Kamrin, 2017]. On the other end of the complexity spectrum, several researchers today highlight the insufficient predictive power of classical terramechanics models as applied to planetary rover wheel design [Kumar et al., 2003, Lun et al., 1984], and their limitations to specific flow geometries [Haeri et al., 2020], though they are computationally efficient.

1.2 Problem Statement

A promising direction for modeling granular flows in robot-terrain interactions is using methods from continuum mechanics [Dunatunga and Kamrin, 2015] which compromise between computational efficiency and accuracy. Furthermore, a new direction in physics simulations is the use of geometric machine learning in real-time approaches [Sanchez-Gonzalez et al., 2018]. An efficient and accurate continuum method could provide extensive training data for such approaches. The objective of this research is to advance continuum and machine learning methods for modeling rigid body-driven granular flows, for application to space robotics as well as terrestrial industrial machines. For offline design applications, this research aims to develop a continuum-based modeling approach that captures reaction forces and the effect of gravity on granular flows, with accuracy and modest computational complexity. Also, for real-time control applications, this work considers the development of a subspace machine learning simulation approach.

1.3 Related Work

This section presents high-level related work from the literature on continuum methods for granular flow modeling and then on real-time physics modeling.

1.3.1 Continuum Modeling of Granular Flows

The two main aspects of continuum methods affecting their accuracy and efficiency are the choices of constitutive model and numerical solver which are investigated here.

Constitutive Models. Constitutive models are used to calculate the material internal forces. They relate stress and strain in continuum material modeling. The best models accurately represent such complex systems of physical interactions with reduced-dimensionality representations. For granular materials, they should generally cover quasi-static elastic (solid-like), intermediate (visco) plastic (fluid-like), and inertial disconnected (gas-like) regimes [Kamrin, 2010]. Thus, it is first important to determine the flow regimes and categorize them with regard to flow characteristics [Tardos et al., 2003]. A survey of past and present rovers and their corresponding

1.3. RELATED WORK

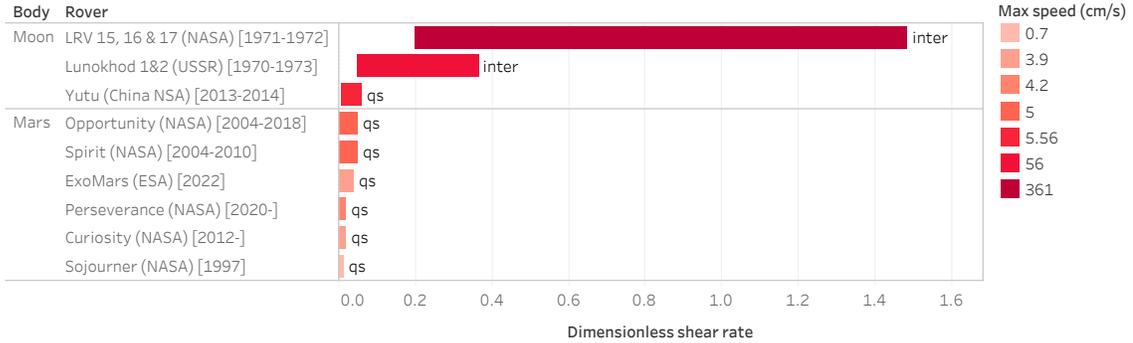


Figure 1.1: Flow regimes survey of Martian and Lunar rovers with different grain diameters, gravities and driving speed (qs and inter stand for quasi-static and intermediate regimes).

flow regimes is shown in Figure 1.1. The information is based on measuring the minimum and maximum dimensionless shear rate presented in [Tardos et al., 2003]. It depicts that all Mars rovers from 1997 to 2020 have been designed to work in the quasi-static regime. It also shows that some Lunar rovers move in the intermediate regime. Therefore, the most important regimes in planetary applications are quasi-static and intermediate regimes. A combination of the aforementioned flow regimes is also applicable to industrial excavation operations [Haeri et al., 2020].

For the two extreme regimes, elastic and disconnected, models have been developed based on soil mechanics [Schofield and Wroth, 1968, Nedderman, 1992, Elaskar and Godoy, 1998, Lagr e and Lhuillier, 2006] and the kinetic theory of gases [Lun et al., 1984], respectively. However, the middle (visco) plastic regime is more challenging where the granular material flows most like a liquid. For many years, for this regime, plastic models have been developed by combining various yield criteria and plastic flow rules [Drucker and Prager, 1952, Bonet et al., 2016, Kl ar et al., 2016]. In contrast, recently, from fluid mechanics literature, viscoplastic models have been made progress invoking a fluid-like flow approach with an appropriate yield criterion [Jop et al., 2006, Henann and Kamrin, 2013, Chen and Baladi, 1985].

In fact, for the elastic regime, Cauchy elasticity, hyperelasticity and hypoelasticity are the general techniques used and can be considered as a framework for the development of (visco) plasticity techniques. In Cauchy elasticity, the current state of stress depends only on the current state of deformation, and it must be combined

1.3. RELATED WORK

with a yield criterion (e.g. Mohr-Coulomb, Drucker-Prager, etc.). While this is the most common framework for granular materials (especially for soils) [Chen and Baladi, 1985], one issue with this linear model is that it may violate the energy equation of thermodynamics. In hyperelasticity, the nonlinear constitutive model is based on the existence of a strain energy function where thermodynamics laws are always satisfied. This framework keeps track of the deformation gradient and is widely used in the material point method (MPM) [Jiang et al., 2016]. Conversely in Hypoelasticity, the stress is a function of the current state of stresses and strains as well as their rates. Thus, hypoelastic models consider reversibility but they typically can not be formulated in a thermodynamically consistent manner [Dunatunga, 2014]. For the inertial regime, while models based on kinetic theory for gases can be used, it is also possible to define some saturation density functions to make granular particles free of stress [Dunatunga and Kamrin, 2015].

For the plastic part of constitutive modeling, incremental theory and/or the deformation theory [Chen and Baladi, 1985] can be used. In deformation theory, the relationship is between the plastic strain and the stress, while in incremental theory it is between the plastic strain rate and the stress and stress rate. Both theories assume the soil is elastic until the stress satisfies a yield criterion. Within the incremental theory, materials are divided into elastic-perfectly plastic and strain hardening ones. While the yield criterion in the former is a function of stress only, in the latter it is a function of plastic strain as well. In the flow theory in plasticity, the incremental change in plastic strain, $d\gamma$, is defined to include an elastic and a plastic part (denoted with e and p superscripts, respectively) as $d\gamma = d\gamma^e + d\gamma^p$.

For the plastic regime, flow theory provides a plastic flow rule relating the current stress and plastic strain and their incremental changes. This relation is generally assumed to be homogeneous and linear in the incremental changes of stress and plastic strain. This assumption ignores viscosity effects and makes the behavior time-independent [Chen and Baladi, 1985]. In this regime, however, granular materials have similarities with viscoplastic fluids such as Bingham and Herschel-Bulkley fluids [Jop et al., 2006] which are time- (or rate-) dependent as

$$\frac{d\gamma}{dt} = \frac{d\gamma^e}{dt} + \frac{d\gamma^p}{dt} = \dot{\gamma}^e + \dot{\gamma}^p. \quad (1.1)$$

In summary, plastic models can suffer from rate-independency [Chen and Baladi,

1.3. RELATED WORK

Models	Features		Hardening	Rate-dependency (viscosity)	Size-dependency (grain diameter)	Secondary rheology (quasi-static flow)	Flow cessation
Plastic	e.g. DP with Projection		✓*				
Viscoplastic	Local		✓	✓			
	Nonlocal (NGF)	Steady	✓	✓	✓	✓	
		Dynamical	✓	✓	✓	✓	✓

Figure 1.2: Comparison of constitutive models for intermediate regime. The *-symbol indicates that the feature is an addition to the base model.

1985] and in some cases, they may have issues with modeling strain hardening [Klár et al., 2016]. Viscoplastic models eliminate some numerical difficulties associated with plastic models, such as hardening [Abdullah, 2011]. Among viscoplastic constitutive models, although local models (e.g. [Jop et al., 2006]) lack robustness in their ability to predict all flow phenomena, the nonlocal models are accepted as highly predictive in different flow regimes [Kamrin, 2019]. Nonlocality brings the grain size effect [Kamrin, 2020], secondary rheology (where flow induces flow), and flow cessation (where flow stops even though the stress appears to be above the yield limit) into granular flows. Based on three criteria, discussed in [Kamrin, 2019], including ease of implementation, physical motivation, and predictive capability, the nonlocal granular fluidity (NGF) model [Kamrin and Koval, 2012, Henann and Kamrin, 2013, Henann and Kamrin, 2014] outweighs the other nonlocal models (partial fluidization theory [Aranson and Tsimring, 2001, Aranson and Tsimring, 2002, Volfson et al., 2003], I-gradient model [Bouزيد et al., 2013, Bouزيد et al., 2015], and self-activation model [Pouliquen and Forterre, 2009]). A feature comparison of the models for the intermediate regime is shown in Figure 1.2.

The hyperelastic three-dimensional FEM [Henann and Kamrin, 2016] and hypoelastic two-dimensional MPM [Dunatunga, 2014, Dunatunga, 2017] implementations of the viscoplastic steady NGF model have already provided promising results in various flow configurations. A hyperelastic numerical approach of the unsteady (also called dynamical) form of this model developed for MPM ensures the conservation of energy and the occurrence of flow cessation, respectively.

Numerical Solvers. In continuum mechanics, analytical methods (AM) such as [Lagrée and Lhuillier, 2006] seem to be efficient alternatives to numerical methods.

1.3. RELATED WORK

However, they might be inaccurate depending on the model simplification assumptions. These assumptions can even make the model inflexible to different conditions and configurations. Among the several continuum-based numerical methods to solve the governing equations of granular flows, finite difference, for fluid-like behavior, and finite element, for solid-like behavior, are the common base methods. Both methods can yield good results in certain cases. However, the finite difference method (FDM) has difficulties with extensional disconnection and static regimes, while the finite element method (FEM) has issues when mesh distortion becomes large [Dunatunga and Kamrin, 2015]. A feature comparison of the methods for granular flow modeling is shown in Figure 1.3. Note that there exist variants of these methods that can alleviate their issue(s) at the cost of additional computations and/or accuracy reduction. For instance, the mesh distortion issue in FEM can be addressed by re-meshing techniques [Khoei, 2005, Trädegård et al., 1998, Habraken and Cescotto, 1990].

MPM is a modern approach that combines the advantages of both FDM and FEM [Zhang et al., 2014] without having their disadvantages. In other words, MPM is similar to the FEM, but also incorporates the complementary advantage of FDM by keeping an undeformed Cartesian background grid appropriate for large-deformation problems as well [Zhang et al., 2014]. In addition to the grid, MPM consists of particles which carry information (mass and momentum) during the simulation. The particles can freely deform and, at the end of each time step, transfer the information to the grid nodes and vice versa. The governing equations in the weak form can be solved on the active grid nodes (the ones that have mass), identical to FEM. The grid, then, transfers the updated information back to the particles while it remains undeformed. Some recent lines of research have proved MPM as one of the most accurate and efficient methods for granular flow modeling [Dunatunga and Kamrin, 2017, Dunatunga and Kamrin, 2015, Hu et al., 2018, Klár et al., 2016]. Furthermore, instability and non-conserved (angular) momentum issues observed in the earlier variants including smoothed particle hydrodynamics (SPH) [Monaghan, 1992], reproducing kernel particle method (RKPM) [Liu et al., 1995], fluid implicit particle (FLIP) [Bridson, 2015], and particle-in-cell (PIC) [Harlow, 1964] are addressed and fixed in MPM by using proper particle-grid transfer schemes [Jiang et al., 2015, Fu et al., 2017].

1.3. RELATED WORK

Methods	Features	Accuracy		Efficiency	Flexibility
		Disconnection handling	Distorsion handling		
Terramechanics Models				✓	
DEM			✓		✓
Continuum Methods	FDM		✓	✓	✓
	FEM	✓		✓	✓
	MPM	✓	✓	✓	✓

Figure 1.3: Comparison of methods for granular flow modeling, indicating which feature(s) each method has.

From the efficiency point of view, MPM has two additional steps for transferring properties from particles to grid and vice versa. Furthermore, MPM can have more than one Gauss quadrature at each grid element (particles which are not attached to the grid). Thus, in MPM, computational cost per time step is higher than in FEM. However, MPM, by having constant grid element size (i.e. characteristic element length), requires less number of time steps for large-deformation problems, making MPM more efficient for these particular problems. In terms of accuracy, there might be some loss (dissipation error) in MPM transfer steps (it can be vanished using higher-order transfer schemes [Fu et al., 2017] while sacrificing efficiency). However, by not suffering from grid distortion, MPM can achieve higher accuracy for large-deformation problems in which static regimes may also possibly exist.

1.3.2 Real-Time Physics Modeling

Although continuum methods can be simultaneously accurate and efficient, they may not be real-time for 3D configurations with existing computer processors, especially in hardware-constrained applications like planetary rovers. Machine learning has begun to be applied to the problem of reducing the run time of simulating complex physics. A lower-dimensional representation for a mapping from inputs to simulation outputs can be obtained via dimensionality reduction methods. Further, deep learning neural networks are being used to learn such mappings. In the following, three important aspects of such approaches are reviewed.

Data. Since machine learning-based physics simulation methods require hours of

1.3. RELATED WORK

data at various initial and boundary conditions, numerical methods will be needed to complement and may even outweigh the use of experiments. To generate fluid simulations as training data, several lines of work have utilized smoothed particle hydrodynamics (SPH [Monaghan, 1992]) [Sanchez-Gonzalez et al., 2018], fluid implicit particle (FLIP [Bridson, 2015]) [Kim et al., 2019, Um et al., 2018], finite volume method (FVM) [Thuerey et al., 2020], or finite difference method (FDM) [Özbay et al., 2019]. Also, some general methods such as position based dynamics (PBD [Macklin et al., 2014]) [Holden et al., 2019, Li et al., 2019], and projective dynamics (PD [Bouaziz et al., 2014]) [Narain et al., 2016, Weiler et al., 2016] can be used for different types of material simulation (e.g. sand, snow, jelly). However, the continuum force-based MPM, has already been discussed in §1.3.1 as one of the most accurate and efficient methods for our purpose i.e. granular flow modeling. MPM [Hu et al., 2018] has also been recently used to produce sand simulations for the purpose of training a simulator [Sanchez-Gonzalez et al., 2018].

Reduced Data. In physics simulations, model reduction methods are utilized to capture the effective degrees of freedom of a physics system [Harmon and Zorin, 2013, Huang et al., 2011, Sifakis and Barbic, 2012, von Radziewsky et al., 2016]. This is with the hope to reduce both processing time and memory space usage corresponding to the system. Some work has also applied subspace methods to the equations of motion in FEM solvers [An et al., 2008, Pan et al., 2015]. However, it is difficult to handle collisions in such approaches, and further modifications [Teng et al., 2015] significantly reduce its efficiency. Recent literature [Holden et al., 2019] has shown that principal component analysis (PCA) could be an effective choice to capture the primary modes of elastic object deformations.

Learning. Machine learning (ML) approaches have been recently shown as real-time alternatives to traditional numerical methods to learn from data [Holden et al., 2019, Ladicky et al., 2017]. One crucial challenge here is to overcome the understanding of the underlying dynamics principles in the physical systems automatically. Some ML approaches have been applied to differential equations such as Poisson equation [Özbay et al., 2019]. Several recent ML-based approaches have also been utilized for fluid simulations, including: regression forest [Ladicky et al., 2017, Ladický et al.,

2015], multi layer perceptron (MLP) [Wiewel et al., 2019, Um et al., 2018], autoencoder (AE) [Thuerey et al., 2020, Wang et al., 2020, Morton et al., 2018] convolutional neural network (CNN) [Kim et al., 2019, Tompson et al., 2017], continuous convolution (CConv) [Ummenhofer et al., 2020], generative adversarial network (GAN) [Xie et al., 2018], and loss function-based method [Prantl et al., 2020].

These ML-based approaches are useful for a specific state of materials (gas-like, solid-like, or specifically fluid-like). To make them work in the other states, they need extra investigations and usually more different types of data. However, graph networks (GN) [Battaglia et al., 2018], a type of graph neural network [Scarselli et al., 2008], have been shown to be capable of simulating different states of materials while being simple to implement. Dynamic interaction network [Li et al., 2019], as a GN variant, has been proposed for general simulations including elastic and rigid objects. But, it also requires additional computations and data to simulate elastoplastic materials. In fact, the current position and velocity are no longer sufficient as inputs; additionally, the resting position and an extra network type (hierarchical modeling) are required. A very recent work on graph network simulator (GNS) [Sanchez-Gonzalez et al., 2018], has outperformed some other recent approaches [Li et al., 2019, Ummenhofer et al., 2020] in terms of accuracy and ease of implementation.

1.4 Contributions

The thesis makes the following principal contributions:

- Develops an accurate and efficient continuum method, 3D material point method (MPM) with hyperelastic dynamical nonlocal granular fluidity (NGF), for modeling granular flows.
- Identifies MPM as an appropriate numerical solver for the non-trivial task of capturing the effect of gravity, and explains why MPM can accomplish it.
- Generates experimental data via robotic equipment to quantitatively and qualitatively validate MPM, and generates large training datasets using this validated MPM.

- Enhances a real-time and high-accuracy machine learning approach, graph network simulator (GNS) with principal component analysis (PCA), for simulating large-scale rigid body-driven granular flows and predicting the interaction forces, while being trainable via single desktop GPU with moderate VRAM.

1.5 Thesis Outline

This thesis is organized in the following manner: Chapter 2 will first present a background on the nonlocal theory and material point method. Next, it will develop a hyperelastic numerical approach of the dynamical nonlocal granular fluidity for material point method (MPM). Then, the approach will be validated by multiple experimental test cases. The chapter will end by evaluating continuum numerical solvers in capturing the effect of gravity. Chapter 3 will first provide a background on graph neural networks (GNN), and principal component analysis (PCA). Then, it will develop various modules in the training phase of the subspace graph network simulator (GNS). The chapter will end by providing quantitative and qualitative roll-out results compared with the ground truth MPM results. Chapter 4.1 will conclude the thesis by summarizing the approaches presented in this thesis and suggesting potential further research topics. The source codes of the developed algorithms are available on GitHub as referred in the context.

Chapters 2 and 3 are meant to be stand-alone. These chapters will each present their own relevant background, theory, and result sections. Also, some symbols are reused and mean different things in each of these chapters. The symbols are hyperlinked in the context, and are summarized alongside their definitions in the List of Notations.

1.6 Publications

This thesis contains materials which were or will be published in the following (available here¹).

- Journal Papers

¹aminhaeri.com/publications

1.6. PUBLICATIONS

1. Haeri, A., Skonieczny, K. (2021). Three-Dimensional Granular Flow Continuum Modeling via Material Point Method with Hyperelastic Nonlocal Granular Fluidity. *Computational Mechanics*. Revision requested.
 2. Haeri, A., Skonieczny, K. (2021). Gravity Sensitivity of Continuum Numerical Solvers for Granular Flow Modeling. *Granular Matter*. Revision requested.
 3. Haeri, A., Skonieczny, K. (2021). Subspace Graph Physics: Real-Time Rigid Body-Driven Granular Flow Simulation. *Engineering Applications of Artificial Intelligence*. Submitted.
- Conference Papers
 1. Haeri, A., Tremblay, D., Holz, D., Teichmann, M., Skonieczny, K. (2021). Experimental Exploration And Real-time Simulation Of The Soil Flow In Blade-Soil Interactions. *International Society For Terrain-Vehicle Systems (ISTVS)*. Canada.
 2. Haeri, A., Skonieczny, K. (2021). Accurate and Real-time Simulation of Rover Wheel Traction. *IEEE Aerospace Conference*. US. **Best Paper In Software and Computing Track Award**.
 3. Haeri, A., Skonieczny, K. (2021). Granular Flow Modeling of Robot-Terrain Interactions In Reduced Gravity. *ASCE Earth and Space Conference*. US.
 4. Haeri, A., Tremblay, D., Skonieczny, K., Holz, D., Teichmann, M. (2020). Efficient Numerical Methods for Accurate Modeling of Soil Cutting Operations. *International Symposium on Automation and Robotics in Construction (ISARC)*. Japan.
 - Conference Oral Presentation
 1. Haeri, A., Skonieczny, K. (2019). Granular Flow Modeling of Robot-Terrain Interactions In Reduced Gravity. *IMA Conference on Dense Granular Flows*. University of Cambridge, UK.

Chapter 2

Continuum Modeling of Granular Flows

2.1 Introduction and Background

This section introduces the nonlocal theory and then derives the governing equations and procedure of material point method.

2.1.1 Nonlocal Granular Fluidity

A constitutive model known as the local model, introduced by [Jop et al., 2006], captures some characteristics of dense granular flows - including yield criterion and shear rate dependency - by an analogy to viscoplastic fluids (e.g. Bingham fluids). The local model proposes a relation for the friction coefficient as a function of the inertial number, I , for fast ($10^{-3} \lesssim I \lesssim 10^{-1}$) and dense granular flows, given by

$$\mu = \mu(I) = \mu_s + \frac{\mu_2 - \mu_s}{I_0/I + 1} \quad (2.1)$$

where μ_s is the static (minimum) friction coefficient that causes flow to happen, μ_2 and I_0 are the friction coefficient saturation value and a material constant to describe the material rheology, respectively. The inertial number, I , is given by

$$I = \dot{\gamma}^p \sqrt{\frac{\rho_s d^2}{p}} \quad (2.2)$$

2.1. INTRODUCTION AND BACKGROUND

where $\dot{\gamma}^p$, d , p , and ρ_s are the equivalent plastic shear strain rate, grain diameter, mean normal stress and grain density, respectively. Note that the volume fraction ϕ depends linearly on the inertial number $\phi = \phi(I)$ [Jop et al., 2005], but can be assumed to be constant for a large range of low I . This is a simplifying assumption for well-developed Lagrangian steady-state flows [Henann and Kamrin, 2013].

The inertial number can also be computed by the inverse of (2.1) and the linearization around μ_s [Dunatunga, 2014] given by

$$I = I_0 \frac{\mu - \mu_s}{\mu_2 - \mu_s}. \quad (2.3)$$

Using (2.2) and (2.3) gives the local equivalent plastic shear strain rate

$$\dot{\gamma}^p = \dot{\gamma}^p_{\text{loc}}(p, \mu) = \begin{cases} \sqrt{p/(\rho_s d^2)}(\mu - \mu_s)/b, & \mu \geq \mu_s \\ 0, & \mu < \mu_s. \end{cases} \quad (2.4)$$

This relation will be used in the NGF model presented in the following, where $b = (\mu_2 - \mu_s)/I_0$. Also, it shows that the yield criterion for having flow in the local model takes the form of Drucker-Prager criterion.

The local model works well in the intermediate regime ($\mu \geq \mu_s$); however, it captures no flow in the quasi-static regime ($\mu < \mu_s$), and has problems for slow ($I \lesssim 10^{-3}$) flows [Henann and Kamrin, 2013]. To capture flows in the quasi-static regime, a non-local granular rheology is proposed in [Henann and Kamrin, 2014] consistent with modern continuum thermodynamics in an elasto-viscoplastic context. In contrast to the previous constitutive models, in this model the stress depends on the strain and the gradient of strain as well. To consider the effect of nonlocality, this introduces a new scalar variable, g , called granular fluidity and is governed by the partial differential equation shown in (2.5). The granular fluidity (g) relates the equivalent plastic shear strain rate ($\dot{\gamma}^p$) to the friction coefficient (μ) via $g = \dot{\gamma}^p/\mu$.

$$t_0 \dot{g} = A^2 d^2 \nabla^2 g - (\mu_s - \mu)g - b \sqrt{\frac{\rho_s d^2}{p}} \mu g^2 \quad (2.5)$$

where $\nabla^2 g = \sum_{i=1}^3 \frac{\partial^2 g}{\partial x_i^2}$ and for a constant time-scale t_0 , a dimensionless material parameter called nonlocal amplitude A , the time domain t , the i -th component of the spatial domain \mathbf{x}_i .

2.1. INTRODUCTION AND BACKGROUND

This model can also be used in a steady-state form, proposed in [Kamrin and Koval, 2012] and [Henann and Kamrin, 2013] and shown in (2.6). It can be considered as an extension of the local model with modifications addressing its deficiencies.

$$\nabla^2 g = \frac{1}{\xi^2}(g - g_{loc}) \quad (2.6)$$

where $g_{loc} = \dot{\gamma}^p_{loc}/\mu$, $\mu = \tau/p$ and

$$\xi(\mu) = \frac{A}{\sqrt{|\mu - \mu_s|}}d \quad (2.7)$$

for shear stress τ , and the cooperativity length ξ which should be calibrated for each flow geometry by A . The cooperativity length is for plastic rearrangement, thereby imposing a length scale (d , i.e. grain diameter) on the flow.

The microscopic basis of the model is that flow induces flow; i.e., plastic deformations cause stress fluctuations that can induce plastic events in neighboring material. It makes the model work in the quasi-static flow regime in addition to the intermediate regime. An important feature to note is that when μ is very close to μ_s , the cooperativity length is very large. This suggests that when a granular material is in the quasi-static regime ($\mu \leq \mu_s$), but shear loaded, motion will diffuse over a large range through the network of granular contacts. A small perturbation at one location can be transmitted through this network and through force rearrangement of many of its neighbors [Henann and Kamrin, 2013].

The generic dynamical form of NGF is derived based upon some fundamental assumptions. The elastic strain is assumed to be small. This, as a kinematical restriction, assumes the rotation tensor R^e is the dominant part of the polar decomposition of the elastic deformation gradient, as opposed to the stretch tensor U^e , $F^e = R^e U^e \approx R^e$. The deformation gradient is defined by $F = \frac{\partial x}{\partial X}$ where $x = \chi(X, t)$ is the mapping function between the undeformed and deformed states. The position of points in the undeformed and deformed states is denoted by X and x , respectively. Furthermore, the plastic flow is assumed to be incompressible and irrotational. It considers the critical state of materials and ignores the transient dilations and compactions.

2.1.2 Material Point Method

Governing Equations. Among the conservation laws, mass conservation is automatically satisfied in MPM. Also, the symmetric stress tensor and hyperelasticity will satisfy the angular momentum and energy conservations, respectively. The strong-form unsteady conservation of linear momentum is considered as the governing equation given by

$$\phi\rho\dot{v} = \text{div}T + \phi\rho G \quad (2.8)$$

where \dot{v} is the material derivative¹ of velocity, T is Cauchy stress tensor, ϕ is volume fraction, G is gravitational acceleration, and ρ is again bulk density. MPM uses the weak formulation of the governing equation. It can be obtained by multiplying the strong form by the test function q and integrating by parts over the material (undeformed) domain Ω ; given by

$$\frac{1}{\Delta t} \int_{\Omega} \rho \Delta v q \, dV = \int_{\Omega} \rho G q \, dV - \int_{\Omega} T \nabla q \, dV \quad (2.9)$$

Here, the traction on the boundary is ignored by assuming zero first-order boundary conditions. In order to discretize the spatial terms in the governing equation, the Moving Least Squares (MLS) shape function [Hu et al., 2018] is used. It can speed up MPM by eliminating the need for explicitly calculating the weighting function derivative. Furthermore, it is consistent with the APIC (affine particle-in-cell [Jiang et al., 2015]) particle-grid transfer scheme in a way that MLS-MPM uses the $\frac{\partial v}{\partial x}$ quantity from APIC, required in the deformation gradient update, to reduce computational complexity.

The left hand side of equation (2.9) is first transferred from the continuum domain Ω to the particle domain Ω_p by

$$\int_{\Omega} \rho v q \, dV = \sum_p \int_{\Omega_p} \rho v q \, dV \quad (2.10)$$

Using the shape function, the continuous functions (v and q) are approximated by

¹ $D(\cdot)/Dt = d(\cdot)/dt + v\nabla(\cdot)$

2.1. INTRODUCTION AND BACKGROUND

the grid node information and it converts to the following form

$$\begin{aligned} \sum_p \int_{\Omega_p} \rho v q dV &\approx \sum_{p,i,j} m_p \Phi_i(x_p) \Phi_j(x_p) v_j q_i \\ &\approx \sum_{p,i} m_p \Phi_i(x_p) v_i q_i \end{aligned} \quad (2.11)$$

where i, j and p are the nodal and particle indexes, and $\Phi_i(x)$ is MLS shape function given by

$$\Phi_i(x) = \kappa_i(x_p) P^T(x - x_p) M^{-1}(x_p) P(x_i - x_p) \quad (2.12)$$

with P and κ_i as the polynomial basis and localized weighting function, respectively, and M defined as

$$M(x_p) = \sum_{i \in B_x} \kappa_i(x_p) P(x_i - x_p) P^T(x_i - x_p) \quad (2.13)$$

where B_x is a set of grid nodes satisfying $\kappa_i(x_p) \neq 0$. Here, the density function was simplified as $\int_{\Omega_p} \rho dV = m_p$. Mass lumping approximation [Jiang et al., 2016] was also used to eliminate $\Phi_j(x_p)$ by summing it over node j where $\sum_j \Phi_j(x_p) = 1$.

The derivative of the test function is required to be computed in the right hand side of the governing equation. Hence, adopting a linear polynomial basis $P(x - x_p) = [1, x - x_p]$ and quadratic B-splines as the weighting function $\kappa_i = N_i$, will give the test function derivative

$$\nabla q = M_p^{-1} N_i(x_p) (x_i - x_p) \quad (2.14)$$

Similarly, the right hand side can also be expressed in the particle domain.

$$\begin{aligned} \int_{\Omega} \rho G q dV - \int_{\Omega} T \nabla q dV = \\ \sum_p \left(\int_{\Omega_p} \rho G q dV - \int_{\Omega_p} T \nabla q dV \right) \end{aligned} \quad (2.15)$$

By substituting (2.14) in (2.15) and using the mass lumping, the following is obtained

$$\begin{aligned} \sum_p \left(\int_{\Omega_p} \rho G q dV - \int_{\Omega_p} T \nabla q dV \right) = \\ \sum_{p,i} [m_p G q_i - V_p M_p^{-1} T_p N_i(x_p) (x_i - x_p)] \end{aligned} \quad (2.16)$$

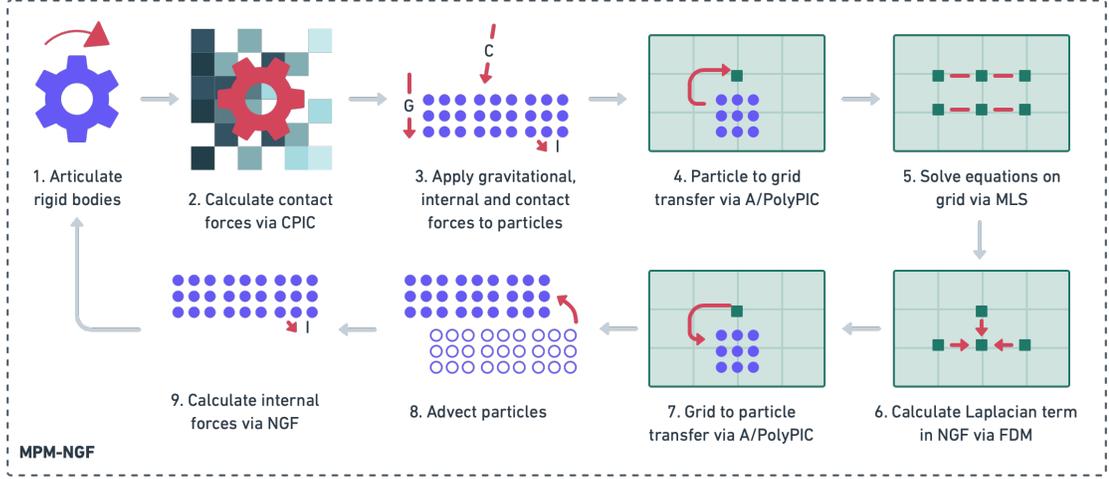


Figure 2.1: Simulation procedure (i.e. algorithm) overview for material point method (MPM) with nonlocal granular fluidity (NGF) model.

where V_p is the particle volume, for the quadratic B-splines $M_p^{-1} = 4/\Delta x^2$, and a one-point quadrature rule is used to express stress continuous function T at a particle (T_p).

Simulation Procedure. Figure 2.1 shows the procedure used to generate the MPM results. While Step 7 is not strictly required by the MPM procedure itself, it is introduced to compute the Laplacian term used in NGF model. Step 9, the calculation of particle internal forces via NGF model, will be explained in detail in §2.2. Upon initialization, after setting initial values: i) The 3D background is discretized by an Eulerian grid using a sparse paged grid [Setaluri et al., 2014] which stores the grid on a 1D memory span and uses Virtual Memory to avoid allocating unused regions. ii) Rigid bodies (e.g. inner and outer cylinders) are meshed and the elements get occupied by particles (at least one per element). iii) Deformable materials (e.g. granular material) are discretized by Lagrangian particles using Poisson Disk Sampler [Bridson, 2007], an efficient algorithm with linear time complexity ($O(n)$) that guarantees particles are no closer than a set distance.

Rigid body interaction (Steps 1 and 2). After initialization, the algorithm enters a loop which repeats every timestep (e.g. $1e4$ times per second). First, the center of mass (COM) and attitude of the rigid bodies are articulated using dynamics equations given either the scripted linear and angular velocities (as inputs) or the current

2.1. INTRODUCTION AND BACKGROUND

state of dynamic rigid bodies, alongside rigid collisions detection (if any). Then the contacts between rigid bodies and deformable materials' particles are detected using CPIC (Compatible Particle-In-Cell) [Hu et al., 2018]. This algorithm constructs a narrow-band colored unsigned distance field (CDF) around a rigid body to recognize whether grid nodes in near-boundary particle's support kernel are outside the rigid body (compatible) or not. Hence, the near-boundary particle normal vectors and their distances to rigid body are also estimated via the first-order MLS on the CDF constructed. With this, two-way particle-rigid body interactions are handled by applying relative impulses both on particles and on the COM of rigid bodies based on the Coulomb model of friction. For instance, equations (2.17) to (2.19) compute the impulse from a particle ($m_p \Delta v_{c_p}$) on a rigid body as shown in Figure 2.2(left). The opposite impulses from a rigid body ($m_r \Delta v_{c_r}$) can be derived with a similar approach for various contact conditions (e.g. sticky, slip, etc.).

$$\Delta v_{c_p} = N_i(x_p) [v_p - (v_r(x_i) + \max(0, |v_{rp,t}| + \mu v_{rp} \cdot n_p) v_{rp,t} / |v_{rp,t}|)] \quad (2.17)$$

where μ and n_p are the external friction coefficient and the normal vector from the surface of the rigid body to the particle, respectively; the subscripts p , i and r denote particle, grid node, and rigid body indices, and

$$\begin{aligned} v_{rp} &= v_p - v_r(x_i) \\ v_{rp,t} &= v_{rp} - (v_{rp} \cdot n_p) n_p \\ v_r(x_i) &= v_{com} + \omega_r \times (x_i - x_{com}) \end{aligned} \quad (2.18)$$

This impulse and also the particle internal force (computed in step 10) then contribute to the linear and angular velocity of the rigid body via

$$\begin{aligned} \Delta v_{com} &= m_r^{-1} [m_p \Delta v_{c_p} + f_p \Delta t] \\ \Delta \omega_r &= I_r^{-1} [(x_i - x_{com}) \times (m_p \Delta v_{c_p} + f_p \Delta t)] \end{aligned} \quad (2.19)$$

Particle momentum contribution (Step 3). Next, gravitational forces, contact forces, plus the internal forces which have been computed in the previous timestep, contribute to the momentum of particles via $(mv)_p + m_p \Delta v_p$ for

$$\Delta v_p = (G \Delta t + \Delta v_{c_r} + (f_p / m_p) \Delta t) \quad (2.20)$$

2.1. INTRODUCTION AND BACKGROUND

where the mass of a particle is neglected against the mass of the rigid body. This is a valid assumption for our applications where the flow is not highly dynamic.

Particle-to-grid transfer (Step 4). The background Cartesian grid is essential for accurately solving the momentum equation, calculating spatial derivatives (e.g. velocity spatial gradient and Laplacian term in unsteady nonlocal model), and detecting collisions. Thus, momentums should be transferred from particles to grid nodes. This is done by APIC (i.e. first-order PolyPIC, also called PolyPIC-4 in 3D) transfer scheme. The particle granular fluidity (g) introduced in [Henann and Kamrin, 2014] is also directly transferred via a zero-order scheme. This newly introduced scalar variable relates the equivalent plastic shear strain rate ($\dot{\gamma}^p$) to the friction coefficient (μ) via $g = \dot{\gamma}^p / \mu$.

PolyPIC with a quadratic grid interpolation in 3D, shown in 2D in Figure 2.2(right), transfers momentums (with $n_l = 4$) and granular fluidities (with $n_l = 1$) as follows

$$(mv)_i = \sum_p N_{ip} m_p \sum_l^{n_l=4} s_l C_{pl}, g_i = \sum_p N_{ip} g_p \quad (2.21)$$

where for simplicity in using quadratic B-Spline function $N_i(x_p) \equiv N_{ip}$. In contrast to FLIP (Fluid Implicit Particle) and PIC, APIC is stable and has smaller dissipation, respectively [Jiang et al., 2015]. While APIC may suffer from numerical friction, higher number of scalar modes n_l makes the transfer less dissipative and even lossless with $n_l = (n_b + 1)^d$ where $n_b = 2$ for the quadratic interpolation [Fu et al., 2017]. The generalized velocity modes s_l for APIC in 3D are given by

$$s_1 = 1, s_{2:4} = x_{1:3,i} - x_{1:3,p} \quad (2.22)$$

and the mode coefficients C_{pl} are calculated via equation (2.25) (from previous timestep) in grid-to-particle transfer section (for APIC). Note that the granular fluidity of open-state particles are not transferred.

Grid momentum update (Step 5) and Laplacian calculation (Step 6). During solving the momentum equation, the second-order derivative (Laplacian) term in the unsteady nonlocal model can be calculated with the help of background grid and a finite difference (FD) scheme. This technique has also been used in [Dunatunga, 2017, Stomakhin et al., 2014] as an alternative to the weak form of the finite element method (FEM). In fact, the FD scheme is used to reduce the stencil size and to avoid

2.1. INTRODUCTION AND BACKGROUND

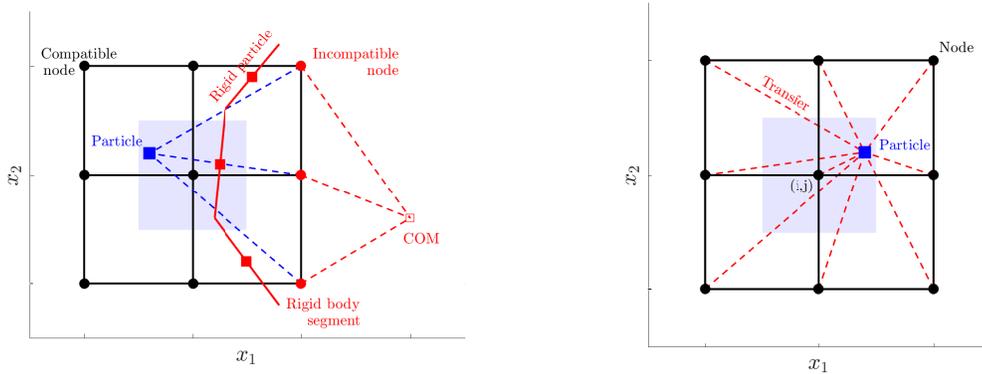


Figure 2.2: **Left**: Near-rigid-boundary particle (blue square) with compatible (black circles) and incompatible (red circles) grid nodes in its kernel support in 2D. Red squares and red unfilled square are rigid body’s particles and COM. Blue area represents area of particles with the same kernel. **Right**: Transferring particle information to grid nodes inside particle’s kernel (9 nodes for quadratic B-Spline interpolation in 2D)

checkerboard-type modes observed in FEM causing oscillations in the granular fluidity field. This term is obtained using a second-order central difference scheme as follows

$$\nabla^2 g_{i,j,k} = \frac{1}{\Delta x^2} (g_{i+1,j,k} + g_{i,j+1,k} + g_{i,j,k+1} - 6g_{i,j,k} + g_{i-1,j,k} + g_{i,j-1,k} + g_{i,j,k-1}) \quad (2.23)$$

For better consistency with the transfer scheme, a 27-point FD scheme can be used [Cheng et al., 2019] instead. It should also be mentioned that, for efficiency, this value is identical for each particle inside a grid element with the center node at (i, j, k) .

Grid-to-particle transfer (Step 7). Next, the boundary conditions can be applied to the boundary grid nodes, and the information on them is transferred back to particles via

$$\{mv, g\}_p = \sum_i N_{ip} \{mv, g\}_i \quad (2.24)$$

Here, the APIC mode coefficients are also computed for the next particle-to-grid

transfer as follows

$$\begin{aligned}
 C_{p1} &= \sum_{k=1}^{n_k} N_{i_k p} v_{i_k} \\
 C_{p2:4} &= \left(\sum_{k=1}^{n_k} N_{i_k p} v_{i_k} (x_{1:3,i_k} - x_{1:3,p}) \right) \frac{4}{\Delta x^2}
 \end{aligned}
 \tag{2.25}$$

where number of grid nodes in the kernel of a particle $n_k = 27$ for quadratic interpolation in 3D. Given equation (2.25), the gradient of quadratic B-Spline can be approximated by $C_{p_{apic}} = \sum_{l=2}^4 C_{pl}$ as follows

$$\nabla N_{ip} = \frac{C_{p_{apic}}}{\sum_i v_i} = \frac{4}{\Delta x^2} (x_i - x_p) N_{ip}
 \tag{2.26}$$

This gradient is reused in equation (2.14). It is also used as the velocity spatial gradient L so that

$$L = \nabla v_p = \sum_i v_i \nabla N_{ip} = C_{p_{apic}}
 \tag{2.27}$$

where this algorithmic optimization reduces the computational load [Hu et al., 2018].

Particle and rigid body advection (Step 8). At this point, particle positions are updated based on the updated velocities by $x_p^{n+1} = x_p^n + v_p^{n+1} \Delta t$. Also, dynamic rigid bodies are advected given their scripted motions and/or impulses from particles.

Particle viscoplasticity (Step 9). Now, while being in the particle loop, the internal forces between deformable material particles can be obtained via a constitutive model. For granular materials, the constitutive model can be the plastic Drucker-Prager model, the viscoplastic nonlocal granular fluidity (NGF) model, etc. Here, we aim to use the NGF model in MPM. It is due to the fact that NGF is more capable in modeling flows (due to viscoplasticity), and in capturing quasi-static regimes (due to nonlocality) than plastic and local models. The procedure and the TC flow results using both Drucker-Prager (DP) and NGF constitutive models are presented in §2.2. Note that in both models, the particles with negative pressure (i.e. tension) are set to be stress-free which turned out to be very important as will be discussed in §2.3. The similarity of the results shows that MPM ability to capture the gravity effect is invariant to the constitutive models.

2.2 Numerical Approach of Nonlocal Constitutive Model

In the case where inertial number $I \gtrsim 10^{-3}$, plastic models and FEM have problems in modeling the rate-dependent intermediate flows. Also, the viscoplastic local model has difficulty in modeling rate-independent quasi-static flows where $I \lesssim 10^{-3}$. Therefore, MPM with viscoplastic nonlocal granular fluidity (NGF) model can be a good candidate approach to model the entire flow.

Here, we develop a hyperelastic numerical approach for the implementation of the dynamical NGF in three-dimensional MPM. The hyperelasticity and dynamical form ensure the conservation of energy and the occurrence of flow cessation, respectively. This approach also includes the rules for modeling material particles in the inertial disconnected regime. The hyperelasticity framework and the finite difference (FD)-based NGF Laplacian term calculation emulate [Henann and Kamrin, 2016] and [Dunatunga, 2017], respectively. However, to be more accurate and consistent with our MPM particle-grid transfer scheme, we recommend a 27-point FD scheme for the Laplacian term. Additionally, some aspects of the handling of disconnected particles are from [Dunatunga, 2017] while we customize them for the hyperelasticity framework. The MPM code package utilized for generating the results is primarily from [Hu et al., 2018], combined in novel ways with additional elements from [Jiang et al., 2015, Setaluri et al., 2014, Bridson, 2007] written in C++ and Python. We implement our approach as an extension within this MPM for modeling granular flows and their interactions with rigid bodies. Our code is publicly available on GitHub here².

2.2.1 Implementation

One novelty in this research is the accurate calculation of internal forces via the dynamical form of viscoplastic Nonlocal Granular Fluidity (NGF) model [Henann and Kamrin, 2014] with hyperelasticity. In fact, this is a thermodynamically consistent version of the nonlocal theory for three-dimensional MPM. Hyperelasticity requires keeping track of the deformation gradient, which is multiplicatively decomposed into

²github.com/haeramin/MPM-NGF

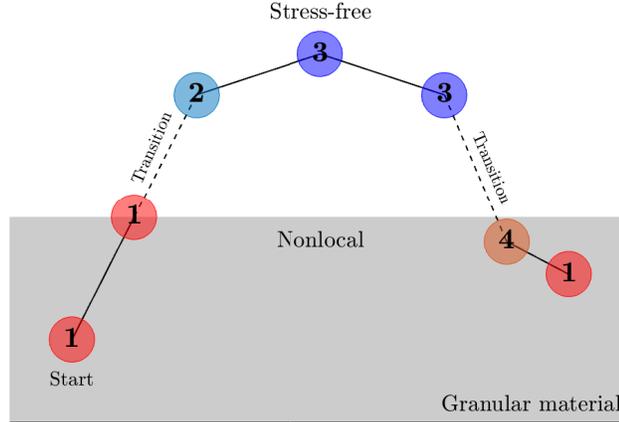


Figure 2.3: Possible states for a particle. Red (1): under compression, light blue (2): newly stress-free, blue (3): stress-free, light red (4): newly under compression. Gray area represents granular material.

elastic and plastic parts. The change rate of deformation gradient can be obtained by

$$\dot{F} = LF \quad (2.28)$$

where $L = \frac{\partial v}{\partial x}$ is approximated by the APIC transfer scheme. Thus, the total deformation gradient is updated by (2.28), and the elastic deformation gradient is calculated by $F^e = F(F^p)^{-1}$. The NGF constitutive model is hence utilized to calculate F^p . By assuming that the viscosity ($1/g$) is time-dependent, the PDE of the dynamical NGF model, shown in (2.5), should be solved for g . Then the equivalent plastic shear strain rate can be obtained via $\dot{\gamma}^p = g\mu$. Note that the friction coefficient μ is computable using the stress tensor, as shown in the details of (2.6).

Since in MPM granular materials can be separated, the disconnected particles should also be modeled. While kinetic theory of gases is capable of this modeling, in most cases it is accurate enough to handle granular gas via pure kinematics (stress-free). To detect this regime, pressure (mean normal stress) should be tracked for every individual particle. Figure 2.3 shows four possible states that can occur for a particle in the next timestep: 1) positive pressure (under compression) 2) first-time negative

2.2. NUMERICAL APPROACH OF NONLOCAL CONSTITUTIVE MODEL

pressure 3) negative pressure 4) first-time positive pressure. Particles in states 1 and 4 are under elasto-viscoplastic deformations (based on the NGF model); whereas particles with the states 2 and 3 are behaved as stress-free particles. However, as state 1 is distinguishable from state 4 based on prior state, and similarly 2 is from 3, the way particles separate from or return to the material should be modeled. Here, our focus is on the return phase, however, the algorithm is extendable for customizing the separation phase.

As depicted in Figure 2.4, first, the following steps are taken for particles in any state. The total deformation gradient is updated via the solution from the first-order forward Euler method applied to equation (2.28) – while one can use the analytical solution $\exp(\Delta t L)F$ – as follows ($\mathbf{1}$ is unity matrix)

$$F_{n+1} = F_n(\mathbf{1} + \Delta t L_{n+1}). \quad (2.29)$$

Now, with the current (i.e. at timestep $n+1$) total deformation gradient and by calling the previous (i.e. at timestep n) plastic deformation gradient, the elastic deformation gradient can be computed at trial timestep via

$$F_{tr}^e = F_{n+1} F_n^p{}^{-1}. \quad (2.30)$$

where the trial timestep refers to a middle timestep (i.e. between n and $n+1$) when the plastic deformation is frozen. Consequently, the trial shear stress differs from the current shear stress as shown in Figure 2.5(left) [Askari and Kamrin, 2016]. Now from the right polar decomposition of elastic deformation gradient, one can use the stretch part U^e to compute the Hencky elastic strain at the trial timestep, given by

$$E_{tr}^e = \ln(U_{tr}^e). \quad (2.31)$$

This logarithmic mapping, also shown in Figure 2.5(right), considers only particles with small elastic strain to be under elasto-viscoplastic deformation. Otherwise, it causes particles with large strain to be stress-free. This is consistent with the assumption made in the nonlocal theory as small elastic strain described before.

As suggested in [Henann and Kamrin, 2016], it is convenient to use an elastic Gibbs free energy; Mandel stress M^e is hence utilized to construct pressure p , shear stress τ , direction of plastic flow N^p , and Cauchy stress T . Trial Mandel stress is

2.2. NUMERICAL APPROACH OF NONLOCAL CONSTITUTIVE MODEL

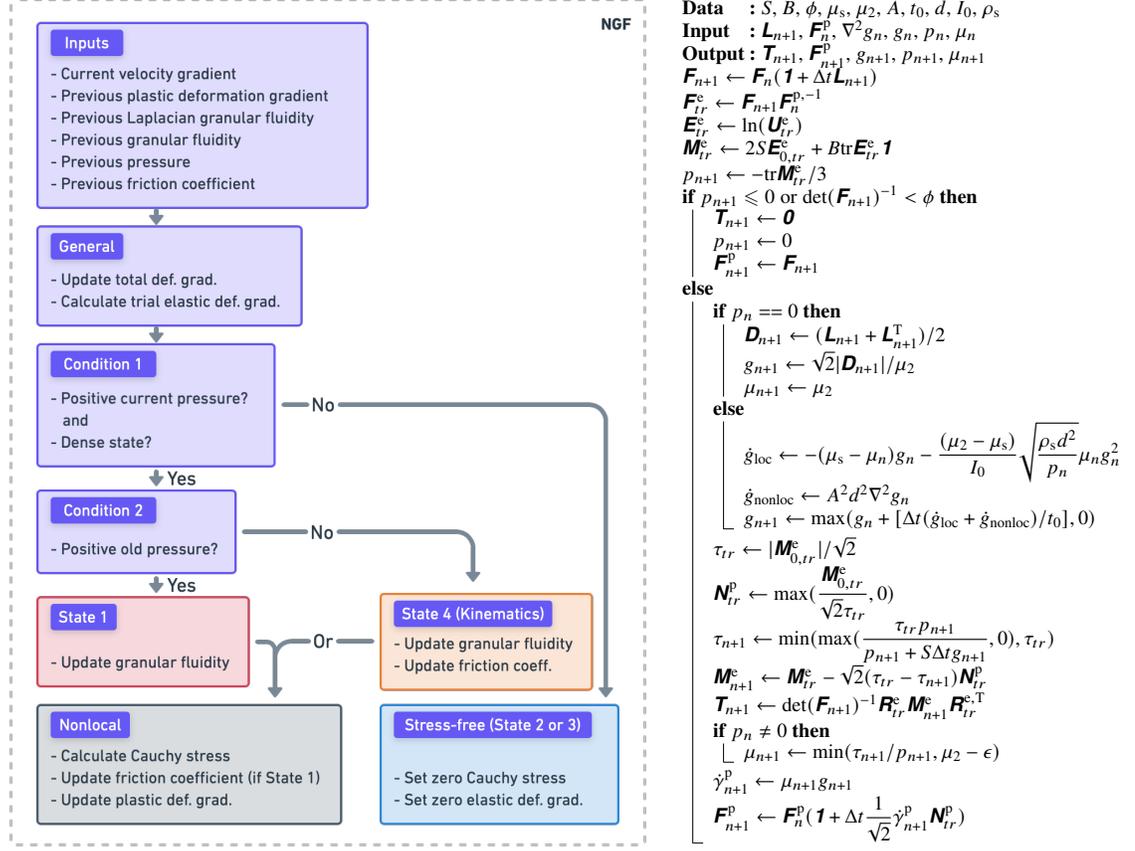


Figure 2.4: Overview (left) and pseudo code (right) of our nonlocal granular fluidity numerical approach for MPM.

computed by a common constitutive law in non-linear elasticity known as generalized Hookean law [Papadopoulos, 2017]

$$\mathbf{M}_{tr}^e = 2S \mathbf{E}_{0,tr}^e + B \text{tr} \mathbf{E}_{tr}^e \mathbf{1} \quad (2.32)$$

where E_0 is the deviatoric part of Hencky strain, and S and B are shear and bulk moduli. The current (and trial) pressure is finally computed by

$$p_{n+1} = p_{tr} = -\text{tr} \mathbf{M}_{tr}^e / 3. \quad (2.33)$$

Now depending on the updated pressure p_{n+1} , and the density change given by

$$J = \det(\mathbf{F}_{n+1}), \quad (2.34)$$

2.2. NUMERICAL APPROACH OF NONLOCAL CONSTITUTIVE MODEL

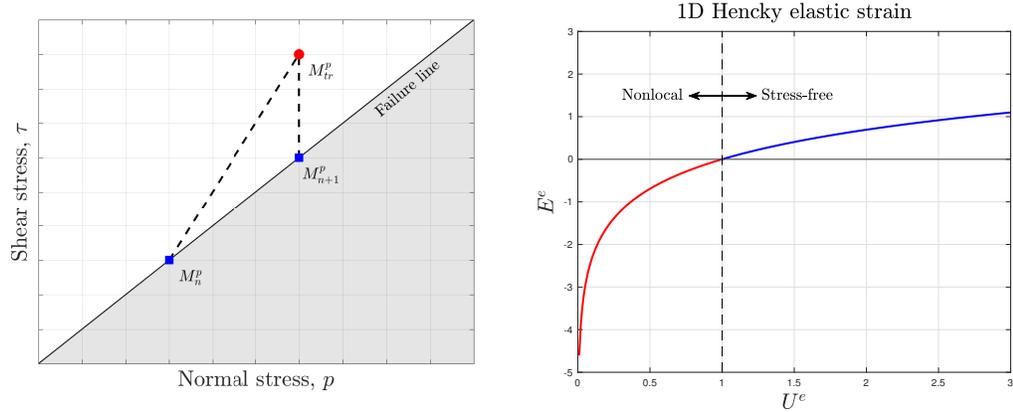


Figure 2.5: Top: Trial timestep illustration while plastic flow is frozen. Bottom: Particle state illustration with logarithmic Hencky elastic strain in 1D.

the state of the particle is determined. If the pressure is negative or the current density is below the bulk density ($J^{-1} < \phi$), the particle is in state 2 or 3. Thus, setting zero Cauchy stress tensor makes the particle stress-free and the motion is purely given by kinematics. In addition, as there would be no elastic deformation here, the current plastic deformation gradient is updated via the current total deformation gradient, $F_{n+1}^P = F_{n+1}$. Note, resetting the current pressure to zero is only for tagging the particle.

Otherwise, it means that the particle is in state 1 or 4. In the case that the previous state of the particle was stress-free ($p_n = 0$), the particle is now in state 4; and the granular fluidity should be calculated via kinematics given by

$$g_{n+1} = \sqrt{2}|D_{n+1}|/\mu_2 \quad (2.35)$$

where an additive decomposition of the velocity spatial gradient $L = D + W$ results in the stretch part $D = (L + L^T)/2$. In addition, the current friction coefficient is set to the friction coefficient saturation value (μ_2) [Dunatunga, 2017] based on the $\mu(I)$ -rheology [Jop et al., 2006].

However, if the previous pressure was non-zero (state 1), the granular fluidity is normally updated using equation (2.5) and the value of Laplacian term obtained from

equation (2.23) as follows

$$g_{n+1} = g_n + \Delta t [A^2 d^2 \nabla^2 g_n - (\mu_s - \mu_n) g_n - \frac{(\mu_2 - \mu_s)}{I_0} \sqrt{\frac{\rho_s d^2}{p_n}} \mu_n g_n^2] / t_0. \quad (2.36)$$

Then, the remaining associated trial quantities are calculated as follows

$$\begin{aligned} \tau_{tr} &= |M_{0,tr}^e| / \sqrt{2} \\ N_{tr}^p &= \frac{M_{0,tr}^e}{\sqrt{2} \tau_{tr}}. \end{aligned} \quad (2.37)$$

It is assumed that the deviator of Mandel stress and the direction of plastic flow are codirectional. Hence, the plastic flow rule is non-associated and prevents plastic volume change.

In order to compute the Cauchy stress, it is required to update shear stress via the updated granular fluidity, and then update Mandel stress by

$$\begin{aligned} \tau_{n+1} &= \frac{\tau_{tr} p_{n+1}}{p_{n+1} + S \Delta t g_{n+1}} \\ M_{n+1}^e &= M_{tr}^e - \sqrt{2} (\tau_{tr} - \tau_{n+1}) N_{tr}^p. \end{aligned} \quad (2.38)$$

Finally, the Cauchy stress is computed for particles in state 1 or 4 as follows

$$T_{n+1} = J^{-1} R_{tr}^e M_{n+1}^e R_{tr}^{e,T} \quad (2.39)$$

where R^e is the rotation tensor of the polar decomposition of the elastic deformation gradient. Now with the non-zero previous pressure the current friction coefficient is updated via

$$\mu_{n+1} = \tau_{n+1} / p_{n+1}, \quad (2.40)$$

otherwise, it remains μ_2 . The equivalent plastic shear strain rate is then calculated via

$$\dot{\gamma}_{n+1}^p = \mu_{n+1} g_{n+1} \quad (2.41)$$

which results in the plastic part of velocity spatial gradient [Henann and Kamrin, 2016] known as (visco)plastic flow rule given by

$$L_{n+1}^p = \frac{1}{\sqrt{2}} \dot{\gamma}_{n+1}^p N_{tr}^p. \quad (2.42)$$

Therefore, the updated plastic deformation gradient is given by the forward Euler solution of equation (2.28) as follows

$$F_{n+1}^p = F_n^p(\mathbf{1} + \Delta t L_{n+1}^p). \quad (2.43)$$

After all, the particle internal force can be calculated via the updated Cauchy stress tensor as $f_p = V_p \operatorname{div} T_p$. There are some extra instructions in the pseudo code shown in Figure 2.4(right) regarding programming limitations which were not included in the aforementioned steps.

2.2.2 Validation

In this section, the implemented NGF model is validated by our excavation experiments. We also compare the NGF results with the results from a recent modified Drucker-Prager plasticity model [Klár et al., 2016] for the excavation case. In addition, we model an industrial excavator and explain the three states the granular material simultaneously experiences in a practical case. Moreover, 3D wheel-soil and silo are modeled to further verify the behavior of the granular flow governed by this constitutive equation. Another validation is also done by simulations to compare to microgravity Taylor-Couette (TC) flow experiments [Murdoch et al., 2013a, Murdoch et al., 2013b, Murdoch et al., 2013c] (for more details see [Haeri and Skonieczny,]).

The simulations were performed on an Intel Core i7-6700 3.4GHz Quad-Core processor. Houdini Apprentice [SideFX, 2020] is utilized for the visualizations. From a high performance computing (HPC) point of view, multi-threading (via Intel TBB) and vectorization (via explicit SIMD) were utilized in the code, in addition to some algorithmic improvements explained in §2.1.2. These techniques make the current MPM 2x faster than a traditional MPM [Hu et al., 2018]. Although the NGF model appends an extra PDE to the governing equations, our numerical approach does not significantly increase the simulation run-time ($\approx 5\%$, relative to algebraic or parametric models).

The NGF parameters shown in table 2.1 are set based on the physical properties in our experiments and previous work [Dunatunga and Kamrin, 2015] except when noted otherwise. Some of the parameters are subject to the Von Neumann stability condition of the NGF model’s PDE (Poisson equation) where $\Delta t < (\Delta x^2 t_0)/(2A^2 d^2)$

2.2. NUMERICAL APPROACH OF NONLOCAL CONSTITUTIVE MODEL

S [MPa]	B [MPa]	ρ_s [kg/m ³]	d [mm]	t_0 [s]	μ_s	μ_2
{5.8, 1}	{12.5, 10}	{2583, 2550}	{0.3, 4}	1e-4	{0.70, 0.38}	{0.96, 0.64}

Table 2.1: Material parameters for NGF model. First set members are for excavation, industrial excavator, wheel and silo models (unless otherwise noted), and second ones are for Taylor-Couette flow model. Also, $A = 0.48$ and $I_0 = 0.278$.

Model	E [MPa]	Δt [s]	Δx [m]	Particle # (ppc)	Runtime [sec/sec]	MPE [%]
Excav.	15.00	4.0e-5	3.3e-3	230K (8)	770	-0.5
Excav.*	0.15	3.5e-4	4.0e-3	15K (1)	11	15.8
Indust.	0.15	1.0e-4	3.3e-3	256K (8)	1300	–
Wheel	15.00	1.0e-5	3.3e-3	50K (4)	2522	5.2
Silo	0.15	1.0e-4	4.0e-3	32K (8)	857	–
TC	3.00	1.0e-4	4.0e-3	500K (4)	1565	0.9

Table 2.2: Specifications for simulations of 20-sec 5-cm depth excavation, 6-sec industrial excavator, 20-sec 20% slip wheel, 12-sec silo, and 22-sec One-G Taylor-Couette (TC) flow. Mean percentage errors (MPE) in excavation forward force, wheel load, and TC flow angular velocity are specified relative to experiments. * indicates excavation model with relaxed Young’s modulus for significant computational efficiency yet acceptable accuracy.

[Dunatunga, 2017]. Also, the computational time spent in all the MPM-NGF simulation models are provided in table 2.2. MPM runs can be substantially sped up by reducing the number of particles per cell (ppc) and/or increasing the grid spacing (Δx), which all result in a lower particle count. Using a relaxed Young’s modulus and a larger timestep (Δt) would also contribute to this while satisfying the stability conditions of elasticity and the NGF model’s PDE [Dunatunga and Kamrin, 2015, Dunatunga and Kamrin, 2017, Henann and Kamrin, 2016]. An implicit MPM solver could rectify this and is a possible direction of future work. The mean percentage error (MPE) of the excavation forward force, wheel load, and TC flow angular velocity are also specified relative to the experiments.

Wheel. We validated the model via wheel-on-soil experiments. To this end, our

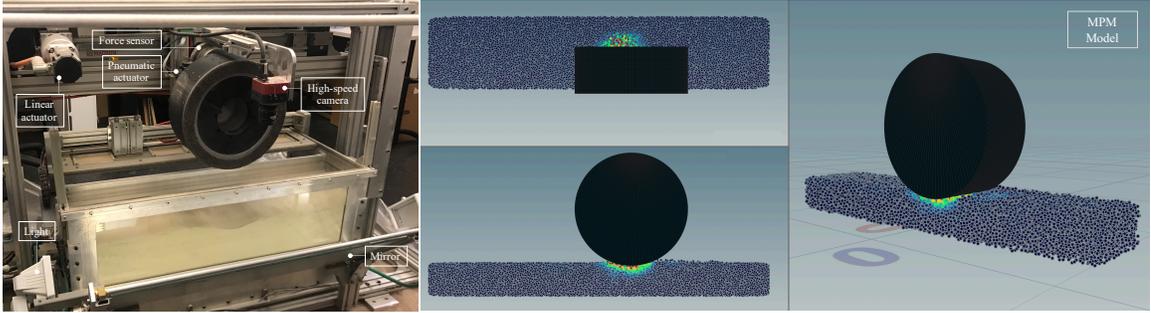


Figure 2.6: **Left:** Robotic equipment and experiment setup. **Right:** Wheel-soil model in MPM (right).

specialized robotic single-wheel apparatus was utilized [Niksirat et al., 2020]. The rover wheel is driven via synchronized control of a horizontal linear actuator (Macron Dynamics R6S driven by a Kollmorgen AKM23C motor and AKD-P00306 driver) and a wheel motor (Maxon RE35 with a MaxPos 50/5 driver) in an instrumented soil bin with a dimension of 90 x 19.3 x 35 cm. A vertical load is applied to the wheel while allowing free vertical motion. Data collection includes forces in the forward (drawbar pull, DP, i.e. net traction) and vertical (wheel load, W) directions via an ATI Delta IP60 sensor, and vertical wheel displacement via an ALPS slide potentiometer. The wheel is located against a transparent window in the soil bin. A high-speed 4 MP monochrome camera with 16 mm EFL lens and Core2 digital video recorder observes wheel–soil interactions through a glass sidewall in the soil bin via a mirror tilted at 45 degrees. The images are captured, processed, and analyzed using the Soil Optical Flow Technique (SOFT) [Skonieczny et al., 2014]. Experiment setup in the apparatus and its model in MPM is shown in Figure 2.6.

The smooth wheel is 30 cm in diameter and 12.5 cm in width. The soil in the experiment is a NASA Glenn Research Center lunar soil simulant GRC-1. The relative density used is 44.6 +/- 7.2%. This is calculated based on the cone index gradient of 5.30 +/- 0.6 kPa/mm using the correlation in [Oravec et al., 2010]. Thus, the corresponding internal friction angle can be obtained as 35 deg. The median grain diameter and density are 0.3 mm and 2583 kg/m³. Using the triaxial test performed by [Oravec et al., 2010] the estimated Young’s and shear moduli are 15 MPa and 5.8 MPa, respectively. Also, the measured external friction coefficient between the wheel and soil is approximately 0.4. The amount of traction a wheel produces is related to

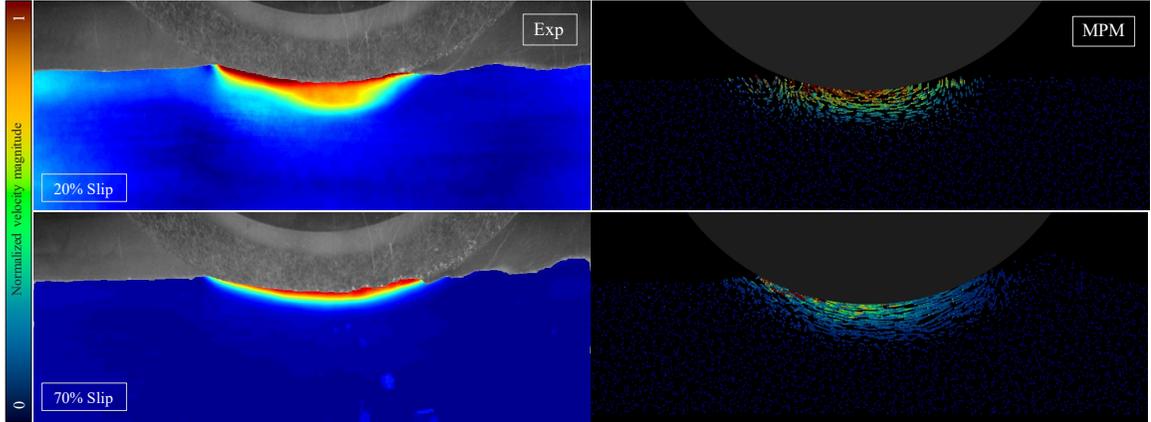


Figure 2.7: Average normalized subsurface soil velocity magnitude between 5 and 6 second marks from experiment (left) and MPM-NGF (right) with 20% (top) and 70% (bottom) slips.

how much it slips. Thus, we tested with three different slips including $s = 20\%$, 40% , and 70% where $s = 100(1 - v/r\omega)$ is a function of wheel radius r , horizontal speed v , and rotational speed ω . Here, we varied the horizontal velocity to change the amount of slip. Moreover, we exerted a normal load of 164 N on the wheel. This load consists of the wheel unit weight plus applied force on the wheel unit.

A visual quantitative comparison is shown in Figure 2.7. This shows the average subsurface soil motion between the 5 and 6 second marks in the experiment and simulation (MPM-NGF) with 20% and 70% slips. The grain/particle velocities are normalized with the tangential velocity of the wheel rim. The velocities visualized in colors clearly depict the static (blue) and highly dynamic (red) parts. The soil deformation underneath and behind the wheel (right side) from experiment and simulation are comparable.

Furthermore, the wheel-soil interactions from simulation here are evaluated by the forces measured in the experiments. Figure 2.8(left) compares the drawbar pull and wheel load forces from MPM-NGF and experiment with 20% slip. The quantitative force values are in good agreement with the experimental forces in both the forward and vertical directions. The MPE of the wheel load force in the simulation compared to the experimental result, provided in table 2.2, does not exceed 6% magnitude. In addition to the overall trend here, as mentioned, MPM-NGF is also able to capture drops and rises in force at various timesteps. The NGF parameters are set based

2.2. NUMERICAL APPROACH OF NONLOCAL CONSTITUTIVE MODEL

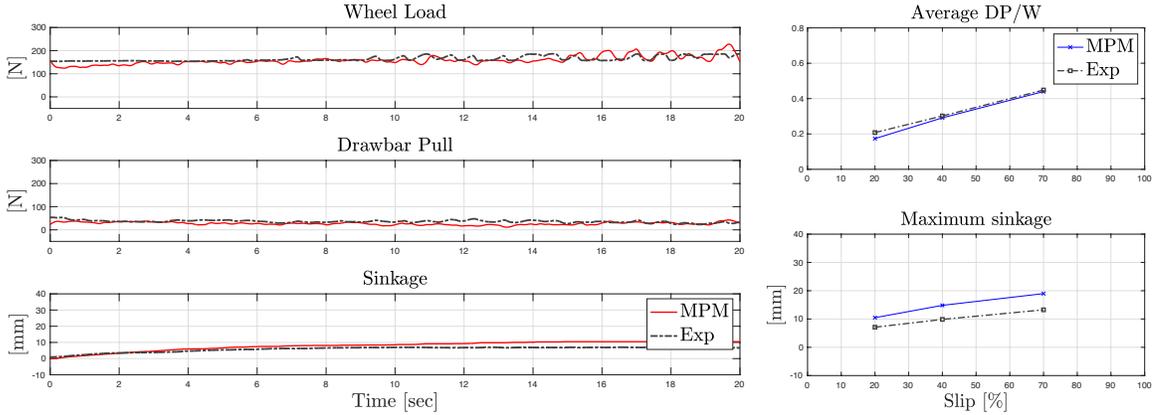


Figure 2.8: **Left:** Interaction forces (drawbar pull and wheel load) and sinkage from experiment and MPM-NGF with 20% slip. **Right:** Average drawbar pull to wheel load ratio and maximum sinkage at different slips from experiment and MPM-NGF.

on the physical properties in the experiment. In the simulation, to settle the wheel quickly and prevent wheel vertical bouncing, we applied an exponential damping to the vertical motion of the wheel for one second prior to start. The wheel upward motion was also significantly damped out during the wheel forward motion. This means, the wheel could move only downward (but not upward) freely.

Moreover, Figure 2.8(right) compares the average drawbar pull to wheel load ratio (DP/W) and maximum sinkage at different slips from experiment and MPM-NGF. The DP/W ratio is a common metric in the field of terramechanics that refers to the load a vehicle can tow relative to vertical load. Wheel sinkage, as a function of slip, is also a measure of the response of the terrain to a specific loading, and affects wheel performance. The simulation DP/W ratio is in a very good agreement with the experiment. Also, due to the implementation of the wheel upward damping, the maximum sinkage is slightly higher in the simulation than in the experiment. However, MPM-NGF has been able to capture the overall trends.

Excavation. We also set up an excavation experiment to validate the MPM with NGF model. It consists of a sandbox positioned under a 3-degree-of-freedom motorized unit to which an excavation tool is attached. For this experiment and for the numerical simulation presented here, the excavation tool is a flat plate (blade) as depicted in Figure 2.9. The rake angle of the blade can be set manually and it remains constant during the run. The excavator can be moved horizontally and

2.2. NUMERICAL APPROACH OF NONLOCAL CONSTITUTIVE MODEL

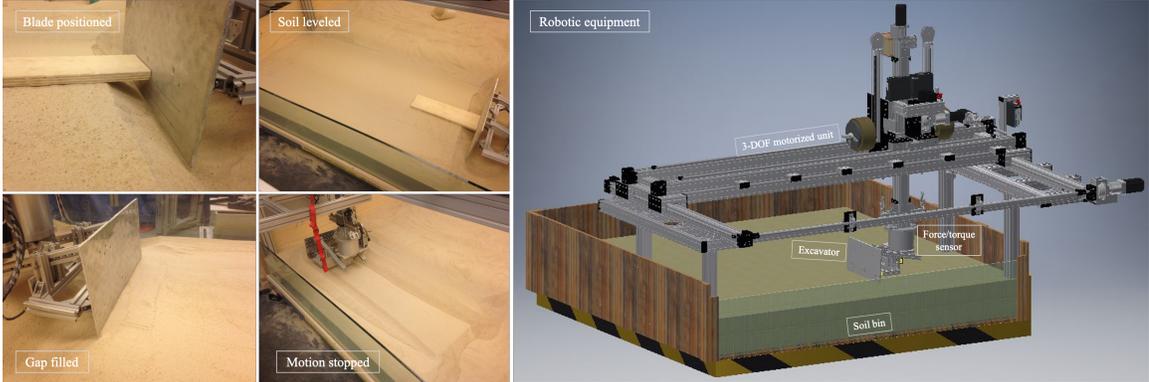


Figure 2.9: Robotic equipment and experiment setup.

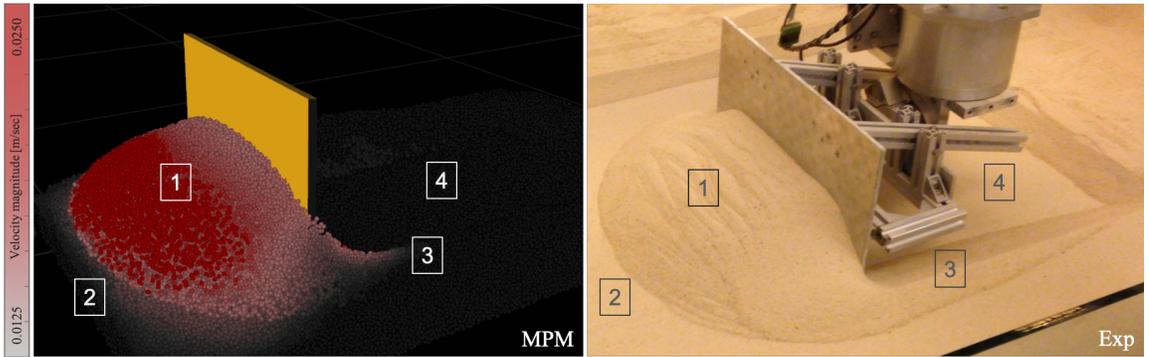


Figure 2.10: Geometries of soil deformations in MPM-NGF and experiment for 5-cm depth.

vertically independently. The motors are controlled such that the impulses from the soil flow do not affect the trajectory of the excavator. The excavator is installed on a Delta IP60 (ATI Industrial Automation Inc.) 6-axis force-torque sensor. The blade trajectory is composed of three segments: first, a downward ramped motion at the start to dive into the soil to a specific depth, then a long-duration horizontal motion, and finally an upward ramp at the very end. Two tests were done based on this trajectory but at different (2-cm and 5-cm) depths.

The soil in the experiment is the GRC-1 with the aforementioned properties. Also, the measured external friction angle between the blade and soil is ~ 30 deg. The setup of the experiment is shown in Figure 2.9(left).

A qualitative and visual comparison is shown in Figure 2.10. This shows the soil geometry in the end of the second trajectory segment in the 5-cm MPM-NGF and

2.2. NUMERICAL APPROACH OF NONLOCAL CONSTITUTIVE MODEL

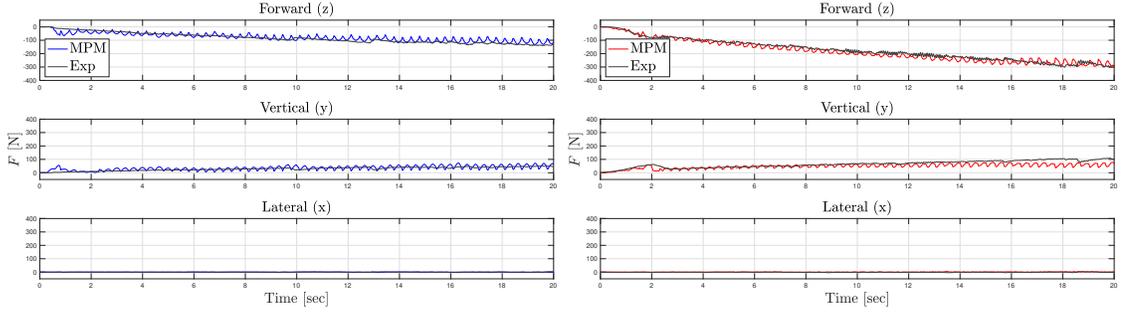


Figure 2.11: Interaction forces from MPM-NGF and experiment for 2-cm (left) and 5-cm (right) depths.

experiment. Four visual criteria are emphasized for a better analogy. In general, the soil behavior in MPM-NGF is predicted similar to the one in the experiment. The particle velocities visualized in colors clearly depict the static (gray) and dynamic (red) parts. The MPM-NGF velocity field corresponds well with the experiment.

Furthermore, the tool-soil interactions in the simulations here are evaluated by the forces measured in the experiments. Figure 2.11 compares all the forces from MPM-NGF and experiment. The quantitative force values are in good agreement with the experimental forces in the three (forward, vertical and lateral) directions. The MPE of the forward force of the 5-cm depth case in the simulation compared to the experimental result is provided in table 2.2. The errors do not exceed 1% and 16% magnitude for the original and faster simulations, respectively. Due to the dimensional analysis used to model the large experimental cases, the scaled MPM-NGF results seem to have slightly boosted oscillation amplitude. However, in addition to the overall trend, MPM-NGF is able to capture drops and rises in force at various timesteps of the two experiments. This can highlight the unsteady form of the MPM solver as well as the NGF constitutive model.

Also, we tried modeling the excavation case with a recent modified Drucker-Prager (DP) plasticity model (with projection) [Klár et al., 2016]. This model showed its capability in modeling the Taylor-Couette flow [Haeri and Skonieczny,]. In addition to the material parameters mentioned before, the hardening parameters were set as $h_0 = 44$, $h_1 = 9$, $h_2 = 0.2$, and $h_3 = 9$. Parameters h_0 and h_3 are responsible for the minimum and maximum friction angle relative to the plastic deformation (here 35 and 44 degrees, respectively). Also, h_1 and h_2 define the change rate of the friction

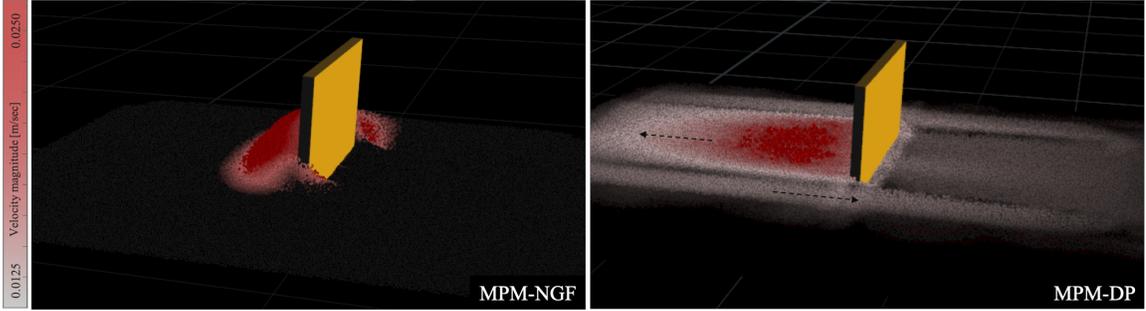


Figure 2.12: Excavation modeling with MPM-NGF (left) and MPM-DP (right). Soil modeled with DP experiences unexpected fluid-like behaviors in untouched areas with no accumulating pile.

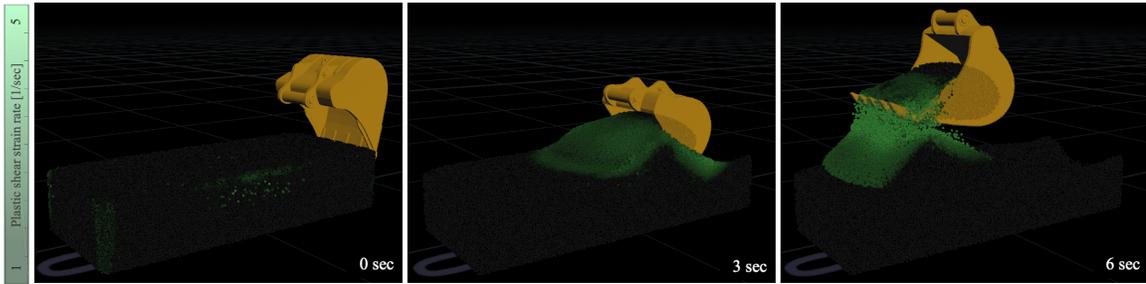


Figure 2.13: Industrial excavator modeled via MPM-NGF.

angle [Mast et al., 2015]. We found the model insensitive to the depths we were interested in (i.e. 2 cm and 5 cm). The fluid-like behavior of the soil modeled with this plastic model is shown in Figure 2.12(right) compared to the soil deformation modeled with the viscoplastic NGF model (left). It seems that the DP model has issue with the hardening part, because as soon as we increased the depth, the soil started to pile up in front of the excavator yet with unexpected fluid-like behaviors in very distant areas, and to the sides of the blade even reflected soil flow in the opposite direction. With the help of visualization of the particle velocities from black (0 m/s) to white (0.0125 m/s) and red (0.0250 m/s), the deformations are clearly visible in Figure 2.12. MPM with DP plasticity would clearly require further development in future work, if there is a chance for it to properly capture the experimental results.

We further modeled an industrial excavation with MPM-NGF shown in Figure 2.13. It demonstrates the ability of MPM-NGF to model the three states that a granular material can experience during soil cutting operations [Haeri et al., 2020].

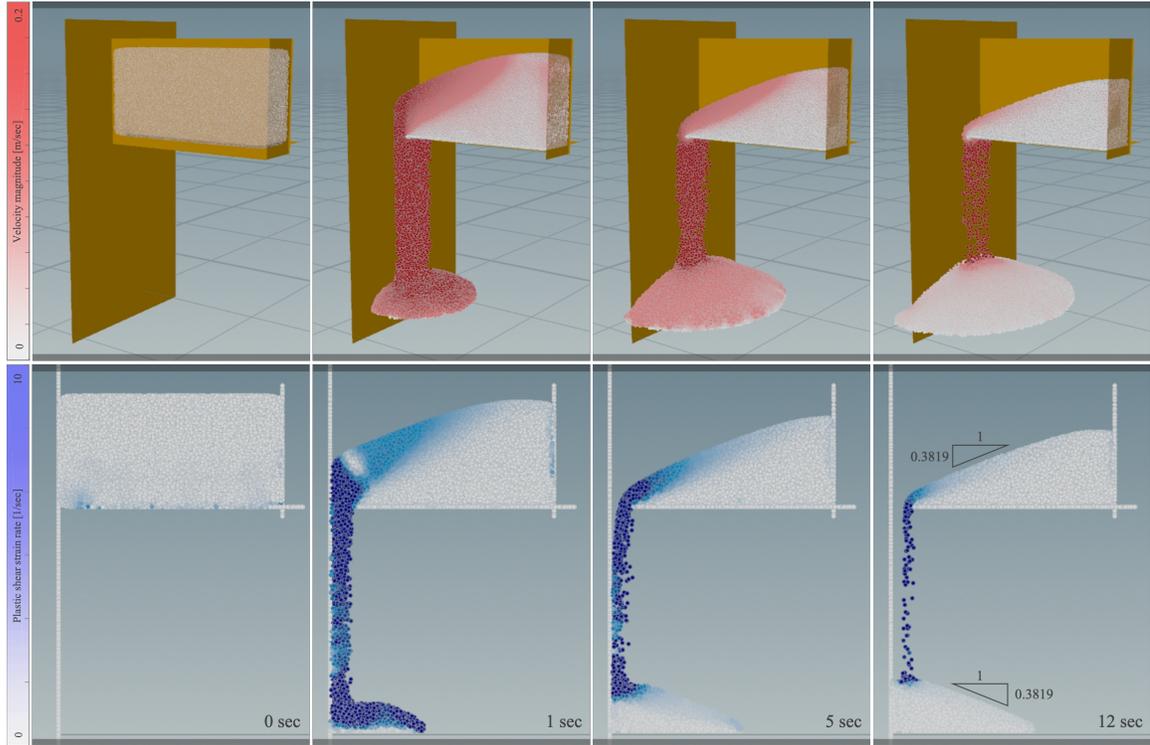


Figure 2.14: **Top:** Velocity field in 3D silo. **Bottom:** Plastic shear strain rate field shown in a thin layer of 3D silo.

The colors indicate the equivalent plastic shear strain rate ($\dot{\gamma}^P$) from black (0 s^{-1}) to light green (5 s^{-1}) for each particle. This enables us to track the soil deformations in the particle's body frame as opposed to the velocity magnitude in the global frame. At the 6 second mark, the soil inside the bucket and untouched areas are under solid-like elastic deformation. The separated particles leaking from the bucket are under gas-like stress-free deformation. Finally, the particles piled up after returning to the bin are under fluid-like viscoplastic deformation.

Silo. Similar to recent work in the literature [Dunatunga and Kamrin, 2015], we modeled drainage of a silo here but via the NGF model and in 3D. The material parameters are given in table 2.1 except for $\mu_s = 0.3819$. The bottom boundary is rough and sides are smooth. This case again demonstrates the fact that the model can capture the three states of matter. Figure 2.14(top), a perspective 3D view of the silo, shows the velocity magnitude of each individual particle from white (0 m/s) to red (0.2 m/s) at 0, 1, 5, and 12 second marks. Also, a side 2D view of a thin layer in

the middle is shown in Figure 2.14(bottom); it visualizes the equivalent plastic shear strain rate from white (0 s^{-1}) to blue (10 s^{-1}). The slope of the soil pile at the 12 second mark matching the friction angle provides further evidence of the accuracy of the model.

Taylor-Couette Flow. We also validated the model via the experimental results that have been produced by flying a TC flow apparatus (shown in Figure 2.15(left)) aboard a reduced gravity aircraft [Murdoch et al., 2013a]. The results will be presented and discussed in §2.3.

2.3 Gravity Sensitivity of Continuum Numerical Solvers

FEM and MPM are chosen to be studied for granular materials under varying gravity. The nonlocal granular fluidity (NGF) model [Haeri and Skonieczny, 2021, Henann and Kamrin, 2014] is utilized as the constitutive model. These choices ensure capturing both quasi-static and intermediate regimes which are crucial in the current research. Also, for reference, a simple and computationally efficient analytical method that does not require numerical solvers is also considered.

A candidate theoretical framework for understanding rover wheel-soil interaction under different gravity conditions, is Taylor Couette (TC) flow, which models viscous granular flow between a rotating inner and stationary outer concentric cylinder, shown in Figure 2.15. Parallels can be drawn between the rotating inner cylinder and a rotating wheel [Higgs et al., 2006]. This approach is justified by similarities to rotational aspects of wheel motion observed experimentally in [Skonieczny et al., 2014]; in particular, velocity gradients in the granular terrain resemble viscous flow. The goal in this section is to explore the best solver that can successfully model experimental results that have been produced by flying a TC flow apparatus aboard a reduced gravity aircraft [Murdoch et al., 2013a]. Note that the simulation conditions (e.g. cell dimensions, boundary conditions, etc.) are all from the experiment [Murdoch et al., 2013a].

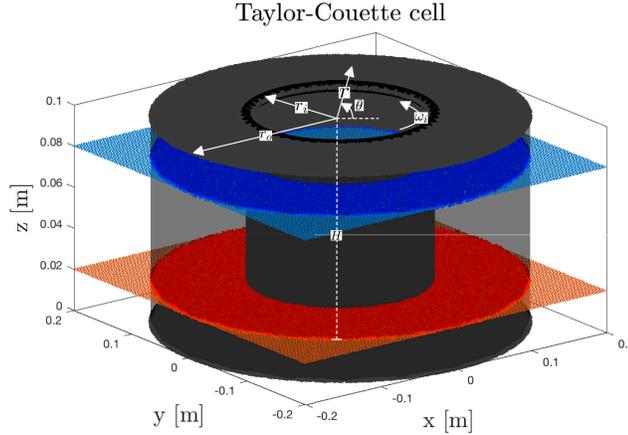


Figure 2.15: Taylor Couette cell, with chosen top (blue) and bottom (red) reference surfaces. The inner cylinder is moving and the outer cylinder is fixed.

2.3.1 Reduced-gravity TC flow experiment

In [Murdoch et al., 2013a] a TC shear cell was designed to investigate granular flow due to rotational shear forces in reduced-gravity environments. As of 2013, it was mentioned that nothing was known regarding TC flow in weak gravity. To reduce gravity during the experiments, a microgravity condition on a parabolic flight was utilized. There were 93 parabolas (each 22 sec) in one flight giving ≈ 30 minutes of microgravity ($\leq 0.05g_{Earth}$) in total. Two high-speed cameras, imaging the top and bottom of the transparent cell, and image processing tools were used to collect and process data in a specific region.

The TC geometry is shown in Figure 2.15 and consists of two cylinders; the outer cylinder has a radius (r_o) of 195 mm, the inner one has a radius (r_i) of 100 mm, and both are rough with a layer of particles while the floor is smooth. The gap between the two cylinders is filled to a height (H) of 100 mm with 4-mm diameter (d) spherical beads. The inner cylinder angular velocity (ω_i) is 0.025 rad/sec. Thus, the maximum dimensionless shear rate (DSR), where shear rate $\dot{\gamma}$ is equal to inner cylinder angular velocity ω_i , can be obtained for both One-G and Micro-G conditions. DSRs reveal that the flows are in the quasi-static regime (i.e. $DSR < 0.15$ [Tardos et al., 2003]).

$$\max(DSR) = \omega_i \sqrt{d/G} = \begin{cases} 0.0005, & \text{One-G} \\ 0.0020+, & \text{Micro-G} \end{cases} \quad (2.44)$$

2.3. GRAVITY SENSITIVITY OF CONTINUUM NUMERICAL SOLVERS

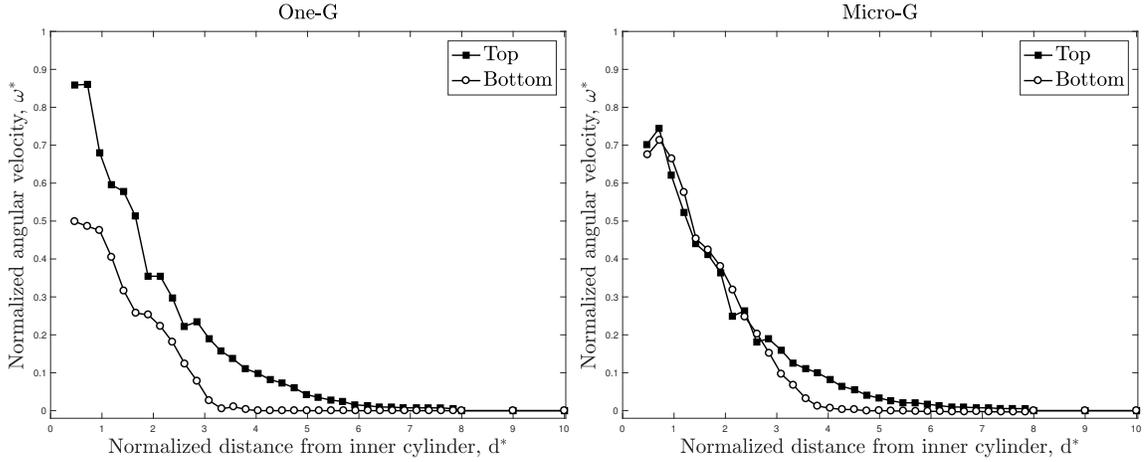


Figure 2.16: Comparison of normalized angular velocity profiles for the top and bottom surfaces in One-G (left) and Micro-G (right). The velocity profiles shown only extend up to 10 particle diameters from the inner cylinder as further away there is very little particle motion [Murdoch et al., 2013a].

Figure 2.16 shows a comparison between the angular velocity profiles on the top and bottom surfaces in One-G and Micro-G conditions [Murdoch et al., 2013a]. On the ground, the top surface speed is higher than bottom surface speed, while in microgravity the top and bottom surface speeds are similar. However, in both conditions the bottom surface speed decays faster indicating it has a narrower shear band (a narrow zone of large relative particle motion bounded by static regions). Also, in One-G the shear band is much smaller in the ordered grains on the bottom surface than in the disordered grains of the top surface (see Figure 2.23). Moreover, the bottom surface velocity in One-G is notably lower than in Micro-G. In different gravity, the bottom surface is under the influence of different overburden weight from the particles above ($m = 13.5$ kg).

Based upon these results, two relative trends are defined by us: (1) The velocity difference between top and bottom surfaces, which is larger in One-G than in Micro-G; (2) The shear band length difference between top and bottom surfaces, which is again larger in One-G than in Micro-G. In addition to capturing these trends, an accurate model should achieve the correct velocity profile values quantitatively as well.

2.3.2 Analytical Method for reduced-gravity TC flow

An analytical method (AM) put forward by [Lagrée and Lhuillier, 2006] proposes two constitutive equations for two elements in the stress tensor: σ_{rr} and $\sigma_{r\theta}$ (expressed in cylindrical coordinates). The assumptions of this constitutive model are valid for dry and dense (i.e. high, but strictly less than maximum, volume fractions) granular TC flows. The equation for σ_{rr} considers the effects of the disorder pressure $P^*F(\phi)$ (where ϕ is volume fraction), and the impact pressure. The equation for $\sigma_{r\theta}$ also includes these same two effects plus the influence of the friction coefficient $\mu(\phi)$ and the sliding contact dissipation $\mu_T(\phi)$. Ultimately, they are a function of grain density ρ_s , grain diameter d , dilatancy $\mu_N(\phi)$, angular velocity rate $\frac{\partial\omega}{\partial r}$, and radial distance to the moving inner cylinder as well. Some parts contain constants that are obtained from previous experiments.

$$\begin{aligned}\sigma_{rr} &= P^*F(\phi) + \rho_s d^2 \mu_N(\phi) \left(r \frac{\partial\omega}{\partial r}\right)^2 \\ \sigma_{r\theta} &= \mu(\phi)\sigma_{rr} + \rho_s d^2 \mu_T(\phi) \left(r \frac{\partial\omega}{\partial r}\right)^2\end{aligned}\tag{2.45}$$

where

$$\begin{aligned}F(\phi) &= \log(1/(1 - \varphi)), \mu_N(\phi) = \mu_{N_0}/(1 - \varphi)^2 \\ \mu_T(\phi) &= \mu_{T_0}/(1 - \varphi)^2, \mu(\phi) = \mu_0 \\ \varphi &= (\phi - \phi_m)/(\phi_M - \phi_m)\end{aligned}$$

with ϕ_m and ϕ_M being the minimum and maximum volume fractions, considered as random loose and close packings, respectively. Note that μ_{N_0} and μ_{T_0} represent the dilatancy and contact dissipation, respectively, at minimum volume fraction. Because of the very slow motion ($\omega < 1$ rad/sec), the model is appropriate for quasi-static and intermediate regimes given the grain diameters and gravity considered in the survey (shown in Figure 1.1).

In the AM, using the proposed constitutive models for the two elements in the stress tensor, and applying simplifying assumptions on momentum equations [Lagrée and Lhuillier, 2006], gives an integral relation for calculating the angular velocity at

2.3. GRAVITY SENSITIVITY OF CONTINUUM NUMERICAL SOLVERS

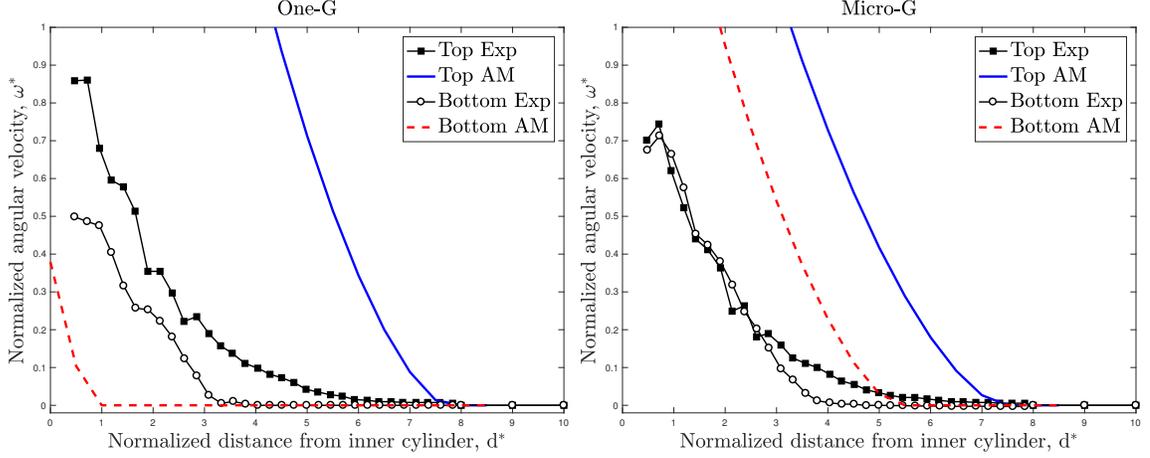


Figure 2.17: Normalized angular velocity versus normalized distance from inner cylinder in One-G and Micro-G from AM and experiment for top and bottom surfaces.

each vertical position, z , in the TC flow as follows

$$\omega(r') = \int_{r'}^{r'_0} \sqrt{\frac{P^*}{\mu_{N_0} \rho_s d^2}} (q(x) + q(x) e^{-P(z)/P^*} e^{q^2(x)}) dx/x \quad (2.46)$$

where

$$P(z) = K(P_c + \rho G(H - z)),$$

$$q(x) = \sqrt{\frac{\mu_{N_0}}{\mu_{T_0} P^*}} \sqrt{\tau_i/x^2 - \mu_0 P(z)}$$

for $r' = r/r_i$, $r'_o = r_o/r_i$ and where K is coefficient of earth pressure at rest (i.e. the ratio of the horizontal normal stress to the vertical normal stress), P_c is confining pressure applied by the top plate, ρ is bulk density, G is gravitational acceleration in z -direction, P^* is a constant, and τ_i is shear stress applied to the inner cylinder. Recall from above that r_i is inner cylinder radius, r_o is outer cylinder radius, and H is the height of the TC filled with granular material.

Figure 2.17 shows the normalized angular velocity $\omega^* = \omega/\omega_i$ versus normalized distance from inner cylinder $d^* = (r - r_i)/d$ in different gravity conditions for the top and bottom surfaces compared with the experiment. It reveals that it is possible to achieve some qualitative correspondence to the experiment in terms of the relative trends defined earlier, as larger differences between top and bottom velocities are

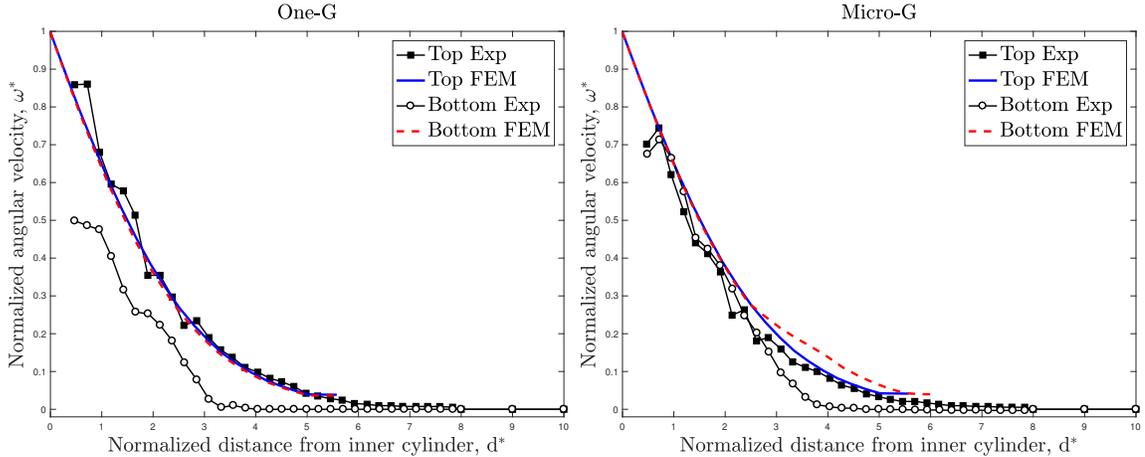


Figure 2.18: Normalized angular velocity profiles in One-G and Micro-G from FEM and experiment for top and bottom surfaces

observed in One-G than in Micro-G. However, in terms of absolute angular velocity the results do not correspond to the experimental results.

2.3.3 Finite Element Method for reduced-gravity TC flow

Here, FEM is used to model the steady-state TC flow by allowing a small rotation of the inner cylinder (4 degrees) in the TC cell. The code used here to generate the results is provided by [Henann and Kamrin, 2016], written in Fortran within the Abaqus User Element subroutine, and accompanied by the steady form of Nonlocal Granular Fluidity model. Figure 2.18 compares results of the FEM versus the experiment. In contrast to the AM, in terms of absolute velocity, FEM is accurate and, on average, consistent with the experimental results. However, it captures little to no discrepancy among velocity profiles in varying gravity and vertical position.

A study in [Henann and Kamrin, 2016] emphasizes that high Froude numbers ($Fr \approx 1$, rate-dependent) cause differences in the velocity profiles at different vertical positions (z), and the velocity profile in rate-independent cases ($Fr \leq 10^{-3}$) shows no z -dependence. In both cases the pressure is only z -dependent. In accordance with this, the Froude numbers ($Fr = r_i \omega_i (Gd)^{-0.5}$) of 0.013 and 0.056 for the One-G and Micro-G experiments, reveal the flows are not quite rate-independent, but also not large enough to be strongly z -dependent. The lack of z -dependence in the velocities is evident in the FEM results shown in Figure 2.19.

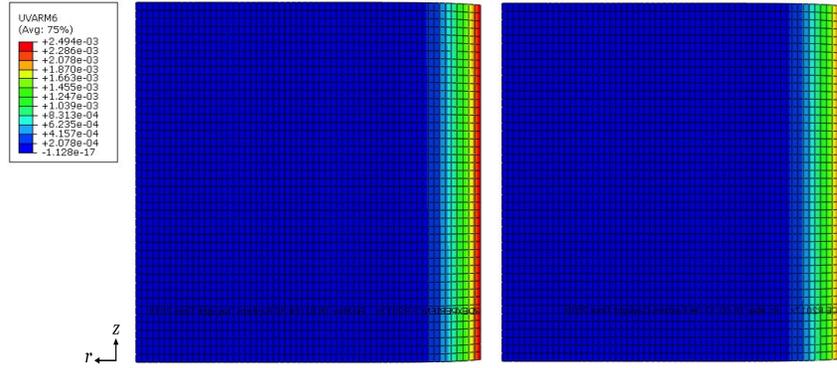


Figure 2.19: Tangential velocity ($v_\theta = r\omega$) in r-z plane in both One-G (left) and Micro-G (right) conditions from FEM

2.3.4 Material Point Method for reduced-gravity TC flow

Figure 2.20 compares the velocity profiles between the MPM and the experiment (introduced in §2.3.1), similarly to how the AM and FEM were compared in §2.3.2 and §2.3.3. MPM captures absolute values, shear band lengths and relative curves at different vertical positions and gravity conditions. Figure 2.21 shows the normalized angular velocity profiles for the top and bottom surfaces in One-G and Micro-G from MPM and the experiment separately. In that, error bounds (MPM) and error bars (experiment) show one standard deviation. Also, the radial velocity for the top surface in One-G, shown in Figure 2.22, is only three percent of the inner cylinder velocity in MPM; the experimental result is also very low (approximately 0 after $d^*=5$, considering experimental error).

Particle motions at the top and bottom surfaces in One-G at the 22 second mark (i.e. near the end of the Micro-G parabola) are shown in Figure 2.23(left). Colors from white to red represent normalized velocity magnitude. Two small images in the middle are long exposure images from the experiment with blurred areas indicating moving grains. In both MPM and the experiment, the shear band at top is larger than at the bottom in One-G. Figure 2.23(right) shows similar results but in Micro-G. Shear band lengths are close to equal at top and bottom in Micro-G.

Furthermore, the shear band lengths at different vertical positions of the cell can be seen in Figure 2.24 showing moving areas in red. In contrast to FEM (Figure 2.19), in MPM (and also the experiment) there is clearly more deformation at top near the

2.3. GRAVITY SENSITIVITY OF CONTINUUM NUMERICAL SOLVERS

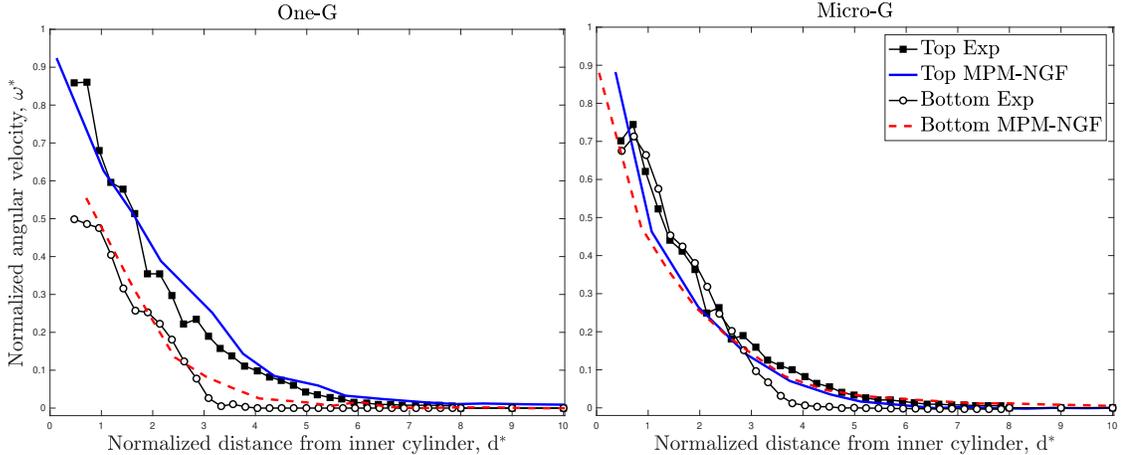


Figure 2.20: Normalized angular velocity profiles in One-G and Micro-G from MPM with NGF model and experiment for top and bottom surfaces.

inner cylinder in One-G, and similar deformation at top and bottom in Micro-G.

The plastic Drucker-Prager (DP) constitutive model (with projection) [Klár et al., 2016] is also used to show the MPM gravity sensitivity to constitutive models. The same as before, Figure 2.25 compares the normalized angular velocity profiles but this time between MPM with DP model and the experiment. The results are again in a good agreement with the experiment. Further, the mean absolute percentage errors (MAPE) of the both constitutive models relative to the experiment in One-G and Micro-G for top and bottom surfaces are shown in Table 2.3. This can emphasize the MPM capability in capturing the gravity effect regardless of the constitutive model.

In order to examine the steadiness of the flow, the shear reversal case in the experimental reference was also modeled. This case examines TC flow while the inner cylinder changes its rotation direction at the middle of the experiment (at 11 sec). Figure 2.26 represents a time track of normalized mean angular velocity of three near-inner cylinder radial bins (1, 2, and 3) and one near-outer cylinder radial bin (10) for top surface in One-G and Micro-G. The results reach the steady state very soon (after ~ 2 sec); while there is a brief transient state just after the shear reversal. This transient weakening is stronger in Micro-G in both the MPM and experiment.

2.3. GRAVITY SENSITIVITY OF CONTINUUM NUMERICAL SOLVERS

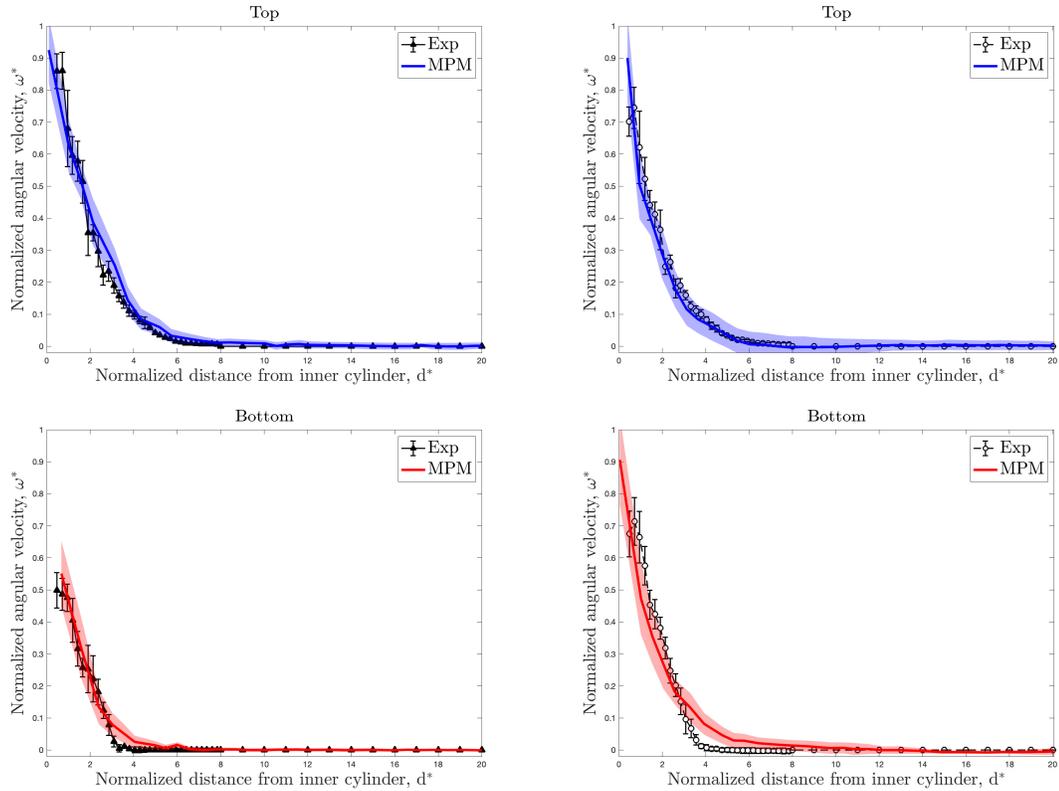


Figure 2.21: Normalized angular velocity profiles in One-G (left) and Micro-G (right) from MPM versus experiment for top and bottom surfaces.

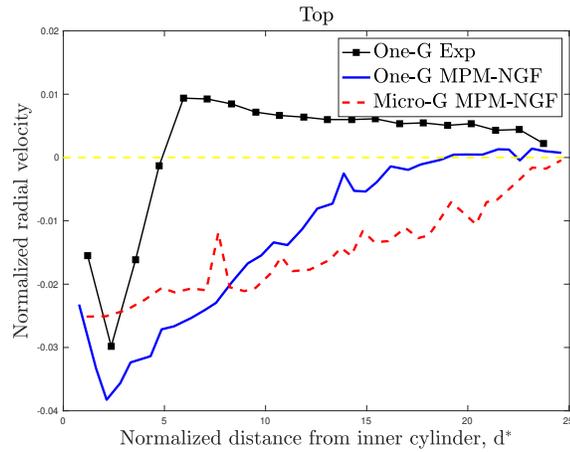


Figure 2.22: Comparison of normalized radial velocity in One-G from MPM with experiment for top surface. No data is available from microgravity experiment.

2.3. GRAVITY SENSITIVITY OF CONTINUUM NUMERICAL SOLVERS

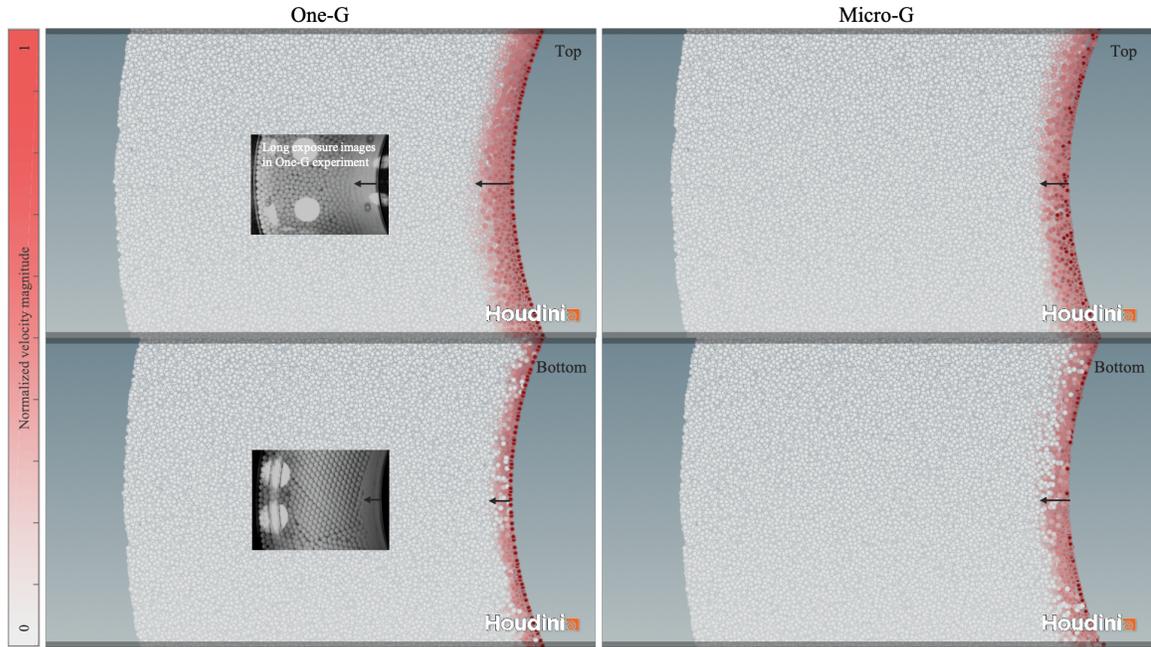


Figure 2.23: Shear band illustration in One-G (left pair) and Micro-G (right pair), based on particle normalized velocity magnitude at top (left) and bottom (right) at the 22 second mark from MPM versus experimental images [Murdoch et al., 2013a].

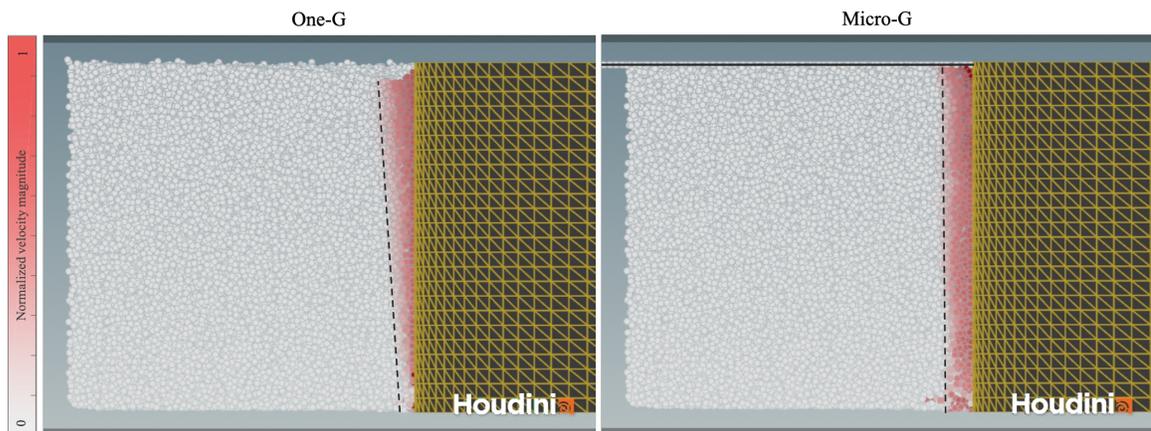


Figure 2.24: Moving areas in vertical plane, in One-G (left) and Micro-G (right) from MPM.

2.3. GRAVITY SENSITIVITY OF CONTINUUM NUMERICAL SOLVERS

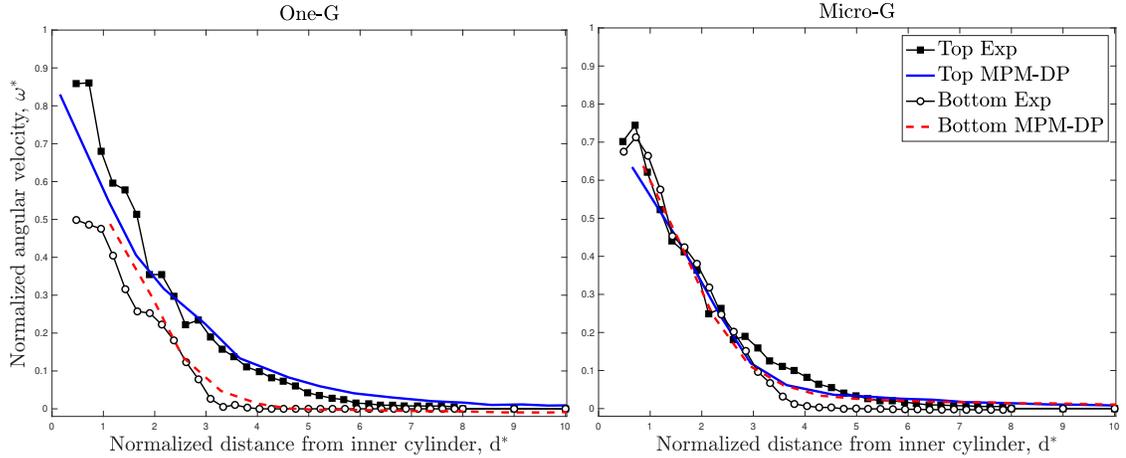


Figure 2.25: Normalized angular velocity profiles in One-G and Micro-G from MPM with DP model and experiment for top and bottom surfaces.

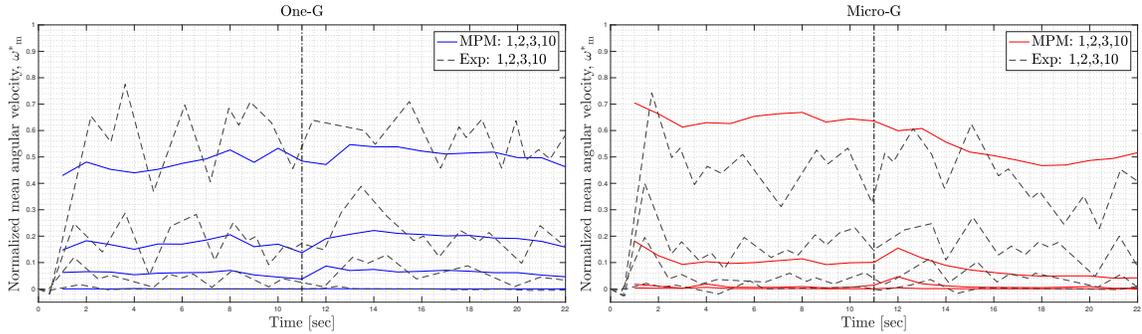


Figure 2.26: Shear reversal results for top surface in One-G (top) and Micro-G (bottom).

Table 2.3: Mean absolute percentage error (MAPE) of velocity field in TC flow from DP and NGF models relative to experiment in One-G and Micro-G for top and bottom surfaces.

Model	One-G Top	One-G Bottom	Micro-G Top	Micro-G Bottom
NGF	0.88%	0.92%	1.50%	1.85%
DP	0.93%	0.99%	1.24%	1.28%

2.3.5 Discussion

Gravity, in MPM, is first applied on particles. The constitutive model calculates particle stress and the transfer scheme transfers momentum and stress from particles to grid nodes. By solving the momentum equation, grid information is updated, and affects particle momentum and deformation through the transfer scheme. In this way, gravity influences particle velocity, grid velocity, and finally particle deformations.

Gravity might cause z-dependency in the Taylor-Couette (TC) flow velocity field as observed in the aforementioned experimental and MPM results. Although there exist experiments reporting a height-invariant flow field in TC flow [MiDi, 2004, Bocquet et al., 2001], for some conditions and within the resolution of the measurements [Mueth et al., 2000], some experimental conditions result in a clear z-dependence in the velocity field [Murdoch et al., 2013a, Murdoch et al., 2013b]. We speculate that some of the factors that may influence the degree of z-dependence in an experiment could include floor friction (as it can't be totally frictionless in reality), TC cell aspect ratio, properties of particular granular materials, etc. In MPM, the results agree with the experiment in [Murdoch et al., 2013a] as shown in Figures 2.20 and 2.21 using the nonlocal granular fluidity model and Figure 2.25 using the Drucker-Prager model.

Whether a simulation develops a z-dependency or not appears to be driven by the presence of z- and r-dependence in the pressure field. Most of the particles in the shear band are under negative pressure (i.e. tension) in MPM³. As mentioned, these particles are set to be stress-free (i.e. zero pressure). In contrast, FEM [Henann and Kamrin, 2016] does not experience the negative pressure field in the shear band with the specified rotation angle (4 degrees). The effects of this can be observed in the normalized mean pressure field of the vertical plane shown in Figure 2.27 for both MPM and FEM in One-G. In these cases, only the MPM pressure field is both z- and r-dependent. This can also be observed using Drucker-Prager in MPM.

As a result, the (z- and r-dependent) pressure field can affect the inertial number field. The resulting MPM and FEM inertial number fields are shown in Figure 2.28.

³The MPM pressure field may have some analogy to viscous fluid where flow near a moving inner cylinder is under negative pressure [Vittoz et al., 2019].

2.3. GRAVITY SENSITIVITY OF CONTINUUM NUMERICAL SOLVERS

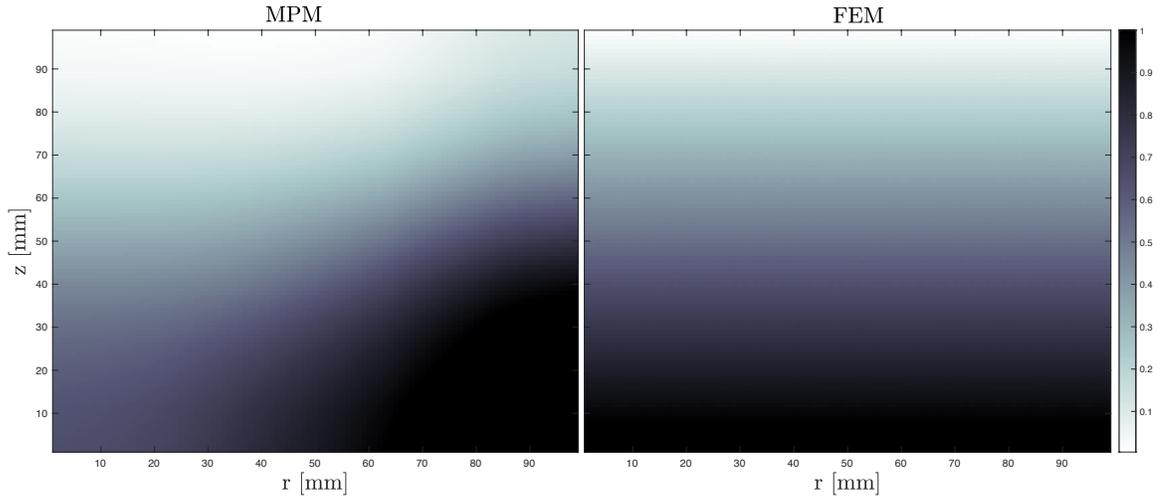


Figure 2.27: Normalized mean pressure field in vertical (i.e. r-z) plane in MPM and FEM.

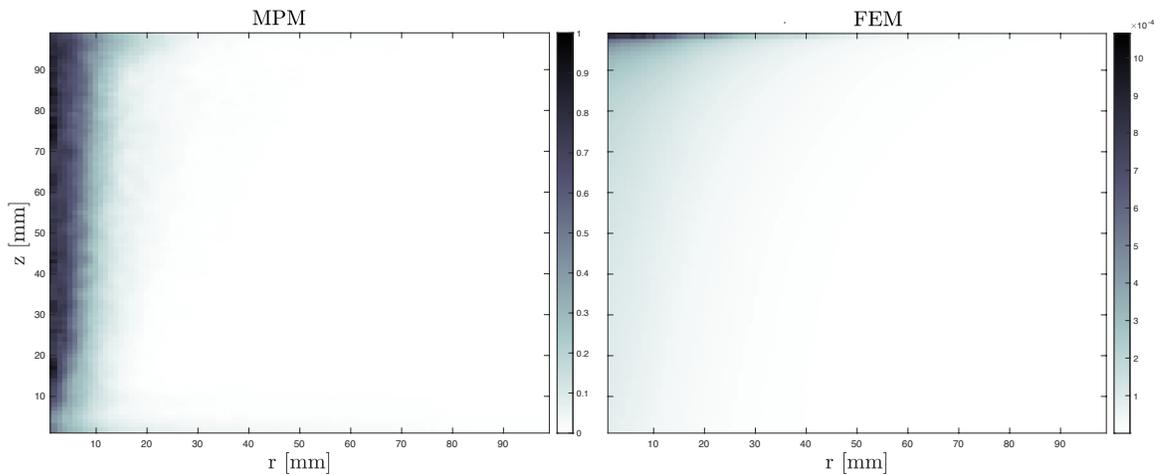


Figure 2.28: Mean inertial number field in vertical (i.e. r-z) plane in MPM and FEM.

Unlike FEM, The rate-dependent MPM flow near the inner cylinder makes the velocity field z -dependent similar to the way that a high Froude number can cause this [Henann and Kamrin, 2016].

Given the z -dependent experimental velocity field, it seems that the MPM, unlike FEM, is capable of producing the true inertial number field. In the TC flow, most of the zones experience slow flows ($I \lesssim 10^{-3}$), but some do experience flow with higher I . In the case where $I \gtrsim 10^{-3}$, FEM has problems in modeling the rate-dependent intermediate flows. Therefore, MPM is shown to be a good candidate to model the

2.3. GRAVITY SENSITIVITY OF CONTINUUM NUMERICAL SOLVERS

entire flow in various gravity conditions.

Chapter 3

Real-Time Physics Modeling

Offline, yet efficient, modeling of the interaction of robots and granular flows is so far made possible by a continuum method comprising a modern constitutive model, nonlocal granular fluidity (NGF), and a state-of-the-art numerical solver, material point method (MPM). This could contribute to the robot design applications including wheel and sample bucket design, as well as computer graphics applications such as animation production. As an example, Autodesk Maya’s Bifrost is recently equipped with an embedded MPM engine for visual effect creation. Online simulation, however, is essential for training operators in the construction and space industry. For instance, Lunar exploration rovers can be controlled (with a 4-second delay) by these on-ground operators. Moreover, maintaining traction remains one of the challenges that semi-autonomous Mars exploration rovers encounter while roving on Martian deformable terrains. Such terrains consist of granular regolith under reduced gravity conditions. The real-time simulation of wheel-soil interactions, that accurately takes gravity effects into account, can improve rovers’ online mobility control as well.

Data-driven methods have begun to be applied to similar (specially fluid) simulations. Some researchers [Holden et al., 2019, Harmon and Zorin, 2013, Huang et al., 2011, Sifakis and Barbic, 2012, von Radziewsky et al., 2016] are applying dimensionality reduction methods, such as principal component analysis (PCA) [Hotelling, 1933], to get a lower-dimensional representation of the physics data. Others [Sanchez-Gonzalez et al., 2018, Li et al., 2019] are using graph neural networks (GNNs) [Gori et al., 2005, Scarselli et al., 2008, Scarselli et al., 2009, Li et al., 2016], such as graph

networks (GNs) [Battaglia et al., 2018], to learn the underlying physical dynamics more accurately. Metrics of success for these methods have to date focused on qualitative visual appearance. In this chapter, we first provide a background on GNs and PCA. Then, we develop a real-time learning simulation approach for granular flow modeling of robot-terrain interactions. Finally, we present the results of two configurations including Excavation and Wheel. There are two distinctions here: (1) extensive training data will be generated via our accurate MPM-NGF verified by experimental data, and (2) the training and rollout runtime and memory usage of GNs will be significantly reduced by learning the subspace dynamics (i.e. subspace learning [Champion et al., 2019]) using the reduced data obtained by applying PCA to high-dimensional physics data.

3.1 Introduction and Background

First, §3.1.1 provides a background on graph neural networks (GNNs). We introduce graph machine learning, and explain the GNNs (under the field of graph machine learning) and their necessity. Furthermore, we discuss the expressive power of GNNs and introduce graph network as generalized GNNs. We also explain one limitation of GNNs and why a non-learning model reduction method can help. Then, §3.1.2 provides a background on the principal component analysis (PCA) [Brunton and Kutz, 2019] and the dimensionality reduction aspect of it. We emphasize that although we select GNNs and PCA for our implementation, other neural network architectures and model reduction methods could potentially be used, respectively.

3.1.1 Graph Neural Networks

Graphs are general data structures for describing complex systems. Graphs are composed of entities (also known as nodes) and pairwise interactions (edges). In many cases, nodes and edges can have attribute (feature) information differentiating them from each other. Node-level features can be structure-based (capturing topological properties of the local neighbourhood) and importance-based (capturing the importance of nodes). In addition, edge-level features can be distance-based and neighbourhood overlap-based. There are also graph kernel methods for defining graph-level

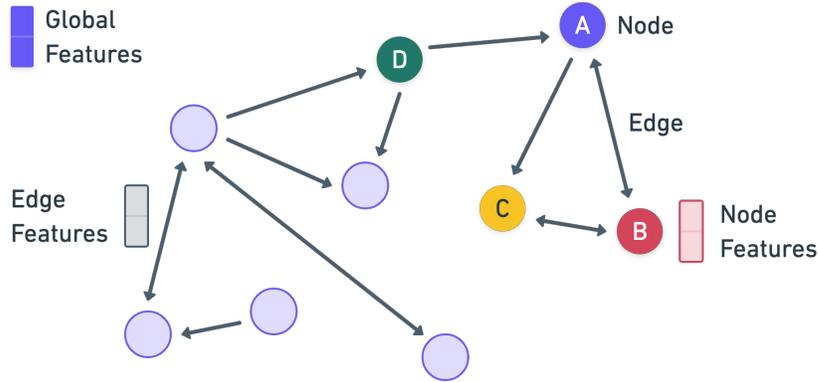


Figure 3.1: A sample graph with nodes, directed edges, and features.

features such as graphlets [Shervashidze and Borgwardt, 2009], Weisfeiler-Lehman (WL) kernels [Shervashidze et al., 2011, Weisfeiler and Lehman, 1968], and random walk kernels [Kashima et al., 2003]. A sample graph is shown in Figure 3.1.

Machine learning with graphs enables us to do different node-, edge-, and graph-level tasks on graph data. These include, but are not limited to, node classification [Jumper et al., 2020], relation prediction [Ying et al., 2018, Zitnik et al., 2018], graph generation [Konaklieva, 2014, Stokes et al., 2020], and graph evolution [Sanchez-Gonzalez et al., 2018]. Traditional machine learning pipelines require *hand-engineered* features extracted based on manually computed graph statistics and kernels (e.g. node degree, neighborhood overlap). These hand-engineered features are non-adaptive through a learning process, and time-consuming to process [Hamilton, 2020]. Graph representation learning is an alternative approach to learning over graphs. In this approach, the aim is to automatically learn the features from encoded graph structural information.

Shallow embedding approaches are one way to learn the features [Hamilton et al., 2017b]. In a shallow *Encoder-Decoder* framework, the Encoder maps nodes (i.e. their IDs) to a latent space called vector *embeddings*. Then, the decoder reconstructs graph statistics from the node embeddings. The Encoder-Decoder framework can be optimized by random walk approaches. In other words, these approaches can provide reference values for minimizing the Decoder loss (by measuring graph-based similarities). DeepWalk [Perozzi et al., 2014] and node2vec [Grover and Leskovec, 2016] are two examples of random walk approaches with different decoders, similarity

3.1. INTRODUCTION AND BACKGROUND

measures, and loss functions. The shallow embedding approaches, however, have the following drawbacks:

1. No parameter sharing: Every node has its own unique embedding which is computationally inefficient.
2. No node feature incorporation: They cannot leverage node features in the Encoder.
3. No generalization: They cannot generate embeddings for unseen nodes –those nodes not observed in training. This means they are inherently transductive [Hamilton et al., 2017b]).

Graph neural networks (GNNs), on the other hand, overcome the aforementioned limitations. GNNs also use embeddings, but are more generalized. A basic GNN is the generalization of convolutional neural network (CNN) beyond structured grid and sequence data to non-Euclidean data [Bruna et al., 2014]. In fact, GNNs can handle data with arbitrary size and complex topological structure, varying node ordering, and often dynamic features. In a deep *message-passing* framework, GNNs determine a node’s *computational graph* and propagate information through the graph. Below, we summarize how to propagate information across a graph to compute node features using a neural network.

Message-Passing Framework. In this framework, a graph is defined as $G = (E, V, u)$. The u is a global feature. The $V = \{v_i\}_{i=1:N^v}$ is a set of nodes where v_i are node features. The $E = \{(e_i, r_i, s_i)\}_{i=1:N^e}$ is a set of edges where e_i are edge features, r_i is the index of receiver node, and s_i is the index of sender node. Also, N^e and N^v are the number of edges and nodes, respectively. Here the global, node and edge feature embeddings are $h_u = f_u(u)$, $h_{v_i} = f_v(v_i)$ and $h_{e_i} = f_e(e_i)$, respectively, where $f(\cdot)$ ’s are arbitrary functions. Note that the global and edge embeddings are only used in the graph network as a generalized message-passing framework.

Every node has its own computational graph based on its local neighbourhood. The computational graph forms a tree structure by unfolding the neighborhood around the target node [Hamilton, 2020]. It can have arbitrary depth (i.e. message-passing steps). A one-step computational graph for a single node is shown in Figure 3.2 (given the graph shown in Figure 3.1).

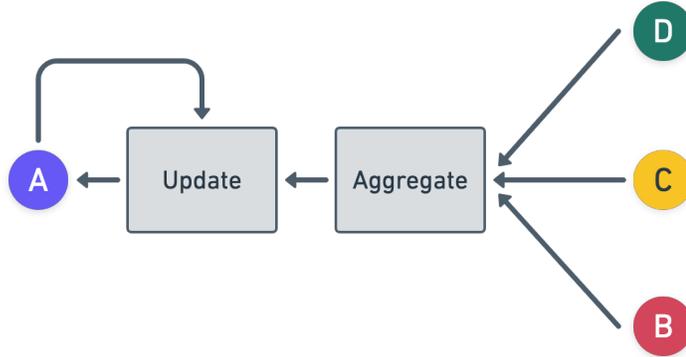


Figure 3.2: Illustration of one-step computational graph of node A in graph shown in Figure 3.1.

The message-passing framework is defined upon the computational graph. A basic framework consists of arbitrary differentiable node Update and Aggregate functions as follows

$$h_{v_i}^{(k+1)} = \text{Update}^{(k)} \left(h_{v_i}^{(k)}, \text{message}_{N(v_i)}^{(k)} \right) \quad (3.1)$$

for

$$\text{message}_{N(v_i)}^{(k)} = \text{Aggregate}^{(k)} \left(\{h_{v_j}^{(k)} | \forall v_j \in N(v_i)\} \right) \quad (3.2)$$

where $\text{message}_{N(v_i)}^{(k)}$ is the message that is aggregated from node v_i 's graph neighborhood, $N(v_i)$, at message-passing step k . Also, the Update function combines the aggregated message with the previous target node embeddings.

The Update and Aggregate functions are applied per node. The Update function can be defined via a neural network (with nonlinear activation(s) to add expressiveness) in the GNN message-passing framework. There are different ways to aggregate the features, such as mean pooling in graph convolutional network (GCN) [Kipf and Welling, 2016], and max pooling in GraphSage [Hamilton et al., 2017a]. Also, since the Aggregate function takes a set as input, the framework is permutation equivariant by design. This means that the Aggregate function output is permuted in a consistent way in response to permutations of the input [Hamilton, 2020]. The expressiveness power of the Update function and the choice of the Aggregate function will be discussed later in the current section.

In this framework GNN is inductive: (1) the parameters are shared, (2) the number

of parameters is sublinear within the size of graph (e.g. node features size, $|V|$), and (3) it can be generalized to unseen nodes. Specifically, GNNs are a certain class of general neural architectures. It means that graph structure is the input to the neural architecture (instead of being part of it), and the model parameters are shared to respect the invariance properties of the input graph [You et al., 2020]. GNN examples include GCN, GraphSage, and Graph Attention Network (GAT) [Veličković et al., 2018] with different message-passing architectures.

Expressiveness (Theory). It is desirable for GNNs to have a high level of *expressiveness*. The expressive power of GNNs is specified as the ability to distinguish different computational graphs (i.e. graph structures).

Isomorphic nodes are defined as the nodes that have the same computational graphs (i.e. same features and neighbourhood structure). GNNs are unable to distinguish two isomorphic nodes since they consider node features, not node IDs/indexes. Also, an *injective* function maps every unique input to a different output. The most expressive GNN maps every computational graph into different node embeddings, injectively.

Hence, the Aggregate function should be injective. That is, the expressiveness of GNNs can be characterized by the expressiveness of the Aggregate function over multisets (i.e. sets with repeating elements). For instance, GCN and GraphSage are not maximally powerful GNNs as they use non-injective mean and max pooling functions for aggregation, respectively. Theorem 1 recently proposed by [Xu et al., 2019] indicates that a sum pooling operation is an injective function over multisets.

Theorem 1 *Assume \mathcal{X} is countable. There exists a function $f : \mathcal{X} \rightarrow \mathbb{R}^n$ so that $h(X) = \sum_{x \in X} f(x)$ is unique for each multiset $X \subset \mathcal{X}$ of bounded size. Moreover, any multiset function g can be decomposed as $g(X) = \Phi(\sum_{x \in X} f(x))$ for some function Φ .*

Also, the universal approximation theorem [Hornik et al., 1989] states that a multi-layer perceptron (MLP) with sufficiently large hidden dimensionality (i.e. from 100 to 500 nodes) and appropriate non-linearity (e.g. ReLU activation) can approximate any continuous function with an arbitrary accuracy. The Graph Isomorphism Network (GIN), developed based on the aforementioned theorems [Xu et al., 2019], is one of the most expressive GNNs in the class of message-passing GNNs, where its expressiveness

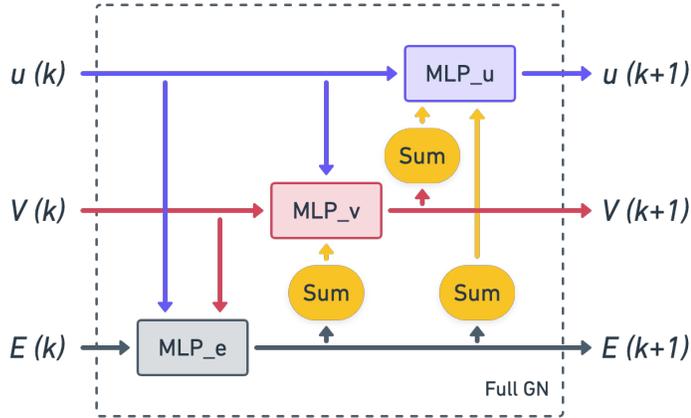


Figure 3.3: Algorithm overview of full graph network [Battaglia et al., 2018].

is upper bounded by WL test. It has the following relation:

$$h_{v_i}^{(k+1)} = \text{MLP}^{(k)}\left(h_{v_i}^{(k)}, \text{Sum}^{(k)}\left(\{h_{v_j}^{(k)} \mid \forall v_j \in N(v_i)\}\right)\right) \quad (3.3)$$

Therefore, sum pooling (Sum) and MLP will be used in a generalized message-passing framework. Note that we can also add rich node features to improve the expressiveness of the GNNs.

Graph Networks. A generalized message-passing framework is known as a graph network (GN) [Battaglia et al., 2018]. The important aspect in graph networks is that, during message passing, in addition to node embeddings h_{v_i} , we generate the edge embeddings h_{e_i} , as well as a global embedding h_u corresponding to the entire graph.

This allows the framework to easily integrate edge- and graph-level features. Also, recent work [Barceló et al., 2020] has shown GN to have benefits in terms of expressiveness compared to a standard GNN. Generating edge and global embeddings during message-passing also makes it trivial to define loss functions based on graph or edge-level classification tasks [Hamilton, 2020].

The algorithm overview of a one-step full GN architecture is shown in Figure 3.3 and Algorithm 1. Multi-layer perceptrons (MLPs) and element-wise sum pooling (Sum) are used as the Update and Aggregate functions for maximal expressiveness. The message-passing neural network (MPNN, also called interaction network) architecture [Gilmer et al., 2017], a simplified version of the full GN architecture, will be used for the physics simulator explained in §3.2.3.

Algorithm 1: Full graph network [Battaglia et al., 2018]

Input : $E^{(k)}, V^{(k)}, u^{(k)}$
Output : $E^{(k+1)}, V^{(k+1)}, u^{(k+1)}$
for $i = 1$ **to** N^e **do**
 $h_{e_i}^{(k+1)} \leftarrow \text{MLP}_e \left(h_{e_i}^{(k)}, h_{v_{r_i}}^{(k)}, h_{v_{s_i}}^{(k)} \right)$
for $i = 1$ **to** N^v **do**
 $\bar{h}_{e_i}^{(k+1)} \leftarrow \text{Sum} \left(\{ (h_{e_j}^{(k+1)}, i, s_j^{(k+1)}) \}_{j=1:N^e} \right)$
 $h_{v_i}^{(k+1)} \leftarrow \text{MLP}_v \left(\bar{h}_{e_i}^{(k+1)}, h_{v_i}^{(k)}, h_u^{(k)} \right)$
 $\bar{h}_e^{(k+1)} \leftarrow \text{Sum} \left(\{ (h_e^{(k+1)}, r_i^{(k+1)}, s_i^{(k+1)}) \}_{i=1:N^e} \right)$
 $\bar{h}_v^{(k+1)} \leftarrow \text{Sum} \left(\{ (h_{v_i}^{(k+1)}) \}_{i=1:N^v} \right)$
 $h_u^{(k+1)} \leftarrow \text{MLP}_u \left(\bar{h}_e^{(k+1)}, \bar{h}_v^{(k+1)}, h_u^{(k)} \right)$

Limitation. GNNs might be infeasible for large-scale applications on a GPU with mid-level limits on memory (i.e. VRAM of 10-20GB). Aside from mini-batch, one solution to this is to consider linear activation functions in GNNs. This has worked well for node classification benchmark [Wu et al., 2019]. Using this, one can pre-process graph features on a CPU (with large memory capability) offline.

However, this will limit the expressive power of GNNs due to the lack of non-linearity in generating the embeddings. In this research, we propose applying a non-learning, CPU-runnable dimensionality reduction method, PCA, to physics graph data in order to:

1. Significantly reduce the size of graph data,
2. Reduce the required message-passing steps to only one, and
3. Eliminate the need for computationally expensive edge construction algorithms.

3.1.2 Principal Component Analysis

Principal Component Analysis (PCA) is a non-parametric linear dimensionality reduction method [Pearson, 1901, Hotelling, 1933]. It provides a data-driven, hierarchical coordinate system to re-express high-dimensional correlated data. The resulting

3.1. INTRODUCTION AND BACKGROUND

coordinate system geometry is determined by principal components. These are sometimes called bases, modes or latent variables of the data. These modes are uncorrelated (orthogonal) to each other, but have maximal correlation with the *observations* [Brunton and Kutz, 2019]. Particularly, the goal of this method is to compute a subset of ranked principal components to summarize the high-dimensional data while retaining trends and patterns.

In PCA, dimensionality reduction mapping is defined as $f : S \rightarrow S'$. The matrix $S \in \mathbb{R}^{m \times n}$ includes m observations of n fullspace dimensions. And the matrix $S' \in \mathbb{R}^{m \times r}$ is m observations of r subspace modes. This method is under the following assumptions:

1. Linearity: $f(S) = SU$ where $U \in \mathbb{R}^{n \times r}$ is transformation matrix (or loadings).
2. The first two moments of data are sufficient statistics: The only probability distribution class that can be fully described by the mean and variance are exponential distributions (e.g. normal distribution). Thus, the probability of *dimensions* must be exponentially distributed.
3. Large variances represent important dynamics: The modes with larger associated variances have interesting dynamics.
4. The principal components are orthogonal: The dot products of the columns of the reduced data are zero, $S'_{*i} \cdot S'_{*j} = 0$.

There are two approaches to obtaining the reduced data using PCA: (1) singular value decomposition (SVD), and (2) eigenvalue decomposition (EVD). Here, we will use the EVD approach since we can have an easier access to the indexes of the modes in the full data if required.

First, the mean value \bar{s} over all elements of the full data $S_{m \times n}$ is computed via

$$\bar{s} = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n S_{ij} \quad (3.4)$$

and the mean matrix is $\bar{S}_{m \times n} = \bar{s}J_{m \times n}$ where $J_{m \times n}$ is an all-one matrix. The mean-subtracted data $B_{m \times n}$ is then given by

$$B_{m \times n} = S_{m \times n} - \bar{S}_{m \times n}. \quad (3.5)$$

Now covariance matrix $C_{n \times n}$ of the mean-subtracted data B is

$$C_{n \times n} = \frac{1}{n-1} B^T B \quad (3.6)$$

where it measures the degree of the linear relationship between two dimensions. In the covariance matrix, large values on the diagonal correspond to interesting dynamics. Large off-diagonal values correspond to high redundancy.

Next, the eigenvalues λ_i and eigenvectors w_i of $C_{n \times n}$ are calculated by solving the following equations

$$|C - \lambda_i I_n| = 0 \quad (3.7)$$

$$C w_i = \lambda_i w_i. \quad (3.8)$$

which are guaranteed to exist, since $C_{n \times n}$ is Hermitian.

Now we obtain the sorted eigenvector matrix $W = [w_1 \dots w_n]_{n \times n}$ based on the sorted eigenvalues in descending order. The transformation matrix $U_{n \times r}$ is then given by

$$U = \begin{bmatrix} w_1 & \dots & w_n \end{bmatrix}_{n \times r} \quad (3.9)$$

where the reduced data matrix is finally calculated via $S'_{m \times r} = (S_{m \times n} - \bar{S}_{m \times n})U_{n \times r}$.

It should be noted that both the strength and weakness of PCA is that it is a non-parametric method. In cases where standard PCA fails due to the existence of non-linearly in data, kernel PCA [Schölkopf et al., 1998] can be used. Compared to an autoencoder neural network, kernel PCA still does not require learning/optimization and prior specification of size of the latent space.

3.2 Subspace Graph Network Simulator

For engineering applications, the underlying physics govern the system response. A learning physics simulation can be discussed and assessed in terms of the following factors:

1. Accuracy: An accurate method ideally learns from clean and accurate datasets (§3.2.1).

3.2. SUBSPACE GRAPH NETWORK SIMULATOR

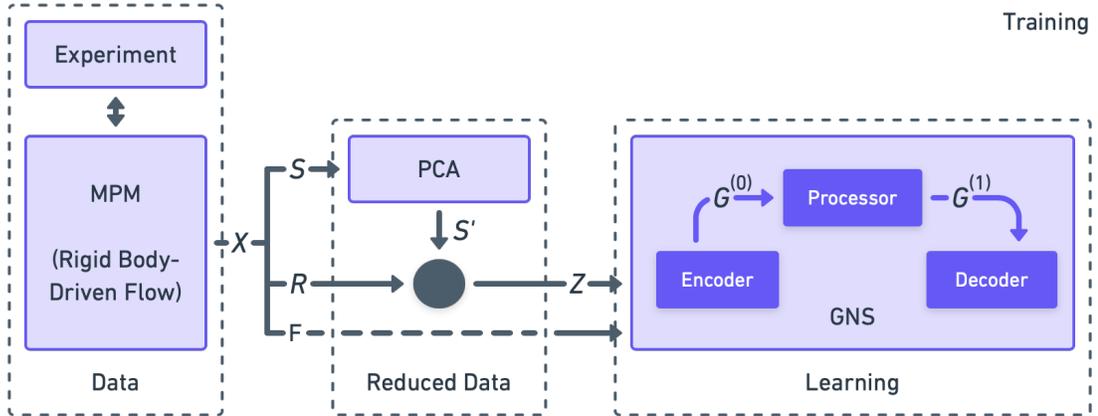


Figure 3.4: Training in our subspace GNS. Data X includes time series of flow states S , rigid states R , and interaction forces F . Also, S' and Z are reduced flow states and reduced system states. Here, states are particle positions.

2. Speed: A fast method can be achieved via reduced, memory-efficient data and/or by using reusable modules that can be run on modern accelerator hardware such as GPU (§3.2.2).
3. Generalization: Strong generalization is achievable by imposing *relational inductive bias* on the learning model. Models with strong inductive biases are more data efficient and can generalize much better to unseen scenarios (§3.2.3).
4. Differentiability: Differentiable models such as neural networks are appropriate for inverse problems (e.g. control systems).

In this section, we develop a real-time learning simulation approach for modeling rigid body-driven elasto-viscoplastic (and sometimes stress-free) granular flows. Focus on this application does not limit its potential applicability to other materials, not considered here. The approach will benefit from material point method (MPM), principal component analysis (PCA), and graph network simulator (GNS), and is decomposed to a training phase and a rollout phase. Such an approach has not been developed for granular flows before.

The training phase is illustrated in Figure 3.4. First, we generate training datasets via MPM verified by experiments. Then, we apply PCA to the full data in a pre-processing step (on CPU). As a result, depending on the application, a desired quality

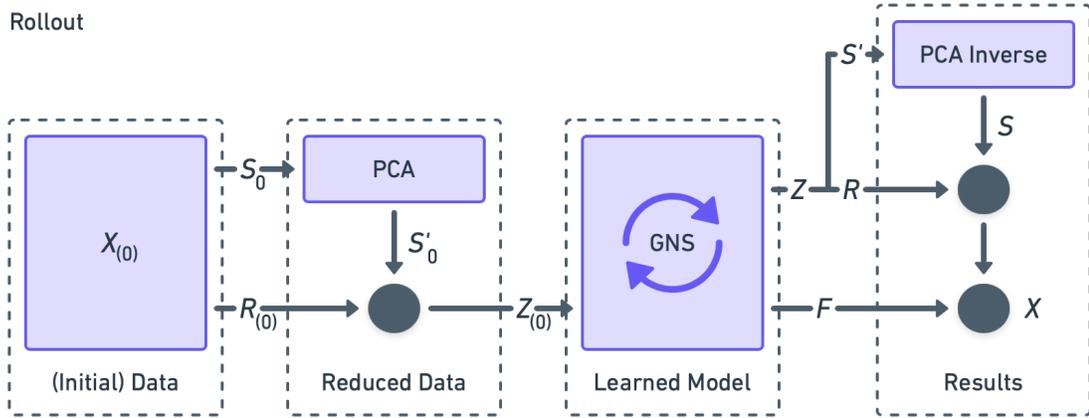


Figure 3.5: Rollout in our subspace GNS. Initial Data $X_{(0)}$ includes initial flow states S_0 and (initial) rigid states $R_{(0)}$. Also, S'_0 and $Z_{(0)}$ are reduced initial flow states and reduced (initial) system states. In the end, results X include time series of flow states S , rigid states R , and interaction forces F . Also, S' and Z are reduced flow states and reduced system states. Here, states are particle positions.

(i.e. considering both accuracy and memory-efficiency) can determine the number of PCA modes. Finally, we train GNS (a GN model with Encoder-Processor-Decoder scheme on GPU) using data representing subspace elasto-viscoplastic granular flow. The modules shown in the figure including Data (§3.2.1), Reduced Data (§3.2.2), and Learning (§3.2.3) will be elaborated in the following sections.

In the rollout phase, depicted in Figure 3.5, we use the learned subspace model to compute the granular flow-rigid body interaction forces (on CPU). Also, the fullspace granular flows are computed in a post-processing step (on GPU) which is useful for visualization purposes. Rollout is equivalent to a testing phase. The rollout phase will be explained in §3.2.4.

3.2.1 Data

In the previous chapter, MPM was validated by experimental configurations including wheel-soil, excavation, and Taylor-Couette flow. This has been shown to provide good matching with experimental results with runtime on the order of 10 to hundreds of times real-time. Now, hundreds of numerical simulations can be produced for the proposed machine learning approach.

3.2. SUBSPACE GRAPH NETWORK SIMULATOR

Table 3.1: Datasets specifications. Motion types include (1) a discrete ramp-horizontal-ramp motion, and (2) a continuous curved motion. Also, Grav., Fric., and Dia. stand for gravity, soil internal friction angle, and wheel diameter.

Dataset	Variables				
Excavation	Angle [deg]	Depth [cm]	Speed [cm/s]	Motion	
	{0,4,10,31,45}	{2,4,5,8,10}	{1,4,8,10,15}	{1,2}	
Wheel	Grav. [m/s ²]	Fric. [deg]	Load [N]	Dia. [cm]	Slip [%]
	{1.62,3.72,9.81}	{30,37,43}	{100,164,225}	{5,15,30}	{20,40,70}

Here, we generate two datasets: Excavation and Wheel. Each dataset has ~ 75000 data frames gathered with 60 Hz of data acquisition rate (as temporal discretization is not constrained by stability conditions, in data-driven methods). The Excavation dataset contains 250 examples with a single soil type and a blade cutting at various depths, speeds, rake angles (i.e. relative to vertical), and motion types. The Wheel dataset is slightly more complex. This includes 243 examples with soils having different internal friction angles, and wheels with various diameters operating at various normal load and slip (percentage of the wheel’s rotary motion not translating to forward linear motion). There are also multiple gravity conditions representing the Moon, Mars, and Earth. These variables are summarized in Table 3.1. Moreover, some examples are shown in Figure 3.6.

A dataset $X = \{X_i\}_{i=1:c}$ includes c examples of $X_i = \{S_i, R_i, F_i\}$ consisting of a time series of the positions of granular flow S_i and the rigid body R_i interacting with it, and a time series of the total interaction forces F_i (applied to the center of mass of the rigid body), where

$$S_i = \begin{bmatrix} x_1^1 & y_1^1 & z_1^1 & \dots & x_1^m & y_1^m & z_1^m \\ x_2^1 & y_2^1 & z_2^1 & \dots & x_2^m & y_2^m & z_2^m \\ \vdots & \vdots & \ddots & & \vdots & \vdots & \vdots \\ x_n^1 & y_n^1 & z_n^1 & \dots & x_n^m & y_n^m & z_n^m \end{bmatrix}_{3m \times n}^T \quad (3.10)$$

with m and n as the numbers of timesteps (frames) and flow particles, respectively.

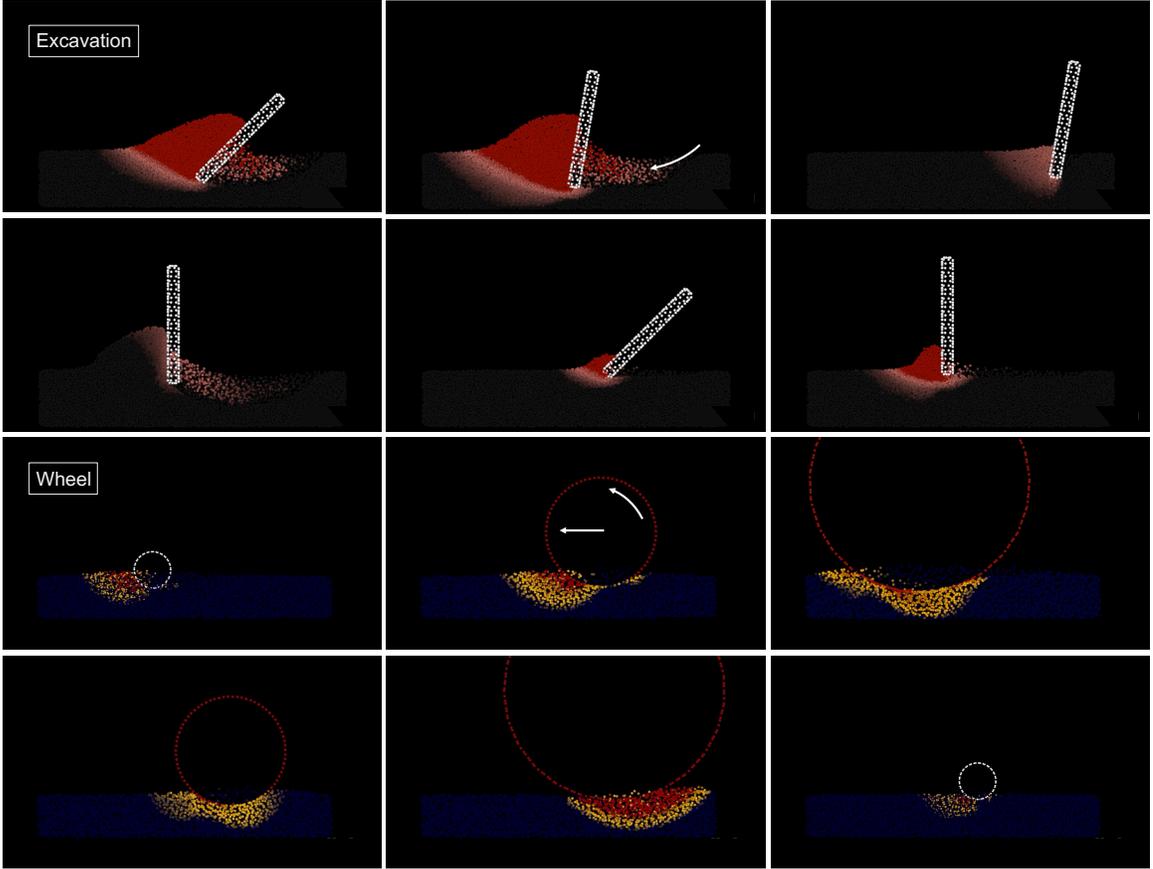


Figure 3.6: Side view (cross-section) of some Excavation and Wheel training examples with various initial and boundary conditions generated via MPM.

In fact, each column in S_i represents a time series of particle positions in 3D. The R_i has also a similar structure. Moreover, the interaction forces F_i are given by

$$F_i = \begin{bmatrix} F_x^1 & F_x^2 & \dots & F_x^m \\ F_y^1 & F_y^2 & \dots & F_y^m \\ F_z^1 & F_z^2 & \dots & F_z^m \end{bmatrix}_{m \times 3}^T. \quad (3.11)$$

We divide the dataset into two subsets: the training split including 90% and test split including 10% of the examples. We work with the training split in the training phase, and with the test split in the rollout phase. Also, a validation split is not required as we will use already tuned model hyperparameters [Sanchez-Gonzalez et al., 2018].

Note that as we separately construct the S_i and R_i data matrices, the particle

types (i.e. flow, rigid/boundary, etc.) are recognizable. They are used as node features in the graph network explained in §3.2.3.

3.2.2 Reduced Data

Physical systems in MPM can be described by particles each with three coordinates (degrees of freedom) in 3D. The minimum number of particles are subject to stability conditions (discussed in §2.2.2). Thus, such systems are often high-dimensional for large scale problems. Note, to conserve the particle coordinates, we define the dimensions by the particles rather than the system’s degrees of freedom. However, the effective dimensions of a physical system can be far smaller than the system dimensionality. PCA, as a model reduction method, can capture the effective dimensions of the physical system. In fact, PCA enables us to calculate the complexity of the system.

Reduced Data Computation. We aim to apply PCA to the granular flow data, and pass the rigid body as is ([Holden et al., 2019]). Therefore, we compute the PCA transformation matrix $U_{n \times r}$ using the procedure described in §3.1.2 where r is the number of modes we select, and the data matrix is S_{train} including the flow states of each example in the training split as follows

$$S_{\text{train}} = \begin{bmatrix} S_1 & S_2 & \dots & S_{c_{\text{train}}} \end{bmatrix}_{3c_{\text{train}}m \times n}^T, \quad (3.12)$$

where c_{train} is the number of examples in the training split.

Now we can compute the reduced flow states $S' = \{S'_i\}_{i=1:c}$ by applying the PCA transformation $U_{n \times r}$ separately to each example in $S = \{S_i\}_{i=1:c}$ as follows

$$S'_i = (S_i - \bar{S}_{\text{train}})U. \quad (3.13)$$

Finally, we construct the system states $Z = \{Z_i\}_{i=1:c}$, where

$$Z_i = \begin{bmatrix} S'_i & R_i \end{bmatrix}. \quad (3.14)$$

We omit the subscript i hereinafter for simplicity, and decompose the states for each example, Z_i , into multiple $(d + 1)$ timestep windows of states $Z^{t-(d-1), \dots, t, t+1}$ as the inputs to the Learning module (in §3.2.3). Note, the Z^{t+1} are as the target (or label) states. The F^{t+1} is also passed to the Learning module as the target interaction force.

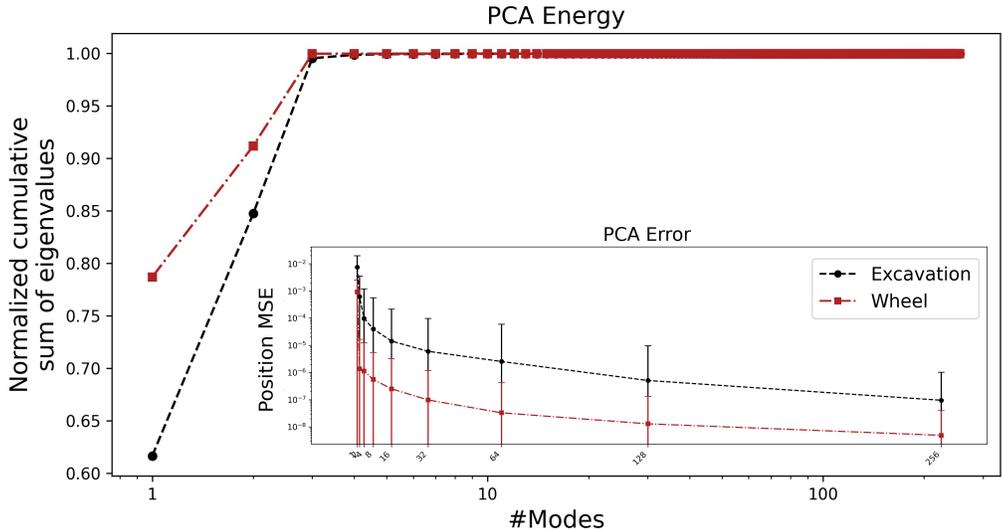


Figure 3.7: **Outset:** Normalized cumulative sum of the transformation matrix eigenvalues (modes) representing their energy. **Inset:** Position MSE between original full data and mapped-back full data. Errors bar represent one standard deviation.

Mode Number Selection. Each example in the Excavation and Wheel datasets is composed of 27000 and 6000 particles (or dimensions), respectively. To quantitatively determine the complexity of the physics data, we compute the normalized cumulative sum of the (sorted) eigenvalues corresponding to the columns (i.e. modes) of the transformation matrix U via

$$\text{Energy} = \frac{\sum_{i=1}^r \lambda_i}{\sum_{i=1}^n \lambda_i}. \quad (3.15)$$

This is used to select an appropriate number of modes, (r), that can capture most of the Energy (i.e. variance) of the data [Brunton and Kutz, 2019]. Figure 3.7(outset) shows that the very first modes (< 10 out of 27K and 6K dimensions) can be used to construct the Excavation and Wheel data in subspace (i.e. PCA space) while keeping more than 99% of the data variance.

Moreover, to measure the position mean squared error (MSE), we (1) map the data to the subspace using different numbers of modes ($r = 1, 2, 4, \dots, 256$), (2) map back the reduced data to the fullspace, and (3) compare the mapped-back data with the original data. Based upon the results shown in Figure 3.7(inset), we choose 8 modes to use as inputs to the graph network (discussed in §3.2.3) for learning the subspace

dynamics of the granular flows. This will reduce the time and memory complexity of the graph network in both the training and rollout phases, while keeping position MSE below 10^{-4} .

3.2.3 Learning

Graph neural networks (GNNs) should “express” as many *inductive biases* as possible. Inductive biases allow learning algorithms to prioritize one solution over another, independent of the observed data [Mitchell, 1980]. Also, *relational reasoning* involves manipulating structured representations (graphs) of entities (nodes) and relations (edges), using rules (functions) for how they can be composed [Battaglia et al., 2018]. Graph networks (GNs) carry strong relational inductive biases which guide their approach towards learning about entities and relations [Battaglia et al., 2018]. They, in fact, impose constraints on relationships among graph nodes in a learning process. Furthermore, some non-relational inductive biases are used in deep learning, including non-linear activations, dropout [Srivastava et al., 2014], normalization [Ioffe and Szegedy, 2015], etc. all imposing constraints on the outcome of learning.

The inductive biases can improve the *combinatorial generalization* of GNs, an important aspect of a learning physics simulation approaches (factor 3 listed in §3.2). The principle of combinatorial generalization is constructing new predictions from known observation. This has been at the heart of the broader field of artificial intelligence [Chomsky, 1957]. In addition, the GNs’ node- and edge-level functions are shared across all edges and nodes, respectively. This means GNs automatically support a form of combinatorial generalization [Battaglia et al., 2018].

In terms of physics modeling, GNs have analogies with physical systems (in Newtonian mechanics) [Shlomi et al., 2021, Cranmer et al., 2020]. If we assume that the graph nodes are like particles, then the particle internal forces, net internal forces, and accelerations, are analogous to the graph edges, Sum functions, and node Update function, respectively. These can be used iteratively to compute the positions in the next timestep. Here we aim to learn the subspace dynamics of the physical systems via graph network simulator (GNS) proposed by [Sanchez-Gonzalez et al., 2018]. GNS has an Encoder-Processor-Decoder scheme which is a combination of the Encoder-Decoder and message-passing frameworks introduced in §3.1.1.

3.2. SUBSPACE GRAPH NETWORK SIMULATOR

GNS is a simulator $s_\theta : Z^{t-(d-1), \dots, t-1, t} \rightarrow Z^{t+1}$ with learning parameters θ . It predicts the future states of a physical system \hat{Z}^{t+1} given a finite time history (d timesteps) of the states $Z^{t-(d-1), \dots, t-1, t}$. As mentioned in §3.2.2, Z includes subspace particle positions and particle type information. GNS modules including Encoder, Processor, and Decoder will be described below along with minor but effective customization given our reduced data. Afterwards, the training specifications will be detailed.

Encoder. The Encoder : $Z^{t-(d-1), \dots, t-1, t} \rightarrow G^{(0)}$ embeds the particle states $Z^{t-(d-1), \dots, t-1, t}$ as an initial latent graph $G^{(0)} = (E^{(0)}, V^{(0)})$ in the current timestep t . The $V^{(0)} = \{h_{v_i}^{(0)}\}_{i=1:N^v}$ is a set of node embeddings $h_{v_i}^{(0)} = \text{MLP}_{\text{enc}, v}(v_i^{(0)})$ where $v_i^{(0)}$ are node features. The $E^{(0)} = \{(h_{e_i}^{(0)}, r_i, s_i)\}_{i=1:N^e}$ is a set of edge embeddings $h_{e_i}^{(0)} = \text{MLP}_{\text{enc}, e}(e_i^{(0)})$ where r_i is the index of receiver node, s_i is the index of sender node, and e_i^t are edge features. Note that the global feature u (such as gravity and internal friction angle) has been omitted, as all required features can be incorporated into the node embeddings in our application.

Here, particle types and a finite time history of particle velocities ($d-1$; velocities computed from particle positions) are considered as the node features v_i . Also, relative positions between the sender and receiver nodes are added as the edge features e_i . This choice of feature augmentation makes the position encoding spatial equivariant. In other words, this causes GNNs to perform better in position-aware tasks as GNNs have limited power in capturing the position of a given node with respect to all other nodes of the graph [You et al., 2019]. The general solution is to construct the relative position by anchoring a point as a reference. In GNS, the anchoring points are also chosen to be relative (e.g. sender nodes).

In the current approach, we construct a complete graph (i.e. all nodes are connected with all other nodes via two-way directed edges). This is because we are using physics particles as nodes in subspace and their relationships in this space might not be proximity-based (this is an open question for us). As a result, it eliminates the need to use a nearest neighbor algorithm for edge construction.

Processor. The Processor : $G^{(0)} \rightarrow G^{(1)}$ is used as a message-passing step to compute the updated latent graph $G^{(1)} = (E^{(1)}, V^{(1)})$ given the initial latent graph $G^{(0)}$. This uses interaction network architecture [Gilmer et al., 2017] – a simplified

3.2. SUBSPACE GRAPH NETWORK SIMULATOR

Algorithm 2: Interaction network

Input : $E^{(0)}, V^{(0)}$
Output : $E^{(1)}, V^{(1)}$
for $i = 1$ **to** N^e **do**
 $h_{ei}^{(1)} \leftarrow \text{MLP}_{\text{proc},e} \left(h_{ei}^{(0)}, h_{vr_i}^{(0)}, h_{vs_i}^{(0)} \right)$
for $i = 1$ **to** N^v **do**
 $\bar{h}_{ei}^{(1)} \leftarrow \text{Sum} \left(\{ (h_{ej}^{(1)}, i, s_j) \}_{j=1:N^e} \right)$
 $h_{vi}^{(1)} \leftarrow \text{MLP}_{\text{proc},v} \left(h_{vi}^{(0)}, \bar{h}_{ei}^{(1)} \right)$

version of the full GN architecture. The equations in the interaction network are shown in Algorithm 2.

In addition to the spatial equivariance inductive bias (in the Encoder, already described above), the Processor introduces other physics-informed inductive biases in the processor including:

1. Permutation equivariance: The Update functions (MLPs) are over a set of embeddings. The set representation does not impose ordering (whereas a vector does).
2. Pairwise interactions: The edge Update function ($\text{MLP}_{\text{proc},e}$) captures the pairwise interactions between nodes (inter-particle forces in physics).
3. Superposition principle: The sum pooling as the Aggregate function behaves similar to superposition (superposition principle in physics).
4. Local interactions: The node Update function ($\text{MLP}_{\text{proc},v}$) captures the local interaction. It means that the effects of nearby nodes are larger than the farther nodes on a target node (nonlocality in physics).
5. Universal rules: The node and edge Update functions (MLPs) are shared for each node and edge, respectively. Hence they can learn the underlying universal rules (the laws of physics).

Also, another inductive bias (inertial motion) will be incorporated in the Decoder. These physics-informed inductive biases make GNS as general-purpose as possible for physics simulations. Note that, as a customization specific to modeling subspace

3.2. SUBSPACE GRAPH NETWORK SIMULATOR

physics, we use one message-passing step ($N^k = 1$) since our graph is complete and thus messages can reach any node from any other node in a single step.

Decoder. The Decoder : $G^{(1)} \rightarrow \hat{o}^{t+1}$ extracts the dynamics information \hat{o}^{t+1} from the updated latent graph $G^{(1)}$. We define and compute the output \hat{o}^{t+1} via

$$\hat{o}^{t+1} := \begin{bmatrix} \hat{a}_1^{t+1} & \cdots & \hat{a}_{N^v}^{t+1} \\ \hat{F}_1^{t+1} & \cdots & \hat{F}_{N^v}^{t+1} \end{bmatrix}_{6 \times N^v} = \text{MLP}_{\text{dec}}(G^{(1)}) \quad (3.16)$$

as a combination of particle (node) accelerations \hat{a}_i^{t+1} and particle interaction force contributions \hat{F}_i^{t+1} to the total interaction force $\hat{F}^{t+1} = \sum_{i=1}^{N^v} \hat{F}_i^{t+1}$ applied to the center of mass of the rigid body. The MLP_{dec} is applied to each particle separately. Also, the predicted particle states \hat{Z}^{t+1} can then be computed based on the predicted particle positions \hat{z}_i^{t+1} given by

$$\hat{z}_i^{t+1} = z_i^t + \Delta t \dot{z}_i^t + \Delta t^2 \hat{\ddot{z}}_i^{t+1} \quad (3.17)$$

where $\dot{z}_i^t = (z_i^t - z_i^{t-1})/\Delta t$, $\hat{\ddot{z}}_i^{t+1} = \hat{a}_i^{t+1}$, and $\Delta t = 1$. Basically, GNS focuses on the non-trivial, higher-order term (i.e. particle acceleration \ddot{z}) of equation (3.17). In fact, this term is a function of the particle internal force f (as $f = m\ddot{z}$, with m here being mass). And f is a function of particle stress tensor T including the effect of all elasto-visco-plastic (or sometimes stress-free) deformations (as $f = V \text{div} T$, with constant particle mass and volume, V ; not to be confused with the set of nodes V). In GNS, it is also possible to include the effect of external forces by adding gravity, external friction coefficient, etc. to the node features. With these, GNS fundamentally becomes a force-based simulator like MPM by predicting the right hand-side of the momentum equation (as opposed to position-based methods like PBD).

From another point of view, GNS predicts the deviation from the current position leading to a more stable prediction. Note that here, the accelerations are assumed to be constant between the timesteps. So the accelerations between timesteps t and $t+1$ are shown by $\hat{\ddot{z}}_i^{t+1}$ (or \hat{a}_i^{t+1}). Also in equation (3.17), it is not necessary to include the 1/2 coefficient since $\hat{\ddot{z}}_i^{t+1}$ is the output of an MLP that can learn this coefficient.

Specifications. As explained above, five MLPs are used as the Update functions in GNS. These include $\text{MLP}_{\text{enc},e}$, $\text{MLP}_{\text{enc},v}$, $\text{MLP}_{\text{proc},e}$, $\text{MLP}_{\text{proc},v}$, and MLP_{dec} whose parameters (i.e. weights) θ should be optimized. Each MLP has two hidden layers (with ReLU activations) and an output layer each with size of 128, except for the

3.2. SUBSPACE GRAPH NETWORK SIMULATOR

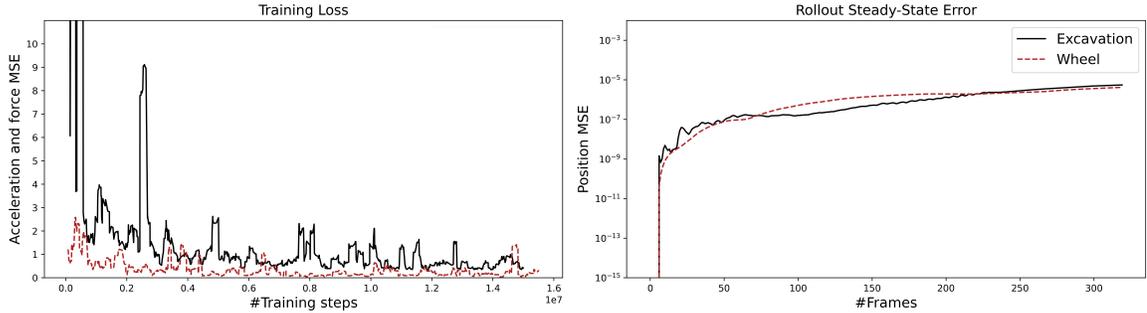


Figure 3.8: Training loss convergence in ~ 15 M steps (left) and steady-state rollout error of positions for both Excavation and Wheel.

MLP_{dec} whose output layer size is 6 given our defined output vector (i.e. concatenated vector of particle acceleration and interaction force contribution).

We define the loss function L as the sum of the mean squared error (MSE) of the accelerations a and squared error of the interaction force F given by

$$L(Z^t, Z^{t+1}; \theta) = \left(\sum_{i=1}^{N^v} \|\hat{a}_i^{t+1} - a_i^{t+1}\|^2 \right) / N^v + \|\hat{F}^{t+1} - F^{t+1}\|^2 \quad (3.18)$$

where F^{t+1} is the target interaction force, and the target accelerations a_i^{t+1} are approximated using a temporal second-order central finite difference scheme on the positions given by

$$a_i^{t+1} = \ddot{z}_i^{t+1} = \frac{z_i^{t-1} - 2z_i^t + z_i^{t+1}}{\Delta t^2} \quad (3.19)$$

where z_i^{t+1} are the target positions and $\Delta t = 1$. Adam optimizer [Kingma and Ba, 2017] is used to minimize the loss function over a mini-batch of size 2. The exponential learning rate is set to decay from 10^{-4} to 10^{-6} through a maximum of 20M training steps. We used the tuned model hyperparameters from [Sanchez-Gonzalez et al., 2018].

To accelerate the training process, the data is element-wise normalized to zero mean and unit variance with pre-computed means and variances of the training split. Also, a random-walk noise, $\text{Normal}(0, 0.0003)$, is applied to the node positions and velocities [Sanchez-Gonzalez et al., 2018]. This can resolve error accumulation issues; a roughly steady-state rollout Position MSE was achieved, as shown in Figure 3.8(right), for both Excavation and Wheel. The training loss convergence achieved in ~ 15 M steps is also shown in Figure 3.8(left).

The training was performed on an NVIDIA GeForce RTX 3080 10GB GDDR6X graphics card. The post-processing visualizations that will be described in §3.2.4, were done in Python. The open-source code of the GNS model has been developed primarily via TensorFlow [Abadi et al., 2016] and Graph Nets library [Battaglia et al., 2018] and published by Google DeepMind [Sanchez-Gonzalez et al., 2018]. We have, however, customized it in a way to be feasible and appropriate for robotics applications by adding real-time 3D force prediction functionality. Our code is publicly available on GitHub here¹.

3.2.4 Rollout

The training phase described in the previous section prepares the model for the rollout phase depicted in Figure 3.5. The rollout consists of the Initial Data, pre-processing Reduced Data, Learned Model, and post-processing Results modules.

Initial Data includes an initial d -timestep history of flow states S_0 . Also, the rigid states should be included. The rigid states can be either initialized (R_0) or scripted/interactive (R). The former means that the model will predict the trajectory of the rigid body interacting with flow as well; this choice should be consistent with the model already trained. Next, in Reduced Data, the initial flow states is reduced to a subspace representation (S'_0) using PCA (see equation (3.13)). The flow and rigid states $Z_{(0)} = \{S'_0, R_{(0)}\}$ are then input to the learned GNS model. This model predicts reduced flow states S' and full interaction forces F applied to the center of mass of the rigid body (and full rigid states R if applicable). The reduced flow states can then be mapped back (S) using PCA Inverse as follows

$$S = (S' U^T) + \bar{S}_{\text{train}}. \tag{3.20}$$

For interactive applications, the pre-processing PCA and post-processing PCA Inverse steps can be performed using GPU, and the subspace GNS using CPU [Holden et al., 2019]. Note that here states include particle positions.

In the following, we will examine the efficiency and accuracy of our approach using an example in the test split of the Excavation and Wheel datasets. Then, we present

¹github.com/haeriamin/Subspace-Graph-Physics

3.2. SUBSPACE GRAPH NETWORK SIMULATOR

Table 3.2: MPM (Physics), subspace GNS (Ours) and fullspace GNS (Ref) runtime comparison. GPU¹, GPU², CPU¹, and CPU² are NVIDIA GeForce RTX 3080 10GB, NVIDIA Tesla P6 16GB, Intel Core i7-10700 2.9GHz 8-Core, and Intel Core i7-6700 3.4GHz Quad-Core, respectively. One CPU¹ core and four CPU² cores are used.

Dataset	Method	Training [sec/step]		Rollout [sec/sec]		Simulation
		GPU ¹	GPU ²	CPU ¹	GPU ¹	CPU ²
Excavation	Physics	–	–	–	–	270
	Ref	OOM	1.000	–	OOM	–
	Ours	0.016	0.064	0.350	0.440	–
Wheel	Physics	–	–	–	–	240
	Ref	OOM	0.500	–	OOM	–
	Ours	0.017	0.068	0.380	0.460	–

some further results to compare the ground truths and predictions in subspace and over multiple examples in the test split.

Efficiency. Table 3.2 provides the runtime of our approach, subspace GNS, in training and rollout phases using CPU (with one core) and GPU (including TFRecord file deserialization). It also includes the fullspace GNS runtime as reference. Due to out-of-memory (OOM) issue, the fullspace GNS with our large-scale configurations was not trainable on the aforementioned GPU (RTX 3080 10GB). However, using an NVIDIA Tesla P6 16GB graphics card, we were able to measure the fullspace GNS runtime on the Excavation and Wheel datasets. The results show that our approach is real-time (also $\sim 700x$ faster than MPM) whereas the fullspace GNS is not even feasible for large-scale 3D physics configurations.

Accuracy. We separately assess the accuracy of PCA and GNS. To this end, we compare the distribution of PCA and GNS position MSEs in Excavation and Wheel configurations. Particularly, we compute the MSE of each particle’s positions through time (using 8 PCA modes). To obtain the PCA error, we (1) apply PCA to the original MPM data, (2) apply PCA Inverse to the resulting reduced MPM data, and (3) compute the MSE between the resulting mapped-back data (i.e. PCA Inverse(PCA(MPM))) and original MPM data. To obtain the GNS error, we (1) perform GNS using the reduced MPM data (one example in test split), (2) apply

3.2. SUBSPACE GRAPH NETWORK SIMULATOR

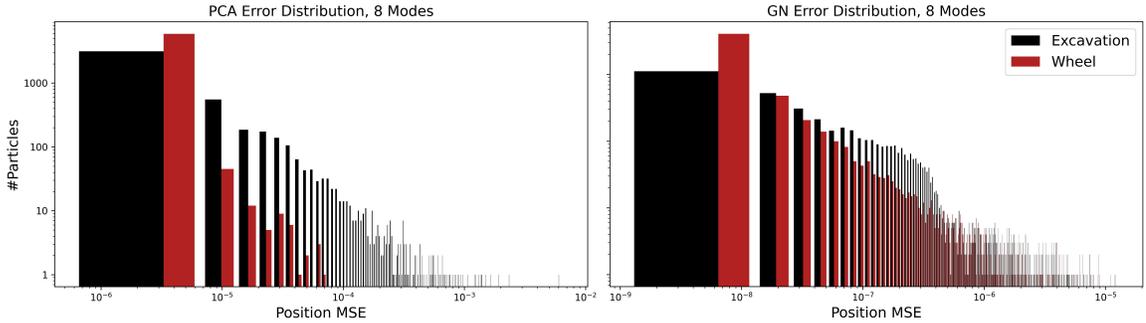


Figure 3.9: Distribution of PCA and GNS position MSEs in Excavation and Wheel configurations. Note that the varying width of bars is only due to logarithmic scaling of the x-axis.

PCA Inverse to the resulting predicted reduced GNS data, and the reduced data, and (3) compute the MSE between resulting mapped-back predicted data (i.e. PCA Inverse(predicted GNS); recall GNS works in the reduced subspace) and mapped-back data (i.e. PCA Inverse(PCA(MPM))). Thus they are separate errors.

The histograms shown in Figure 3.9 indicate that the position MSE ranges from 10^{-6} to 10^{-2} in PCA, and from 10^{-9} to 10^{-5} in GNS. But most particles have the lowest error in these ranges. This is somehow expected as most of the particles are stationary in both the Excavation and Wheel configurations. However, the notable point here is that the order of PCA error is higher than the GNS error. This suggests that a more accurate dimensionality reduction method with the same mode number can improve the accuracy of the final particle positions. The rest of the current section will focus on the accuracy of GNS in predicting the reduced data (i.e. in subspace learning).

The specifications of each example chosen from the Excavation and Wheel test splits are shown in Table 3.3. Note that in the Wheel configuration, the model learns the motion of the rigid body (i.e. wheel) as well. Also, the properties of flow (i.e. soil) and rigid body in the Wheel dataset are varying. These make the Wheel configuration more challenging for the GNS to learn.

Now we assess the GNS accuracy in predicting particle positions and velocities in subspace. Figure 3.10 shows side views of the Excavation and Wheel ground truth and predicted simulations in time (using the test examples introduced above). Red to black colors indicate the particle velocity magnitudes. There is a good agreement

3.2. SUBSPACE GRAPH NETWORK SIMULATOR

Table 3.3: Test example specifications. Motion type 2 indicates blade has continuous curved motion. Also, Grav., Fric., and Dia. stand for gravity, soil internal friction angle, and wheel diameter.

Dataset	Variables				
Excavation	Angle [deg]	Depth [cm]	Speed [cm/s]	Motion	
	45	10	10	2	
Wheel	Grav. [m/s^2]	Fric. [deg]	Load [N]	Dia. [cm]	Slip [%]
	3.72	30	100	15	40

between the ground truth and predicted elasto-viscoplastic flows.

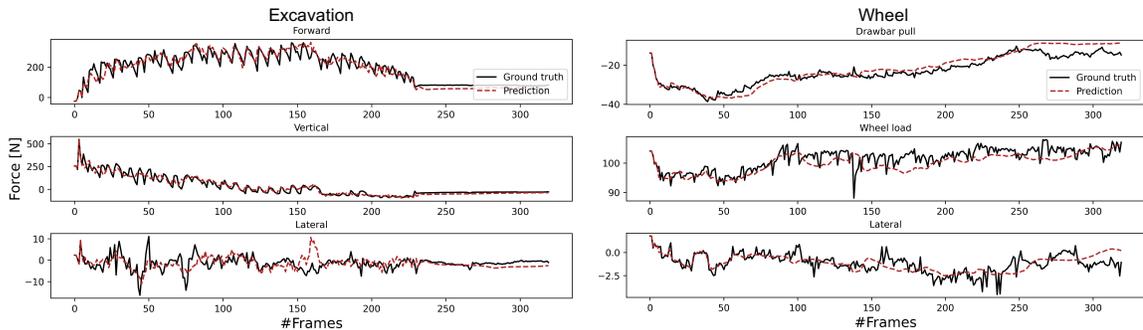


Figure 3.11: Ground truth and predicted Excavation and Wheel interaction forces. They are plotted in forward (drawbar pull), vertical (wheel load), and lateral directions.

Note that the results have been mapped back by PCA Inverse. Also, arrows with blue to green colors represent the directions and magnitudes of interaction forces applied to the center of mass of the blade or wheel.

We further assess the GNS accuracy in predicting interaction forces in subspace through time. It should be noted that the PCA accuracy does not directly affect the GNS force prediction accuracy as the force prediction is done in subspace via GNS. In other words, as long as the GNS learns the subspace dynamics accurately the force prediction will be accurate as well. Figure 3.11 shows the ground truth and prediction interaction forces in the aforementioned test examples of the Excavation and Wheel datasets. The GNS model has predicted the Excavation forces more

3.2. SUBSPACE GRAPH NETWORK SIMULATOR

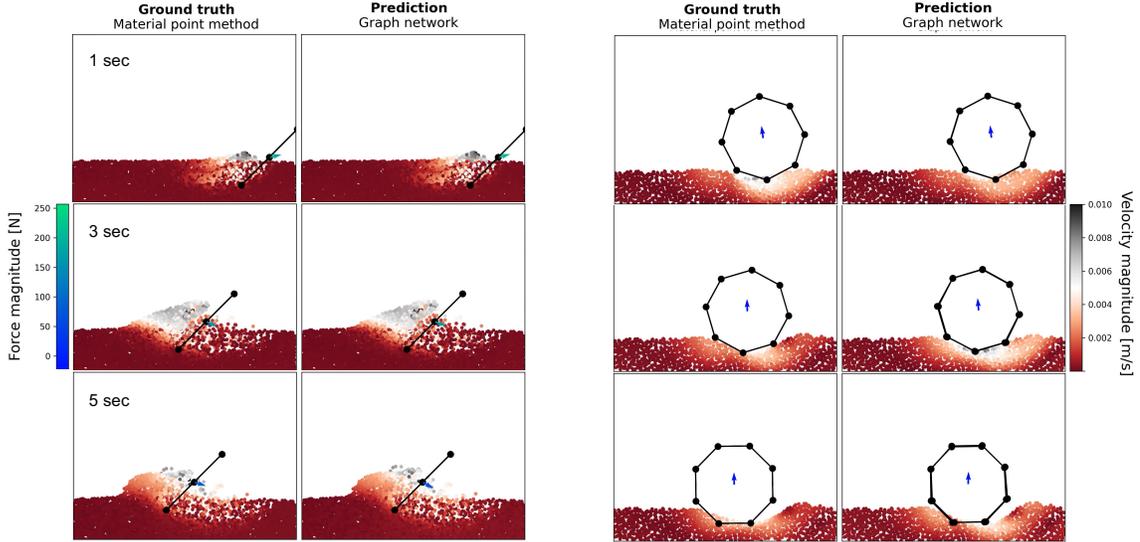


Figure 3.10: Side views of the Excavation and Wheel (cross-section) ground truth and predicted simulations in time (at 1, 3, and 5 second marks). Red to black colors indicate the particle velocity magnitudes. Arrows with blue to green colors represent directions and magnitudes of interaction forces.

accurately than the Wheel forces (see Table 3.4). As mentioned the Wheel dataset is more non-trivial to learn. Therefore, we suggest training multiple Wheel GNS models separately by datasets each with the same soil properties and/or gravity condition (like the Excavation dataset).

More Results. Here, we visualize GNS subspace prediction in 3D. Figure 3.12 reveals the underlying subspace dynamics predicted by GNS in time (at 1, 3, and 5 second marks) for both the Excavation and Wheel configurations. The subspace velocities of the 8 particles as well as the fullspace interaction forces are shown. This demonstrates the existence of dynamics in subspace that GNS has been able to learn. Such subspace dynamics have also been discussed in [Champion et al., 2019]. Methods such as symbolic regression can be used to extract the potential subspace parametric governing equation for further analysis [Cranmer et al., 2020]. In addition, the 3D mapped-back results (by PCA Inverse) are shown in Figure 3.13. As already explained and shown in these figures, the forces themselves are not reduced by PCA. The forces are not inputs to the GNS model; they are predicted based on the learned subspace dynamics.

We also provide the results of multiple examples in test splits. Figures 3.14 and

3.2. SUBSPACE GRAPH NETWORK SIMULATOR

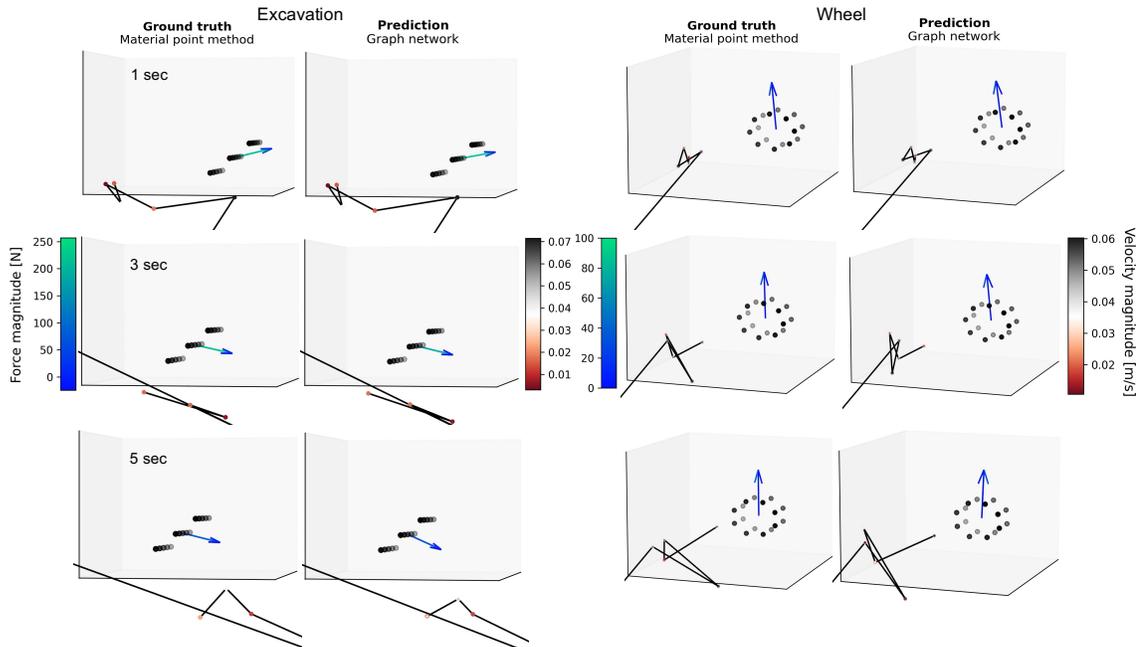


Figure 3.12: Underlying subspace dynamics predicted by GNS in time for both the Excavation and Wheel configurations (vs ground truth). Subspace particle velocity magnitudes and fullspace interaction forces are visualized.

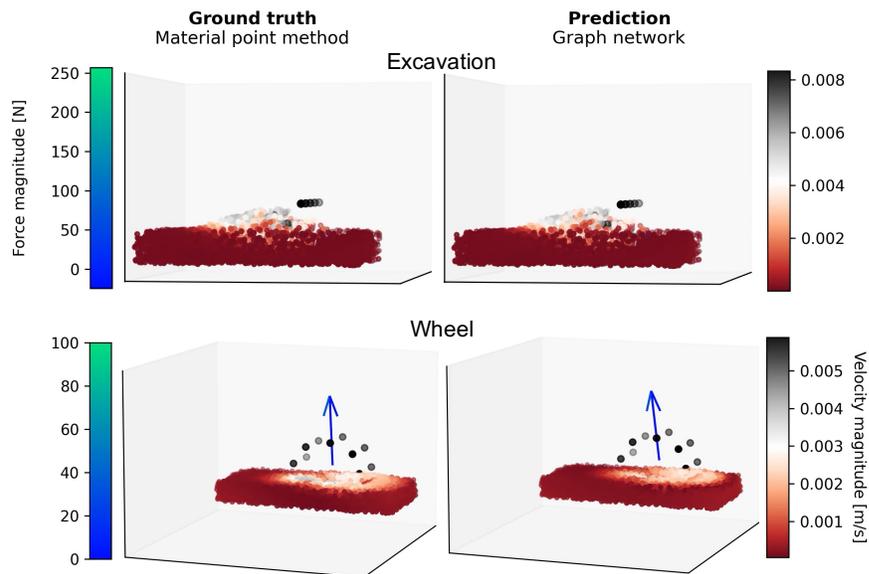


Figure 3.13: Underlying fullspace dynamics predicted by GNS at 3 second mark for both the Excavation and Wheel configurations (vs ground truth). Fullspace particle velocity magnitudes and interaction forces are visualized.

3.15 show the mapped-back particle positions and velocities, and interaction forces at 3 second mark of 6 test examples in Excavation and Wheel datasets, respectively. As observed in the Excavation results, we can even change the discretization of the rigid body and still achieve accurate results (there are 20 rigid particles in number 2 and 4 whereas 15 in others). Although this should be examined further, in the current research this could be an indication that GNS models the particle interactions similarly to how a physics simulation approach such as MPM does. More research on this has been detailed in [Sanchez-Gonzalez et al., 2018].

Furthermore, the time history of Excavation and Wheel interaction forces in 9 test examples are shown in Figures 3.16 and 3.17, respectively. Each plot of the figures include forces in forward, vertical and lateral directions. In most of the examples, the predictions are very well matched with the ground truths (in the test split).

Finally, we summarize each example’s position and force prediction errors in Table 3.4. It includes position MSE and force MPE (mean percentage error) of the 9 examples in the test split. Also, MPM force MPEs (relative to experiment) are provided for reference. Forces are in forward and vertical directions in the Excavation and Wheel configurations, respectively. As already discussed, force prediction is more accurate in the Excavation dataset due to no variation in the soil and blade properties in this dataset (i.e. only the blade motion is varying). Also, because of higher motion in the Excavation dataset, the mean position MSE is higher in Excavation than in Wheel.

Table 3.4: Position MSE and force MPE of test examples in subspace GNS rollout. MSEs (mean squared error) and MPEs (mean percentage errors) are specified relative to MPM. Also, MPM force MPEs are relative to experiment. Forces are in forward and vertical directions in Excavation and Wheel respectively.

Test Example	Position MSE $\times 10^{-4}$		Force MPE [%]	
	Excavation	Wheel	Excavation	Wheel
1	0.01	0.01	4.5	10.9
2	0.01	0.02	-5.3	-16.2
3	0.00	0.00	0.3	-18.5
4	0.02	0.03	-10.5	-22.8
5	2.41	0.05	11.6	-21.7
6	0.02	0.09	-2.4	-5.1
7	2.86	0.03	-17.2	0.2
8	0.01	0.02	-1.9	26.6
9	0.03	0.03	-5.1	-31.0
Mean	0.60	0.03	—	—
MPM	—	—	-0.5	5.2

3.2. SUBSPACE GRAPH NETWORK SIMULATOR

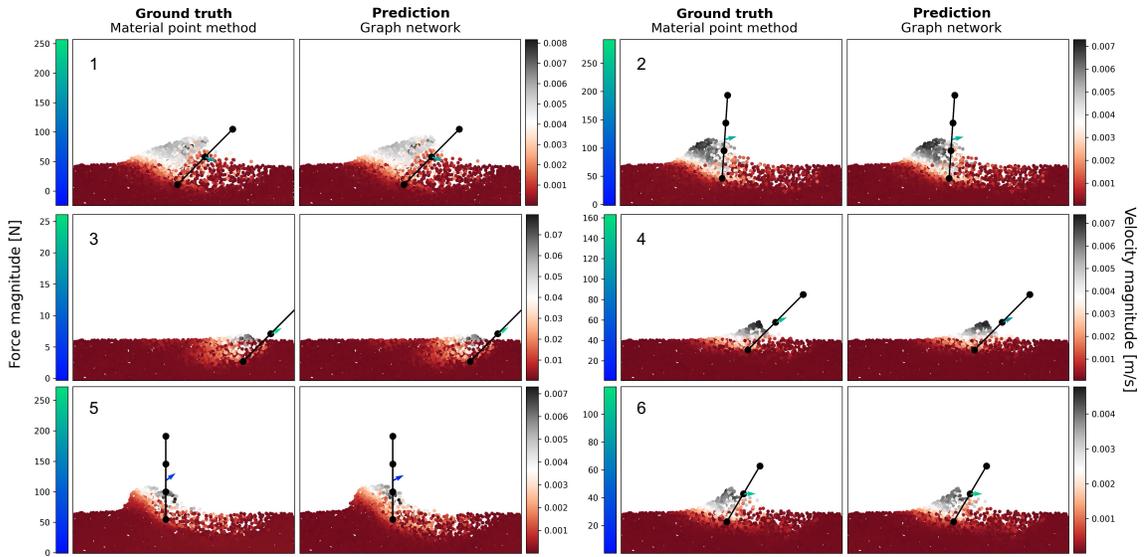


Figure 3.14: Side views of 6 Excavation test examples at 3 second mark. Red to black colors indicate the particle velocity magnitudes. Arrows with blue to green colors represent directions and magnitudes of interaction forces.

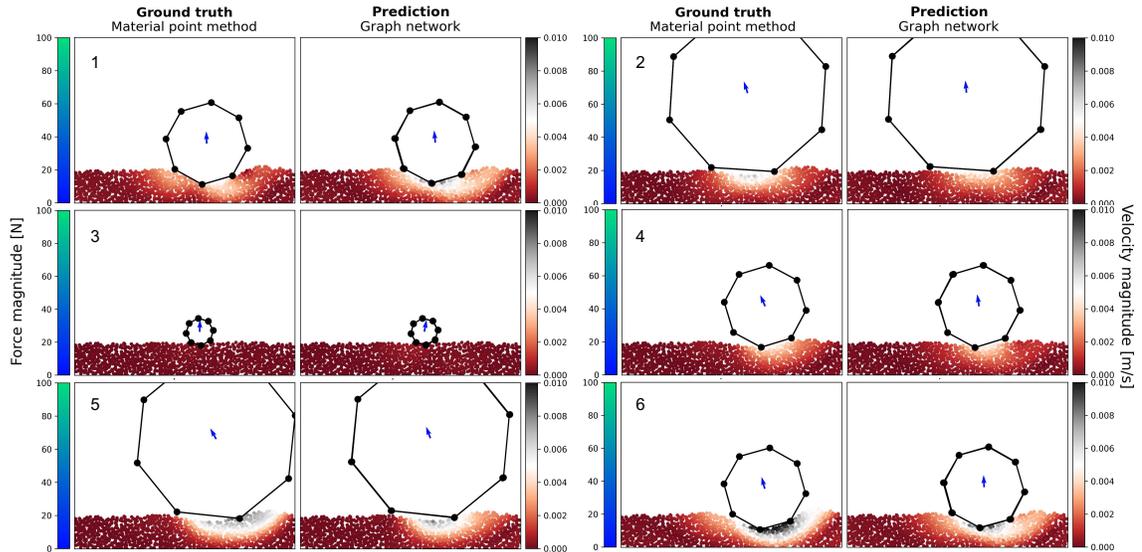


Figure 3.15: Side views of 6 Wheel test examples at 3 second mark. Red to black colors indicate the particle velocity magnitudes. Arrows with blue to green colors represent directions and magnitudes of interaction forces.

3.2. SUBSPACE GRAPH NETWORK SIMULATOR

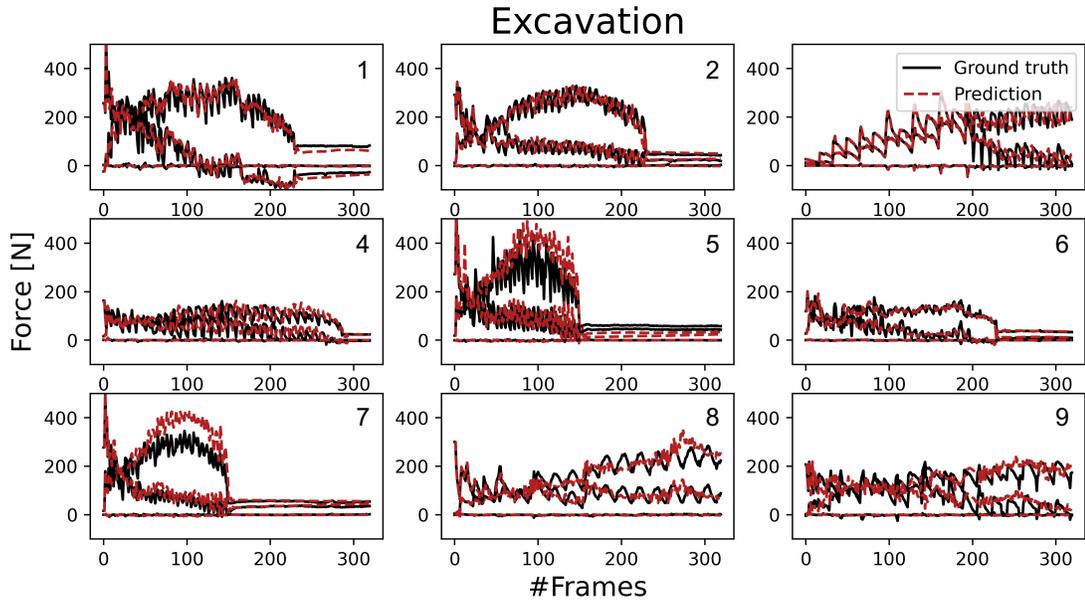


Figure 3.16: Interactions forces in 9 Excavation test examples. Each plot contains forces in forward, vertical and lateral directions.

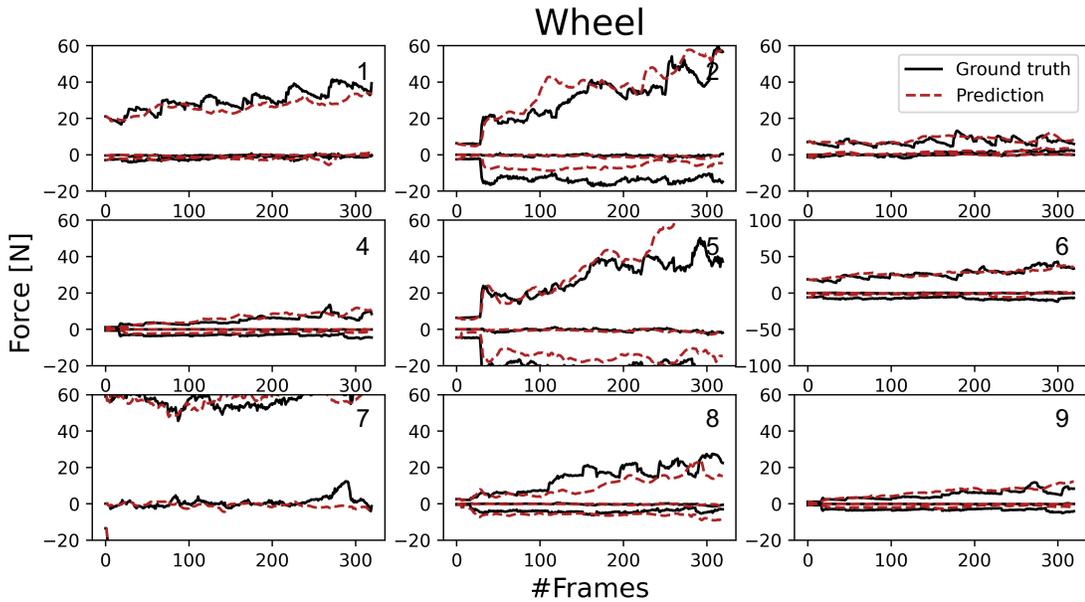


Figure 3.17: Interactions forces in 9 Wheel test examples. Each plot contains forces in forward, vertical and lateral directions.

Chapter 4

Conclusion

This research has addressed existing challenges in robot-terrain interactions by advancing continuum modeling of granular flows as well as real-time physics modeling. It arrives at a novel methodology for predicting complex rigid body-driven granular flows, along with key engineering values such as the interaction forces, in real-time on GPU-enabled computer systems. This work achieves state-of-the-art accuracy in predicting wheel-soil and excavation interactions, and it is the first to the best of our knowledge to successfully apply machine learning accelerations (using graph neural networks in conjunction with a model reduction method) to these classes of problems.

For offline design applications, this research aimed to develop a continuum-based modeling approach that captures interaction forces and the effect of gravity on granular flows, with accuracy and modest computational complexity. To this end, we developed a continuum method comprising a modern constitutive model, nonlocal granular fluidity (NGF), and a state-of-the-art numerical solver, material point method (MPM). Specifically, we designed a numerical approach, within a hyperelasticity framework, to implement the dynamical form of the viscoplastic NGF constitutive model in three-dimensional MPM. This approach is thermodynamically consistent which satisfies the conservation of energy. Also, the dynamical form includes the nonlocal effect of flow cessation. Excavation and Wheel data, both quantitative measurements and qualitative visualization, are collected experimentally via our robotic equipment to evaluate the model with respect to the flow geometry as well as interaction forces. The granular flow deformations including solid-like elastic, fluid-like

viscoplastic, and gas-like stress-free from experiments and simulations were comparable. Moreover, the MPEs (mean percentage errors) of the Excavation and Wheel forces in the simulation compared to the experimental result, did not exceed 1% and 6% magnitude, respectively. In addition to the overall force trends, MPM-NGF was able to capture drops and rises in forces at various timesteps.

Furthermore, this research considered Taylor Couette (TC) flow geometry as a theoretical framework to explore the gravity sensitivity of continuum numerical solvers including finite element method and MPM, as well as an analytical method. It compared these against an experiment done by flying a TC flow apparatus aboard a reduced gravity aircraft presented in literature. It finally identified MPM as an appropriate continuum solver to model granular flows under differing gravity. In fact, the modeling of stress-free particles allowed MPM to accurately generate a z- and r-dependent pressure field in TC flow (vertical and radial directions, respectively). As a result, this realistic pressure field could affect the inertial number field. Unlike finite element method (FEM), the rate-dependent MPM flow made the velocity field z-dependent. This led MPM to be able to model the the TC flow with under 2% MAPE (mean absolute percentage error). These results and observations can benefit the modeling of granular flows in robot-terrain interactions in reduced gravity.

For real-time physics modeling, this work considered the development of a subspace machine learning simulation approach. To generate training datasets, we utilized our high-fidelity (non-real-time) continuum method comprising MPM and NGF verified by experiments. Principal component analysis (PCA) was used to reduce the dimensionality of data. This showed that the first few principal components (< 10 out of thousands of dimensions) can be used to construct the physics data in subspace while keeping more than 99% of the data variance. A graph network simulator (GNS) was trained to learn the underlying subspace dynamics. In subspace GNS, we constructed a complete graph due to the fact that the relationship between physics particles as nodes in subspace might not be proximity-based. The complete graph allowed us to use only one message-passing step, and to omit the use of a time-consuming nearest neighbor algorithm for edge construction. The learned subspace GNS was then able to predict subspace particle accelerations and fullspace interaction forces with good accuracy and roughly steady-state rollout error (position MSE

below 10^{-4}). Force prediction was more accurate in the Excavation dataset due to no variation in the flow and rigid body properties in this dataset. More importantly, PCA significantly enhanced the time and memory efficiency of GNS in both training and rollout. That is, using 8 principal components enabled GNS to be trained using a single desktop GPU with moderate 10GB of VRAM. This also made the GNS real-time on large-scale 3D physics configurations (i.e. 700x faster than our continuum method which were used to produce the datasets).

4.1 Limitations and Future Work

Although significant progress has been made in accurate and real-time simulation of rigid-body driven granular flows, several improvements can still be made for more accurate and efficient approaches.

- **Continuum Modeling of Granular Flows**

1. Implementation of an implicit and coupled MPM approach for solving the governing PDEs (i.e. momentum and nonlocal equations) would lead to the numerical approach being unconditionally stable. This could potentially allow for larger timesteps and consequently faster runtime. However, large timesteps might still cause instability due to the cell-crossing issue in MPM [Dunatunga, 2017].
2. Appending the recently developed transient granular rheology (via extended fabric tensor relations) [Parra and Kamrin, 2019] would enable MPM to model the unsteady pre-critical state behaviors in granular flows.
3. Capturing sub-grid level contacts between rigid body and flow particles in MPM could more accurately result in two-way interactions as done by [Azevedo et al., 2016]. Moreover, an implicit rigid body articulation-MPM coupling would be desirable.
4. Further research on the MPM and nonlocal model parameters, and on the cell-crossing issue in the current MPM (moving least squares) could provide more in-depth insights for future MPM accuracy improvements.

Furthermore, higher-order MPM schemes and a double-precision C++ implementation will increase the accuracy of MPM results [Hu et al., 2018]. This, with all the accuracy-related directions mentioned above, would lead to better GNS predictions for real-time applications.

- **Real-Time Physics Modeling**

1. An autoencoder can be used to reduce the dimensionality of data while capturing non-linearity in data. However, it requires learning procedure and an *a priori* defined size of latent space. These are important limiting factors for high-dimensional data using limited-memory GPU hardware. As an alternative, kernel PCA can be employed to capture non-linearity in data without the above-mentioned restrictions [Schölkopf et al., 1998].
2. Since particle relations might not be proximity-based in subspace, a complete graph is constructed in subspace GNS. However, it may be more memory-efficient to find the potential relations between the selected principal components and construct the effective edges e.g. using the neural relational inference (NRI) model proposed by [Kipf et al., 2018]. It can be even more computationally efficient if one can find an analytical relation between the (sorted) principal components (as the time complexity of GNS edge update is proportional to edge number, $O(N^e)$).
3. The existence of subspace dynamics has recently been discussed in [Champion et al., 2019]. Also, a combination of fullspace GNS and symbolic regression has recently been proposed to extract dynamics equations using pure data [Cranmer et al., 2020]. So such approaches can be used to capture the potential subspace analytical governing equation for further analysis, and for potentially even better generalization than subspace GNS [Cranmer et al., 2020].
4. More research should be done to explore the generalization of subspace GNS. For instance, we might be able to (1) train the subspace GNS using a specific number of principal components, and then (2) perform rollouts with different number of principal components depending on the accuracy required.

4.1. LIMITATIONS AND FUTURE WORK

5. To potentially further reduce the GNS steady-state rollout error, a loss function over a finite next time horizon can be used as suggested by [Holden et al., 2019].
6. Combining GNS with numerical solvers and/or physics-informed loss functions will also be a promising direction for even better generalization in specific applications. Such approaches have already been developed with classical deep learning architectures [Karniadakis et al., 2021, Raissi et al., 2019, Haghighat et al., 2021, Lu et al., 2019].

Bibliography

- [Abadi et al., 2016] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems.
- [Abdullah, 2011] Abdullah, W. S. (2011). Viscoplastic finite element analysis of complex geotechnical problems. *Journal of Civil Engineering*, 5.
- [An et al., 2008] An, S. S., Kim, T., and James, D. L. (2008). Optimizing cubature for efficient integration of subspace deformations. *ACM Trans. Graph.*, 27(5).
- [Aranson and Tsimring, 2002] Aranson, I. and Tsimring, L. (2002). Continuum theory of partially fluidized granular flows. *Phys Rev E*.
- [Aranson and Tsimring, 2001] Aranson, I. S. and Tsimring, L. S. (2001). Continuum description of avalanches in granular media. *Phys. Rev. E*, 64:020301.
- [Askari and Kamrin, 2016] Askari, H. and Kamrin, K. (2016). Intrusion rheology in grains and other flowable materials. *Nature Materials*, 15(12):1274–1279.
- [Azevedo et al., 2016] Azevedo, V. C., Batty, C., and Oliveira, M. M. (2016). Preserving geometry and topology for fluid flows with thin obstacles and narrow gaps. *ACM Trans. Graph.*, 35(4).

BIBLIOGRAPHY

- [Barceló et al., 2020] Barceló, P., Kostylev, E. V., Monet, M., Pérez, J., Reutter, J. L., and Silva, J.-P. (2020). The logical expressiveness of graph neural networks. In *ICLR*.
- [Battaglia et al., 2018] Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G., Vaswani, A., Allen, K., Nash, C., Langston, V., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M., Vinyals, O., Li, Y., and Pascanu, R. (2018). Relational inductive biases, deep learning, and graph networks.
- [Bocquet et al., 2001] Bocquet, L., Losert, W., Schalk, D., Lubensky, T. C., and Gollub, J. P. (2001). Granular shear flow dynamics and forces: Experiment and continuum theory. *Phys. Rev. E*, 65:011307.
- [Bonet et al., 2016] Bonet, J., Gil, A. J., and Wood, R. D. (2016). *Nonlinear Solid Mechanics for Finite Element Analysis: Statics*. Cambridge University Press.
- [Bouaziz et al., 2014] Bouaziz, S., Martin, S., Liu, T., Kavan, L., and Pauly, M. (2014). Projective dynamics: Fusing constraint projections for fast simulation. *ACM Trans. Graph.*, 33(4).
- [Bouzid et al., 2013] Bouzid, M., Trulsson, M., Claudin, P., Clément, E., and Andreotti, B. (2013). Nonlocal rheology of granular flows across yield conditions. *Phys. Rev. Lett.*, 111:238301.
- [Bouzid et al., 2015] Bouzid, M., Trulsson, M., Clément, E., Claudin, P., and Andreotti, B. (2015). Nonlocal rheology in dense granular flows. *Eur Phys J E*, 38:1–15.
- [Bridson, 2007] Bridson, R. (2007). Fast poisson disk sampling in arbitrary dimensions. In *ACM SIGGRAPH*, page 22–es, New York, NY, USA. Association for Computing Machinery.
- [Bridson, 2015] Bridson, R. (2015). Fluid simulation for computer graphics. *CRC Press*.

BIBLIOGRAPHY

- [Bruna et al., 2014] Bruna, J., Zaremba, W., Szlam, A. D., and LeCun, Y. (2014). Spectral networks and locally connected networks on graphs. *CoRR*, abs/1312.6203.
- [Brunton and Kutz, 2019] Brunton, S. L. and Kutz, J. N. (2019). *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press.
- [Champion et al., 2019] Champion, K. P., Lusch, B., Kutz, J., and Brunton, S. (2019). Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences of the United States of America*, 116:22445 – 22451.
- [Chen and Baladi, 1985] Chen, W. F. and Baladi, G. Y. (1985). In *Soil plasticity: Theory and implementation*. Elsevier.
- [Cheng et al., 2019] Cheng, D., Chen, B., and Chen, X. (2019). A robust optimal finite difference scheme for the three-dimensional helmholtz equation. *Mathematical Problems in Engineering*, 2019:13.
- [Chomsky, 1957] Chomsky, N. (1957). *Syntactic structures*. The Hague: Mouton.
- [Cranmer et al., 2020] Cranmer, M., Sanchez-Gonzalez, A., Battaglia, P., Xu, R., Cranmer, K., Spergel, D., and Ho, S. (2020). Discovering symbolic models from deep learning with inductive biases. *ArXiv*, abs/2006.11287.
- [Drucker and Prager, 1952] Drucker, D. C. and Prager, W. (1952). Soil mechanics and plastic analysis or limit design. *Quarterly of Applied Mathematics*, 10(2):157–165.
- [Dunatunga, 2014] Dunatunga, S. (2014). *A nonlocal dense granular flow model implemented in the Material Point Method*. MSc dissertation, Massachusetts Institute of Technology.
- [Dunatunga, 2017] Dunatunga, S. (2017). *A framework for continuum simulation of granular flow*. PhD dissertation, Massachusetts Institute of Technology.

- [Dunatunga and Kamrin, 2015] Dunatunga, S. and Kamrin, K. (2015). Continuum modelling and simulation of granular flows through their many phases. *Journal of Fluid Mechanics*, 779:483–513.
- [Dunatunga and Kamrin, 2017] Dunatunga, S. and Kamrin, K. (2017). Continuum modeling of projectile impact and penetration in dry granular media. *Journal of Mechanics Physics of Solids*, 100:45–60.
- [Elaskar and Godoy, 1998] Elaskar, S. A. and Godoy, L. A. (1998). Constitutive relations for compressible granular materials using non-newtonian fluid mechanics. *International Journal of Mechanical Sciences*, 40(10):1001–1018.
- [Fu et al., 2017] Fu, C., Guo, Q., Gast, T., Jiang, C., and Teran, J. (2017). A polynomial particle-in-cell method. *ACM Trans. Graph.*, 36(6).
- [Fujiwara et al., 2006] Fujiwara, A., Kawaguchi, J., Yeomans, D. K., Abe, M., Mukai, T., Okada, T., Saito, J., Yano, H., Yoshikawa, M., Scheeres, D. J., Barnouin-Jha, O., Cheng, A. F., Demura, H., Gaskell, R. W., Hirata, N., Ikeda, H., Kominato, T., Miyamoto, H., Nakamura, A. M., Nakamura, R., Sasaki, S., and Uesugi, K. (2006). The rubble-pile asteroid itokawa as observed by hayabusa. *Science*, 312(5778):1330–1334.
- [Gilmer et al., 2017] Gilmer, J., Schoenholz, S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural message passing for quantum chemistry. *ArXiv*, abs/1704.01212.
- [Gori et al., 2005] Gori, M., Monfardini, G., and Scarselli, F. (2005). A new model for learning in graph domains. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, 2:729–734 vol. 2.
- [Grover and Leskovec, 2016] Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks.
- [Habraken and Cescotto, 1990] Habraken, A. M. and Cescotto, S. (1990). An automatic remeshing technique for finite element simulation of forming processes. *International Journal for Numerical Methods in Engineering*, 30(8):1503–1525.

- [Haeri and Skonieczny,] Haeri, A. and Skonieczny, K. *Granular Flow Modeling of Robot-Terrain Interactions in Reduced Gravity*, pages 51–61.
- [Haeri and Skonieczny, 2021] Haeri, A. and Skonieczny, K. (2021). Three-dimensional granular flow continuum modeling via material point method with hyperelastic nonlocal granular fluidity. *Submitted to Computational Mechanics*.
- [Haeri et al., 2020] Haeri, A., Tremblay, D., Skonieczny, K., Holz, D., and Teichmann, M. (2020). Efficient numerical methods for accurate modeling of soil cutting operations. In "Osumi, Hisashi", F. H. T. K., editor, *Proceedings of the 37th International Symposium on Automation and Robotics in Construction (ISARC)*, pages 608–615, Kitakyushu, Japan. International Association for Automation and Robotics in Construction (IAARC).
- [Haghighat et al., 2021] Haghighat, E., Raissi, M., Moure, A., Gomez, H., and Juanes, R. (2021). A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 379:113741.
- [Hamilton, 2020] Hamilton, W. L. (2020). Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159.
- [Hamilton et al., 2017a] Hamilton, W. L., Ying, R., and Leskovec, J. (2017a). Inductive representation learning on large graphs. In *NeurIPS Proceedings*.
- [Hamilton et al., 2017b] Hamilton, W. L., Ying, R., and Leskovec, J. (2017b). Representation learning on graphs: Methods and applications. *ArXiv*, abs/1709.05584.
- [Harlow, 1964] Harlow, F. H. (1964). The particle-in-cell computing method for fluid dynamics. *Methods Comput. Phys.*, 3:319–343.
- [Harmon and Zorin, 2013] Harmon, D. and Zorin, D. (2013). Subspace integration with local deformations. *ACM Trans. Graph.*, 32(4).
- [Henann and Kamrin, 2013] Henann, D. L. and Kamrin, K. (2013). A predictive, size-dependent continuum model for dense granular flows. *Proceedings of the National Academy of Sciences*, 110(17):6730–6735.

- [Henann and Kamrin, 2014] Henann, D. L. and Kamrin, K. (2014). Continuum thermomechanics of the nonlocal granular rheology. *International Journal of Plasticity*, 60:145–162.
- [Henann and Kamrin, 2016] Henann, D. L. and Kamrin, K. (2016). A finite element implementation of the nonlocal granular rheology. *International Journal for Numerical Methods in Engineering*, 108(4):273–302.
- [Higgs et al., 2006] Higgs, C. F., Jasti, V., Racusen, C., Heller, C., Cohen, N., and Tichy, J. (2006). *On the Behavior of Granular Materials in Rough Wheel Contacts on Mars*, pages 1–8.
- [Holden et al., 2019] Holden, D., Duong, B. C., Datta, S., and Nowrouzezahrai, D. (2019). Subspace neural physics: Fast data-driven interactive simulation. In *Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA ’19, New York, NY, USA. Association for Computing Machinery.
- [Hornik et al., 1989] Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.
- [Hotelling, 1933] Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:498–520.
- [Hu et al., 2018] Hu, Y., Fang, Y., Ge, Z., Qu, Z., Zhu, Y., Pradhana, A., and Jiang, C. (2018). A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Trans. Graph.*, 37(4).
- [Huang et al., 2011] Huang, J., Tong, Y., Zhou, K., Bao, H., and Desbrun, M. (2011). Interactive shape interpolation through controllable dynamic deformation. *IEEE Transactions on Visualization and Computer Graphics*, 17(7):983–992.
- [Ioffe and Szegedy, 2015] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML’15, page 448–456. JMLR.org.

BIBLIOGRAPHY

- [ISEC, 2013] ISEC (2013). The global exploration roadmap. *International Space Exploration Coordination Group*.
- [Jiang et al., 2015] Jiang, C., Schroeder, C., Selle, A., Teran, J., and Stomakhin, A. (2015). The affine particle-in-cell method. *ACM Trans. Graph.*, 34(4).
- [Jiang et al., 2016] Jiang, C., Schroeder, C., Teran, J., Stomakhin, A., and Selle, A. (2016). The material point method for simulating continuum materials. In *ACM SIGGRAPH 2016 Courses*, SIGGRAPH '16, New York, NY, USA. Association for Computing Machinery.
- [Jop et al., 2005] Jop, P., Forterre, Y., and Pouliquen, O. (2005). Crucial role of sidewalls in granular surface flows: consequences for the rheology. *Journal of Fluid Mechanics*, 541:167–192.
- [Jop et al., 2006] Jop, P., Forterre, Y., and Pouliquen, O. (2006). A constitutive law for dense granular flows. *Letters*, 441.
- [Jumper et al., 2020] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Tunyasuvunakool, K., Ronneberger, O., Bates, R., Zidek, A., Bridgland, A., Meyer, C., Kohl, S., Potapenko, A., Ballard, A., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Steinegger, M., Pacholska, M., Silver, D., Vinyals, O., Senior, A., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2020). High accuracy protein structure prediction using deep learning. In *Fourteenth Critical Assessment of Techniques for Protein Structure Prediction (Abstract Book)*.
- [Kamrin, 2010] Kamrin, K. (2010). Nonlinear elasto-plastic model for dense granular flow. *International Journal of Plasticity*, 26(2):167–188.
- [Kamrin, 2019] Kamrin, K. (2019). Non-locality in granular flow: Phenomenology and modeling approaches. *Frontiers in Physics*, 7:116.
- [Kamrin, 2020] Kamrin, K. (2020). *Quantitative Rheological Model for Granular Materials: The Importance of Particle Size*, pages 153–176. Springer International Publishing, Cham.

BIBLIOGRAPHY

- [Kamrin and Koval, 2012] Kamrin, K. and Koval, G. (2012). Nonlocal constitutive relation for steady granular flow. *Phys. Rev. Lett.*, 108:178301.
- [Karniadakis et al., 2021] Karniadakis, G., Kevrekidis, I., Lu, L., and et al. (2021). Physics-informed machine learning. *Nat Rev Phys*, 3:422–440.
- [Kashima et al., 2003] Kashima, H., Tsuda, K., and Inokuchi, A. (2003). Marginalized kernels between labeled graphs. In *ICML*.
- [Khoei, 2005] Khoei, A. R. (2005). Error estimation and adaptivity. In Khoei, A. R., editor, *Computational Plasticity in Powder Forming Processes*, pages 175–209. Elsevier, Oxford.
- [Kim et al., 2019] Kim, B., Azevedo, V. C., Thuerey, N., Kim, T., Gross, M., and Solenthaler, B. (2019). Deep fluids: A generative network for parameterized fluid simulations. *Computer Graphics Forum*, 38(2):59–70.
- [Kingma and Ba, 2017] Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.
- [Kipf et al., 2018] Kipf, T., Fetaya, E., Wang, K.-C., Welling, M., and Zemel, R. (2018). Neural relational inference for interacting systems.
- [Kipf and Welling, 2016] Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907.
- [Klár et al., 2016] Klár, G., Gast, T., Pradhana, A., Fu, C., Schroeder, C., Jiang, C., and Teran, J. (2016). Drucker-prager elastoplasticity for sand animation. *ACM Trans. Graph.*, 35(4).
- [Konaklieva, 2014] Konaklieva, M. (2014). Molecular targets of β -lactam-based antimicrobials: Beyond the usual suspects. *Antibiotics*, 3:128 – 142.
- [Kovács et al., 2020] Kovács, L. L., Ghotbi, B., González, F., Niksirat, P., Skonieczny, K., and Kövecses, J. (2020). Effect of gravity in wheel/terrain interaction models. *Journal of Field Robotics*, 37(5):754–767.

- [Kumar et al., 2003] Kumar, J., Lakshmana Rao, C., and Massoudi, M. (2003). Couette flow of granular materials. *International Journal of Non-Linear Mechanics*, 38(1):11–20.
- [Ladicky et al., 2017] Ladicky, L., Jeong, S., Bartolovic, N., Pollefeys, M., and Gross, M. (2017). Physicsforests: Real-time fluid simulation using machine learning. In *ACM SIGGRAPH 2017 Real Time Live!*, SIGGRAPH '17, page 22, New York, NY, USA. Association for Computing Machinery.
- [Ladický et al., 2015] Ladický, L., Jeong, S., Solenthaler, B., Pollefeys, M., and Gross, M. (2015). Data-driven fluid simulations using regression forests. *ACM Trans. Graph.*, 34(6).
- [Lagrée and Lhuillier, 2006] Lagrée, P.-Y. and Lhuillier, D. (2006). The couette flow of dense and fluid-saturated granular media. *European Journal of Mechanics - B/Fluids*, 25(6):960–970.
- [Li et al., 2019] Li, Y., Wu, J., Tedrake, R., Tenenbaum, J. B., and Torralba, A. (2019). Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids.
- [Li et al., 2016] Li, Y., Zemel, R., Brockschmidt, M., and Tarlow, D. (2016). Gated graph sequence neural networks. In *Proceedings of ICLR'16*.
- [Liu et al., 1995] Liu, W. K., Jun, S., and Zhang, Y. F. (1995). Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, 20(8-9):1081–1106.
- [Lu et al., 2019] Lu, L., Jin, P., and Karniadakis, G. (2019). Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *ArXiv*, abs/1910.03193.
- [Lun et al., 1984] Lun, C. K. K., Savage, S. B., Jeffrey, D. J., and Chepuruiy, N. (1984). Kinetic theories for granular flow: inelastic particles in couette flow and slightly inelastic particles in a general flowfield. *Journal of Fluid Mechanics*, 140:223–256.

BIBLIOGRAPHY

- [Macklin et al., 2014] Macklin, M., Müller, M., Chentanez, N., and Kim, T.-Y. (2014). Unified particle physics for real-time applications. *ACM Trans. Graph.*, 33(4).
- [Mast et al., 2015] Mast, C. M., Arduino, P., Mackenzie-Helnwein, P., and Miller, G. R. (2015). Simulating granular column collapse using the material point method. *Acta Geotechnica*.
- [MiDi, 2004] MiDi, G. (2004). On dense granular flows. *Eur. Phys. J. E*, 14:341–365.
- [Mitchell, 1980] Mitchell, T. M. (1980). The need for biases in learning generalizations.
- [Monaghan, 1992] Monaghan, J. J. (1992). Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30(1):543–574.
- [Morton et al., 2018] Morton, J., Jameson, A., Kochenderfer, M. J., and Witherden, F. (2018). Deep dynamical modeling and control of unsteady fluid flows. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- [Mueth et al., 2000] Mueth, D. M., Debregeas, G. F., Karczmar, G. S., Eng, P. J., Nagel, S. R., and Jaeger, H. M. (2000). Signatures of granular microstructure in dense shear flows. *Nature*, 406(6794):385–9.
- [Murdoch et al., 2013a] Murdoch, N., Rozitis, B., Green, S., Lophem, T., Michel, P., and Losert, W. (2013a). Granular shear flow in varying gravitational environments. *Granular Matter*, 15:129–137.
- [Murdoch et al., 2013b] Murdoch, N., Rozitis, B., Green, S. F., Michel, P., de Lophem, T.-L., and Losert, W. (2013b). Simulating regoliths in microgravity. *Monthly Notices of the Royal Astronomical Society*, 433(1):506–514.
- [Murdoch et al., 2013c] Murdoch, N., Rozitis, B., Nordstrom, K., Green, S. F., Michel, P., de Lophem, T.-L., and Losert, W. (2013c). Granular convection in microgravity. *Phys. Rev. Lett.*, 110:018307.

- [Narain et al., 2016] Narain, R., Overby, M., and Brown, G. E. (2016). Admm projective dynamics: Fast simulation of general constitutive models. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '16, page 21–28, Goslar, DEU. Eurographics Association.
- [Nedderman, 1992] Nedderman, R. M. (1992). *Statics and Kinematics of Granular Materials*. Cambridge University Press.
- [Niksirat et al., 2020] Niksirat, P., Daca, A., and Skonieczny, K. (2020). The effects of reduced-gravity on planetary rover mobility. *The International Journal of Robotics Research*, 39(7):797–811.
- [Oravec et al., 2010] Oravec, H., Zeng, X., and Asnani, V. (2010). Design and characterization of grc-1: A soil for lunar terramechanics testing in earth-ambient conditions. *Journal of Terramechanics*, 47(6):361–377.
- [Pan et al., 2015] Pan, Z., Bao, H., and Huang, J. (2015). Subspace dynamic simulation using rotation-strain coordinates. *ACM Trans. Graph.*, 34(6).
- [Papadopoulos, 2017] Papadopoulos, P. (2017). In *Introduction to continuum mechanics*. Department of Mechanical Engineering University of California, Berkeley.
- [Parra and Kamrin, 2019] Parra, E. R. and Kamrin, K. (2019). Capturing transient granular rheology with extended fabric tensor relations. *Granular Matter*, 21.
- [Pearson, 1901] Pearson, K. F. (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.
- [Perozzi et al., 2014] Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 701–710, New York, NY, USA. Association for Computing Machinery.
- [Pouliquen and Forterre, 2009] Pouliquen, O. and Forterre, Y. (2009). A non-local rheology for dense granular flows. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1909):5091–5107.

- [Prantl et al., 2020] Prantl, L., Chentanez, N., Jeschke, S., and Thuerey, N. (2020). Tranquil clouds: Neural networks for learning temporally coherent features in point clouds.
- [Raissi et al., 2019] Raissi, M., Perdikaris, P., and Karniadakis, G. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707.
- [Rozitis et al., 2014] Rozitis, B., MacLennan, E., and Emery, J. P. (2014). Cohesive forces prevent the rotational breakup of rubble-pile asteroid. *Nature*, 512:174–176.
- [Sanchez-Gonzalez et al., 2018] Sanchez-Gonzalez, A., Heess, N., Springenberg, J. T., Merel, J., Riedmiller, M., Hadsell, R., and Battaglia, P. (2018). Graph networks as learnable physics engines for inference and control. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4470–4479. PMLR.
- [Scarselli et al., 2008] Scarselli, F., Gori, M., Tsoi, C., Hagenbuchner, M., and Monfardini, G. (2008). The graph neural network model. *IEEE Transactions on Neural Network*, 20:61–80.
- [Scarselli et al., 2009] Scarselli, F., Gori, M., Tsoi, C., Hagenbuchner, M., and Monfardini, G. (2009). Computational capabilities of graph neural networks. *IEEE Transactions on Neural Network*, 20:81–102.
- [Schoefield and Wroth, 1968] Schoefield, A. and Wroth, P. (1968). Critical state soil mechanics. *Pergamon, Oxford*.
- [Schölkopf et al., 1998] Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Non-linear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319.
- [Setaluri et al., 2014] Setaluri, R., Aanjaneya, M., Bauer, S., and Sifakis, E. (2014). Spgrid: A sparse paged grid structure applied to adaptive smoke simulation. *ACM Trans. Graph.*, 33(6).

BIBLIOGRAPHY

- [Shervashidze and Borgwardt, 2009] Shervashidze, N. and Borgwardt, K. (2009). Fast subtree kernels on graphs. In *NIPS*.
- [Shervashidze et al., 2011] Shervashidze, N., Schweitzer, P., van Leeuwen, E. J., Mehlhorn, K., and Borgwardt, K. M. (2011). Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(77):2539–2561.
- [Shlomi et al., 2021] Shlomi, J., Battaglia, P., and Vlimant, J.-R. (2021). Graph neural networks in particle physics. *Machine Learning: Science and Technology*, 2(2):021001.
- [SideFX, 2020] SideFX (2020). Houdini apprentice 18.0 [computer software]. <https://www.sidefx.com>.
- [Sifakis and Barbic, 2012] Sifakis, E. and Barbic, J. (2012). Fem simulation of 3d deformable solids: A practitioner’s guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*, SIGGRAPH ’12, New York, NY, USA. Association for Computing Machinery.
- [Skonieczny et al., 2014] Skonieczny, K., Moreland, S. J., Asnani, V. M., Creager, C. M., Inotsume, H., and Wettergreen, D. S. (2014). Visualizing and analyzing machine-soil interactions using computer vision. *Journal of Field Robotics*, 31(5):820–836.
- [Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.
- [Stokes et al., 2020] Stokes, J. M., Yang, K., Swanson, K., Jin, W., Cubillos-Ruiz, A., Donghia, N. M., MacNair, C. R., French, S., Carfrae, L. A., Bloom-Ackermann, Z., Tran, V. M., Chiappino-Pepe, A., Badran, A. H., Andrews, I. W., Chory, E. J., Church, G. M., Brown, E. D., Jaakkola, T. S., Barzilay, R., and Collins, J. J. (2020). A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702.e13.
- [Stomakhin et al., 2014] Stomakhin, A., Schroeder, C., Jiang, C., Chai, L., Teran, J., and Selle, A. (2014). Augmented mpm for phase-change and varied materials. *ACM Trans. Graph.*, 33(4).

- [Tardos et al., 2003] Tardos, G. I., McNamara, S., and Talu, I. (2003). Slow and intermediate flow of a frictional bulk powder in the couette geometry. *Powder Technology*, 131(1):23–39.
- [Teng et al., 2015] Teng, Y., Meyer, M., DeRose, T., and Kim, T. (2015). Subspace condensation: Full space adaptivity for subspace deformations. *ACM Trans. Graph.*, 34(4).
- [Thuerey et al., 2020] Thuerey, N., Weißenow, K., Prantl, L., and Hu, X. (2020). Deep learning methods for reynolds-averaged navier–stokes simulations of airfoil flows. *AIAA Journal*, 58(1):25–36.
- [Tompson et al., 2017] Tompson, J., Schlachter, K., Sprechmann, P., and Perlin, K. (2017). Accelerating eulerian fluid simulation with convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML’17, page 3424–3433. JMLR.org.
- [Trädegård et al., 1998] Trädegård, A., Nilsson, F., and Östlund, S. (1998). Fem-remeshing technique applied to crack growth problems. *Computer Methods in Applied Mechanics and Engineering*, 160(1):115–131.
- [Um et al., 2018] Um, K., Hu, X., and Thuerey, N. (2018). Liquid splash modeling with neural networks. *Computer Graphics Forum*, 37(8):171–182.
- [Ummenhofer et al., 2020] Ummenhofer, B., Prantl, L., Thurey, N., and Koltun, V. (2020). Lagrangian fluid simulation with continuous convolutions. In *International Conference on Learning Representations*.
- [Veličković et al., 2018] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks.
- [Vittoz et al., 2019] Vittoz, L., Oger, G., de Lefte, M., and Le Touzé, D. (2019). Comparisons of weakly-compressible and truly incompressible approaches for viscous flow into a high-order cartesian-grid finite volume framework. *Journal of Computational Physics: X*, 1:100015.

- [Volfson et al., 2003] Volfson, D., Tsimring, L. S., and Aranson, I. S. (2003). Order parameter description of stationary partially fluidized shear granular flows. *Phys. Rev. Lett.*, 90:254301.
- [von Radziewsky et al., 2016] von Radziewsky, P., Eisemann, E., Seidel, H.-P., and Hildebrandt, K. (2016). Optimized subspaces for deformation-based modeling and shape interpolation. *Comput. Graph.*, 58(C):128–138.
- [Wang et al., 2020] Wang, R., Kashinath, K., Mustafa, M., Albert, A., and Yu, R. (2020). Towards physics-informed deep learning for turbulent flow prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '20*, page 1457–1466, New York, NY, USA. Association for Computing Machinery.
- [Weiler et al., 2016] Weiler, M., Koschier, D., and Bender, J. (2016). Projective fluids. In *Proceedings of the 9th International Conference on Motion in Games, MIG '16*, page 79–84, New York, NY, USA. Association for Computing Machinery.
- [Weisfeiler and Lehman, 1968] Weisfeiler, B. and Lehman, A. (1968). A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia*, 2(9):12–16.
- [Wiewel et al., 2019] Wiewel, S., Becher, M., and Thuerey, N. (2019). Latent space physics: Towards learning the temporal evolution of fluid flow. *Computer Graphics Forum*, 38(2):71–82.
- [Wu et al., 2019] Wu, F., Zhang, T., de Souza Jr. au2, A. H., Fifty, C., Yu, T., and Weinberger, K. Q. (2019). Simplifying graph convolutional networks.
- [Xie et al., 2018] Xie, Y., Franz, E., Chu, M., and Thuerey, N. (2018). Tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow. *ACM Trans. Graph.*, 37(4).
- [Xu et al., 2019] Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2019). How powerful are graph neural networks? *ArXiv*, abs/1810.00826.

- [Ying et al., 2018] Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*.
- [You et al., 2020] You, J., Leskovec, J., He, K., and Xie, S. (2020). Graph structure of neural networks. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10881–10891. PMLR.
- [You et al., 2019] You, J., Ying, R., and Leskovec, J. (2019). Position-aware graph neural networks.
- [Zhang et al., 2014] Zhang, X., Chen, Z., and Liu, Y. (2014). *The material point method*. Elsevier.
- [Zhu et al., 2005] Zhu, H., Wu, Y., and Yu, A. (2005). Discrete and continuum modelling of granular flow. *China Particuology*, 3(6):354–363.
- [Zitnik et al., 2018] Zitnik, M., Agrawal, M., and Leskovec, J. (2018). Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466.
- [Özbay et al., 2019] Özbay, A. G., Laizet, S., Tzirakis, P., Rizos, G., and Schuller, B. (2019). Poisson cnn: Convolutional neural networks for the solution of the poisson equation with varying meshes and dirichlet boundary conditions.

List of Notations

γ	Shear strain
$\dot{\gamma}^p$	Equivalent plastic shear strain rate
κ	Localized weighting function
λ	Eigenvalue
μ	Friction coefficient
μ_2	Maximum friction coefficient (saturation value)
μ_s	Minimum (static) friction coefficient
ϕ	Volume fraction
Φ	Shape function
ρ	Density
ρ_s	Grain density
τ	Shear stress
θ	GNS model parameters
a	Node/subspace particle acceleration
A	Nonlocal amplitude
B	Material bulk modulus (Chapter 2) Mean-subtracted data (Chapter 3)
c	Number of examples in each dataset
C	APIC mode coefficient (Chapter 2) Covariance (Chapter 3)

List of Notations

- d Grain diameter (Chapter 2)
Number of history timesteps (Chapter 3)
- D Stretch part of velocity spatial gradient
- e Graph edge features
- E Hencky elastic strain (Chapter 2)
Set of graph edge features or embeddings (Chapter 3)
- f Particle internal force
- F Deformation gradient (Chapter 2)
Interaction force applied to COM of rigid body (Chapter 3)
- g Granular fluidity
- G Gravitational acceleration (Chapter 2)
Graph (Chapter 3)
- h_e Graph edge embeddings
- h_u Graph global embeddings
- h_v Graph node embeddings
- I Inertial number
- L Velocity spatial gradient (Chapter 2)
Loss function (Chapter 3)
- m Particle mass (Chapter 2)
Number of observations in data (Chapter 3)
- M Mandel stress
- n Number of dimensions in data

List of Notations

N	Quadratic B-splines (Chapter 2) Graph node neighborhood (Chapter 3)
N^e	Number of graph edges
N^k	Number of message-passing steps
N^v	Number of graph nodes
o	GNS Decoder output
p	Mean normal stress
P	Polynomial basis
q	Test function
r_i	Index of graph receiver node of edge i
r	Number of subspace modes
R	Rotation part of deformation gradient (Chapter 2) Rigid states (Chapter 3)
s_i	Index of graph sender node of edge i
S	Material shear modulus (Chapter 2) Flow states (Chapter 3)
\bar{S}	Mean of flow states
S'	Reduced flow states
T	Cauchy stress
u	Graph global feature
U	Stretch part of deformation gradient (Chapter 2) PCA transformation (Chapter 3)
v	Velocity (Chapter 2) Graph node features (Chapter 3)

List of Notations

- V Particle volume (Chapter 2)
Set of graph node features or embeddings (Chapter 3)
- x Position in deformed state, material coordinate
- X Position in undeformed state, spatial coordinate (Chapter 2)
MPM Data (Chapter 3)
- z Node/subspace particle position
- Z Reduced flow and rigid states