Data-driven model order reduction for granular media

Erik Wallin[1] and Martin Servin[1]

[1]Umeå University

May 14, 2020

Abstract

We investigate the use of reduced-order modelling to run discrete element simulations at higher speeds. Taking a data-driven approach, we run many offline simulations in advance and train a model to predict the velocity field from the mass distribution and system control signals. Rapid model inference of particle velocities replaces the intense process of computing contact forces and velocity updates. In coupled DEM and multibody system simulation the predictor model can be trained to output the interfacial reaction forces as well. An adaptive model order reduction technique is investigated, decomposing the media in domains of solid, liquid, and gaseous state. The model reduction is applied to solid and liquid domains where the particle motion is strongly correlated with the mean flow, while resolved DEM is used for gaseous domains. Using a ridge regression predictor, the performance is tested on simulations of a pile discharge and bulldozing. The measured precision is about 90% and 65%, respectively, and the speed-up range between 10 and 60.

1 Introduction

Computational modelling of granular dynamics has important applications in both science and engineering, but is challenging due to the complex nature of granular medias. The discrete (or distinct) element method (DEM) is perhaps the most versatile numerical method for it. It supports the three granular phases, solid, liquid, and gas. It can capture both discrete and collective phenomena, that depend on contact parameters, particle shape and arrangements. DEM simulations are, however, computationally intense, which limits the practical applicability.

There are basically three methods of accelerating a DEM simulation. Firstly, the computational speed can be increased by parallelization and use of specialized hardware [13, 20, 6, 21, 7, 15]. Secondly, changing from explicit to implicit time-integration allows for much larger time-steps than the limit set by the time-period of free vibrations for particles of given mass and contact stiffness. The computational bottleneck is then shifted from collision detection to solving the equations of motion and contact force computation. Depending on the system properties and error tolerance this may be very advantageous [19].

The third way is to employ some form of model order reduction, where the original system is substituted with an approximation that require fewer variables and computational operations per simulated unit of time. Normally, model order reduction is seen as a projection from a high-dimensional space to a low-dimensional subspace, where the time-integration can be performed with manageable computational intensity. Once advanced in time, the solution can be projected back to the original high-dimensional space. The process introduces a model reduction error, that may or may not be acceptable for the intended purpose of the simulation.

In the present paper we explore the possibility of accelerating DEM simulation using data-driven model reduction. The idea is to perform numerous detailed simulations of a system in advance, train a model to predict new system states and use these to advance a running simulation faster in time than the original simulations. The question is whether and how this is at all feasible, and what speed-up and precision can be achieved.

The data-driven approach has the drawback that a certain amount of resolved simulations must be performed in advance to generate training data and build a model. The question then arises what are the advantages of the method. Simulations that must run at real-time speed is one type of application that may benefit from using this technique. Specific examples are simulators for operator training, system testing with hardware-in-the-loop or embedded simulations serving a model-based controller. Another class of applications is surrogate models [5] for simulation-based planning and optimization in parameter spaces too large to be covered with full-resolution simulation but manageable with a reduced-order model trained on a comparatively small set of resolved simulations.

To increase the knowledge about the challenges and opportunities of accelerating DEM simulations using data-driven model order reduction, we have developed and tested a realisation of this idea. First, a general method is described, and error measures are introduced. Next, this is implemented, using a ridge regression model for predicting the velocity field. The computationally expensive process of computing contact forces is substituted by rapid inference of the model to output the velocity field at the particle positions. This approach can be expected to perform well in the solid and liquid regime but poorly in the gaseous regime, where individual particle motion is not strongly correlated with the mean flow. Therefore, we investigate an adaptive model order reduction technique, where the system is decomposed in solid/liquid, and gaseous parts. Gaseous sub-domains are integrated using full resolution DEM while the reduced-order model is applied to the solid/liquid sub-domains. The method is tested on two different systems, a pile with controlled feed and gravity-driven discharge flow, and a blade cutting and pushing through a particle bed like a bulldozer blade. In both cases a model is trained to predict the velocity field from the given input signals and the current mass distribution. In the bulldozer case, the force on the blade is also predicted. The precision and computational speed-up are analysed on these systems.

We do not explore the many alternatives that machine learning offer for the velocity predictor, e.g. deep neural networks, that can be trained to predict complex fluid flows [9]. The present work is a first step. As such it is natural to investigate the performance of a plain regression model and building knowledge for employing more advanced machine learning algorithms.

1.1 Previous work

In the DEM literature there are only a few examples of model order reduction. Boukouvala et al. [3] explored discrete element reduced-order modelling for particle mixing in a blade blender. By principal component analysis (PCA) of simulation snapshots, sampled in a regular grid covering the mixer interior, models were built for predicting the particle velocity field and blade force as function of the mixer control parameters (blade speed and geometry). In turn this was used to develop a surrogate model for optimization of mixing performance based on a relatively small set of time-consuming DEM simulations. This work was later extended by Rogers et al. [16], considering also the effect of dynamic response from changes in the control parameters. The reduced-order model was not used to accelerate the DEM simulations themselves.

In [18], Servin and Wang developed an adaptive model order reduction technique which substitute particles that collectively move as a single rigid body, with six degrees of freedom rigid aggregates of the corresponding mass, momentum, and contact shape. Different strategies for predicting when the aggregate should split in smaller constituents was investigated. The method is severely limited by that the reduced model support only rigid body modes of motion.

Recently, Zhong and Sun [24] investigated a reduced-order model for granular materials under small viscoelastic deformations, with fix connectivity between the particles, using proper orthogonal decomposition (POD) of the displacement field. Inspiration for that work was found in [23] and [10].

Pseudo-particle modelling [4] may be considered a form of model order reduction. Each pseudo-particle represents the collective effect of many small real particles. The pseudo-particle shape and contact parameters are considered model parameters that are calibrated to give the material the approximate bulk-mechanical properties of the original system.

We have not found any example in the DEM literature where the state of a fine-resolution particle model is projected to a pseudo-particle subspace and projected back after time-integration. Ideas of this type may be found in the literature of computer graphics [8] for the purpose of visual appearance and with no analysis of the reduction error.

2 Model order reduction

Let us first summarize the classical meaning of model order reduction [2]. Consider a dynamical system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{1}$$

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) \tag{2}$$

with vectors of state $\boldsymbol{x}(t) \in \mathbb{R}^n$, input $\boldsymbol{u}(t) \in \mathbb{R}^m$ and output $\boldsymbol{y}(t) \in \mathbb{R}^q$. The problem of model order reduction can be stated as follows. Find a lower order model

$$\dot{\hat{\boldsymbol{x}}} = \hat{\boldsymbol{f}}(\hat{\boldsymbol{x}}, \boldsymbol{u}) \tag{3}$$

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{g}}(\hat{\boldsymbol{x}}, \boldsymbol{u}) \tag{4}$$

that produce approximately the same observations

$$\|\boldsymbol{y} - \hat{\boldsymbol{y}}\| \leq \varepsilon_{\mathrm{r}} \|\boldsymbol{u}\| \tag{5}$$

to an accuracy ε_{r} for all input signals $\boldsymbol{u} \in \mathcal{U}$ relevant to the particular application. The idea is that the subspace where the reduced state vector lives, $\hat{\boldsymbol{x}}(t) \in \mathbb{R}^r$, is of much lower dimension than the original system space, i.e., $r \ll n$. Note that the observation vector, $\hat{\boldsymbol{y}} \in \mathbb{R}^q$, must have the same dimensions as in the original system or there must exist a projection operator to that space.

The standard methods for computing approximate low-order models are the SVD-based and Krylov-based approximation methods [2]. The proper orthogonal decomposition (POD) method is a special case of SVD-based model order reduction that is particularly popular in computational mechanics and fluid dynamics. However, granular media modelled using DEM differ from many other dynamical and physical systems in that the connectivity of the variables changes frequently and unpredictably. Therefore, a non-standard model order reduction approach is necessary.

3 A reduced-order discrete element method

In this section we describe a reduced-order model for granular media simulation using the discrete element method.

3.1 Resolved DEM

We first briefly describe the standard discrete element method. We will refer to this as *resolved DEM.*

Each of the N_{p} particles, indexed $a \in \mathcal{N}$, has a position $\boldsymbol{x}^a(t) \in \mathbb{R}^3$, velocity $\boldsymbol{v}^a(t) = \dot{\boldsymbol{x}}^a$, scalar mass m^a and diameter d^a. For clarity of the exposition, we ignore the rotational degrees of freedom. The equations of motion are

$$\dot{\boldsymbol{x}} = \boldsymbol{v} \tag{6}$$

$$\boldsymbol{M}\dot{\boldsymbol{v}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{v}, t) \tag{7}$$

with system position vector $\boldsymbol{x} \in \mathbb{R}^{3N_{\mathrm{p}}}$, velocity vector $\boldsymbol{v} \in \mathbb{R}^{3N_{\mathrm{p}}}$, diagonal mass matrix $\boldsymbol{M} \in \mathbb{R}^{3N_{\mathrm{p}} \times 3N_{\mathrm{p}}}$. The force $\boldsymbol{f}$ is the sum of external forces and contact forces. Each of the N_{c} particle-particle contacts, indexed by $n \in \mathcal{N}_{\mathrm{c}}$, have a contact position $\boldsymbol{x}_{\mathrm{c},n}$ and pairwise contact force $\boldsymbol{f}_n^{ab} \in \mathbb{R}^3$ on particle a from particle b. Each contact force has one normal and two tangential (friction) components. The computationally intense part is the numerical integration of the velocity which involve contact detection and contact force computation. Thus, we ascribe the system a *computational dimensionality* of $N_{\mathrm{d}} = 3N_{\mathrm{p}} + 3N_{\mathrm{c}}$.

Numerical integration is normally performed using an explicit method with small time-steps, that resolve the natural oscillation frequency given the particle mass and stiffness. The computational bottleneck is the collision detection and force calculation from the contact overlap and relative velocity. For some systems it is more beneficial to use the time-implicit method referred to as nonsmooth contact dynamics, or nonsmooth DEM [14, 19]. This allows for large time-step integration, moving the computational bottleneck to the process of solving the nonlinear equations (variational inequalities) from the Signorini-Coulomb and Newton contact laws.

3.2 Reduced DEM

Let the particles be divided in two subsystems, A and B, such that $\boldsymbol{x} = [\boldsymbol{x}_A, \boldsymbol{x}_B]$ and $\boldsymbol{v} = [\boldsymbol{v}_A, \boldsymbol{v}_B]$. The force is divided as $\boldsymbol{f} = [\boldsymbol{f}_A + \boldsymbol{f}_{AB}, \boldsymbol{f}_B + \boldsymbol{f}_{BA}]$, where $\boldsymbol{f}_{AB}$ is the interfacial forces on A from B, and $\boldsymbol{f}_A$ and $\boldsymbol{f}_B$ denote forces acting only on particles in A and B, respectively. The computational dimensionality of the system can be decomposed as $N_\mathrm{d} = 3(N_\mathrm{p}^A + N_\mathrm{p}^B) + 3(N_\mathrm{c}^A + N_\mathrm{c}^B + N_\mathrm{c}^{AB})$. Assume that the particles in subsystem B move according to a known velocity field $\boldsymbol{u}(\mathbf{x}, t)$. Each particle $b \in \mathcal{N}_B$ thus has a known velocity $\boldsymbol{v}_b = \boldsymbol{u}(\mathbf{x}, t)_{\mathbf{x}=\boldsymbol{x}_b}$ at coordinate $\mathbf{x}$ and time t. We abbreviate this as $\dot{\boldsymbol{x}}_B = \boldsymbol{u}(\boldsymbol{x}_B, t)$. Consequently, the particle acceleration is $\dot{\boldsymbol{v}}_b = \partial_t \boldsymbol{u}(\boldsymbol{x}_b, t) + \boldsymbol{v}_b \cdot \nabla \boldsymbol{u}(\mathbf{x}, t)_{\mathbf{x}=\boldsymbol{x}_b}$. This eliminates the equation for $\dot{\boldsymbol{v}}_B$ and leave us with the following, reduced, set of equations of motion

$$\begin{bmatrix} \dot{\boldsymbol{x}}_A \\ \dot{\boldsymbol{x}}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_A \\ \boldsymbol{u}(\boldsymbol{x}_B, t) \end{bmatrix} \tag{8}$$

$$\boldsymbol{M}_A \dot{\boldsymbol{v}}_A = \boldsymbol{f}_A + \boldsymbol{f}_{AB}. \tag{9}$$

If the velocity field $\boldsymbol{u}(\mathbf{x}, t)$ can be computed with negligible effort, the computational intensity of the system is that of integrating Eq. (9). The dimensionality of the reduced model is then $N'_\mathrm{d} = 3N_\mathrm{p}^A + 3(N_\mathrm{c}^A + N_\mathrm{c}^{AB})$. Assuming each particle is in contact with a handful of other particles the *reduction factor* become $R \equiv N'_\mathrm{d}/N_\mathrm{d} \gtrsim N_\mathrm{p}^A/N_\mathrm{p}$.

3.3 Model reduction errors

There can be two sources of errors in the described reduced-order model. Firstly, the model velocity field $\boldsymbol{u}'(\mathbf{x}, t)$ may deviate from the true mean velocity $\boldsymbol{u}(\mathbf{x}, t)$. Secondly, the particle velocities may deviate from the mean velocity field. One quantity that captures the level of fluctuations is the so-called *granular temperature*, $T(\mathbf{x}, t) = \left\langle [\boldsymbol{v}_a - \boldsymbol{u}(\mathbf{x}, t)]^2 \right\rangle$, where $\langle \ldots \rangle$ denote averaging over particles in a small volume, centred at $\boldsymbol{x}$. For the mean squared deviation of the particle velocities from the model velocity field we observe

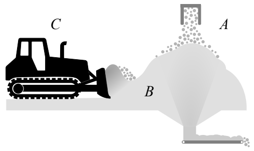

Figure 1: Illustration of a granular system divided in a high-resolution part (A), reduced-order part (B) and coupling with a multibody system (C).

$$\left\langle \|\boldsymbol{v}_a - \boldsymbol{u}'\|^2 \right\rangle \leq \left\langle \|\boldsymbol{v}_a - \boldsymbol{u}\|^2 \right\rangle + \left\langle \|\boldsymbol{u} - \boldsymbol{u}'\|^2 \right\rangle. \tag{10}$$

We therefore introduce the *granular temperature error*

$$\mathcal{E}_T(t) \equiv \left[\frac{1}{V_B} \int_{V_B} w(\boldsymbol{x}) \frac{\left\langle \|\boldsymbol{v}_a - \boldsymbol{u}\|^2 \right\rangle}{v_0^2} \mathrm{d}^3\boldsymbol{x} \right]^{1/2} \tag{11}$$

and the *velocity error* associated with the model reduction

$$\mathcal{E}_v(t) \equiv \left[\frac{1}{V_B} \int_{V_B} w(\boldsymbol{x}) \frac{\left\langle \|\boldsymbol{u} - \boldsymbol{u}'\|^2 \right\rangle}{v_0^2} \mathrm{d}^3\boldsymbol{x} \right]^{1/2} \tag{12}$$

where the integrals are over the volume enclosing the reduced subsystem B and v_0 is a characteristic velocity for the system that the model should be able to resolve. The errors can be computed with no weight, $w(\boldsymbol{x}) = 1$, or weighted by the local mass density, $w(\boldsymbol{x}) = \rho(\boldsymbol{x})/\rho_0$, relative to a nominal bulk density ρ_0 to suppress harmless errors in dilute region. If a surface height function $z = h(x, y)$ is tracked during a simulation, it can be interesting to analyse the *surface height error*

$$\mathcal{E}_h(t) \equiv \left[\frac{1}{V} \int_A [h(x, y) - h'(x, y)]^2 \, \mathrm{d}A \right]^{1/2} \tag{13}$$

where $h'(x,y)$ is the surface height function using the reduced model, A is the projected area of the system in the xy plane and V is its volume enclosed by $h(x,y)$ and some reference surface $h_0(x,y)$.

3.4 Extension to multibody systems

Consider the presence also of a rigid multibody system C with position $\boldsymbol{x}_C$, velocity $\boldsymbol{v}_C$ and mass $\boldsymbol{M}_C$. The multibody system has articulation joints and actuators that are represented by a constraint vector $\boldsymbol{g}_C(\boldsymbol{x}_C, \boldsymbol{v}_C, t) = 0$ with Jacobian $\boldsymbol{G}_C = \partial \boldsymbol{g}_C / \partial \boldsymbol{x}_C$. The forces on the multibody system are the constraint force $\boldsymbol{G}_C \boldsymbol{\lambda}_C$, external force $\boldsymbol{f}_C$ and contact forces $\boldsymbol{f}_{CA}$ and $\boldsymbol{f}_{CB}$ from system A and B. The extended system has the following equations of motion

$$\begin{bmatrix} \dot{\boldsymbol{x}}_A \\ \dot{\boldsymbol{x}}_B \\ \dot{\boldsymbol{x}}_C \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_A \\ \boldsymbol{u}(\boldsymbol{x}_B, t) \\ \boldsymbol{v}_C \end{bmatrix} \tag{14}$$

$$\begin{bmatrix} \boldsymbol{M}_A \dot{\boldsymbol{v}}_A \\ \boldsymbol{M}_C \dot{\boldsymbol{v}}_C \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_A + \boldsymbol{f}_{AB} + \boldsymbol{f}_{AC} \\ \boldsymbol{f}_C + \boldsymbol{f}_{CA} + \boldsymbol{f}_{CB} + \boldsymbol{G}_C \boldsymbol{\lambda}_C \end{bmatrix} \tag{15}$$

$$0 = \boldsymbol{g}_C(\boldsymbol{x}_C, \boldsymbol{v}_C, t). \tag{16}$$

The multibody system has some velocity $\boldsymbol{v}_{BC}$ at the interface between system B and C, which is a contributing cause of the velocity field $\boldsymbol{u}(\mathbf{x}, t)$ in B. The contact force $\boldsymbol{f}_{CB}$ on system C from B is either computed from a contact model or an additional output of the reduced-order model.

3.5 Adaptively reduced DEM

The purpose of the model reduction is fast simulation with sufficient precision. Fast simulation is achieved by minimizing the number of dynamic particles, N_{p}^A, that are simulated with high computational intensity in system A. High precision requires that system B does not include too large domains with high granular temperature, where the individual particle motion deviate substantially from the mean flow. The solution is to adaptively control which regions and what particles are simulated with the reduced DEM and resolved DEM, keeping the reduction factor R and the effect of the granular temperature error $\mathcal{E}_T(t)$ at minimum. This is carried out as follows, with reference to the illustration in Fig. 1.

The reduced model velocity field, $\boldsymbol{u}'(\mathbf{x}, t)$, is assumed to be known. Each particle a has a speed $\Delta v_a = \|\boldsymbol{v}_a - \boldsymbol{u}'\|$ relative to the velocity field. Particles are kept dynamic and part of system A as long as their relative speed exceeds a threshold value $\Delta v_a > \varepsilon v_0$ for some error tolerance ε. Particles with relative speed below the threshold value, $\Delta v_a \leq \varepsilon v_0$, are simulated with the reduced model in system B. Now, contacts in granular media are strongly dissipative. Consequently, particles in A that repeatedly collide with particles in B will have a velocity that quickly approach the velocity field and become part of system B.

It is easy to conceive extensions to this basic scheme. Particles in B that receive a large impulse can be made dynamic and part of system A. If the received impulse is large enough, and the particle lack contact support, it will no longer co-move with the velocity field and remain dynamic. Otherwise, it will merge back. If it is possible to predict regions of high granular temperature, the particles there can be kept dynamic. This is useful at outlets, belt conveyor endpoints or when releasing material from a digging tool, where the flow transitions quickly from solid or dense liquid phase into gaseous phase and free fall. Depletion in mass density is, for instance, a good indicator that the granular temperature is about to increase.

4 A velocity field predictor

Reduced-order DEM simulation, as outlined in Sec. 3, rely on a lower-order model for predicting the velocity field in the granular media. In this section we present a regression model for discrete velocity field prediction and the techniques we use for sampling training and test data from resolved DEM simulations.

4.1 Coarse-graining

Coarse-graining is a technique for sampling and averaging particle states to obtain macroscopic fields. The field values at any coordinate $\mathbf{x}$ is a weighted average of a discrete set of particle and contact variables in a neighbourhood controlled by a coarse-graining function $\phi(\mathbf{x}, R)$ with smoothing length R. For sampling velocity fields when running the reduced-order model, we choose a Heaviside coarse-graining function. For particles near the coarse-graining boundary, the mass is weighted by the volumetric overlap approximated using a bounding box. In offline field analysis we use a Gaussian function $\phi(\mathbf{x}, R) = (\sqrt{2\pi}R)^{-3}\exp(-|\mathbf{x}|^2/2R^2)$. Here we apply a cut-off at $|\mathbf{x}| = 3R$ for practical reasons, which cause a truncation error of 0.01. The mass density field is computed as $\rho(\mathbf{x}, t) = \sum_a m^a \phi(\mathbf{x} - \boldsymbol{x}^a(t), R)$. The velocity field is obtained by $\boldsymbol{u}(\mathbf{x}, t) = \boldsymbol{p}(\mathbf{x}, t)/\rho(\mathbf{x}, t)$, where the momentum density field is first computed as $\boldsymbol{p}(\mathbf{x}, t) = \sum_a m^a \boldsymbol{v}^a \phi(\mathbf{x} - \boldsymbol{x}^a(t), R)$.

4.2 Discretization

We use a regular grid with N_{v} cells, voxels, with side lengths $\boldsymbol{L} = (L_x, L_y, L_z)$, where we denote the shortest of these $L_{\min}$. Each voxel, indexed $\boldsymbol{i} = (i, j, k)$, has a centre point coordinate $\mathbf{x}_{\boldsymbol{i}}$. The mass density and velocity fields are represented in discrete form by their values in the voxel centres, $\rho_{\boldsymbol{i}} = \rho(\mathbf{x}_{\boldsymbol{i}})$ and $\boldsymbol{u}_{\boldsymbol{i}} = \boldsymbol{u}(\mathbf{x}_{\boldsymbol{i}})$. The grid size is limited by the particle diameters d according to $L_{\min} > d_{\max}$, where $d_{\max}$ is the largest particle. The smoothing length for coarse-graining is equal (Heaviside) or somewhat larger (Gaussian) than the size of the voxels.

4.3 Sampling

Data is sampled from resolved DEM simulations. The instantaneous velocity field at a time t is stored in a vector $\boldsymbol{U}(t) = [\boldsymbol{u}_{\boldsymbol{i}}(t)] \in \mathbb{R}^{3N_{\mathrm{v}}}$, the mass density field in a vector $\boldsymbol{P}(t) = [\rho_{\boldsymbol{i}}(t)] \in \mathbb{R}^{N_{\mathrm{v}}}$, and the control signal vector $\boldsymbol{J}(t) = [j_i(t)] \in \mathbb{R}^{N_{\mathrm{j}}}$ that drive the system. These system state time instances are referred to as *snapshots*. A data *sample* is a time-average of snapshots between time t_n and $t_{n+N_\tau-1} \leq \tau$, e.g.,

$$\boldsymbol{U}_n = \frac{1}{N_\tau}\sum_{k=0}^{N_\tau-1}\boldsymbol{U}(t_{n+k}). \tag{17}$$

Any reaction force on a selected body or contact surface, b, may also be sampled, and we denote it $\boldsymbol{F}(t) = [\boldsymbol{f}_b(t)] \in \mathbb{R}^{N_{\mathrm{f}}}$, where N_{f} is the total number of sampled force components.

4.4 Regression model

We are searching for a model that predict the discrete velocity field $\boldsymbol{U} \in \mathbb{R}^{3N_{\mathrm{v}}}$, and possibly also the reaction force $\boldsymbol{F} \in \mathbb{R}^{N_{\mathrm{f}}}$, from a given mass density field $\boldsymbol{P} \in \mathbb{R}^{N_{\mathrm{v}}}$ and control signal $\boldsymbol{J} \in \mathbb{R}^{N_{\mathrm{j}}}$. This is approached as a regression problem, $\boldsymbol{y} = \boldsymbol{\phi}(\boldsymbol{x})$, with predictor variable $\boldsymbol{x} = [\boldsymbol{P}, \boldsymbol{J}] \in \mathbb{R}^{N_{\mathrm{v}}+N_{\mathrm{j}}}$ and response variable $\boldsymbol{y} = \boldsymbol{U} \in \mathbb{R}^{3N_{\mathrm{v}}}$ or $\boldsymbol{y} = \boldsymbol{F} \in \mathbb{R}^{N_{\mathrm{f}}}$ for the velocity and force prediction, respectively. The natural start is to first consider a linear regression model

$$\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1\boldsymbol{x} + \boldsymbol{\varepsilon} \tag{18}$$

with model parameters $\boldsymbol{\beta}_0 \in \mathbb{R}^{3N_{\mathrm{v}}}$ and $\boldsymbol{\beta}_1 \in \mathbb{R}^{3N_{\mathrm{v}}\times(N_{\mathrm{v}}+N_j)}$ and error term $\boldsymbol{\varepsilon}$, for the case of the velocity response variable. There is, however, good reason to believe that a purely linear model cannot capture the behaviour and our numerical experiments also confirmed this. The velocity field and reaction force is expected to depend nonlinearly on the mass and control signal. Therefore, we make the following nonlinear model ansatz

$$\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1\mathtt{vec}[\boldsymbol{H}(\boldsymbol{P})\boldsymbol{J}^T] + \boldsymbol{\varepsilon} \tag{19}$$

where $\boldsymbol{H} : \mathbb{R}^{N_{\mathrm{v}}} \to \mathbb{B}^{N_{\mathrm{v}}}$ is the Heaviside function, component-wise returning an occupancy value 0 or 1 depending on whether the mass density in the voxel is nonvanishing. The vectorization operator `vec( )` produce a regression variable $\boldsymbol{x} \in \mathbb{R}^{N_{\mathrm{v}}N_j}$ out of the matrix $\boldsymbol{H}(P)\boldsymbol{J}$ with dimension $N_{\mathrm{v}} \times N_j$. The model parameters is $\boldsymbol{\beta}_1 \in \mathbb{R}^{3N_{\mathrm{v}}\times N_{\mathrm{v}}N_j}$. We make the same ansatz for the force response variable.

It can be expected that this model suffers from *multicollinearity*, i.e., that the predictor variables are correlated according to the measurement data. One way to handle this is to apply *Ridge regression* which adds a penalty term $\lambda\|\boldsymbol{\beta}\|_2^2$ to the regression loss function, where $\boldsymbol{\beta}_0$ and $\boldsymbol{\beta}_1$ have been combined in $\boldsymbol{\beta}$ as is customary. The regression loss function becomes

$$\mathcal{L} = \|\boldsymbol{y} - \boldsymbol{\beta}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 \tag{20}$$

with penalty parameter λ, that is a hyperparameter to be calibrated. Another way to treat the multicollinearity would be to apply *principal component regression.* In this case one performs PCA on the predictor variables and omit the low-order principal components. Ridge regression accomplish the same effect, but without dimensional reduction.

5 Numerical experiments

To test the model order reduction technique, numerical experiments are performed on two systems. We use a nonsmooth discrete element method as described in [19] using the software AGX Dynamics [1]. The reduced-order model is implemented in Python using NumPy and the model training is performed using scikit-learn [12]. The experiment steps, summarized in Fig. 2, are as follows. First, numerous ground-truth simulations are run. System state snapshots are recorded, and coarse-grained data samples are produced. Each data sample with index n holds a discrete representation of the mass density field $\boldsymbol{P}_n$, velocity field $\boldsymbol{U}_n$, forces $\boldsymbol{F}_n$ and control signal $\boldsymbol{J}_n$. The data samples are split in training data and test data by the ratio 80/20. The model parameters, $\boldsymbol{\beta}$, that minimize the loss function, Eq. (20), are computed using `sklearn.linear_model.Ridge`. The regularization that give the best trade-off between training and test error is chosen manually. The trained models are exported and used in DEM simulations for model order reduction. Validation is made by running fully resolved ground-truth simulations and recording validation snapshots. These are compared to snapshots recorded from reduced-order simulations starting from the same initial state and running with identical control signals.

5.1 Pile with a discharge flow

A quasi 2D pile is confined by inclined sidewalls and vertical rear and front walls, as shown in Fig. 3. The inflow of material is controlled by an emitter above the pile, feeding material at variable flow rate. There is a 1.9 m wide outlet where the sidewalls meet. This is also the distance between the front and rear walls. The discharge flow at the outlet is controlled with a control signal $j(t)$. Particles become kinematic at the outlet, moving with a velocity $\boldsymbol{v}_{\text{out}} = [0, 0, -j(t)]$.

The particles are spherical with diameter 0.1 m, 0.16 m, 0.22 m, and 0.3 m, distributed by the mass ratio of 0.3, 0.5, 0.15, and 0.05, respectively, relative to the total mass. The specific mass density is 2500 kg/m^3, elasticity 10^8 Pa, coefficient of restitution 0.0, friction coefficient 0.5 and rolling resistance coefficient 0.2. The sidewalls have inclination 20° and share the same contact parameters as the particles. Frictionless boundary condition is applied on the vertical (front/rear) walls. The simulations are run with 0.017 s time-step and 200 projected Gauss-Seidel solver iterations. Grid dimensions are $24 \times 1.9 \times 15$ m with $40 \times 3 \times 25$ voxels. The predicted velocity field is a mean field, trained on coarse-grained data that involve both spatial and temporal averaging. Consequently, the particles in the reduced domain may be integrated with a different time-step than used in resolved DEM simulation. The Courant-Friedrichs–Lewy condition imply a time-step around 1 s or smaller for the given voxel size and flow rates. We use 0.17 s time-step for integrating the reduced-order model.

Depending on the confinement geometry and material parameters, the granular media in a pile or silo is discharged either through funnel flow or mass flow. In funnel flow, the material divides into stagnant zones with no motion, and flow zones with shear flow stretching from the outlet to the surface of the pile. In mass flow there are no stagnant zones and all particles are in motion during discharge. The pile in Fig. 3 exhibits funnel flow during discharge. For modelling the velocity field, we assume that the bulk flow is quasi-stationary and depend only on the current outflow control signal $j(t)$ and on the mass distribution $\rho(\mathbf{x}, t)$. Since the model converts the mass density into binary occupancy, the model can generally take the surface height function, $h(x, y)$, as input and directly compute the occupancy underneath it. This is useful when running the model coupled to a real system instrumented with range sensors.

It is important that the training data cover the system state space, that is spanned by the vector $[\boldsymbol{v}(\mathbf{x}), \rho(\mathbf{x}), j]$. We generate data samples from 2500

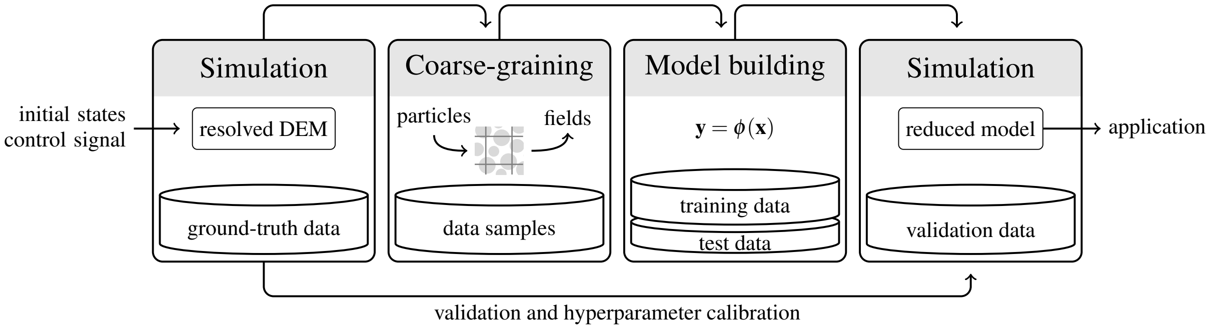

Figure 2: The different steps of developing a reduced-order model.

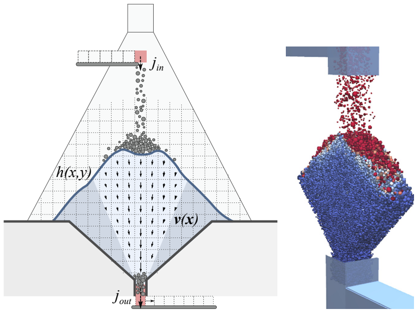

Figure 3: A pile with a discharge flow.

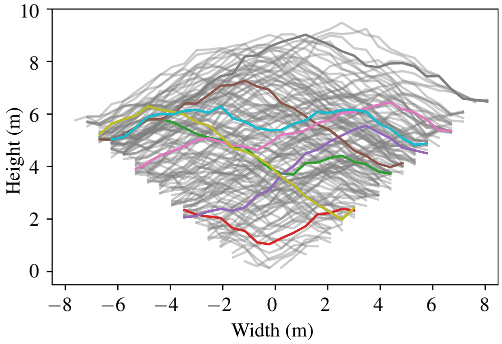

Figure 4: Distribution of surface profiles for the 150 initial states, with some highlighted samples.

full-resolution simulations, with varying outlet velocity, starting from 150 different initial states. An initial state is a certain material distribution, created by different combinations of mass flow at the outlet and inlet. Also, the position of the inlet is varied. The surface profiles for the 150 initial states can be seen in Fig. 4.

During each discharge simulation, the outlet velocity is kept constant at values between 0 and 0.5 m/s and there is no inflow. The duration of each simulation is 30 s. Data samples with 1 s time average are produced from recorded snapshots. To avoid sampling any transient flow after engaging the outlet, the initial part of simulation is discarded. The discharge simulations result in $\lesssim 50,000$ data samples, constructed with latin hypercube sampling uniformly from the 150 initial states and 2500 outlet velocities.

A number of models for predicting the velocity field

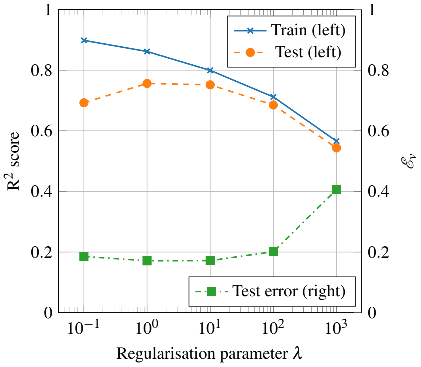

Figure 5: The R^2 score as a function of the regularization parameter λ for the pile training- and test data (left), and the average model reduction velocity error for the test data (right).

are generated with ridge regression parameter in the range 10^{-1} to 10^3. The performance of these models is evaluated in multiple ways. i) The prediction score on the training and test data are compared to see how well the models can generalize to unseen states. ii) The ability to predict velocity fields is examined by comparing coarse-grained resolved DEM fields to corresponding predicted fields, in a complete discharge of a chosen pile state. iii) The same pile discharge is used to compare when each particle pass the outlet (exit time) in reduced-order DEM c.f. resolved DEM. iv) The adaptive reduced-order DEM technique is evaluated by comparison to resolved DEM during more extensive filling and discharging with identical inflow/outflow signals.

The prediction score on the training and test data can be seen in Fig. 5. The best performance on the unseen test data occurs for regularization parameter values between 1 and 10. We chose $\lambda = 10$ as our preferred model and will focus on the performance of this.

Fig. 6 shows snapshots from a ground truth resolved DEM simulation of the discharge of a pile state with constant outflow velocity 0.5 m/s. The columns are snapshots from 10, 20, 30 and 40 seconds into the simulation, with the fields time-averaged over 1 s. In the third and fourth row the velocity field from the ground truth simulation and the model prediction ($\lambda = 10$) are shown together with the mass density. In the fifth row the difference between the ground truth and predicted velocity field is shown, and we observe a good agreement. The lower panel shows the model reduction velocity error, defined in Eq. (12), as a function of time for the duration of the pile discharge. This evaluates the performance of the velocity prediction for all the produced models. We can see that the model reduction velocity error is around 10% for the $\lambda = 10$ model but increases to 20% towards the end when the amount of remaining material is small. In the second row we observe that the granular temperature error is elevated especially near the outlet, indicating an irregular flow there. The time-evolution of the granular temperature error is also included in the lower panel. It varies mostly between 0.5 and 1.

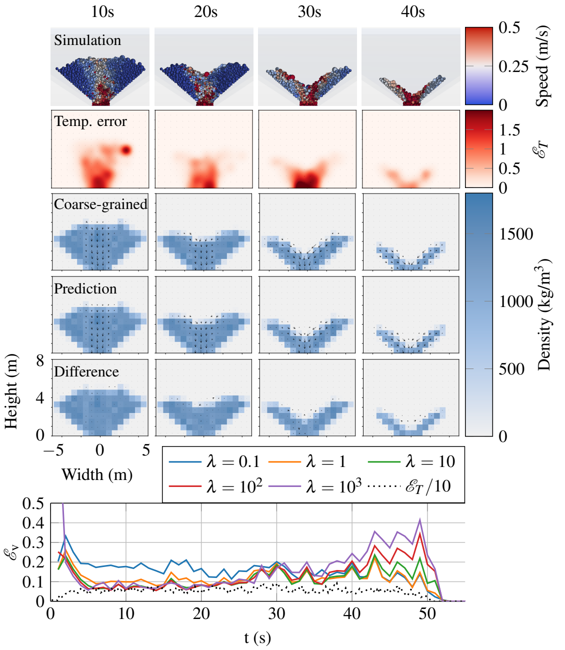

Figure 6: Sample results, comparing ground truth velocity fields from DEM simulations with predicted velocity fields from the reduced-order model.

The ability to predict the velocity field is necessary for reduced-order DEM simulation, but not sufficient. When the system is time-integrated using the reduced-order model the errors may drift or cause instability. Therefore, we examine also the performance of the model when used to propagate the system forwards in time during a pile discharge. Starting from the same state as in Fig. 6 and running with the same control signal, particle positions are integrated using linear interpolation of the velocity field to the particle positions. We compare the particles exit time in the two cases, the ground truth resolved DEM simulation and the reduced-order DEM case. The results can be seen in Fig. 7. The orange line indicates identical exit times for the two cases. Most particles are concentrated on this line, with standard deviation 3.4 s and mean absolute deviation 2.1 s, which is a relative error around 10%. The distributions of number of outflow particles per time unit are also in fair agreement.

Finally, we evaluate the performance of the adap-

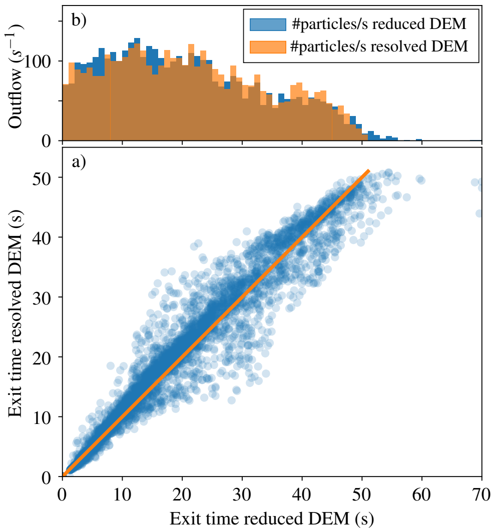

Figure 7: Comparison of residence times from pile discharge using fully resolved DEM simulations and the reduced-order model. The upper panel show the number of discharged particles per second for the two cases.

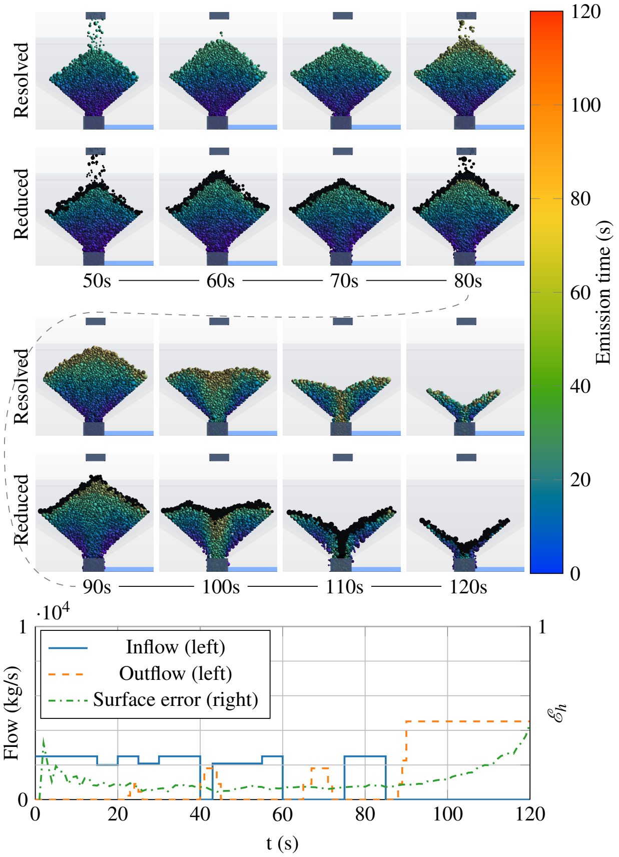

Figure 8: Comparing ground truth simulations and model reduction simulations with identical emitter/outlet signals.

tive reduced-order DEM technique, where reduced DEM is used below the surface of the pile, while resolved DEM is used for particles that are in free fall, impacting and flowing rapidly on the surface. The idea is to run one resolved DEM simulation and one adaptive reduced-order DEM simulation using identical control signals. The tests start from an empty container, building up a pile with variable inlet and outlet signals. The control signals can be seen in the lower panel of Fig. 8, with the outflow velocity converted to an estimated mass outflow per unit time. The outflow velocity is initially kept within the domain of the training data (0 to 0.5 m/s), but is in the end set well outside this domain, at 1 m/s. A comparison of the two cases can be seen in Fig. 8, with snapshots at every 10 seconds after building up the pile. The fully resolved DEM simulations (ground truth) can be seen above the corresponding adaptive reduced-order model case. The particles are colour coded according to time of emission, with the dynamic particles in the adaptive reduced-order model case displayed in black. We track the surface of the pile and let particles down to a depth of 0.5 m be dynamic. One can see that the adaptive reduced-order model performs well, with the surface profiles being similar between the two cases for essentially all times. The surface height error can be seen, as a function of time, in the lower panel of Fig. 8. It is well below 10% during most of the simulations but increase towards the end when there is less remaining material.

It is interesting to study the granular temperature error, defined in Eq. (11), to gain insight in the deviation of particle motion simulated with the full resolution model and the reduced-order model. Two snapshots, from time 53 s and 95 s, are presented in Fig. 9. As expected, the granular temperature error

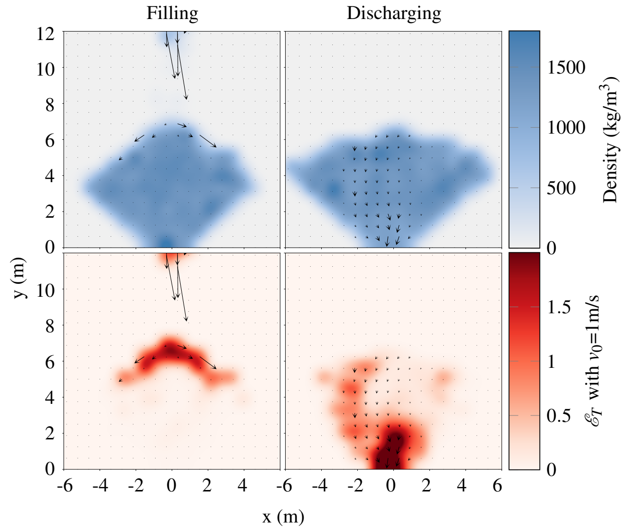

Figure 9: The temperature error (bottom row) from a full resolution simulation of a pile during filling (left column) and discharging (right column). The velocity vector field and mass density field (top row) are included for reference.

is elevated on the surface of the pile when there is an incoming flow, thus motivating the use of resolved DEM there. During discharge, the granular temperature error is elevated around the outlet, not surprising given that the largest particle diameter is 1/6 of the outlet width. There is also signs of correlated velocity fluctuations on a longer length and time scale than that of individual particle rearrangements. This limits the possibility for the reduced models to achieve a low velocity error $\mathcal{E}_v$, even if the model accurately predicts the mean fields. Since this occurs near the outlet, it has marginal effect on the material above.

Sample videos are available at `http://umit.cs.umu.se/ddgranular/`.

5.2 Bulldozing blade

As a second test we simulate a blade driven horizontally, cutting the surface of a granular bed like a bulldozer blade, see Fig. 10. The simulated flow is consistent with the theory of soil mechanics, that predict the formation of a wedge-shaped failure zone in front of the blade [11]. When pushed forward, the soil fails along a localized shear band that stretch from the cutting edge of the blade up to the free surface. Outside the failure zone the material is at rest. Inside the failure zone the material moves forward and upward and may form a circulating flow. The granular temperature is elevated at the front and at the cutting edge where impacts are frequent. A model is trained to predict the velocity field and the reaction force on the blade from the horizontal velocity and the mass distribution in front of the blade. The blade is 1.6 m wide and is attached with a 6-degree-of-freedom constraint to a kinematic body having velocity $\boldsymbol{v} = [j(t), 0, v_z(t)]$. The constraint force holding the blade relative to the kinematic body $\boldsymbol{f} = [f_x, f_y, f_z]$ is measured during the simulation. The shape of the blade is that of two rectangular plates joined along their long edge at an angle of 35°. To avoid sampling of an unnecessarily large domain a coarse-graining grid is co-moving with the blade (velocities are in the world frame and the voxel field is moved to the position of the blade at each iteration). The dimensions of the grid is $1.0 \times 2.5 \times 1.0$ m with $8 \times 5 \times 8$ voxels. Data samples are time-averaged over 1/6 s. In this numerical experiment the ground truth simulations are run with a time-step of 0.005 s and 250 projected Gauss-Seidel solver iterations, and the particle contact parameters are identical to the pile experiment.

During simulations using the adaptive model order reduction, the particles inside the co-moving grid are assigned the velocity predicted by the velocity field using linear interpolation between the voxel centres. Outside the grid, the velocity field is assumed to be zero. Particles exiting the co-moving grid become dynamic, simulated using resolved DEM, until they are at relative rest to the particle bed.

To generate data samples, 250 simulations are run where the blade is pushed with different constant velocities, between 0 and 1.5 m/s, and with different cutting depth, ranging between 0 and 0.2 m. The model is trained using exclusively horizontal motion. As for the pile, a number of models are generated with different ridge regression parameter values, here ranging from 10^{-3} to 10^3. The prediction score on the training and test data can be seen in Fig. 11 and

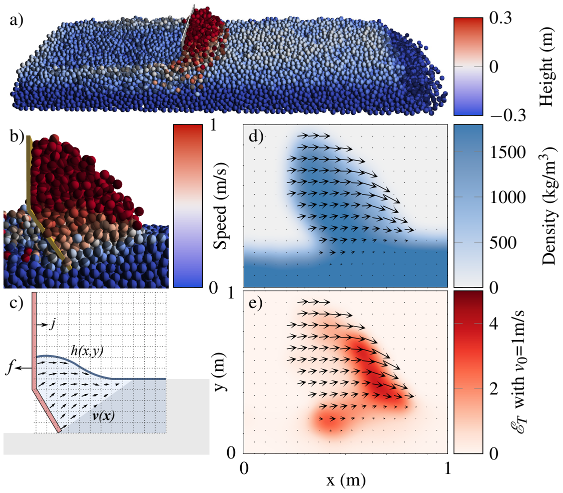

Figure 10: A blade pushing a granular bed simulated with fully resolved DEM. A time instance is shown in 3D overview (a) and cross-section view (b). The predictor model for the velocity field and reaction force from the control signal and mass distribution is illustrated in (c). Also shown is the mass density (d) and granular temperature error (e), with the mean velocity vector field superimposed.

we again pick $\lambda = 10$ as our preferred model, with the best generalization to the test data.

The models are evaluated with regard to i) the ability to predict velocity fields, ii) the ability to predict the force holding the blade, and iii) the ability to propagate the system forwards in time. This is all considered for a standardized blade trajectory, which can be seen in the upper panel of Fig. 13.

The ability to predict the velocity fields is evaluated by comparing coarse-grained ground truth velocity fields to that predicted from the corresponding mass densities, as seen in Fig. 12. The first row show states at 1 s intervals from a resolved DEM simulation with the blade cutting soil at 1 m/s and 0.15 m depth. The second row show the corresponding mass density and velocity field obtained by coarse-graining. The velocity field predicted by the trained model is shown in the third row, and the fourth row show the difference between the simulated and predicted velocity fields. The predicted velocity fields are generally close to the measured ones inside the active zone but can differ considerably on the surface. This is consistent with the observation, in Fig. 10 (e), that the granular temperature is elevated there. There are also large differences at 1s, during the build up phase. The evolution of the model reduction velocity error, $\mathcal{E}_{\text{v}}$, over time and for the different regularization parameters is shown in the bottom of the figure. For the $\lambda = 10$ model the error is at 0.35.

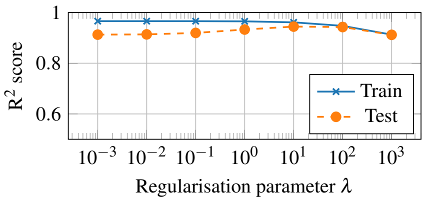

Figure 11: The R^2 score as a function of the regularization parameter λ for the blade training- and test data.

The capability of predicting the force required to push the blade through the particle bed is also examined. A fully resolved DEM simulation is performed using the described trajectory to produce ground truth data of the blade force. From the same simulation, mass density field and control signal are used as input to the model for predicting the blade force. Both forces, time-averaged over 1/6 s, are plotted in Fig. 13. The path of the blade (top figure) is illustrated with 1 s intervals. The predicted force is overall in good agreement with the ground truth but has some problems when the blade is lowered into and raised from the bed, which was not represented in the training data. Also, it was found that training data with the blade moving above the bed, without any mass in sampling region, was necessary for the model to predict the weight of the blade, i.e., the vertical force when the blade is reversed at the end of the bulldozing cycle.

Predicting the velocity field is a necessary but not sufficient functionality. It does not imply that the reduced-order model can propagate the system forward in time with similar precision. Velocity errors in the build up phase may lead to unphysical material distributions, in which case it is of little value that the model is good at predicting stationary flows.

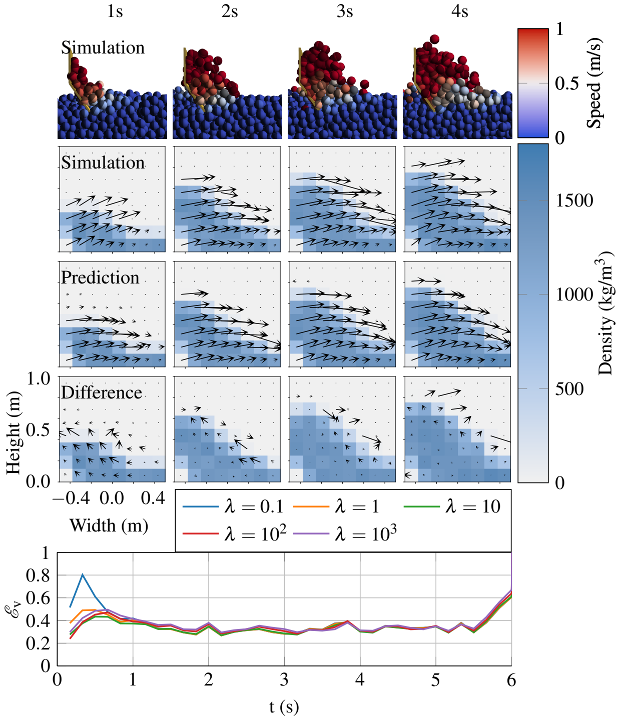

Figure 12: Sample results, comparing ground truth velocity fields with the velocity fields predicted by the reduced-order model.

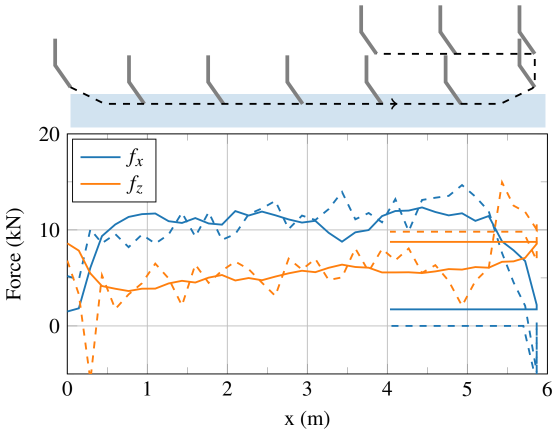

Figure 13: Sample data of the horizontal (red) and vertical (blue) components of the force to push the blade, simulated using resolved DEM (solid lines) and the reduced model (dashed lines).

Also, there are regions of elevated granular temperature, lower right in Fig. 10, at the blade's cutting edge and the front surface of the pile. Since the particle velocities deviate from the mean flow in those regions, the model should not be able to predict them accurately. Therefore, we compare reduced-order DEM and resolved DEM simulations with identical blade trajectories, to evaluate the performance in propagating the system forwards in time. The reduced-order DEM is simulated with the same blade trajectory as previously, depicted in Fig. 13, but with the larger time-step 0.017 s. Fig. 14 shows the two surfaces after a completed bulldozing cycle, and the difference between them. The surfaces has slots and side windrows of similar depth, height and width. In the reduced DEM simulation, the resulting pile is not pushed as far to the end of the slot as in the resolved DEM simulation. This can be understood by that the particles have no inertia in the reduced-order model, and the predictor do not consider the vertical motion when the blade is lifted. Instead, the material simply stays still when the blade is stopped and lifted, and subsequently fall down in a pile when exiting the voxel field.

The reduced-order model captures the force on the blade well, the displacement of material fairly well but the particle velocities only to partial extent. Unfortunately, it is also found that the model is sensitive to the training data. The problem is that the flow during the build up phase is rather different from the stationary flow, and it turns out it is hard to find a balance in representing them both with the model. Depending on which temporal parts of the training data are included, more or less emphasis can be put on these parts. This imbalance typically results in either particles not building up properly in front of the blade, or particles not able to comove with the blade, leaking out through the back. This suggests the need for a different model ansatz than Eq. (19).

Sample videos are available at `http://umit.cs.umu.se/ddgranular/`.

5.3 Performance measurements

In Table 1 we summarize the performance measurements from the numerical experiments. The compu-

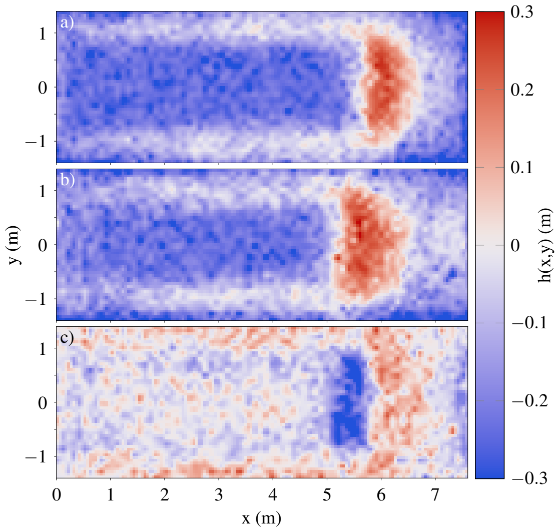

Figure 14: The surface of the material bed after one bulldozing cycle with ground truth simulation (top), granular reduced-order model (middle) and the difference between them (bottom).

tational time for the fully resolved and reduced DEM simulations, t_{res} and t_{red}, are normalized by the real time duration of the experiments, t_{real}. The speed-up factor is the ratio of the resolved computational time to the reduced one. The reduction factor is the ratio of the average number of dynamic particles in the reduced and the resolved DEM simulations. The precision is the inverse of the numerical errors in the tests.

The numerical experiments are performed on a prototype implementation, combining Python and NumPy operations for the reduced-order DEM and AGX Dynamics for resolved DEM and multibody dynamics. We expect there is room for optimization of the computational speed. One opportunity that has not been utilized is that the number of iterations in the projected Gauss-Seidel solver can be decreased with the number of resolved particles (size of the contact network) [19]. In the performed tests the number of solver iterations was kept constant at 200 and 250 in the pile and the blade experiments, respectively. If the reduced DEM simulations instead are run with 20 iterations, appropriate for an error tolerance of 10% on these systems, the speed-up would roughly double from 10 to 20 (Experiment Pile Fig. 8) and from 50 to 100 (Experiment Blade Fig. 12), respectively, and reach the real-time requirement $t_{\mathrm{red}}/t_{\mathrm{real}} < 1$. Disabling the collision detection between particles in the reduced-order model is another optimization that has not been applied here. The measurements of the computational time were made using a single thread on an Intel® Core™ i7-4770 CPU @ 3.40GHz.

6 Discussion

The purpose of the model order reduction is to enable simulation at a higher speed or with increased number of particles while remaining withing given computational bounds. The price for the speed-up is reduced precision and the effort of the simulations that must be carried out in advance to produce training data.

Let us first consider what computational speed-up can be expected. In the extreme case when the whole system can be represented with the reduced model, the main computational steps are: i) do inference on the regression model in Eq. (19) with given known parameters $\boldsymbol{\beta}$; ii) determine the particle velocities by interpolation and update their new position; iii) and, possibly, generate output for the purpose of analysis. The computational complexity of doing inference is that of matrix-vector multiplication of size $\dim(\boldsymbol{\beta}) = 3N_{\mathrm{v}} \times N_{\mathrm{v}} N_{\mathrm{j}}$, which require $6N_{\mathrm{v}}^2 N_{\mathrm{j}}$ floating-point operations. At the present time a powerful desktop CPU delivers about 100 gigaFLOPS and a high-end GPU up to 100 teraFLOPS. It is thus conceivable to evaluate models of size $N_{\mathrm{v}} = 10^3$ (CPU) and $N_{\mathrm{v}} = 10^4$ (GPU) within one millisecond. Hence, it should be possible to simulate fully reduced DEM systems at 60 Hz with up to $N_{\mathrm{p}} = 10^5$ (CPU) and $N_{\mathrm{p}} = 10^6$ (GPU) particles, assuming 10 particles per voxel. Also, the reduced-order DEM is limited by Courant-Friedrichs–Lewy (CFL) condition rather by the time-step used in simulation of the resolved DEM. In the pile and blade experiments the CFL time-step limits are estimated to 1 s and 0.1 s, respectively. That adds another factor 10 to 100 in speed or size of systems that may be simulated in real-time with

Experiment	$t_{\mathrm{res}}/t_{\mathrm{real}}$	$t_{\mathrm{red}}/t_{\mathrm{real}}$	Speed-up	Reduction	Precision
Pile Fig. 6	271/70	13/70	21	0/1700	90%
Pile Fig. 8	1496/120	149/120	10	440/3600	90%
Blade Fig. 12	630/8	11/8	57	250/16000	65%

Table 1: Performance number measured in the numerical experiments using resolved and reduced DEM simulations.

the reduced model.

When the system has regions with granular temperature domains that require resolved DEM, this part of the simulation easily becomes the computational bottleneck. The number of particles that can be simulated in real-time with resolved DEM is up to 10^4, the precise number depending on particle size, velocity, contact parameters and numerical integration technique [19]. With a reduction factor of 1/10, we expect that the pile system can be simulated in real-time with $N_\mathrm{p} = 10^5$ particles. Compared to this theoretical performance estimate, the prototype implementation is underperforming in the Pile test in Fig. 8 by a factor of 10 and can potentially be optimised to reach a speed-up factor of 100.

The precision in the pile experiments is about 10%. This is not a high precision but acceptable for many applications, especially when there are no other alternatives that deliver real-time performance. Also, the uncertainties behind the ground truth (resolved) model are in many practical situations also of this magnitude. There are several ways the reduced model can be improved, besides providing it with more or better training data. The velocity field in the real system has fluctuations and transients that cannot be captured by a model assuming quasi-stationary flow. To achieve higher precision, the missing flow dynamics must be included either at the level of the velocity field predictor or at the particle level, like in the spot-model by Rycroft et al. [17]. The error due to elevated granular temperature in shear flow can possibly be reduced by adding diffusion terms in the calculation of new particle velocities from the mean field. Models for diffusion in granular flows, as function of the local strain rate, can be found in the literature [22] and can be calibrated using the resolved DEM simulation data. It may also be a good idea to adjust the particle velocities or positions to resolve unphysical contact overlaps that result from the errors in the velocity predictor or from applying a diffusion model. In the prototype implementation used for the numerical experiments, the regions of elevated granular temperature are detected manually. Creating a model for predicting the local granular temperature and how it can be expected to develop would enable more automatic and perhaps more precise adaptive reduced-order DEM simulation.

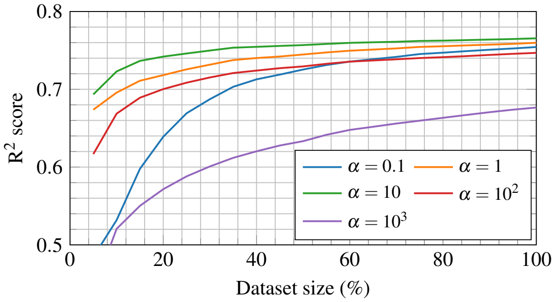

Figure 15: R^2 scores for the pile models on test data, trained on fractions of the training dataset.

Another important question is how much training data is needed to create the reduced-order model. Fig. 15 shows the prediction score on the test data for the pile models, as a function of being trained on increasing fractions of the training set of 50,000 data samples. The most significant increase in the test score occurred when using up to 50% of the training set, with generally a small further increase when using all data. However, the scaling depends on the regularization and none of the cases has really saturated, thus there could still be some improvements by increasing the amount of training data.

In the present study a plain regression model is tested. There is good reason to believe that deep neural networks (DNN) can perform well and train to generalize much better to nonlinear flows and changes in the geometric boundaries. Typically, a DNN has many more parameters and requires larger amounts of training data. That does not necessarily mean many more simulations, but instead using more of the unprocessed data from the full resolution simulations, e.g., the particle motions relative to the mean field, density and the contact force network, to better resolve the fluctuations in the velocity field and reaction forces on objects interacting with the granular media.

7 Conclusion

We introduce a novel technique for model order reduction of DEM simulations of granular media. Using many offline simulations, we train a regression model to predict the velocity field. This model is then used to assign particle velocities, in place of the time-consuming process of collision detection and force computation. An adaptive domain technique is used to apply the reduced-order model in regions with low granular temperature error where individual particle motion coincide well the mean flow, and use resolved DEM simulation in regions with more irregular particle motion. This allows for minimizing the number of particles in the computationally intense process, while still simulating particle motion with realistic mean velocity. The adaptive reduced-order model is applied in two test systems, a gravitational discharge pile and a bulldozer blade cutting and pushing through a particle bed. We measure a computational speed-up of 10 to 20 and 90% precision for the discharge pile. We estimate that it is possible to reach real-time performance with 10^5 particles at 60 Hz and a reduction factor 1/10, corresponding to a speed-up of 100. For the bulldozing blade the speed-up reaches 60 but with a precision of around 65%. Plans for future research includes extending the predictor model beyond plain regression.

Acknowledgements

The project was funded in part by eSSENCE and VINNOVA (grant id 2019-04832) and has been supported by Algoryx Simulation AB. The simulations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC dnr 2019/3-168) at High Performance Computing Center North (HPC2N).

Conflicts of interest

Martin Servin is co-founder of and senior scientist at Algoryx Simulation AB.

References

[1] Algoryx Simulations. AGX dynamics, Sept. 2019.

[2] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems.* Society for Industrial and Applied Mathematics, 2005.

[3] F. Boukouvala, Y. Gao, F. Muzzio, and M. Ierapetritou. Reduced-order discrete element method modeling. *Chemical Engineering Science*, 95:12–26, 2013.

[4] Y. T. Feng and D. R. J. Owen. Discrete element modelling of large scale particle systems—i: exact scaling laws. *Computational Particle Mechanics*, 1(2):159–168, Jun 2014.

[5] A. Forrester, A. Sobester, and A. Keane. *Engineering Design via Surrogate Modeling.* Wiley, 2008.

[6] J. Gan, Z. Zhou, and A. Yu. A gpu-based dem approach for modelling of particulate systems. *Powder Technology*, 301:1172 – 1182, 2016.

[7] Y. He, T. Evans, A. Yu, and R. Yang. A gpu-based dem for modelling large scale powder compaction with wide size distributions. *Powder Technology*, 333:219–228, 2018.

[8] M. Ihmsen, A. Wahl, and M. Teschner. A lagrangian framework for simulating granular material with high detail. *Computers & Graphics*, 37(7):800 – 808, 2013.

[9] J. N. Kutz. Deep learning in fluid dynamics. *Journal of Fluid Mechanics*, 814:1–4, 2017.

[10] C.-H. Lee and J.-S. Chen. Proper orthogonal decomposition-based model order reduction via radial basis functions for molecular dynamics systems. *International Journal for Numerical Methods in Engineering*, 96(10):599–627, 2013.

[11] E. McKyes. *Soil cutting and tillage*. Developments in agricultural engineering. Elsevier, 1985.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[13] T. Preclik and U. Rüde. Ultrascale simulations of non-smooth granular dynamics. *Computational Particle Mechanics*, 2(2):173–196, Jun 2015.

[14] F. Radjai and V. Richefeu. Contact dynamics as a nonsmooth discrete element method. *Mechanics of Materials*, 41(6):715–728, June 2009.

[15] A. D. Rakotonirina and A. Wachs. Grains3d, a flexible dem approach for particles of arbitrary convex shape - part ii: Parallel implementation and scalable performance. *Powder Technology*, 324:18 – 35, 2018.

[16] A. Rogers and M. Ierapetritou. Discrete element reduced-order modeling of dynamicparticulate systems. *AIChE Journal*, 60:3184–94, 2014.

[17] C. H. Rycroft, M. Z. Bazant, G. S. Grest, and J. W. Landry. Dynamics of random packings in granular flow. *Phys. Rev. E*, 73:051306, May 2006.

[18] M. Servin and D. Wang. Adaptive model reduction for nonsmooth discrete element simulation. *Computational Particle Mechanics*, 3(1):107–121, 2016.

[19] M. Servin, D. Wang, C. Lacoursière, and K. Bodin. Examining the smooth and non-smooth discrete element approach to granular matter. *Int. J. Numer. Meth. Engng.*, 97:878–902, 2014.

[20] J. Steuben, G. Mustoe, and C. Turner. Massively Parallel Discrete Element Method Simulations on Graphics Processing Units. *Journal of Computing and Information Science in Engineering*, 16(3), 08 2016. 031001.

[21] Y. Tian, S. Zhang, P. Lin, Q. Yang, G. Yang, and L. Yang. Implementing discrete element method for large-scale simulation of particles on multiple gpus. *Computers & Chemical Engineering*, 104:231 – 240, 2017.

[22] B. Utter and R. P. Behringer. Self-diffusion in dense granular shear flows. *Phys. Rev. E*, 69:031308, Mar 2004.

[23] J. R. Williams and N. Rege. Coherent vortex structures in deforming granular materials. *Mechanics of Cohesive-frictional Materials*, 2(3):223–236, 1997.

[24] X. Zhong and W. Sun. An adaptive reduced-dimensional discrete element model for dynamic responses of granular materials with high-frequency noises. *International Journal for Multiscale Computational Engineering*, 16(4):345–366, 2018.