Mesh Sampling for Finite Element Method

Wanli He

1 Introduction

Finite Element Method (FEM) plays an important role in solving partial differential equations. It is one kind of numerical techniques which formulate the problem into a system of algebraic equations. Solid mechanics and structural dynamics is one such application of FEM, and it has the governing equation in semi-discrete form as shown in Equation 1.

$$M\ddot{u} + f(u) = f^{\text{ext}} \tag{1}$$

In Equation 1, $M \in \mathbb{R}^{n_{\text{dof}} \times n_{\text{dof}}}$ denotes the mass matrix, $f \in \mathbb{R}^{n_{\text{dof}}}$ is the internal force vector, and $f^{\text{ext}} \in \mathbb{R}^{n_{\text{dof}}}$ is the external force vector. Solution $u(t) \in \mathbb{R}^{n_{\text{dof}}}$ is the displacement vector which is a function of time t, and $\ddot{u} \in \mathbb{R}^{n_{\text{dof}}}$ is the acceleration vector. n_{dof} is the number of degrees of freedom (DOFs) in system, which also describes the number of algebraic equations. Equation 1 shows how inertia force balances stress force under the transient system. Depending on the material law, integration time scheme, type of element, etc., Equation 1 has different formulation. For more details of how to formulate the system of algebraic equations, please refer to [1].

The most time consuming part of solving Equation 1 is the operator $f(u)$, especially in nonlinear case. To solve $u = u(t, \Omega)$, where Ω is the continuous spatial domain, FEM instead seeks for an approximate solution at nodal points of sub-domains (elements) $\Omega^e \ \forall e \in \mathcal{V}$, with $\sum_{e \in \mathcal{V}} \Omega^e = \Omega$, where $\mathcal{V}$ is the set of elements in system. This approximation discretizes the spatial domain. We assemble $f^e(u)$ from discrete elements to obtain $f(u)$, and the computation of the internal force is done by Equation 2.

$$f(u) = \sum_{e \in \mathcal{V}} f^e(u) \tag{2}$$

In general, if number of elements $|\mathcal{V}|$ is large, i.e., having fine meshes, the numerical result from Equation 1 is more accurate. However, considering the computational complexity of Equation 2 is $\mathcal{O}(|\mathcal{V}|)$, if $|\mathcal{V}|$ is large, assembling $f(u)$ in Equation 2 is rather computationally expensive. This fact motivates the project. We look for an alternative approach to compute the internal force in Equation 2, which aims to be fast and accurate.

2 Related Work

One existing approach called Hyper-reduced Projection-based Reduced Order Modeling (HPROM) demenstrated in [2][3] solves Equation 1 for complicated problems in real time with high accuracy.

The main idea of HPROM consists of two steps. First, it seeks for the best approximate solution of High Dimensional Model (HDM) in a subspace V so that the solution can be approximated by the reduced coordinate solution via $u(t) = Vy(t)$. Basis V is obtained by running Proper Orthogonal Decomposition (POD) on snapshot matrix X, which is the data collection of snapshot vector u. Basis V contains the compressed information about u, which is very similar to the theory of Principal Component Analysis (PCA). To keep numerical stability, the basis V satisfies $V^T M V = I$. With Galerkin projection, Equation 1 is reduced to Equation 3. The step is called Reduced Ordered Modeling (ROM).

$$\ddot{y} + V^T f(Vy) = V^T f^{\text{ext}} \tag{3}$$

The second step of HPROM is called hyper-reduction[4]. Notice that the reduced internal force term in Equation 3 still scales with the size of HDM, which means that the computational complexity of it is still $\mathcal{O}(|\mathcal{V}|)$. To compute it in real time, it samples a subset of elements by taking the advantage of the sparsity of local internal force vectors.

One improvement of HPROM is to split the global basis V into many local basis as demonstrated in [5]. However, using global basis does excellent job in most cases of FEM due to the nature of structural equations.

3 Method

The HPROM discussed in section 2 highly depends on the training process, i.e., gathering snapshot matrix X. If the data contained in X is not representative, the reduced solution obtained via Equation 3 is not accurate. Thus, we design a modified step 2 by only reducing the assembly cost of internal force vector in Equation 1 without reducing the order of model.

To reduce the assembly cost, we sample a subset of elements $\mathcal{V}'$ form $\mathcal{V}$ and attribute them non-negative weights $\alpha^e > 0 \ \forall e \in \mathcal{V}'$ such that

$$f(u) \approx \sum_{e \in \mathcal{V}'} \alpha^e f^e(u) \tag{4}$$

However, $f^e(u)$ is a sparse vector since it only contains the local information within the element. Thus, the assembly of sampled elements only produces a sparse internal force vector as well if $|\mathcal{V}'|$ is small. This result violates the natural of physics and therefore the single weighting scalar is not a good choice. Alternatively, we can replace the non-negative weight α^e by the weighting matrix W^e to broadcast local information. Finally, the goal of project is to find a subset of elements in system $\mathcal{V}'$ and attribute them weighting matrices $W^e \ \forall e \in \mathcal{V}'$ to build a hypothesis model for internal force vector, as shown in Equation 5.

$$\hat{f}(u) = \sum_{e \in \mathcal{V}'} W^e \hat{f}^e(u) \tag{5}$$

where $W^e \in \mathbb{R}^{n_{\text{dof}} \times n'_{\text{dof}}}$ and $\hat{f}^e(u) \in \mathbb{R}^{n'_{\text{dof}}}$ is the non-zero components (entries) of $f^e(u)$. n'_{dof} is the number of degrees of freedom in the element.

To ensure efficiency, we want $|\mathcal{V}'| \ll |\mathcal{V}|$, i.e., sample very few elements, and to ensure accuracy, we want that the numerical result from Equation 1, with $f(u)$ replaced by hypothesis model $\hat{f}(u)$, is still accurate.

3.1 Data

The gathered data used to train the hypothesis model is split into training set and validation set. Training set contains $\{f^{(i)}, (\hat{f}^{1,(i)}, \hat{f}^{2,(i)}, \dots, \hat{f}^{|\mathcal{V}|,(i)})\}$, for $i = 1, 2, \dots, n$ and validation set contains $\{f^{(j)}, (\hat{f}^{1,(j)}, \hat{f}^{2,(j)}, \dots, \hat{f}^{|\mathcal{V}|,(j)})\}$, for $j = 1, 2, \dots, m$. The data comes from the reference solution by solving Equation 1 and each piece of data, denoted by superscript (i) and (j) corresponds to each time step. Data is shuffled and $n = 40000$ and $m = 10000$.

3.2 Algorithm

The hypothesis model in Equation 5 can be solved by the optimization problem shown in Equation 6

$$\mathcal{V}'^*, W^* = \arg\min_{\mathcal{V}', W} \sum_{i=1}^{n} \text{loss}(f^{(i)}, \sum_{e \in \mathcal{V}'} W^e \hat{f}^{e,(i)}) \tag{6}$$

where $W = \{W^e\} \ \forall e \in \mathcal{V}'$ is the set of weighting matrices and we define loss function as the 2-norm square of the difference, i.e., $\text{loss}(x, y) = ||x - y||_2^2$.

This is not a standard optimization problem but we can solve it by some sub-optimal approaches. Suppose the optimal subset of elements $\mathcal{V}'^*$ is known, to obtain the optimal weighting matrices W^*, we solve the optimization problem with regularization on weighting matrices W, resulting in Equation 7.

$$\begin{aligned} W^* &= \arg\min_{W} J_{\text{train}}(\mathcal{V}'^*) \\ &= \arg\min_{W} \sum_{i=1}^{n} \text{loss}(f^{(i)}, \sum_{e \in \mathcal{V}'^*} W^e \hat{f}^{e,(i)}) + \lambda \sum_{e \in \mathcal{V}'^*} ||W^e||_F^2 \end{aligned} \tag{7}$$

Stochastic Gradient Descent (SGD) is used to solve Equation 7, and the update rule is shown in Equation 8

$$W^e \leftarrow W^e - \alpha[(\sum_{e \in \mathcal{V}'^*} W^e \hat{f}^{e,(i)} - f^{(i)}) \hat{f}^{e,(i)T} + \lambda W^e] \tag{8}$$

The quality of sampled elements is evaluated by Equation 9, which computes the relative global error in validation set of internal force data.

$$J_{\text{val}}(\mathcal{V}'^{*}, W^{*}) = \sqrt{\frac{\sum_{j=1}^{m} \text{loss}(f^{(j)}, \sum_{e \in \mathcal{V}'^{*}} W^{e*} \hat{f}^{e,(j)})}{\sum_{j=1}^{m} ||f^{(j)}||_2^2}} \tag{9}$$

We can then obtain the optimal subset of elements by sampling them iteratively. We stop selection when the number of elements in subset exceeds some threshold or when the quality of sampled elements meets some other threshold. Details are shown in Algorithm 1.

Algorithm 1: Mesh sampling (brute force)

initialize $\mathcal{V}' \leftarrow \varnothing$;
while not converged **do**
  (a) For $i = 1, 2, \ldots, |\mathcal{V}|$ if $i \notin \mathcal{V}'$, let $\mathcal{V}'_i \leftarrow \mathcal{V}' \cup \{i\}$, compute $W_i^* \leftarrow \arg\min J_{\text{train}}$ and evaluate $J_i = J_{\text{val}}(\mathcal{V}'_i, W_i^*)$;
  (b) Set $\mathcal{V}'$ to be the best subset of elements which minimizes J_{val} on step (a), $\mathcal{V}' \leftarrow \arg\min J_i$;
end

However, if $|\mathcal{V}|$ is large, running Algorithm 1 is expensive and we denote it as the brute force approach for mesh sampling. Alternatively, to speed up the iterative process, we can randomly select a subset from the complete set of elements and only sample from the subset. It leads to a greedy approach as shown in Algorithm 2.

Algorithm 2: Mesh sampling (greedy)

initialize $\mathcal{V}' \leftarrow \varnothing$;
while not converged **do**
  (a) Random sample $\mathcal{V}_s \subset \mathcal{V}$;
  (b) For $i \in \mathcal{V}_s$ if $i \notin \mathcal{V}'$, let $\mathcal{V}'_i \leftarrow \mathcal{V}' \cup \{i\}$, compute $W_i^* \leftarrow \arg\min J_{\text{train}}$ and evaluate $J_i = J_{\text{val}}(\mathcal{V}'_i, W_i^*)$;
  (c) Set $\mathcal{V}'$ to be the best subset of elements which minimizes J_{val} on step (b), $\mathcal{V}' \leftarrow \arg\min J_i$;
end

4 Application

We are interested in a dynamic impact problem of a three-dimensional circular bar on a rigid frictionless wall[3], which features an elasto-plastic material and large plastic strains.

Material response of the bar is characterized by a model of J_2 flow theory using a logarithmic free energy function and the von Mises yield criterion, of which the material properties is summarize in table 1. The geometry of the bar is also summarized in table 1. Because of the symmetry, only one quarter of the bar is modeled. 1728 20-noded hexahedral elements are used with displacement-only DOFs. Figure 1 helps visualize the geometry and mesh of the bar. The initial velocity of the bar is 227 m/s. The temporal discretization is performed using the explicit central difference method with variable time-step based on the estimation of the critical stability time-step. The simulation runs until $T = 5 \times 10^{-5}$ s.

Table 1: Circular bar: material and geometric properties

Material properties		Geometry	
Young's modulus E	117 GPa	Length L	32.4 mm
Poisson's ratio ν	0.4	Radius of circular cross-section R	3.2 mm
Yield stress σ_y	0.4 GPa		
Isotropic hardening modulus H	0.1 GPa		
Density ρ	8930 kg/m^3		

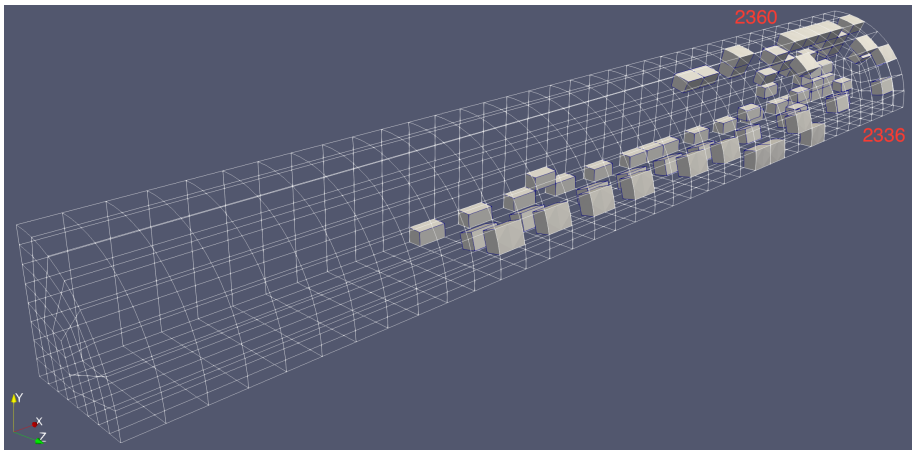


Figure 1: Circular bar: mesh (white lines), sampled elements (white solid elements), and probe locations

We compare two types of models.

Model I: The reference solution $u(t)$ is computed by solving Equation 1 directly, and it stores the internal force data (element-wise and global) for training. The learning rate α is set to be 0.01 and regularization parameter λ is set to be 0.001.

Model II:The mesh sampling solution $\tilde{u}(t)$ is computed by solving Equation 1 but equipped with approximation of internal force with Equation 5.

To truly access the quality of mesh sampling, the global relative error is computed in Equation 10, which quantifies the difference of solution between Model I and model II among all time steps.

$$e = \sqrt{\frac{\sum_t ||u(t) - \tilde{u}(t)||_2^2}{\sum_t ||u(t)||_2^2}} \times 100\% \tag{10}$$

5 Result

For the application described in section 4, the training on Algorithm 2 delivers a subset of 70 sampled elements, i.e. $|\mathcal{V}'| = 70$. The sampled elements can be viewed in Figure 1, which are mostly located at the impacting end. About 4% of elements are sampled. The CPU time and speed-up factors of running Model I and Model II are summarized in Table 2. The speed-up factor indicates that Model II with mesh sampling is able to run $\sim 3\times$ faster than Model I. However, this factor is not as good as the ratio of elements being sampled. The loss of expected efficiency is due to the matrix-vector multiplication when assembling approximated internal force vector. However, Model II does produce very accurate approximated results. Table 2 also summarizes the global relative errors for both displacement and velocity fields in all three directions. The error of displacement is less than 1%, and the error of velocity is one magnitude larger than displacement, which however is expected for FEM application, even for other approximation approaches.

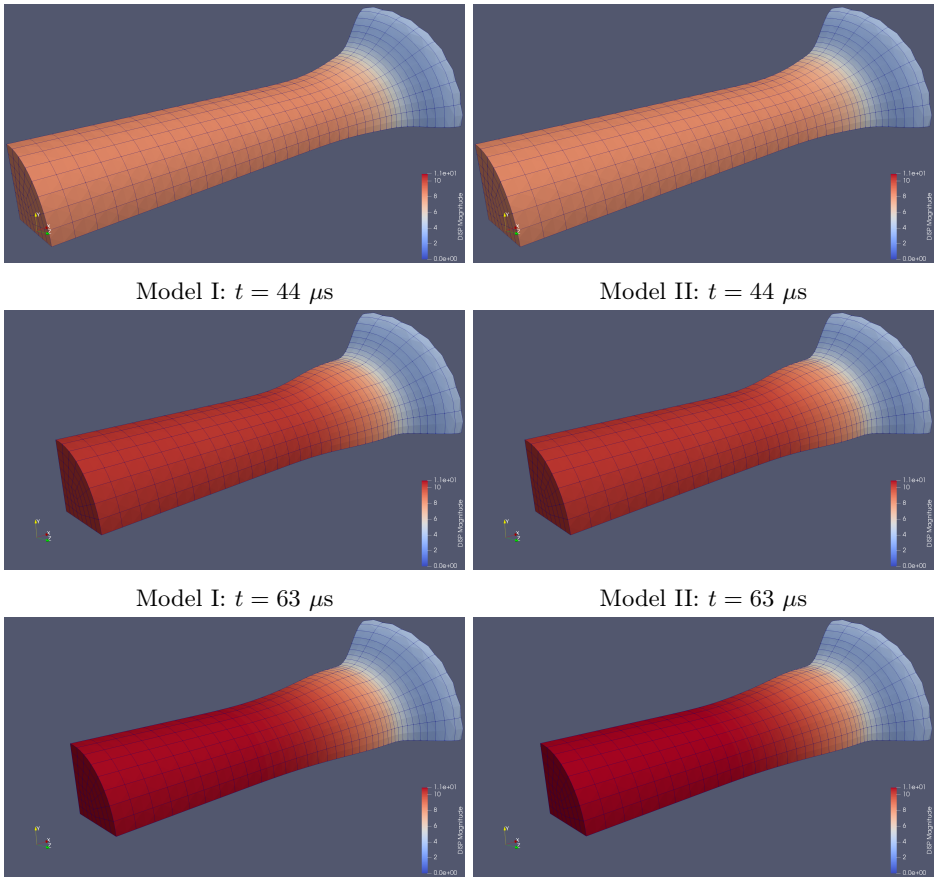
Table 2: Circular bar: overall performance

Efficiency			Accuracy	
	Model I	Model II		Global relative errors (%)
CPU times (s)	1.8×10^3	5.64×10^2	x-displacement	0.46
Speed-up factors	1.0	3.2	y-displacement	0.92
Number of element $\lvert\mathcal{V}\rvert$	1728	70	z-displacement	0.93
			x-velocity	4.0
			y-velocity	9.6
			z-velocity	9.6

Two configurations illustrating the deformation of the bar over time are given in Figure 2, confirming that Model II produces as accurate results as Model I. Figure 3 graphically displays the displacement

and velocity time histories at node 2336. The location of node 2336 is shown in Figure 1, which is at the end of impacting and it has only z-direction displacement and velocity. Again, this figure shows that Model II reproduces the results of Model I with the a good level of accuracy. A notable exception is at the peak of velocity profile. We conclude that the proposed approach of mesh sampling and its training algorithm provides a alternative model of Equation 1. From application and its results, we observe that it produces accurate result but not as fast as expected.

See code here https://github.com/wanlihe/mesh-sammpling



Model I: $t = 44$ μs | Model II: $t = 44$ μs

Model I: $t = 63$ μs | Model II: $t = 63$ μs

Model I: $t = 80$ μs | Model II: $t = 80$ μs

Figure 2: Impact of circular bar: sequence of deformed displacement field at $t = 44$ μs, $t = 63$ μs, and $t = 80$ μs of Model I (left column) and Model II (right column)

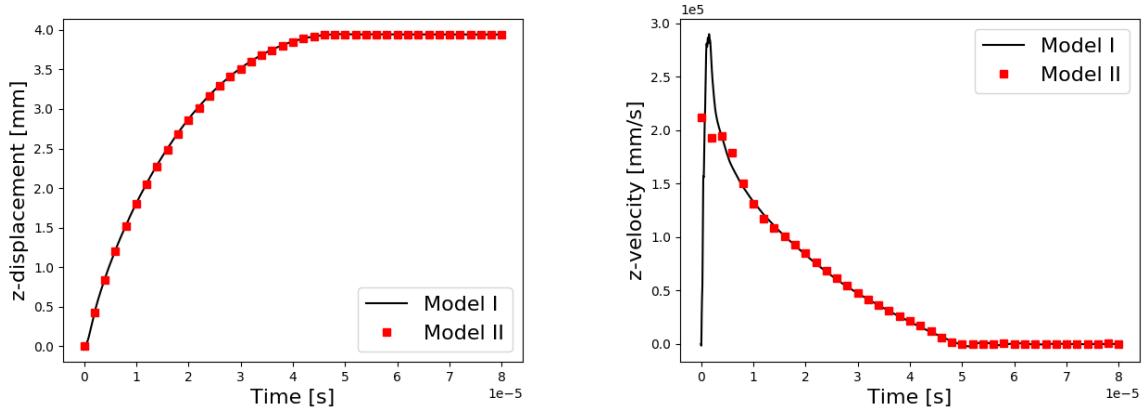


Figure 3: Impact of circular bar: displacements and velocities in z-direction at node 2336

References

[1] T. J. R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. 2000.

[2] M. J. Zahr, P. Avery, and C. Farhat, "A multilevel projection-based model order reduction framework for nonlinear dynamic multiscale problems in structural and solid mechanics," *International Journal for Numerical Methods in Engineering*, vol. 112, no. 8, pp. 855–881, 2017.

[3] C. Farhat, P. Avery, T. Chapman, and J. Cortial, "Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency," *International Journal for Numerical Methods in Engineering*, vol. 98, no. 9, pp. 625–662, 2014.

[4] C. Farhat, T. Chapman, and P. Avery, "Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models," *International Journal for Numerical Methods in Engineering*, vol. 102, no. 5, pp. 1077–1110, 2015.

[5] D. Amsallem, M. J. Zahr, and C. Farhat, "Nonlinear model order reduction based on local reduced-order bases," *International Journal for Numerical Methods in Engineering*, vol. 92, no. 10, pp. 891–916, 2012.