Large Scale FFT-Based Stress-Strain Simulations with Irregular Domain Decomposition

Extended Abstract

Anuva Kulkarni
Electrical and Computer Engineering,
Carnegie Mellon University
Pittsburgh, Pennsylvania
anuvak@andrew.cmu.edu

Franz Franchetti (Advisor)
Electrical and Computer Engineering,
Carnegie Mellon University
Pittsburgh, Pennsylvania
franzf@andrew.cmu.edu

Jelena Kovačević (Advisor)
Electrical and Computer Engineering,
Carnegie Mellon University
Pittsburgh, Pennsylvania
jelenak@cmu.edu

ABSTRACT

In this work, we propose an irregular domain decomposition method to reduce the memory requirement of a Fast Fourier Transform (FFT)-based stress-strain simulation algorithm. Large-scale stress-strain simulations involving parallel FFTs require large amounts of memory and have high communication overhead. Our early results show that both these issues can be addressed using data models and local FFTs without adversely impacting accuracy of the result.

CCS CONCEPTS

• **Theory of computation → Parallel algorithms**; • **Data-intensive parallel algorithms**;

KEYWORDS

FFT, large scale simulations, domain decomposition, GPU

1 INTRODUCTION

Massive parallel supercomputers can run large-scale simulations to model various phenomena in physics, biological sciences and engineering. Simulations involving partial differential equations (PDEs) usually make use of large parallel FFTs, which use all-all communication and hence, have high communication overhead. The high data movement and memory usage is often the chief bottleneck in computation [4]. One such simulation method is the Moulinec-Suquet Composite (MSC)-Basic Scheme, a local stress-strain computation algorithm [2].

In MSC-Basic Scheme, the microstructure (i.e. arrangement of grains in the composite) is discretized onto a regular $N_1 \times N_2 \times N_3$ grid and a PDE with periodic boundary conditions is formulated using the stress-strain constitutive relation and equilibrium conditions. This PDE is solved iteratively using the method of Green's functions [3], which uses FFTs for convolution. In each iteration, the stress and strain fields, which are 3×3 rank-2 symmetric tensors at each grid point, are computed. Increasing the resolution of the grid is desirable since the composite has interesting behavior at grain boundaries, and larger problem sizes can be run with parallelized code. However, this requires large parallel FFT computations (3-D FFT for each tensor component). For serial code alone, Table 1 shows the memory requirement for various simulation sizes (FORTRAN code based on [2] and [1]).

Table 1: Memory requirement of MSC-Basic Scheme (serial version)

Input data size	32^3	64^3	128^3	256^3	512^3	1024^3
Memory required (GB)	0.07	0.55	4.44	35.5	284	2272

Given that grain boundaries are the critical regions of interest, it follows naturally that the material volume be decomposed into grains which can be processed independently in parallel. Grain interior has smooth varying fields that can be approximated. However, grain boundaries must be maintained at full resolution to capture the interaction effects. By processing individual grains using smaller local FFTs, we may incur additional operations but the memory requirement is reduced significantly and all-all communication can be eliminated.

In this work, our first goal is to show that the irregular domain decomposition using grains does not significantly impact the accuracy of the stress and strain fields in each iteration. Once this is established, we will address the boundary interactions issue and move towards a GPU implementation that reduces communication by using models to compactly represent stress and strain fields in the grains (domains). Our method is outlined in Fig. 1. We modify MSC-Basic Scheme into a new algorithm, MSC-Alternate Scheme, which is used in the step indicated by the red box. Additionally, reduced data movement can be achieved in the communication steps (yellow boxes) by transmitting model parameters instead of a large tensor field.

The next section describes more details of MSC-Basic Scheme and MSC-Alternate Scheme. This is followed by some proof-of-concept results.

2 METHOD

In this section, we describe both simulation methods in more detail. First, we include a short description of tensor notation.

2.1 Tensors and Tensor Notation

Einstein notation is used to represent tensor components and operations. Subscripts denote the tensor components. Eg., A_{ij} refers to component (i, j) of the rank-2 tensor A. Repetition of indices

SC'17, November 12-17 2017, Denver, Colorado USA

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/

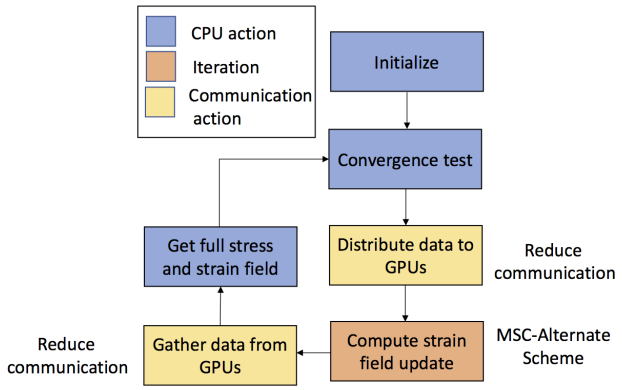

Figure 1: Proposed method

implies a summation over those particular indices. An important tensor operation is the *contraction of indices* (denoted by ':'), which forms lower rank tensors from higher rank tensors. This is done by a sum over repeated indices after multiplication. Thus, $C_{ijkl} : D_{ij} = \sum_i \sum_j C_{ijkl} D_{ij} = E_{kl}$ and yields a rank-2 tensor.

2.2 MSC-Basic Scheme

The pseudocode for this method is as given below. $\epsilon(\mathbf{x})$ and $\sigma(\mathbf{x})$ refer to the strain and stress tensor fields at point $\mathbf{x}$ respectively. $C_{ijkl}(\mathbf{x})$ is the rank-4 stiffness tensor. E is initial average strain. $\hat{\Gamma}_{mnkl}(\boldsymbol{\xi})$ is the Green's operator in Fourier space at frequency point $\boldsymbol{\xi}$. The convergence error is e_s and tolerance error is e_{tol}. $\Delta\epsilon_{kl}$ is the computed perturbation in component (k, l) of the strain tensor. Superscripts indicate iteration number. The iterative scheme continues till convergence is reached. For more details, refer to [2].

Algorithm 1 MSC Basic Scheme

Initialize:
  $\epsilon^0 \leftarrow E$,
  $\sigma^0_{mn}(\mathbf{x}) \leftarrow C_{mnkl}(\mathbf{x}) : \epsilon^0_{kl}(\mathbf{x})$
while $e_s > e_{tol}$ **do**
  $\hat{\sigma}^i_{mn}(\boldsymbol{\xi}) \leftarrow \text{FFT}(\sigma^i_{mn}(\mathbf{x}))$
  Check convergence
  $\Delta\hat{\epsilon}^{i+1}_{kl}(\boldsymbol{\xi}) \leftarrow \hat{\Gamma}_{klmn}(\boldsymbol{\xi}) : \hat{\sigma}^i_{mn}(\boldsymbol{\xi})$
  Update strain: $\hat{\epsilon}^{i+1}_{kl}(\boldsymbol{\xi}) \leftarrow \hat{\epsilon}^i_{kl}(\boldsymbol{\xi}) - \Delta\hat{\epsilon}^{i+1}_{kl}(\boldsymbol{\xi})$
  $\epsilon^{i+1}_{kl}(\mathbf{x}) \leftarrow \text{IFFT}(\hat{\epsilon}^{i+1}_{kl}(\boldsymbol{\xi}))$
  Update stress: $\sigma^{i+1}_{mn}(\mathbf{x}) \leftarrow C_{mnkl}(\mathbf{x}) : \epsilon^{i+1}_{kl}(\mathbf{x})$

2.3 MSC-Alternate Scheme

In MSC-Alternate Scheme, each grain is processed independently using a smaller FFT and the results are combined into the original volume at the end of each iteration. A stress update step is performed by contraction of the stiffness tensor with the updated strain field. In the next iteration, the updated stress field is again decomposed into grains and distributed. Given G grains in the composite, let $(\sigma^i(\mathbf{x}))_j$ be the stress in grain $j \in \text{G}$. The pseudocode for MSC-Alternate Scheme is given by Algorithm 2.

Algorithm 2 MSC Alternate Scheme

Initialize:
  $\epsilon^0 \leftarrow E$,
  $\sigma^0_{mn}(\mathbf{x}) \leftarrow C_{mnkl}(\mathbf{x}) : \epsilon^0_{kl}(\mathbf{x})$
while $e_s > e_{tol}$ **do**
  for each grain $j \in \text{G}$ **do**
    $(\hat{\sigma}^i_{mn}(\boldsymbol{\xi}))_j \leftarrow \text{FFT}((\sigma^i_{mn}(\mathbf{x}))_j)$
    Check convergence
    Update strain: $(\Delta\hat{\epsilon}^{i+1}_{kl}(\boldsymbol{\xi}))_j \leftarrow \hat{\Gamma}_{klmn}(\boldsymbol{\xi}) : (\hat{\sigma}^i_{mn}(\boldsymbol{\xi}))_j$
    $(\Delta\epsilon^{i+1}_{kl}(\mathbf{x}))_j \leftarrow \text{IFFT}((\Delta\hat{\epsilon}^{i+1}_{kl}(\mathbf{x}))_j)$
  Gather step: $\Delta\epsilon^{i+1}_{kl} \leftarrow \sum_j (\Delta\epsilon^{i+1}_{kl})_j$
  Update strain: $\epsilon^{i+1}_{kl}(\mathbf{x}) \leftarrow \epsilon^i_{kl}(\mathbf{x}) - \Delta\epsilon^{i+1}_{kl}(\mathbf{x})$
  Update stress: $\sigma^{i+1}_{mn}(\mathbf{x}) \leftarrow C_{mnkl}(\mathbf{x}) : \epsilon^{i+1}_{kl}(\mathbf{x})$

A MATLAB demo implementation computes error between the outputs of the MSC-Basic Scheme and MSC-Alternate Scheme for various microstructures in a copper dataset. Table 2 shows error for 3 grid sizes, 32^3, 64^3 and 128^3 for different numbers of iterations. We conclude that output MSC-Alternate Scheme is in excellent agreement with that of MSC-Basic Scheme.

Table 2: Error in MSC-Alternate Scheme for different grids

Iter.	% error in stress			% error in strain		
#	32^3	64^3	128^3	32^3	64^3	128^3
1	0.013 %	0.004 %	0.002%	0.012 %	0.003 %	0.003 %
5	0.972 %	0.066 %	0.180 %	1.020 %	0.096 %	0.130 %

2.4 Reduced Communication

An important observation is that the stress and strain fields in the grain interior are smooth but undergo more variation at the grain boundaries. Hence, a model can be fit to the fields in the grain interior while leaving the fields at the boundary uncompressed as approximations at the boundary lead to large errors. Due to lack of space however, modeling results are not described here.

3 CONCLUSIONS

This work demonstrates an alternate scheme designed to reduce memory requirements of MSC-Basic Scheme. Instead of one large 3-D FFT, we perform smaller FFTs in parallel on each of the grains of the composite. Our preliminary results show that the accuracy of stress and strain is not significantly affected by using MSC-Alternate Scheme instead of MSC-Basic Scheme. Approximation to fields in grain interior have been considered here, and the next phase will carefully look at the grain boundary interactions, which must happen at full resolution. Future work also includes a GPU implementation that achieves savings in communication by transmitting models for fields in grains in place of full tensor fields.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Anthony Rollett, Dr Vahid Tari and the staff at Pittsburgh Supercomputing Center for all their assistance and collaboration with this project.

REFERENCES

[1] Ricardo Lebensohn. 2001. N-site modeling of a 3D viscoplastic polycrystal using fast Fourier transform. *Acta Materialia* 49, 14 (2001), 2723–2737.

[2] Hervé Moulinec and Pierre Suquet. 1998. A numerical method for computing the overall response of nonlinear composites with complex microstructure. *Computer methods in applied mechanics and engineering* 157, 1-2 (1998), 69–94.

[3] Toshio Mura. 1983. Micromechanics of Defects in Solids. *The Journal of the Acoustical Society of America* 73, 6 (1983), 2237–2237. https://doi.org/10.1121/1.389536 arXiv:http://dx.doi.org/10.1121/1.389536

[4] Yogish Sabharwal, Saurabh Garg, Rahul Garg, John Gunnels, and Ramendra Sahoo. 2008. Optimization of Fast Fourier Transforms on the Blue Gene/L Supercomputer. *High Performance Computing - HiPC 2008: 15th International Conference, Bangalore, India, December 17-20, 2008. Proceedings* (2008), 309–322. https://doi.org/10.1007/978-3-540-89894-8_29