Algorithm Design for Large Scale Parallel FFT-Based Simulations on Heterogeneous Platforms

Anuva Kulkarni
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh PA 15213
Email: anuvak@andrew.cmu.edu

Franz Franchetti
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh PA 15213
Email: franzf@cmu.edu

Jelena Kovačević
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh PA 15213
Email: jelenak@cmu.edu

***Abstract*—Large scale iterative simulations involving parallel Fast Fourier Transforms (FFTs) have extreme memory requirements and high communication overhead. This prevents scaling to higher grid sizes, which is necessary for high resolution analysis. In this work, we describe an algorithm to overcome these limitations and run stress-strain simulations for larger problem sizes using irregular domain decomposition and local FFTs. Early results show that our method lowers iteration cost without adversely impacting accuracy of the result.**

I. Introduction

Large scale simulations running on machines with many cores are required to study and model various phenomena in physics, biological sciences and engineering. Particularly, simulations involving partial differential equations (PDEs) usually make use of large parallel FFTs, which use all-all communication. However, the chief limiting factors while scaling problem size are prohibitive memory requirements and communication bottlenecks [1], which make high resolution analysis with finer and finer grids impossible. One such method is the Moulinec-Suquet Composite (MSC) Basic Scheme, which is a FORTRAN scheme for local stress-strain computation in composites [2].

In MSC Basic Scheme, the microstructure (i.e. arrangement of grains in the composite) is discretized onto a regular grid and a PDE with periodic boundary conditions is formulated using the stress-strain constitutive relation and equilibrium conditions. The PDE is solved iteratively using convolution with Green's functions [3]. Increasing the resolution is desirable to study interesting behavior at grain boundaries, but larger problems require parallel FFT computations (3-D FFT for each tensor component). For serial code, Table 1 shows the memory requirement for various simulation sizes. This severely limits scaling to larger grid sizes.

In this work, we present MSC Alternate Scheme, an algorithm designed to run stress-strain simulations for large datasets on heterogenous platforms with GPUs. To reduce parallel FFT communication, we operate on different parts of the 3D volume using smaller, local FFTs. To reduce memory requirement, we propose using lossy compression to compactly represent stress field in the grains of the microstructure.

In the background section, we describe the MSC Basic Scheme and pseudocode. The MSC Alternate Scheme is described in the next section, followed by some proof-of-concept results.

TABLE I
Memory requirement of MSC Basic Scheme (serial version) based on [2] and [4]

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

II. Background

In this section, we describe the MSC Basic Scheme in more detail. Note that Einstein notation is used to represent tensor components and operations. Thus, A_{ij} refers to component (i, j) of the rank-2 tensor A. Repetition of indices implies a summation over those particular indices. An important tensor operation is the *contraction of indices* (denoted by ':'). Eg., $C_{ijk\ell} : D_{ij} = \sum_i \sum_j C_{ijk\ell} D_{ij} = E_{k\ell}$ and yields a rank-2 tensor.

The MSC Basic Scheme is a fixed-point iterative numerical method used as an alternative to Finite Element Methods (FEM) to compute local stress and strain fields using Hooke's law. The pseudocode for MSC Basic Scheme is as given below. $\epsilon(\mathbf{x})$ and $\sigma(\mathbf{x})$ are strain and stress tensor fields at point $\mathbf{x}$ respectively. $C_{ijk\ell}(\mathbf{x})$ is the rank-4 stiffness tensor. E is initial average strain. $\hat{\Gamma}_{mnk\ell}(\boldsymbol{\xi})$ 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}. $\Delta\epsilon_{k\ell}$ 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].

The convolution with Green's function requires computation of 3D FFTs of each of the 9 components of the stress field, hence the need for extensive resources for large grids. MSC Basic Scheme is implemented in serial FORTRAN and MPI parallel (using FFTW) versions.

III. Proposed method: MSC Alternate Scheme

This section briefly describes the proposed algorithm, designed to be implemented on a CPU-GPU hardware setup.

Algorithm 1 MSC Basic Scheme

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

A MATLAB-FORTRAN workflow has been used to build a prototype and obtain preliminary results, which are discussed here.

We observe that grain interiors have smooth stress and strain fields that are compressible. Hence, after initialization on the CPU side, an irregular domain decomposition method decomposes the volume into smaller grain volumes with smooth fields. Data models are used to communicate initial stress fields in the smaller volumes to GPUs. In each GPU-based iteration, a method for performing local FFTs on these fields has been developed by us and is used for convolution with the Green's function in the Fourier domain. Then, communication between GPUs serves to transfer parts of the result to respective grains on different GPUs so that stress update for a grain is a self-contained problem. The effect of convolution is summarized by data communicated from different GPUs. This makes the GPU part of the code intrinsically parallel. In this way, stress and strain fields are updated till convergence.

IV. Results

For proof-of-concept results, a simple microstructure test dataset with two types of grain orientations was created using MATLAB for various grid sizes. Grains in each grid of size $N \times N \times N$ are $N/2 \times N/2 \times N/2$ cubes arranged in a periodic lattice. The cubical shape makes it easier to test the prototype with simple windowing techniques. More complicated preprocessing will be used for irregularly shaped grains.

In the MSC Alternate Scheme, n lower cost iterations of the fixed-point method are to be performed on GPUs using data models and local FFTs. A few (4 to 5) high cost iterations are performed using the MSC Basic Scheme to reduce approximation errors in the final answer. The plot in Fig. 1 shows the convergence of stress fields in MSC-Basic Scheme and MSC Alternate Scheme for different values of n in a simulation of size $128 \times 128 \times 128$. We observe that for equal error thresholds for both methods, number of iterations for convergence changes depending on n, but not drastically. Fig. 2 shows mismatch in stress field between the original and proposed method, arising due to approximations. Metrics for iteration cost are not provided here. However, the all-all communication is reduced since GPUs communicate only a few coefficients to summarize the effect of convolution, as opposed to all-all communication for transposes in FFTW.

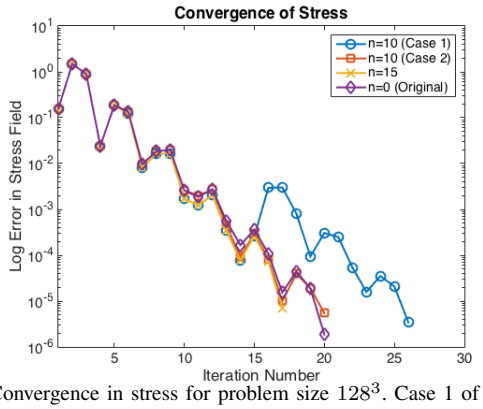

Fig. 1. Convergence in stress for problem size 128^3. Case 1 of n=10 refers to performing iterations 5 to 15 on GPUs, and case 2 refers to performing iterations 10 to 20 on GPUs.

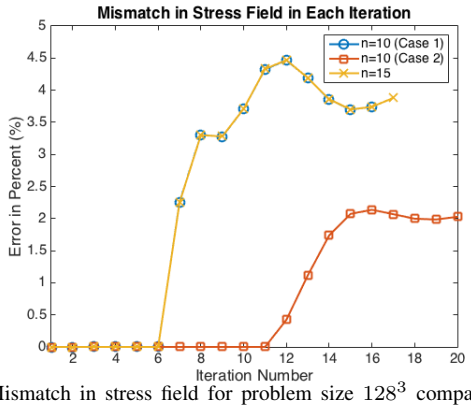

Fig. 2. Mismatch in stress field for problem size 128^3 compared to MSC Basic Scheme, showing convergence to a different local minimum with a small deviation (about $2-4\%$).

V. Conclusions

The proposed MSC Alternate Scheme is a co-design of algorithm and software for heterogenous platforms. It enables scaling of stress-strain simulations to large grids by overcoming high memory requirements and communication bottlenecks. The algorithm uses small local FFTs and data modeling to perform iterations with a lowered cost, which converge to the same solution as the MSC-Basic Scheme with a small accuracy tradeoff, as is seen in proof-of-concept results presented here.

References

[1] Y. Sabharwal, S. Garg, R. Garg, J. Gunnels, and R. K. Sahoo, "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*, pp. 309–322, 2008.

[2] H. Moulinec and P. Suquet, "A numerical method for computing the overall response of nonlinear composites with complex microstructure," *Computer methods in applied mechanics and engineering*, vol. 157, no. 1-2, pp. 69–94, 1998.

[3] T. Mura, "Micromechanics of defects in solids," *The Journal of the Acoustical Society of America*, vol. 73, no. 6, pp. 2237–2237, 1983. [Online]. Available: http://dx.doi.org/10.1121/1.389536

[4] R. A. Lebensohn, "N-site modeling of a 3d viscoplastic polycrystal using fast fourier transform," *Acta Materialia*, vol. 49, no. 14, pp. 2723–2737, 2001.