GPU Acceleration of Finite Element Analysis and Its Application to Large-Scale Structural Topology Optimization

Synopsis Report
In Partial Fulfilment of the Requirements
for the Degree of
Doctor of Philosophy

by

Subhajit Sanfui
156103038

to the

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
November, 2021

1 Introduction

Finite element method (FEM) is a numerical method for approximating solutions of boundary value problems for partial differential equations. Since it offers advantages such as flexibility, adaptability and ease of implementation even for complex geometries, it is widely used across fields such as mechanical engineering, civil engineering, electrical engineering and medical applications to name a few. Finite element analysis (FEA), which is the process of applying concepts of FEM for the analysis of physical systems is known to be a computationally expensive process [Georgescu et al., 2013]. Furthermore, the need for large-scale implementations makes FEA even more computationally demanding especially for dealing with many real-world problems. The aim of this thesis is accelerating FEA on massively parallel graphics processing units (GPUs) with specific focus on its different stages and applications to another computationally expensive process, i.e., structural topology optimization. Due to their suitability for data-parallel and throughput intensive applications, GPUs have been popular in reducing execution time of a large variety of computationally expensive applications. Naturally, several efforts have been made in the literature towards accelerating several steps of FEA on GPU such as meshing, model simplification, numerical integration, assembly and matrix solver.

One of the most important applications of FEA can be found in structural topology optimization, which itself is another computationally expensive process due to the need for high resolution meshes and repeated FEA computations [Deaton and Grandhi, 2014]. Within a given design domain, topology optimization aims to find the optimal distribution of material subject to given boundary conditions and optimization constraints. Density-based topology optimization method is one of the popular methods in which each element of the finite element mesh is assigned a density value which becomes the design variable in the optimization process. Solid isotropic material with penalization (SIMP) and bi-directional evolutionary structural optimization (BESO) are two of the most popular density-based topology optimization methods. Since FEA is the most computationally expensive part in topology optimization, any improvement in its execution time directly impacts the total optimization time.

Through an extensive literature review, we identify several key aspects and research gaps in the acceleration of FEA and for structural topology optimization on modern many-core architectures, which would benefit a significant portion of the scientific community. In this thesis we target to fill some of those research gaps while building a versatile high performance computation tool for the entire linear FEA process and structural topology optimization. The tool incorporates parallelization of entire algorithms with specific focus on the individual building blocks of the algorithms. The primary targets can be summarized into two important challenges as observed in the literature: reduce execution time and reduce memory footprint of parallel applications [Mukherjee et al., 2021]. This is achieved by incorporating a combination of several algorithm-level and high performance computing-based enhancements to the standard FEA and density-based topology optimization methods. The key objectives of the thesis are listed below.

1. Generation of elemental matrices for different finite elements with their assembly and storage on GPUs.

2. Implementation of topology optimization with mesh reduction on GPU using SIMP

method

3. Complete GPU acceleration for large-scale bi-directional evolutionary structural optimization

4. Development of GPU-based hybrid BESO method using mesh reduction approach

5. Performance analysis of the developed tools for benchmark problems and comparison with the state of art

In the following sections, the proposed strategies are discussed in order to achieve the objectives of the thesis.

2 GPU-Accelerated FEA Matrix Generation

The first objective of the thesis is to accelerate the matrix generation stage of FEA using GPU computing. This objective includes generation of elemental stiffness matrices for three dimensional structured and unstructured meshes and for lower and higher order elements on the GPU, along with their assembly into a global stiffness matrix. The primary challenge with this objective is the effective distribution of workload along with an efficient storage scheme for the global stiffness matrix on GPU. For assembly, a three-stage GPU-based FEA matrix generation strategy is presented with the key idea of decoupling the computation of global stiffness matrix indices and values by the use of a novel data structure referred to as the neighbor matrix. The first stage computes the neighbor matrix on the GPU based on the unstructured mesh. Using this neighbor matrix, the indices and values of the global stiffness matrix are computed separately in the second and third stages. The neighbor matrix is computed for different element types.

Two sparse storage formats based on the proposed strategy are also developed by modifying the existing sparse storage formats with the intention of removing the degrees of freedom-based redundancies in the global stiffness matrix. The inherent problem of race condition is resolved through the implementation of coloring and atomics. The proposed strategy is compared with the *state-of-the-art* GPU-based and CPU-based assembly techniques. These comparisons reveal a significant number of benefits in terms of reducing storage space requirements and execution time and increasing performance (GFLOPS). The modified sparse storage formats are found to reduce the storage requirements of global stiffness matrices by more than 50% for different order and types of elements in the mesh. For different proposed implementations on structured and unstructured meshes, speedups of $4\times - 6\times$ and $80\times - 100\times$ are observed over the standard GPU-based and CPU-based implementations. Moreover, using the proposed strategy, it is found that the coloring method is more effective compared to the atomics-based method for the existing as well as the modified storage formats. Furthermore, two different versions for performing numerical integration and assembly in the same and separate kernels are implemented and simulations are run for different mesh sizes having up to 3 million degrees of freedom on a single GPU. The same kernel implementation is found to outperform the separate kernel implementation by 70% to 150% for different element types.

Although assembly-based methods in FEA are relatively simpler to implement, they can become intractable for very-large scale problems, where, despite using the most optimized sparse storage schemes, the memory requirements and cost of data movement become prohibitively high. This leads to another class of solvers called matrix-free or assembly-free methods, where all the FEA computations are carried out at the element level, obviating the need to explicitly store the entire global stiffness matrix at any point of the application. Matrix-free methods are discussed in the following sections in the context of large-scale FEA acceleration of topology optimization problems.

3 Topology Optimization with Reduced Meshes

From the literature of structural topology optimization, it can be observed that several studies have been done to reduce its computational cost using algorithm-level as well as high performance computing (HPC)-based modifications. These individual efforts have brought significant improvements to the conventional density-based topology optimization methods. Typically, completely different approaches are needed for these types of modifications. The algorithm-level modifications need the researcher to focus on the *physics* or *mathematics* of the problem, whereas, for HPC-based modifications, the focus is kept mostly on *efficient implementation* of the standard algorithm on the target architecture. In the second objective of the thesis, we aim to combine these two modifications in order to significantly reduce the computational cost of density-based topology optimization using solid isotropic material with penalization (SIMP) method.

Two of the most successful algorithm-level modifications for GPU-based large-scale topology optimization are the use of matrix-free FEA methods and the reduction of design variables by a local update strategy for the optimization algorithm. In this work, these two algorithm-level modifications are combined with efficient GPU-based acceleration using a novel mesh reduction strategy that aims to reduce the computational complexity of conventional structural topology optimization. In the proposed strategy, the effective number of design variables is reduced by using the concept of *active* nodes and *active* elements in the finite element mesh. A node is considered to be *active* if it is a part of at least one element with non-zero density. An element, on the other hand, is considered to be *active* if it contains at least one *active* node. Nodes and elements that do not satisfy these conditions, are considered *inactive*, and can be expelled from the computation without significant effect on the final outcome. Introducing this concept to GPU-accelerated density-based topology optimization algorithms is not straightforward due to the parallel nature of the computation and the need to dynamically update the finite element mesh at every optimization iteration. Another important challenge is locating the boundary condition nodes after performing mesh reduction to remove the *inactive* nodes and elements on GPU. This issue is addressed by introducing a novel mesh numbering scheme to facilitate parallel identification of *active* nodes using the proposed GPU-based algorithm. The preconditioned conjugate gradient (PCG) solver is further developed using the proposed strategy and the numbering scheme. The proposed strategy is tested on several structural topology optimization problems using SIMP method. The obtained final topologies are found to be identical with the

results from the literature. Results of the proposed strategy demonstrate up to 8× speedup over the standard GPU implementation considering the entire mesh for the SIMP method.

The proposed strategy is also found to be suitable for other topology optimization methods. In the following section, a complete GPU acceleration of the BESO method pipeline is discussed. Furthermore, BESO is also coupled with the mesh reduction strategy developed in this section.

4 Complete GPU Acceleration of Large-Scale BESO Method

For achieving the third objective, a GPU-accelerated implementation for large-scale BESO method is presented. This work addresses the primary challenge of high computational complexity in performing large-scale topology optimization through a complete GPU acceleration of the entire BESO method pipeline. The second major challenge of prohibitively high memory consumption is handled by implementing a matrix-free PCG solver in the finite element analysis stage. An element-by-element strategy has been adopted to parallelize the FEA, sensitivity and compliance calculation, mesh filter, and design update stages. Among all the optimization steps, FEA is observed to consume up to 92% of the total execution time. Consequently, the focus of this work is kept on an efficient parallelization of FEA using PCG method with Jacobi preconditioner. It is observed that the PCG algorithm consists of different linear algebra operations which contribute to all the computations in the algorithm.

- Sparse matrix-vector multiplication (SpMV)
- Inner product (IP)
- A X plus Y (SAXPY)

These individual building blocks of the PCG algorithm are accelerated using suitable strategies that combine the use of the thrust library and custom GPU kernels. Apart from PCG, the other stages of BESO including sensitivity calculation, compliance calculation, mesh filter, stabilization, and design update are also accelerated on GPU using a combination of thrust calls and custom kernels. In order to demonstrate the usability of the implementation for large-scale structures, meshes with up to 70 million nodes are considered. It was found that the GPU memory consumption increased linearly with an increase in the number of nodes in the mesh. Furthermore, it was observed that, even for the largest-sized mesh analyzed, the application only consumed approximately one-sixth of the total GPU memory. Lastly, it is demonstrated that by making small changes to the implementation, it can be applied to solve a large-scale heat transfer problem using BESO.

Following the acceleration of the BESO method on GPU, the mesh reduction strategy is implemented to develop a novel GPU-based hybrid BESO method that combines the soft-kill and hard-kill strategies. The hard-kill formulation uses a solid/void design with discrete values of the densities for the structures. The void elements are removed from the model altogether at every optimization iteration. The soft-kill formulation, on the other hand uses very small values for the densities of void elements with a material interpolation scheme. This solves some of the computational issues with hard-kill at the expanse of increased computation. The proposed hybrid BESO uses the hard-kill approach for the FEA stage and soft-kill approach for all other

stages to eliminate the redundant computational cost incurred due to performing FEA for non-functional degrees of freedom on GPU. This allows the key advantage of the hard-kill approach to be incorporated into the soft-kill without the latter inheriting any of its several drawbacks, such as the breakage of boundary support and other convergence issues. A numbering scheme similar to the one used in Section 3 is implemented for on-the-fly calculation of active DOFs on GPU. The comparison of the hybrid BESO with the standard GPU-accelerated soft-kill BESO using four benchmark problems with more than two million degrees of freedom reveals three key benefits of the proposed hybrid model: reduced execution time with an overall speedup of up to 4×, decreased memory consumption of up to 20%, and improved FEA convergence in terms of PCG iterations, all of which mitigate the major computational issues associated with BESO.

5 Organization of Thesis

The thesis is organized in the following manner.

Chapter 1 begins with the introduction of the problem statement. This is followed by the challenges and a brief description of the previous efforts in the literature. The research gaps are then discussed which lead to the motivation and the objectives of the thesis. Finally the organization of the thesis is discussed.

In **Chapter 2**, the preliminaries including details of GPU computing, FEA, and topology optimization are discussed. This is followed by the literature review of GPU-accelerated FEA and density-based topology optimization.

Chapter 3 presents efficient GPU acceleration of the matrix generation stage in FEA. This includes parallel generation of elemental stiffness, the proposed kernel division strategy for structured and unstructured FEA assembly, and finally the efficient storage of the global stiffness matrix on GPU.

Chapter 4 presents the mesh reduction strategy that is applied to the SIMP method on GPU with matrix-free FEA to achieve the second objective of the thesis. Implementation-level details are provided with focus on the mesh update strategy, mesh renumbering scheme, and parallel computation of active nodes.

Chapter 5 presents the complete GPU acceleration of the BESO method pipeline for completion of the third objective of the thesis. In this chapter, a compact 250-line code for GPU-accelerated FEA is provided that includes all steps of optimization starting from meshing to the plotting of the optimal topology. This chapter also discusses the proposed hybrid BESO method that combines the soft-kill and hard-kill formulation. The strategies for on-the-fly computation of active DOFs along with all the hybrid BESO implementation steps are discussed according to their acceleration strategies.

Chapter 6 presents the important conclusions of this thesis and the future work.

References

Serban Georgescu, Peter Chow, and Hiroshi Okuda. GPU acceleration for FEM-based structural analysis. *Archives of Computational Methods in Engineering*, 20(2):111–121, 2013. ISSN 1134-3060. doi: https://doi.org/10.1007/s11831-013-9082-8.

Joshua D Deaton and Ramana V Grandhi. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization*, 49 (1):1–38, 2014. doi: https://doi.org/10.1007/s00158-013-0956-z.

Sougata Mukherjee, Dongcheng Lu, Balaji Raghavan, Piotr Breitkopf, Subhrajit Dutta, Manyu Xiao, and Weihong Zhang. Accelerating large-scale topology optimization: State-of-the-art and challenges. *Archives of Computational Methods in Engineering*, pages 1–23, 2021. doi: https://doi.org/10.1007/s11831-021-09544-3.

Journal Publications

Published:

- Subhajit Sanfui and Deepak Sharma, 2021, "Symbolic and Numeric Kernel Division for GPU-based FEA Assembly of Regular Meshes with Modified Sparse Storage Formats", ASME Journal of Computing and Information Science in Engineering, 22 (1), 1-12. https://doi.org/10.1115/1.4051123
- Subhajit Sanfui and Deepak Sharma, 2020, "A Three-Stage GPU-based FEA Matrix Generation Strategy for Unstructured Meshes", International Journal of Numerical Methods in Engineering, 121 (17), 3824-3848. https://doi.org/10.1002/nme.6383

In review:

- Subhajit Sanfui and Deepak Sharma, "GPU-based mesh reduction strategy for accelerating structural topology optimization", Applied Soft Computing
- Subhajit Sanfui and Deepak Sharma, "Soft- and Hard-Kill Hybrid GPU-based Bi-Directional Evolutionary Structural Optimization", Structural and Multidisciplinary Optimization
- Subhajit Sanfui and Deepak Sharma, "A 250-line Fully Parallelized CUDA Code for Large-Scale Bi-Directional Evolutionary Structural Optimization using GPU", Structural and Multidisciplinary Optimization

In preparation:

- Subhajit Sanfui and Deepak Sharma, A Review of GPU accelerated FEA for Structural Analysis.

Conference Publication

- Subhajit Sanfui, Shashi Kant Ratnakar, Deepak Sharma, "A Parametric Study on the Convergence Behaviour of SIMP-Based Structural Topology Optimization using GPU", 4^{th} National Conference on Multidisciplinary Design, Analysis, and Optimization (NCMDAO 2021), 7-9 October 2021, IIT Madras, India.
- Subhajit Sanfui and Deepak Sharma, "Exploiting Symmetry in Elemental Computation and Assembly Stage of GPU-Accelerated FEA", Proceedings at the 10th International Conference on Computational Methods (ICCM2019), 9–13 July 2019, Singapore, Eds: G.R. Liu, Fangsen Cui, George Xu Xiangguo, ScienTech Publisher, pp. 641 – 651
- Subhajit Sanfui and Deepak Sharma, "GPU Acceleration of Local Matrix Generation in FEA by Utilizing Sparsity Pattern", In 1st International Conference on Mechanical Engineering (INCON 2018), 4–6 January 2018, Jadavpur University, India.
- Subhajit Sanfui and Deepak Sharma, "A Two-Kernel based Strategy for Performing Assembly in FEA on the Graphic Processing Unit", In IEEE International Conference on Advances in Mechanical, Industrial, Automation and Management Systems, 3-5 February 2017, MNNIT Allahabad, India.

Other Publications:

- Shashi Kant Ratnakar, Subhajit Sanfui and Deepak Sharma, "GPU-based Element-by-Element Strategies for Accelerating Topology Optimization of 3D Continuum Structures Using Unstructured Mesh", ASME Journal of Computing and Information Science in Engineering, 1-17. https://doi.org/10.1115/1.4052892

- Shashi Kant Ratnakar, Subhajit Sanfui and Deepak Sharma, "GPU – based Topology Optimization using Matrix-Free Conjugate Gradient Finite Element Solver with Customized Nodal Connectivity Storage", 2nd International Conference on Future Learning Aspects of Mechanical Engineering (FLAME - 2020), August 5 – 7, 2020, Amity University Uttar Pradesh, Noida, India

- Shashi Kant Ratnakar, Subhajit Sanfui and Deepak Sharma, "SIMP-based Structural Topology Optimization using Unstructured Mesh on GPU", 2nd International Conference on Future Learning Aspects of Mechanical Engineering (FLAME - 2020), August 5 – 7, 2020, Amity University Uttar Pradesh, Noida, India

- Utpal Kiran, Subhajit Sanfui, Shashi Kant Ratnakar, Sachin Singh Gautam, and Deepak Sharma, "Comparative Analysis of GPU-based Solver Libraries for A Sparse Linear System of Equations", in 2nd International Conference on Computational Methods in Manufacturing (ICCMM), 8 – 9 March 2019, IIT Guwahati, India.