Two-Dimensional Adaptive Finite Element Using GPGPU

Amir Hossein Khatami, Saeed Asil Gharebaghi*

Civil Engineering Department, K. N. Toosi University of Technology

* asil@kntu.ac.ir

Abstract

The discretization error is one of the most common errors in the finite element method. One way to reduce this error is by using adaptive methods. Adaptive methods generally involve a large computational load; a technique for reducing this load is the use of data transfer operators. Even with data transfer operators, adaptive methods still require significant time from users. Given the new capabilities provided by graphics processing units (GPUs) for general-purpose computing under the CUDA platform, and the economic efficiency of GPUs compared to standard processors, this paper aims to present an algorithm that can reduce computation time through general-purpose GPU processing. The proposed algorithm begins with an initial finite element analysis using a nearly uniform mesh and refines the mesh intelligently at each step based on the displacement gradient. The conventional algorithm has been improved at several points. The patch formation stage is implemented using the K-nearest neighbor method to facilitate more efficient parallelization. In the data transfer stage, a dynamic method is employed to select the optimal curve from a set of the best curves. The results show that the acceleration of this algorithm increases proportionally with the number of elements. For instance, for a problem with 908 elements, the processing speed for stages one through three increased by factors of 6.6, 9.1, and 12.7, respectively. The total time required for all three stages in serial processing was 96 seconds, which was reduced to 8 seconds using this algorithm.

Keywords: Data Transfer Operator, Adaptive Finite Element Method, GPU, GPGPU, KNN

1 Introduction

To reduce computational cost, data transfer operators are used instead of solving problems from scratch. Various operators have been proposed to transfer information between successive meshes. The weighted average method, although simple, suffers from considerable errors. The dissipative element method and constant transfer operator methods have similar limitations. Zienkiewicz and Zhu introduced the superconvergent patch recovery (SPR) method, significantly reducing errors by fitting higher-order polynomials over superconvergent points [1]. Boroomand and Zienkiewicz extended SPR to elastoplastic problems [2], while Khoei and Asil Gharebaghi developed a three-dimensional version for elastoplasticity, demonstrating good accuracy for nonlinear cases [3-8].

Despite its precision, SPR-based data transfer is computationally expensive. Given the heavy computational load of data transfer operators in adaptive finite element methods, parallel processing can dramatically reduce computation time. Originally designed for graphics, GPUs evolved to support general-purpose computing, notably after NVIDIA introduced CUDA in 2007. This allowed researchers to efficiently perform finite element calculations beyond traditional CPU capabilities. In previous studies, GPU acceleration focused mainly on matrix assembly and solution phases.

2 Methodology

The adaptive meshing algorithm consists of the following major steps:

a. An initial mesh is generated, and the finite element analysis of the problem is performed.
b. Using the displacement values at each node of the old mesh, the norm of the displacement gradient at each point is computed.
c. Based on the displacement gradient at each point of the old mesh, the geometric directions corresponding to the maximum and minimum variations are determined.
d. The new element size is calculated according to these maximum and minimum values, reflecting the appropriate stretching of elements.
e. Using this information, a new mesh is generated such that elements are elongated along the direction of minimal gradient variation and compressed along the direction of maximal variation, resulting in an adaptively refined mesh.
f. Each node of the new mesh is located within the old mesh to find the parent element containing it. Using the shape functions of the old element, displacement values are interpolated at the new node. This completes the first data transfer operator.
g. For each Gauss point of the new mesh, a set of Gauss points from the old mesh is selected such that they are the closest neighbors. This method is referred to as the Gauss point neighborhood.
h. A patch is constructed from these neighboring Gauss points, and stress and strain values at the Gauss point of the new mesh are calculated and stored based on the patch.
i. Improved stress and strain values for the Gauss points of the old mesh are computed using the patch fitting method obtained in the previous step, thus completing the second data transfer operator.
j. The errors in stress and strain are estimated by comparing the recovered values with the original finite element solutions. Unlike the displacement norm, which is evaluated at nodes, the strain error is evaluated at Gauss points.
k. At this stage, the finite element analysis data have been successfully transferred to the new mesh, allowing the analysis to continue without repeating the previous computations.
l. The new mesh is analyzed. If the maximum error values and their distribution are not acceptable, the new mesh is considered as the old mesh, and the entire algorithm is repeated.

3 Results and Discussion

a. The proposed method was tested on several benchmark problems, including Laplace's equation over irregular domains. Results demonstrate that:
b. The GPGPU-accelerated adaptive meshing achieves speedups of 10×–30× compared to conventional CPU-based approaches.
c. The final meshes are well-adapted to solution features, showing higher element densities in regions with steep gradients.
d. The error distribution across the mesh becomes more uniform after each adaptation step.
e. Memory usage remains within acceptable bounds, enabling the handling of meshes with several million elements on a standard GPU.
f. Overall, the method proves robust, efficient, and suitable as shown in Fig. 1 and Table 1.

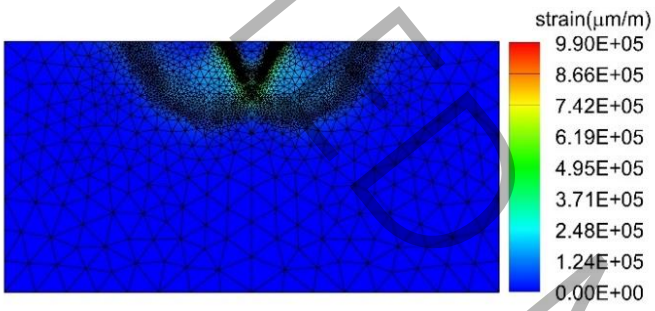

Fig. 1 Critical paths in punch test, captured by the algorithm

Table 1. Elapsed time using CPU and GPU (sec.)

CPU	GPU
3.3	0.5
15.5	1.7
77.2	6.1

4 Conclusion

To effectively accelerate adaptive FEA, the algorithm was modified for GPU parallel execution, notably by replacing traditional patch construction with a K-nearest neighbor (KNN) approach adapted as K Gauss point neighbors. The modified algorithm intelligently refines the mesh, reducing discretization errors automatically at each step. Validation through the Prandtl punch test confirmed its accuracy and practical applicability. The speedup depends on the number of elements; finer meshes yield greater acceleration. In the presented case, refinement stages were accelerated by factors of 6.6, 9.1, and 12.7, reducing total computation time from 96 seconds (serial) to 8 seconds (GPU parallel). Overall, the algorithm significantly enhances performance through GPU parallelization and promises even better results with newer hardware, making it a valuable tool for engineering simulations.

5 References

[1] O.C. Zienkiewicz, J.Z. Zhu, Superconvergence and the superconvergent patch recovery, Finite Elements in Analysis and Design, 19(1) (1995) 11-23.
[2] B. Boroomand, O.C. Zienkiewicz, Recovery procedures in error estimation and adaptivity. Part II: Adaptivity in nonlinear problems of elasto-plasticity behaviour, Computer Methods in Applied Mechanics and Engineering, 176(1) (1999) 127-146.

[3] S. Asil Gharebaghi, A.R. Khoei, Three-dimensional superconvergent patch recovery method and its application to data transferring in small-strain plasticity, Computational Mechanics, 41(2) (2008) 293-312.
[4] S.A. Gharehbaghi, A.R. Khoei, Three-Dimensional Data Transfer Operators in Plasticity Using SPR Technique with C0, C1 and C2 Continuity, Scientia Iranica, 15(5) (2008) -.
[5] A.R. Khoei, S.A. Gharehbaghi, The superconvergence patch recovery technique and data transfer operators in 3D plasticity problems, Finite Elements in Analysis and Design, 43(8) (2007) 630-648.
[6] A.R. Khoei, S.A. Gharehbaghi, Three-dimensional data transfer operators in large plasticity deformations using modified-SPR technique, Applied Mathematical Modelling, 33(7) (2009) 3269-3285.
[7] A.R. Khoei, S.A. Gharehbaghi, A.R. Azami, A.R. Tabarraie, SUT-DAM: An integrated software environment for multi-disciplinary geotechnical engineering, Advances in Engineering Software, 37(11) (2006) 728-753.
[8] A.R. Khoei, S.A. Gharehbaghi, A.R. Tabarraie, A. Riahi, Error estimation, adaptivity and data transfer in enriched plasticity continua to analysis of shear band localization, Applied Mathematical Modelling, 31(6) (2007) 983-1000.