Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2024-0072

Vol. **36**, No. 5, pp. 1157-1185
November 2024

A Simple GPU Implementation of Spectral-Element Methods for Solving 3D Poisson Type Equations on Rectangular Domains and Its Applications

Xinyu Liu[1], Jie Shen[2] and Xiangxiong Zhang[1,*]

[1] *Purdue University, 150 N. University Street, West Lafayette, IN 47907-2067, USA.*
[2] *Eastern Institute of Technology, Ningbo, Zhejiang 315200, P.R. China.*

Received 26 March 2024; Accepted (in revised version) 13 June 2024

Abstract. It is well known since 1960s that by exploring the tensor product structure of the discrete Laplacian on Cartesian meshes, one can develop a simple direct Poisson solver with an $\mathcal{O}(N^{\frac{d+1}{d}})$ complexity in d-dimension, where N is the number of the total unknowns. The GPU acceleration of numerically solving PDEs has been explored successfully around fifteen years ago and become more and more popular in the past decade, driven by significant advancement in both hardware and software technologies, especially in the recent few years. We present in this paper a simple but extremely fast MATLAB implementation on a modern GPU, which can be easily reproduced, for solving 3D Poisson type equations using a spectral-element method. In particular, it costs less than one second on a Nvidia A100 for solving a Poisson equation with one billion degree of freedoms. We also present applications of this fast solver to solve a linear (time-independent) Schrödinger equation and a nonlinear (time-dependent) Cahn-Hilliard equation.

AMS subject classifications: 65F05, 65F60, 65Y05, 65M06, 65M60, 65M70, 65N06, 65N30, 65N35

Key words: 3D Poisson equation, direct solver, spectral element methods, rectangular domain, GPU, tensor matrix multiplication.

1 Introduction

It is well known that the tensor product structure of the discrete Laplacian on Cartesian meshes can be used to invert the Laplacian since 1960s [15]. This approach has been particularly popular for spectral and spectral-element methods [2, 8, 12, 17, 18]. In fact,

*Corresponding author. *Email addresses:* liu1957@purdue.edu (X. Liu), jshen@eitech.edu.cn (J. Shen), zhan1966@purdue.edu (X. Zhang)

this method can be used for any discrete Laplacian on a Cartesian mesh. In this paper, as an example, we focus on the Q^k spectral-element method, which is equivalent to the classical continuous finite element method with Lagrangian Q^k basis implemented with the $(k+1)$-point Gauss–Lobatto quadrature [13,16].

Tensor based solvers naturally fit the design of graphic processing units (GPUs). The earliest successful attempts to accelerate the computation of high order accurate methods in scientific computing communities include nodal discontinuous Galerkin method [11] almost fifteen years ago. These pioneering efforts of GPU acceleration of high order methods, or even those ones published later such as [3] in 2013, often rely on intensive coding.

In recent years, the surge in computational demands from machine learning and neural network based approaches has led to the evolution of modern GPUs. Correspondingly, software technologies have advanced considerably, streamlining the utilization of GPU computing. The landscape of both hardware and software has dramatically transformed, differing substantially from what existed a decade or even just two years ago.

In this paper, we present a straightforward yet robust implementation of accelerating the spectral-element method for three-dimensional discrete Laplacian on modern GPUs. In particular, for a total number of degree of freedoms as large as 1000^3, the inversion of the 3D Laplacian using an arbitrarily high order Q^k spectral-element method, takes no more than one second on one Nvidia A100 GPU card with 80G memory. While this impressive computational speed is naturally contingent on the hardware, it is noteworthy that our approach is grounded in a minimalist MATLAB implementation, ensuring ease of replication. In the Appendix, we give a full MATLAB code for solving a 3D Poisson equation on a rectangular domain using Q^k spectral-element method.

We remark that a similar simple implementation on GPUs can also be achieved using Python using the Python package JAX, which however provides better performance than MATLAB only for single precision computation under TensorFloat-32 precision format. Our numerical tests and comparison on one Nvidia A100 GPU card suggest that MATLAB implementation performs better than Python for double precision computation for large problems like one billion DoFs.

We emphasize that the ability of solving Poisson type equation fast can play an important role in many fields of science and engineering. In fact, a large class of time dependent nonlinear systems, after a suitable implicit-explicit (IMEX) time discretization, often reduces to solving Poisson type equations at each time step (see, for instance, [19]). Therefore, having a simple, accurate and very fast solver for Poisson type equations can lead to very efficient numerical algorithms on modern GPUs for these nonlinear systems which include, e.g., Allen-Cahn and Chan-Hillard equations and related phase-field models [19], nonlinear Schrödinger equations, Navier-Stokes equations and related hydro-dynamic equations through a decoupled (projection, pressure correction etc.) approach [7]. In particular, by using the code provided in the Appendix, one can build, with a relatively easy effort, very efficient numerical solvers on modern GPUs for these time dependent complex nonlinear systems.

The rest of the paper is organized as follows. In Section 2, we give the implementation