INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING, *SERIES B*
Volume 5, Number 1-2, Pages 136–146

ACCELERATING PRECONDITIONED ITERATIVE LINEAR SOLVERS ON GPU

HUI LIU, ZHANGXIN CHEN AND BO YANG

Abstract. Linear systems are required to solve in many scientific applications and the solution of these systems often dominates the total running time. In this paper, we introduce our work on developing parallel linear solvers and preconditioners for solving large sparse linear systems using NVIDIA GPUs. We develop a new sparse matrix-vector multiplication kernel and a sparse BLAS library for GPUs. Based on the BLAS library, several Krylov subspace linear solvers, and algebraic multi-grid (AMG) solvers and commonly used preconditioners are developed, including GMRES, CG, BICGSTAB, ORTHOMIN, classical AMG solver, polynomial preconditioner, ILU(k) and ILUT preconditioner, and domain decomposition preconditioner. Numerical experiments show that these linear solvers and preconditioners are efficient for solving the large linear systems.

Key words. Krylov subspace solver, algebraic multi-grid solver, parallel preconditioner, GPU computing, sparse matrix-vector multiplication, HEC

1. Introduction

Linear and nonlinear solvers play an important role in scientific computing areas. In many scientific applications, the solution of systems of linear algebraic equations dominates the whole simulation time. The black oil simulator [1], for example, may run for weeks or even months depending on the problem size. In this simulator, iterative linear solvers may take up to 98% of the whole simulation time. Therefore, the development of fast and accurate linear solvers and preconditioners are essential to many applications.

Efficient iterative linear solvers, preconditioners and parallel computing techniques have been developed to accelerate the solution of linear systems. Saad developed the GMRES solver, a general solver for unsymmetric linear system [22, 23] and Vinsome designed the ORTHOMIN solver, which was originally developed for reservoir simulation [2], for example. Commonly used preconditioners were also developed, such as Incomplete LU (ILU) factorization, domain decomposition and algebraic multigrid preconditioners [22, 23]. GPUs (Graphics Processing Unit) are usually used for display, since each pixel can be processed simultaneously. GPUs are designed in such a way that they can manipulate data in parallel. They have parallel architectures and their memory speed is very high [5, 6, 7]. In general, GPUs are ten to forty times faster than general CPUs (Central Processing Units) [5, 6, 7], which makes them proper devices for parallel computing, especially for developing fast linear solvers. The architectures of GPUs are different from those of CPUs, and therefore, new algorithms for GPUs should be designed to match the architectures of GPUs. Efforts have been made to utilize GPUs' performance [5, 6, 7, 3, 4, 27, 8, 10, 11]. Bell and Garland investigated different kinds of matrix formats and SpMV algorithms in [3, 4], in which the HYB format matrix and the corresponding SpMV algorithm was also designed. Naumov from NVIDIA developed a fast triangular solver for GPU and the solver was over two times faster

Received by the editors January 12, 2014 and, in revised form, March 20, 2014.
2000 *Mathematics Subject Classification.* 65F30, 65F50, 65Y05, 65Y10, 68W10.
This research was supported by NSERC/AIEE/Foundation CMG and AITF Chairs.

than CPU-based triangular solver. Saad et al. used JAD matrix and developed an efficient sparse matrix-vector multiplication (SpMV) kernel for GPU, and based on the kernel, several Krylov solvers and ILU preconditioner were implemented on GPU. Haase et al. developed a parallel AMG solvers using a GPU cluster [12]. Researchers from NVIDIA also developed an AMG solver on GPU [9, 13]. The setup and solving phases were both run on GPU, which made their AMG very fast. Chen et al. from University of Calgary designed a new matrix format HEC (Hybrid of Ell and CSR), fast SpMV kernel [11], parallel triangular solver [10], parallel preconditioners [8] and the GPU solvers have been applied to reservoir simulation [14, 17]. More details can be read from [5, 6, 7, 3, 4, 27, 8, 10, 11, 15, 24, 28].

In this paper, we introduce our work on developing a general purpose GPU-based parallel iterative linear solver package. We design a flexible matrix format for GPU and its corresponding SpMV algorithm [11]. The matrix is also a good choice for parallel triangular solver [28]. Based on this SpMV and other matrix-vector operations, several Krylov subspace solvers are implemented. For symmetric positive definite matrices, AMG solvers are the most effecient methods, which project a low frequency error to a coarser grid, and convert it to a high frequency error. In this case, the resulting high frequency error is easy to converge again. The convergence rate of these AMG solvers is optimal [18, 19, 20, 21] in terms of the number of iterations. We implement a classical AMG solver, which also serves as a preconditioner. Other commonly used preconditioners are also implemented, including ILU(k), ILUT(p, tol), RAS (Restricted Additive Schwarz method), polynomial and block ILU(k), block ILUT(p, tol). Numerical experiments show that our linear solvers and preconditioners are over ten times faster than CPU-based solvers and preconditioners.

The layout of the paper is as follows. In §2, GPU computing will be introduced briefly. In §3, our work on linear solvers will be proposed. IN §4, parallel preconditioners are presented. Then numerical experiment will be presented in §5.

2. GPU Computing

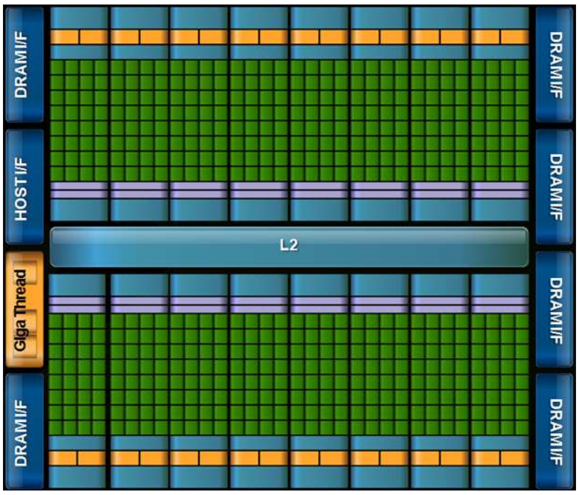

Figure 1: NVIDIA Fermi Architecture.