Journal of Computational and Applied Mathematics 236 (2011) 281–293

Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

# GPU implementation of a Helmholtz Krylov solver preconditioned by a shifted Laplace multigrid method

H. Knibbe *, C.W. Oosterlee, C. Vuik

*Faculty of Electrical Engineering, Mathematics and Computer Science, Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands*

ARTICLE INFO

*Keywords:*
GPU
Helmholtz equation
Krylov solvers
Shifted Laplace multigrid preconditioner

ABSTRACT

A Helmholtz equation in two dimensions discretized by a second order finite difference scheme is considered. Krylov methods such as Bi-CGSTAB and IDR($s$) have been chosen as solvers. Since the convergence of the Krylov solvers deteriorates with increasing wave number, a shifted Laplace multigrid preconditioner is used to improve the convergence. The implementation of the preconditioned solver on CPU (Central Processing Unit) is compared to an implementation on GPU (Graphics Processing Units or graphics card) using CUDA (Compute Unified Device Architecture). The results show that preconditioned Bi-CGSTAB on GPU as well as preconditioned IDR($s$) on GPU is about 30 times faster than on CPU for the same stopping criterion.



## 1. Introduction

Initially driven by the gamer market, GPUs (Graphics Processing Units) recently became suitable for high performance computing applications. A GPU or simply a graphics card is a multi-threaded, many-core processor which was originally developed for graphics processing. With the evolution of General Purpose computing on Graphics Processing Units (GPGPU) it became possible to accelerate a wide range of applications traditionally computed on a CPU (Central Processing Unit). One of the market leaders, Nvidia, developed a parallel computer architecture called CUDA (Compute Unified Device Architecture); see [1]. CUDA is a subset of the C language with some extensions which allows us to program the Nvidia GPUs in an easy way. The more recent initiative of Apple within Khronos Group is called OpenCL (Open Computing Language), that is an open standard and can be used to program CPUs, GPUs and other devices from different vendors. It has been shown that converting a CUDA program to an OpenCL program involves minimal modifications; see [2]. According to Du et al. [3], at this moment CUDA is more efficient on the GPU than OpenCL.

In this paper we focus on iterative solvers for the Helmholtz equation in two dimensions on GPU using CUDA. The Helmholtz equation represents the time-harmonic wave propagation in the frequency domain and has applications in many fields of science and technology, e.g. in aeronautics, marine technology, geophysics, and optical problems. In particular we consider the Helmholtz equation discretized by a second order finite difference scheme. The size of the discretization grid depends on the wave number, that means, the higher the wave number the more grid points are required. For instance to get accurate results with 7-point discretization scheme in three dimensions, at least 10 grid points per wave length have to be used; see [4]. For high wave numbers the discretization results in a very large sparse system of linear equations which cannot be solved with direct methods on current computers within reasonable time. The linear system is symmetric but indefinite, non-Hermitian and ill-conditioned which brings difficulties when solving with basic iterative methods. The convergence of

* Corresponding author.
*E-mail addresses:* hknibbe@gmail.com (H. Knibbe), c.w.oosterlee@cwi.nl (C.W. Oosterlee), c.vuik@tudelft.nl (C. Vuik).

0377-0427/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2011.07.021

the Krylov methods deteriorates with increasing wave number, so the need for preconditioning becomes obvious. In this paper we consider Bi-CGSTAB (see [5]) and IDR($s$) (see [6]) as Krylov solvers.

There have been many attempts to find a suitable preconditioner for the Helmholtz equation, see, for example, [7,8]. Recently the class of shifted Laplace preconditioners evolved, see [9–12]. In this work, we focus on a shifted Laplace multigrid preconditioner introduced in [11,13], to improve the convergence of the Krylov methods.

The purpose of this work is to compare the implementations of the Krylov solver preconditioned by the shifted Laplace multigrid method in two dimensions on a CPU and a GPU. The interest is triggered by the fact that some applications on GPU are 50–200 times faster compared with a CPU implementation (see e.g. [14,15]). However there are no recordings of a Helmholtz solver on a GPU which we present in this paper. There are two main issues: the first one is the efficient implementation of the solver on GPU and the second one is the behavior of the numerical methods in single precision. Nevertheless, even on a modern graphics card with double precision units (for example, Tesla 20 series or Fermi), single precision calculations are still at least two times faster. The first issue can be resolved by knowing the insides of a GPU and CUDA. The second issue can be addressed by using mixed precision algorithms; see e.g. [16].

The paper is organized as follows. In Section 2 we describe the Helmholtz equation and its discretization. Also the components of the solver are described, including Krylov methods such as Bi-CGSTAB and IDR($s$) and the shifted Laplace multigrid method. The specific aspects of the GPU implementation for each method are considered in detail in Section 3 and optimizations for the GPU are suggested. In Section 4 two model problems are defined: with constant and variable wave numbers. We solve those problems with Krylov methods preconditioned by the shifted Laplacian on a single CPU and a single GPU and compare the performance. Finally Section 5 contains conclusions and an outlook of this paper.

## 2. Problem description

The two dimensional Helmholtz equation for a wave problem in a heterogeneous medium is considered

$$-\frac{\partial^2\phi(x,y)}{\partial x^2}-\frac{\partial^2\phi(x,y)}{\partial y^2}-(1-\alpha\mathrm{i})k^2(x,y)\phi(x,y)=g(x,y),\quad x,y\in\Omega \tag{1}$$

where $\phi(x,y)$ is the wave pressure field, $k$ is the wavenumber, $\alpha$ is the damping coefficient, $g(x,y)$ is the source term. The corresponding differential operator has the following form:

$$\mathcal{A}=-\Delta-(1-\alpha\mathrm{i})k^2,$$

where $\Delta$ denotes the Laplace operator. In this paper we consider a rectangular domain $\Omega=[0,X]\times[0,Y]$. There are several boundary conditions:

- Dirichlet boundary conditions $\phi|_{\partial\Omega}=0$,
- Non-reflecting boundary conditions
  - First order radiation boundary condition (described in e.g. [17,18])
$$\left(-\frac{\partial}{\partial\eta}-\mathrm{i}k\right)\phi=0, \tag{2}$$
    where $\eta$ is the outward unit normal component to the boundary. The disadvantage of this boundary condition is that it is not accurate for inclined outgoing waves.
  - Second order radiation boundary condition (described in [18])
$$\mathcal{B}\phi|_{\text{edge}}:=-\frac{3}{2}k^2\phi-\mathrm{i}k\sum_{j=1,j\neq i}^{2}\left(\pm\frac{\partial\phi}{\partial x_j}\right)-\frac{1}{2}\frac{\partial^2\phi}{\partial x_i}=0, \tag{3}$$
$$\mathcal{B}\phi|_{\text{corner}}:=-2\mathrm{i}k\phi+\sum_{i=1}^{2}\left(\pm\frac{\partial\phi}{\partial x_i}\right)=0, \tag{4}$$
    where $x_i$ is a coordinate parallel to the edge for $\mathcal{B}\phi|_{\text{edge}}$. The $\pm$ sign is determined such that for outgoing waves the non-reflecting conditions are satisfied.

In many real world applications the physical domain is unbounded, and artificial reflections should be avoided. Therefore, we consider here the first order radiation boundary condition (2).

### 2.1. Discretization

The domain $\Omega$ is discretized by an equidistant grid $\Omega_h$ with the grid size $h$

$$\Omega_h:=\{(ih,jh)\mid i,j=1,\ldots,N\}.$$

For simplicity we set the same grid sizes in $x$- and $y$-directions. After discretization of Eq. (1) on $\Omega_h$ using central finite differences we get the following linear system of equations:

$$A\phi=g,\quad A\in\mathbb{C}^{N\times N},\ \phi,g\in\mathbb{C}^N. \tag{5}$$