ParCFD'2020
32^{nd} International Conference on Parallel Computational Fluid Dynamics
May-11-13 2020, Nice FRANCE

A HIGHLY PORTABLE HETEROGENEOUS IMPLEMENTATION OF A POISSON SOLVER FOR FLOWS WITH ONE PERIODIC DIRECTION

Àdel ALSALTI-BALDELLOU*†, F. Xavier TRIAS†, Xavier ÁLVAREZ-FARRÉ† and Assensi OLIVA†

*Termo Fluids S.L.
C/ Magí Colet 8, 08204 Sabadell (Barcelona), Spain
e-mail: adel@termofluids.com, web page: http://www.termofluids.com

†Heat and Mass Transfer Technological Center
Technical University of Catalonia, ESEIAAT
C/ Colom 11, 08222 Terrassa (Barcelona), Spain
e-mail: adel@cttc.upc.edu, web page: http://www.cttc.upc.edu

Key words: Poisson solver, FFT, CPU+GPU, Portability

Abstract. The portability of codes has become a major advantage given the continuous development of new architectures for numerical applications, as well as the progressive incorporation of accelerators in modern supercomputers. Following this trend, we have adopted an algebraic approach in the implementation of a Poisson solver for incompressible flows with one periodic direction. This approach, which basically consists of adapting a reduced set of fundamental operations to any architecture (such as the sparse matrix-vector product or the dot product of two vectors), allows us to efficiently port our applications to any heterogeneous supercomputers in an easy manner. More particularly, our three-dimensional solver takes advantage of the existing periodic dimension (by means of a Fourier decomposition) to later execute overlapped data transpositions among devices, which conveniently share the workload with their CPU hosts, and solve the resulting two-dimensional decoupled subsystems.

1 INTRODUCTION

As the HPCG Benchmarck [1] shows, the increasing unbalance between the memory bandwidth and the floating point operations per second (FLOPS) leads, in most scientific applications, to very poor sustained performances. Hence, in the race towards exascale high-performance computing (HPC), the need for higher bandwidths together with more energetically efficient systems, has resulted into the development and use of massively-parallel accelerators such as GPUs, Intel Xeon Phi co-processors or fused CPU-GPU devices.

In this varying context, the portability of codes has become crucial in order to perform efficiently on most heterogeneous platforms. With the aim of overcoming this difficulty, we developed the HPC^2 framework [2] to reduce the number of operations involved in

a simulation to a set of three fundamental ones: sparse matrix-vector product ($SpMV$), the linear combination ($axpy$) and the dot product (dot) of two vectors. Despite its simplicity, the multilevel MPI+OpenMP+OpenCL+CUDA implementation provides a natural adaptation to most current platforms.

The aim of this work is, then, to present a linear solver that successfully follows this algebraic approach and, consequently, is efficiently portable to most current heterogeneous supercomputers. More concretely, the solver presented here has been developed for three-dimensional incompressible flows with one periodic dimension, which are governed by the coupled Navier-Stokes and continuity equations. The application of the well known fractional-step method leads to the implicit solution of the Poisson equation (arising from the incompressibility constraint), which, after an appropriate discretization [3], can be written as:

$$A^{3D}x^{3D} = b^{3D} \tag{1}$$

where x and b correspond, respectively, to the discretized pressure field and minus the divergence of the predictor velocity. A^{3D} stands for the discretized Laplace operator accordingly changed of sign (so that A^{3D} is symmetric and positive semidefinite).

The present work is devoted to the solution of Equation 1 for incompressible flows with one periodic direction, where Fourier decomposition can be applied to boost its solution. A mathematical description of the method is given in Section 2, whereas its implementation is tackled in Section 3.

2 FOURIER DECOMPOSITION OF THE PERIODIC DIRECTION

If we aim to apply a Fourier decomposition along the periodic direction, we are forced to discretize it uniformly. In consequence, we will also assume that our three-dimensional domain arises from a uniform extrusion (along the periodic direction) of a two-dimensional mesh.

As shown in [3], assuming that x is the periodic direction leads to the following decomposition of A^{3D} in terms of the Kronecker product:

$$A^{3D} = \left(A^{2D} \otimes \Omega_x\right) + \left(\Omega^{2D} \otimes A_x\right) \tag{2}$$

where $A^{2D} \in \mathbb{R}^{N^{2D} \times N^{2D}}$ is the discretized Laplace (changed of sign) corresponding to the two-dimensional mesh that is extruded, and $\Omega^{2D} \in \mathbb{R}^{N^{2D} \times N^{2D}}$ is a diagonal matrix whose elements are the areas of the N^{2D} control volumes of that mesh. Analogously, $\Omega_x = \Delta x \mathbb{I}_{N_x} \in \mathbb{R}^{N_x \times N_x}$ and $A_x \in \mathbb{R}^{N_x \times N_x}$ is a symmetric circulant matrix.

Let $\mathcal{Q}_{N_x} \in \mathbb{R}^{N_x \times N_x}$ be the one-dimensional inverse Fourier transform matrix (along the periodic direction). Then, as detailed in [3], A_x can be diagonalized as follows:

$$\Lambda = \mathcal{Q}_{N_x}^{-1} A_x \mathcal{Q}_{N_x} = \mathrm{diag}\left(\lambda_1, \ldots, \lambda_{N_x}\right) \tag{3}$$

Therefore, we can define:

$$\hat{A}^{3D} \coloneqq \left(\mathbb{I}_{N^{2D}} \otimes \mathcal{Q}_{N_x}\right)^{-1} A^{3D} \left(\mathbb{I}_{N^{2D}} \otimes \mathcal{Q}_{N_x}\right) \tag{4}$$

which can straightforwardly be rewritten as:

$$\hat{A}^{3D} = \Delta x \left(A^{2D} \otimes \mathbb{I}_{N_x} \right) + \left(\Omega^{2D} \otimes \Lambda \right) \tag{5}$$

and, reordering the unknowns, we finally reach the following block diagonal expression:

$$\hat{A}^{3D} = \text{diag}\left(\hat{A}_1^{2D}, \ldots, \hat{A}_{N_x}^{2D} \right) = \bigoplus_{i=1}^{N_x} \hat{A}_i^{2D}, \text{ where } \hat{A}_i^{2D} = \Delta x A^{2D} + \lambda_i \Omega^{2D} \tag{6}$$

Hence, we have been able to transform the original coupled three-dimensional system of $N_x N^{2D}$ unknowns of Equation 1 into a set of N_x decoupled two-dimensional subsystems of N^{2D} unknowns (in the Fourier space):

$$\hat{A}_i^{2D} \hat{x}_i^{2D} = \hat{b}_i^{2D} \quad \forall i \in \{1, \ldots, N_x\} \tag{7}$$

such that $\hat{x}^{3D} = \left(\hat{x}_1^{2D}, \ldots, \hat{x}_{N_x}^{2D} \right)$ and $\hat{b}^{3D} = \left(\hat{b}_1^{2D}, \ldots, \hat{b}_{N_x}^{2D} \right)$.

3 HIGHLY PORTABLE IMPLEMENTATION

We can, therefore, decompose our Poisson solver into the following steps:

Algorithm 1 Poisson solver for flows with one periodic direction

1. Fourier transform along the periodic direction: $\hat{b}_{jk} = \mathcal{Q}_{N_x}^{-1} b_{jk}$
2. Data transposition: pencil-like $\rightarrow$ plane-like
3. Parallel solution of the N_x decoupled two-dimensional subsystems: $\hat{A}_i^{2D} \hat{x}_i^{2D} = \hat{b}_i^{2D}$
4. Data transposition: plane-like $\rightarrow$ pencil-like
5. Inverse Fourier transform along the periodic direction: $x_{jk} = \mathcal{Q}_{N_x} \hat{x}_{jk}$

If we wish to make use of the library HPC^2 [2] in order to inherit its high portability, we will need to adopt its algebraic approach in a way that provides us its highest performance. To do so, we overlap communications and calculations while making an extensive use of the highly optimized kernels of HPC^2, in order to preserve its portability.

3.1 One-dimensional Fast Fourier Transform

When it comes to the calculation of the forward and backward discrete Fourier transforms, we will take advantage of the well known Fast Fourier Transform (FFT) algorithm, which can be found implemented in many open-access libraries.

More concretely, we will firstly distribute data in a pencil-like fashion along the periodic direction to increase data locality. The number of one-dimensional FFTs to be calculated by each device will be conveniently set seeking a proper load balancing. Then, each device will compute its one-dimensional FFTs using external libraries (such as FFTW for CPUs, clFFT for OpenCL devices or cuFFT for NVIDIA GPUs).

3.2 Data transposition

One data transposition will need to be executed at every shift from physical to Fourier space, or vice versa. Its high computational cost makes its efficient implementation a crucial property of the solver, forcing the implementation to overlap, as much as possible, FFTs with non-blocking data communications.

With this aim, we have implemented data transpositions (between pencil and plane-like distributions) as SpMVs between permutation matrices and the vectors to be redistributed.

3.3 Solution of decoupled two-dimensional subsystems

Finally, the solution of the decoupled two-dimensional systems may be executed differently depending on the computational resources available. If an iterative method such as the Preconditioned Conjugate Gradient (PCG) is chosen (note that $\hat{A}^{3D}$ is a symmetric positive semidefinite matrix), then it has to be born in mind that, as shown in [3], the spectral condition number of each submatrix $\hat{A}_i^{2D}$ varies depending on the plane and, consequently, also the number of iterations needed to solve each subsystem.

Hence, the workload corresponding to each plane will present discrepancies that may negatively affect load balancing if the implementation does not deal conveniently with the most ill-conditioned planes.

In the conference, we are going to present the Poisson solver in detail, to give the results of the performance analysis executed on different computing systems and to discuss lines of improvement of the numerical algorithm itself. For instance, a particular case of SpMV that takes advantage of the block matrix structure to increase cache reuse.

ACKNOWLEDGMENTS

The work has been financially supported by a competitive R+D project (ENE2017-88697-R) by the Spanish Research Agency. À.A.B is supported by a predoctoral contract (DIN2018-010061) by the Ministry of Science, Innovation and Universities of Spain (MICINN) and X.Á.F. by the by the Government of Catalonia (2019FI_B2-00076).

REFERENCES

[1] J. Dongarra, M. A. Heroux, and P. Luszczek, "A new metric for ranking high-performance computing systems," *National Science Review*, vol. 3, pp. 30–35, mar 2016.

[2] X. Álvarez, A. Gorobets, F. X. Trias, R. Borrell, and G. Oyarzun, "HPC^2 – A fully-portable, algebra-based framework for heterogeneous computing. Application to CFD," *Computers and Fluids*, vol. 173, pp. 285–292, sep 2018.

[3] A. Gorobets, F. X. Trias, M. Soria, and A. Oliva, "A scalable parallel Poisson solver for three-dimensional problems with one periodic direction," *Computers and Fluids*, vol. 39, no. 3, pp. 525–538, 2010.