HPCLatAm 2013, pp. 69-79 (full paper)
Session: GPU Architecture and Applications
C. Garcia Garino and M. Printista (Eds.)
Mendoza, Argentina, July 29-30, 2013

Solving 3D viscous incompressible Navier-Stokes equations using CUDA

Santiago D. Costarelli, Mario A. Storti, Rodrigo R. Paz, Lisandro D. Dalcin,
and
Sergio A. Idelshon

CIMEC - INTEC, Predio CONICET Santa Fe, Colectora Ruta Nac 168, Km 472,
Paraje El Pozo, CP 3000, Santa Fe, Argentina
{santi.costarelli,mario.storti}@gmail.com
http://www.cimec.org.ar

Abstract. A CUDA implementation of the 3D viscous incompressible Navier-Stokes equations is proposed using as advection operator the BFECC (Back and Forth Error Compensation and Correction) scheme. The Poisson problem for pressure is solved with a CG (Conjugated Gradient) preconditioning the system with FFTs (Fast Fourier Transforms). Study cases such as Lid-Driven Cavity and Flow Past Circular Cilinder , both 2D and 3D, are solved in order to check accuracy and obtain performance meassurements.

Keywords: Navier-Stokes FFT CUDA GPGPU BFECC Poisson

1 Introduction

1.1 Fractional Step Method

The equations being solved are the classical incompressible viscous Navier-Stokes equations

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\,\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

where $\mathbf{u}$ is the velocity field, p the pressure field, ρ the density (constant), ν the kinematic viscosity (constant) and $\mathbf{f}$ a body force per unit volume. These equations are going to be solved using several combination of boundary and initial conditions.

Considering $\mathbf{w}_0$ as an approximation to the solution of $\mathbf{u}$ at time $t = n\Delta t$, where Δt is doing reference at time step and n to performed steps, one can obtain the solution at $t + \Delta t$ by performing successively the following operations [Sta99]

– *Force*: Add force terms

$$\mathbf{w}_1(\mathbf{x}, t) = \mathbf{w}_0(\mathbf{x}, t) + \Delta t\ \mathbf{f}(\mathbf{x}, t). \tag{3}$$

– *Advection*: The BFECC method obtains $\mathbf{w}_2$ from $\mathbf{w}_1$ following the characteristics. As a result an unconditionally stable scheme is obtained.
– *Diffusion*: Here we seek for a solution of

$$\frac{\partial \mathbf{w}}{\partial t} = \nu \nabla^2 \mathbf{w}, \tag{4}$$

in the interval $[t, t + \Delta t]$, with $\mathbf{w}(t) = \mathbf{w}_2$, $\mathbf{w}(t + \Delta t) = \mathbf{w}_3$.
– Projection: Finally the projection step is needed in order to make the resulting field divergence free. This is

$$\mathbf{w}_4(\mathbf{x}, t) = \mathbf{w}_3(\mathbf{x}, t) - \nabla p, \tag{5}$$

where p is defined as the solution of [CML80]

$$\nabla^2 p = \nabla \cdot \mathbf{w}_3 \quad in\ D, \tag{6}$$

$$\nabla p \cdot \hat{\mathbf{n}} = \mathbf{w}_3 \cdot \hat{\mathbf{n}} \quad in\ \partial D, \tag{7}$$

$D \in \Re^n$ being the domain on a n-dimensional space, ∂D its boundary and $\hat{\mathbf{n}}$ the outward normal to ∂D.

1.2 Semi-Lagrangian Time Integration

Lets consider for the moment a scalar field F that is being advected by the velocity field $\mathbf{u}$; mathematically

$$\frac{D_{(m)}F}{Dt} = \frac{\partial F}{\partial t} + \mathbf{u} \cdot \nabla F = 0, \tag{8}$$

where $D_{(m)}/Dt$ stands for material derivatives, i.e. following fluid particles.

The approach for solving this kind of equations follows that presented on [SC91]. Considering equation (8) as 1D (higher dimensional expressions follows that on 1D), a splitting reveals that

$$\frac{\partial F}{\partial t} + \frac{dx}{dt}\frac{\partial F}{\partial x} = 0, \tag{9}$$

where

$$\frac{dx}{dt} = u(x, t), \tag{10}$$

that can be solved leading to an ordinary differential equation whose value is constant throughout the streamline. This last equation relates, in particular, points A and C of Figure 1. The objective is to track back the streamline passing through C at time $t + \Delta t$ to the point defined as A at time t. This can be done following the streamline (solid curve) or using the velocity field (dashed curve), in this case as a first order predictor of the previous position; i.e. point A'.

Analysis of stability properties of the Semi-Lagrangian advection scheme shows that it is possible to stably integrate it for CFL numbers greater than unit.

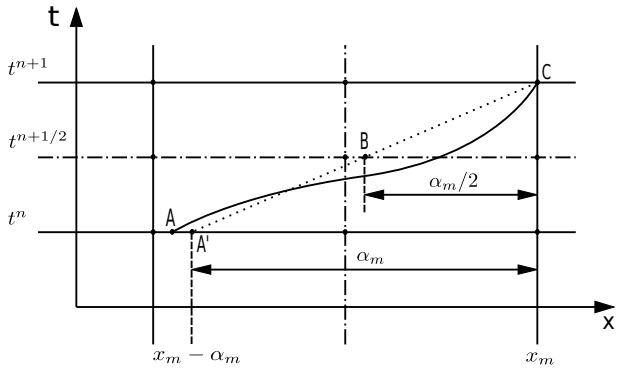

Fig. 1: The point C can be obtained driving along the solid line ($\overline{AC}$), or approximately, using the velocity field as a first order predictor (dashed line $\overline{A'C}$).

1.3 BFECC Method

Following [KLLR07] lets supose that there exists some advection operator $L(.,.)$ performing

$$F^{n+1} = L(\mathbf{u}, F^n), \tag{11}$$

i.e. $L(.,.)$ an upwinding or Semi-Lagrangian advection operator; in this way, consider the Figure 2.

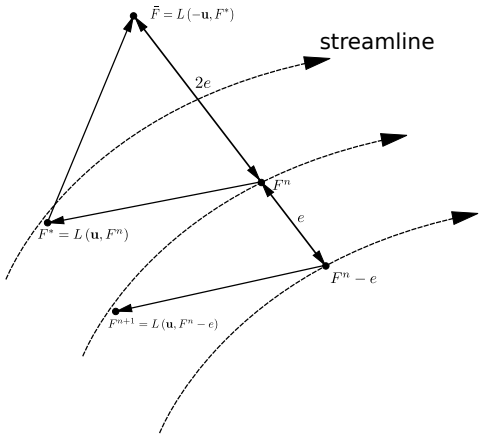

Fig. 2: Schematic BFECC operation over a streamline field and using $L(.,.)$ as the advection operator for the scalar field F.

Supose for the moment that $L(.,.)$ is an exact operator. Following this logic, after applying a forward and backward step one gets the initial field; that is, the

error in the advection is zero. However, this is not true due to numerical errors on the advection operator.

Lets say that the operator contains an error denoted as e. As the same operator is used by backward and forward steps it can be expected an error of $2e$, this is, $\bar{F} = F^n + 2e$, so an explicit expression for e can be readily obtained as

$$e = -\frac{1}{2}\left(F^n - \bar{F}\right). \tag{12}$$

This error can be subtracted from F^n and then advect the field. This method has been proven to be second order accurate in both, time and space [SFK+08].

Considering the advection operator $L(.,.)$ as the Semi-Lagrangian one, BFECC is defined as follows

$$\begin{aligned}
&F^* = L\left(\mathbf{u}, F^n\right)\\
&\bar{F} = L\left(-\mathbf{u}, F^*\right)\\
&F^* = F^n + \left(F^n - \bar{F}\right)/2\\
&F^{n+1} = L\left(\mathbf{u}, F^*\right).
\end{aligned}$$

In this way the order of accuracy of the Semi-Lagrangian scheme can be raised from one to two increasing the amount of work by a factor of three [SFK+08].

1.4 Poisson equation for pressure

This problem is solved using CG as iterative solver preconditioning the system using FFT. A complete analysis in this subject can be found in [SPD+13].

It must be said that, for the case of numerical results that are going to be compared against references, a convergence up to machine precision is used. For performance interests, only up to 3 iterations are used.

2 Tests

2.1 2D lid-driven cavity

This is a classical internal flow test in a square domain with L_c, the domain length, as the characteristic dimension. The shear velocity imposed is fixed at $v = 1$ varying the kinematic viscosity in order to reach the specified Reynolds number, Re_{L_c}, defined as

$$Re_{L_c} = \frac{\rho v L_c}{\mu}. \tag{13}$$

The numerical results obtained at $Re_{L_c} = 1000$ are shown on Figure 3.

The performance obtained, meassured in $[secs/Mcells]$, this is seconds of computation in order to compute one million of nodes, is shown on Table 1.

The main drawback in this study case is the Fourier number, Fo, defined as

$$Fo = \frac{\nu \Delta t}{\Delta x^2}, \tag{14}$$

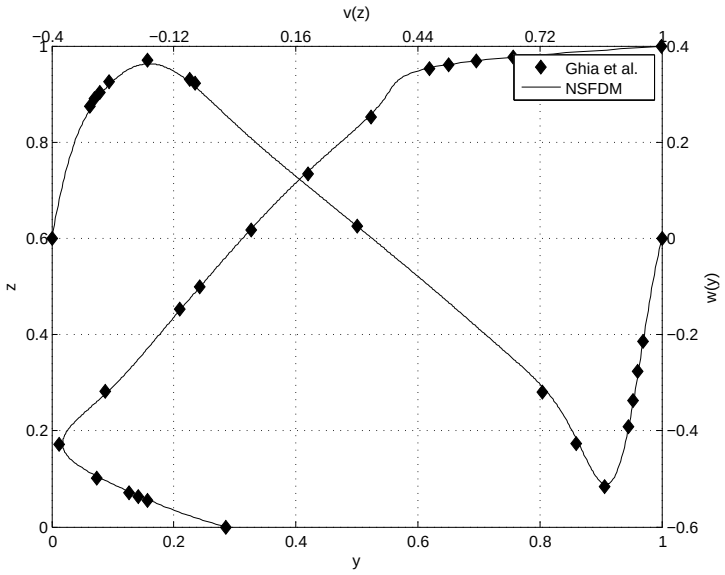

Fig. 3: Results obtained at $Re_{L_c} = 1000$ using a grid of 512×512.

Table 1: 2D lid driven cavity at $Re_{L_c} = 1000$. Performance, meassured in $[secs/Mcells]$, obtained by a GPGPU Nvidia GTX 580.

	Simple	Double
64×64	1.43	1.58
128×128	0.38	0.42
256×256	0.11	0.13
512×512	0.04	0.06

limiting the time step to a CFL (Courant-Friedrichs-Lewy condition) of ≈ 0.48, defining CFL as

$$CFL = \frac{u_{max} \Delta t}{\Delta_{min}}, \tag{15}$$

u_{max} being the maximum velocity magnitude and Δ_{min} the minimum grid size.

2.2 2D flow past circular cylinder

This classical external flow test is performed on a computational domain of $L \times H$ spatial units. The lenght L and height H of the computational domain are related to the diameter D of the cylinder by a relation close to $1:15$, this is, both L and H are 15 times bigger than D. This relation was chosen in order to minimize the adverse effects of boundary condititions on the computation of drag (C_d), lift (C_l) and Strouhal (St) coefficients.

The far field velocity $\mathbf{u}_\infty = (v_\infty, 0)$ is fixed on the inlet, outlet, top and bottom boundaries with $v_\infty = 1$. The kinematic viscosity is adjusted in order to reach a predefined Reynolds number (Re_D).

To compute drag, lift and Strouhal coefficients a momentum balance over a square region is used. As the approximation purposed assumes that the square control surface is very close to the (circular) body, the errors (forces) arised from the square surface can be neglected.

Furthermore, the body is represented as a staircase geometry. No-slip ($\mathbf{u} = \mathbf{0}$) and no-penetration ($\nabla p \cdot \hat{\mathbf{n}} = 0$) are imposed as boundary conditions on the cylinder.

The results and performance obtained at $Re = 1000$ are shown on Table 2 and Table 3 respectively.

Table 2: 2D flow past cylinder at $Re_D = 1000$.

	C_d	C_l	St
Present formulation	1.56	1.3	0.211
PFEM-2 [ING$^+$13]	1.639	1.63	0.2475
FEM [MK01]	1.48	1.36	0.21

Table 3: 2D flow past cylinder at $Re_D = 1000$. Performance, meassured in $[secs/Mcells]$, obtained by a GPGPU Nvidia GTX 580.

	Simple	Double
256×128	0.20	0.22
512×256	0.06	0.09
1024×512	0.03	0.05

In this case, no Fo constrain is encountered, so a $CFL \approx 4$ *to* 5 can be used.

2.3 3D lid-driven cavity

The numerical results obtained at $Re_{L_c} = 1000$ are shown on Figure 4, and the performance obtained is shown on Table 4.

As the main interest of the authors are 3D flows lets analyze the performance obtained. The same argument will hold for every study case of this article. As it can be seen in Table 4, for the case of $128^3 \approx 2$ $[MCells]$ the performance obtained is about 20 $[MCells/sec]$. With this data at hand it is known than $2/20 = 0.1$ $[secs/timestep]$, this is, 10 time steps per second can be performed. As the time step for this case is $\Delta t = 0.01$ $[secs]$ it can be seen that 0.1 $[secs]$ of simulation can be performed in 1 $[sec]$ of computation.

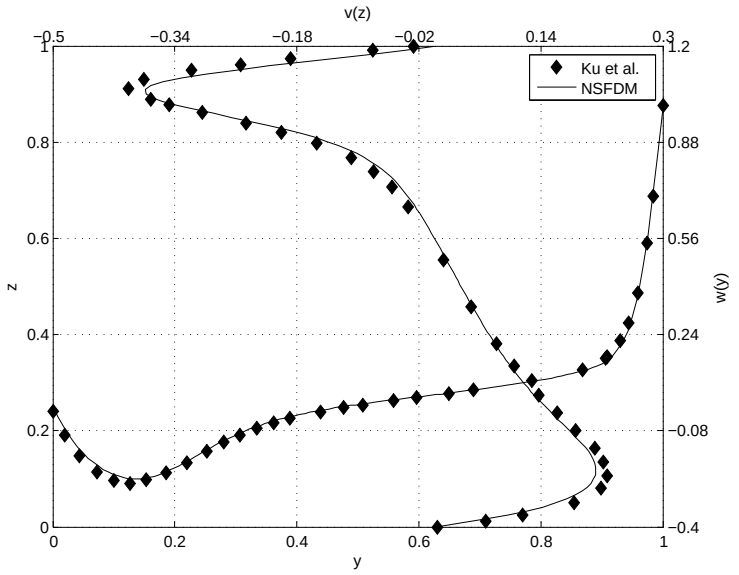

Fig. 4: Results obtained at $Re_{L_c} = 1000$ using a grid of $128 \times 128 \times 128$.

Table 4: 3D lid driven cavity at $Re_{L_c} = 1000$. Performance, meassured in $[secs/Mcells]$, obtained by a GPGPU Nvidia GTX 580.

	Simple	Double
$64 \times 64 \times 64$	0.08	0.19
$128 \times 128 \times 128$	0.05	0.13
$192 \times 192 \times 192$	0.05	0.13

Like the 2D case, the Fo is severely restricting the performance obtained.

2.4 3D flow past circular cylinder

As an extension of the 2D case, the cylinder is now suposed to be infinite at x dimension. In other words, periodic boundary conditions are going to be used in that direction.

The results and performance obtained at $Re_D = 1000$ are shown on Table 5 and Table 6 respectively. Lets do the same analysis of the previos study case. Considering now that no Fo constrain is imposed and $\Delta t = 0.023$ the results shown that 0.23 $[secs]$ of simulation can be performed in 1 $[sec]$ of computation.

Table 5: 3D flow past cylinder at $Re = 1000$.

	CD	CL	St
Experimental	1.00 [Ach72]		0.21 [Ros55]
Present formulation	1.021	0.533	0.183
PFEM-2 [ING$^+$13]	1.16	0.2 to 0.3	0.185
OpenFOAM [ING$^+$13]	1.22	0.5	0.195

Table 6: 3D flow past cylinder at $Re = 1000$. Performance, meassured in $[secs/Mcells]$, obtained by a GPGPU Nvidia GTX 580.

	Simple	Double
$64 \times 256 \times 256$	0.05	0.13

3 Implementation details

The whole Fractinal Step algorithm was implemented in CUDA [1], using the tools provided by Thrust [2] and Cusp [3] for linear algebra operations. The FFT used was that provided by CUDA, CUFFT [4] (for performance tests please see [SPD$^+$13]).

Lets divide the Fractional Sep method, roughly, in the main two steps, *Momentum* (computing the advection) and *Poisson* (solving for pressure). The porcentual amount of work carried by each step, using a previous advection scheme named QUICK [Leo79], is shown on Figure 5. This scheme was implemented using shared memory and other considerations in order to perform as high as possible (the authors considers the implementation of this scheme a fair comparison for BFECC).

It can be seen that, the Poisson equation for pressure is the most time consuming step when QUICK is used as advection scheme.

Lets consider now the same study but using in this case BFECC as advection scheme, the results are shown on Figure 6.

It can be concluded that, considering that BFECC can use CFL numbers as high as ten (or more) times than QUICK, then the computation of the Poisson step is reduced and, as a consequence, the distribution of work changes. Although the rate obtained by BFECC is slower than that of QUICK, BFECC becomes more efficient when $CFL > 2$ is used.

4 Conclusions

A CUDA implementation of the 3D viscous incompressible Navier-Stokes equations was presented and its accuracy and performance were obtained using two

[1] `https://developer.nvidia.com/what_cuda`
[2] `http://code.google.com/p/thrust/`
[3] `http://code.google.com/p/cusp_library/`
[4] `https://developer.nvidia.com/cufft`

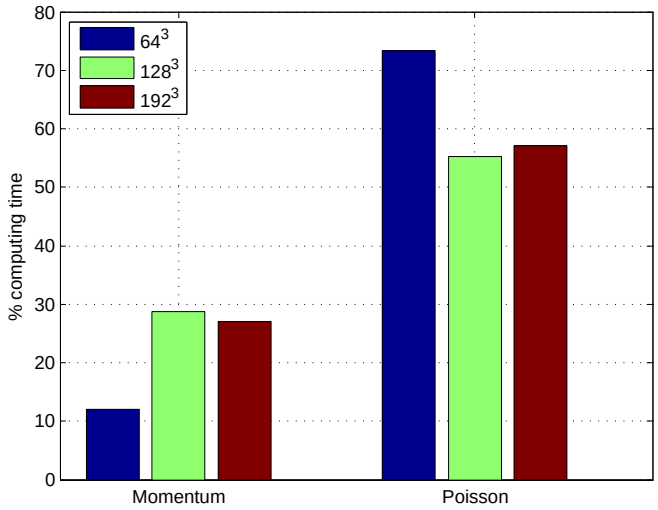

Fig. 5: Porcentual division of the two main computations. Advection scheme: QUICK.

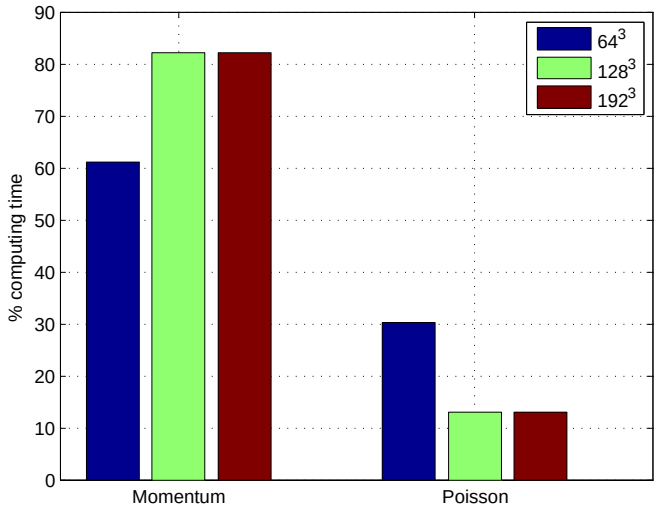

Fig. 6: Porcentual division of the two main computations. Advection scheme: BFECC.

well-known study cases. The results shown good agreement with the references and, when $CFL > 2$, BFECC performs better than the previous advection scheme, QUICK.

It must be recalled that, bodies are stair-case defined and refinements are being explored by the authors at the moment. Also, new ways of solving diffusion equations is being studied too.

References

Ach72. Elmar Achenbach. Experiments on the flow past spheres at very high reynolds numbers. *Journal of Fluid Mechanics*, 54(03):565–575, 1972.

CML80. AJ Chorin, JE Marsden, and A Leonard. A mathematical introduction to fluid mechanics. *Journal of Applied Mechanics*, 47:460, 1980.

ING+13. Sergio Rodolfo Idelsohn, Norberto Marcelo Nigro, Juan Marcelo Gimenez, Riccardo Rossi, and Julio Marti. A fast and accurate method to solve the incompressible navier-stokes equations. *Engineering Computations*, 30-Iss:2:197–222, 2013.

KLLR07. B.M. Kim, Y. Liu, I. Llamas, and J. Rossignac. Advections with significantly reduced dissipation and diffusion. *Visualization and Computer Graphics, IEEE Transactions on*, 13(1):135–144, 2007.

Leo79. Brian P Leonard. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer methods in applied mechanics and engineering*, 19(1):59–98, 1979.

MK01. S Mittal and V Kumar. Flow-induced vibrations of a light circular cylinder at reynolds numbers 10^3 to 10^4. *Journal of sound and vibration*, 245(5):923–946, 2001.

Ros55. Anatol Roshko. On the development of turbulent wakes from vortex streets naca rep. 1191, 25u. *S. Government Printing Office, Washington, DC*, 1955.

SC91. Andrew Staniforth and Jean Côté. Semi-lagrangian integration schemes for atmospheric models-a review. *Monthly Weather Review*, 119(9):2206–2223, 1991.

SFK+08. A. Selle, R. Fedkiw, B. Kim, Y. Liu, and J. Rossignac. An unconditionally stable maccormack method. *Journal of Scientific Computing*, 35(2):350–371, 2008.

SPD+13. Mario A Storti, Rodrigo R Paz, Lisandro D Dalcin, Santiago D Costarelli, and Sergio R Idelsohn. A fft preconditioning technique for the solution of incompressible flow on gpus. *Computers & Fluids*, 74:44–57, 2013.

Sta99. Jos Stam. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 121–128. ACM Press/Addison-Wesley Publishing Co., 1999.