Solving 3D viscous Navier-Stokes equations using CUDA

Santiago D. Costarelli[1] Mario A. Storti[1,2] Rodrigo R. Paz[1,2]
Lisandro D. Dalcin[1,2]

[1] CONICET/CIMEC

[2] Facultad de Ingeniería y Ciencias Hídricas,
Universidad Nacional del Litoral.

HPCLatam 2013
Mendoza, Argentina.

Motivation

The equations being solved are the classical 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.

The objective is

- to develop a real time CFD application.

QUICK's workload distribution

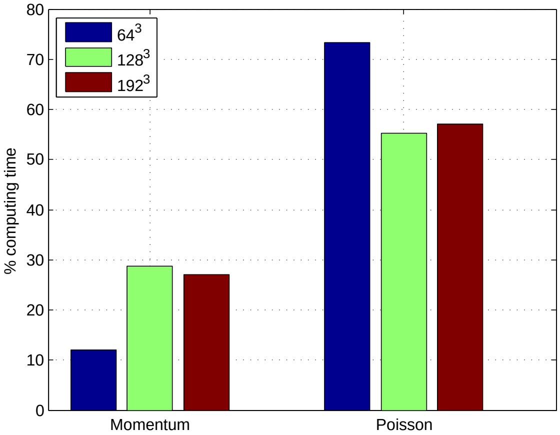

Figura 1: Porcentual workload of the two main computations. Advection scheme: QUICK.

QUICK's workload distribution (cont.)

The most important consideration that is involved with Figure 1 is that the Poisson step is the most time consuming step in the Fractional-Step algorithm used in the solution of Navier-Stokes equations.

In order to speed up the simulations one can choose between many different situations. Thus, one option is

- trying to perform the least amount of Poisson steps;
- but as QUICK needs to satisfy the *CFL* (Courant-Friedrichs-Lewy condition) constraint, some other scheme can be proposed in order to relax this drawback;
- so, the Method of Characteristic (MOC) is used as a solution.

Method of characteristics

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

$$\frac{D_{(m)}F}{Dt} = \frac{\partial F}{\partial t} + \mathbf{u} \cdot \nabla F = 0, \tag{3}$$

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

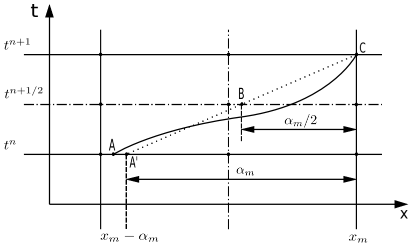

Figura 2: The point C can be obtained driving along the solid line ($\overline{AC}$), or approximately, using the velocity field as a first order predictor (dashed line $\overline{A'C}$).

Method of characteristics (cont.)

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. In fact, in the simulations performed *CFL*'s up to 5 are used.

BFECC method

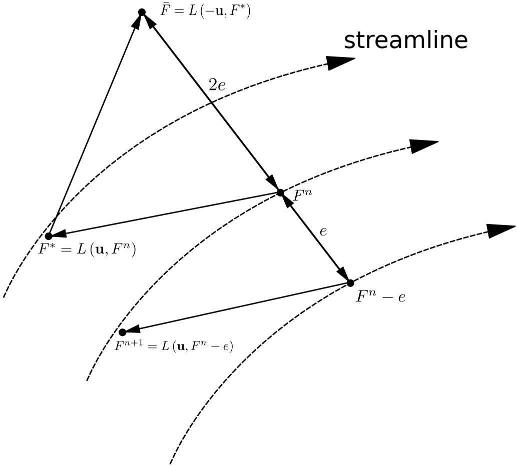

Figura 3: Schematic BFECC operation over a streamline field and using $L(.,.)$ as the advection operator for the scalar field F.

BFECC method (cont.)

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

$$F^{*} = L\left(\mathbf{u}, F^{n}\right) \tag{4}$$

$$\bar{F} = L\left(-\mathbf{u}, F^{*}\right) \tag{5}$$

$$F^{*} = F^{n} + \left(F^{n} - \bar{F}\right)/2 \tag{6}$$

$$F^{n+1} = L\left(\mathbf{u}, F^{*}\right) \tag{7}$$

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.

CUDA 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].

[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

MOC+BFECC's workload distribution

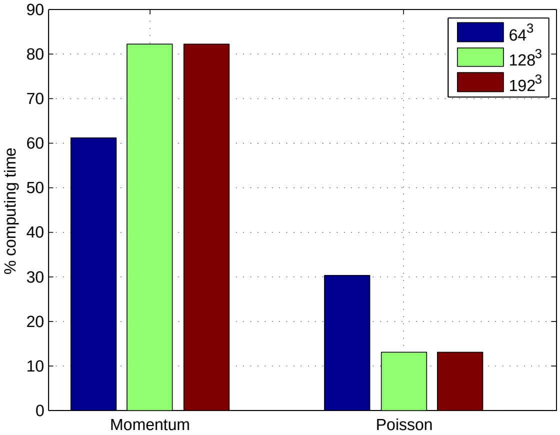

Figura 4: Porcentual workload of the two main computations. Advection scheme: MOC+BFECC.

2D study case: lid-driven cavity

This is a classical internal flow test in a square domain. The shear velocity imposed is fixed at $v = 1$ varying the kinematic viscosity in order to reach the specified Reynolds number, *Re*.

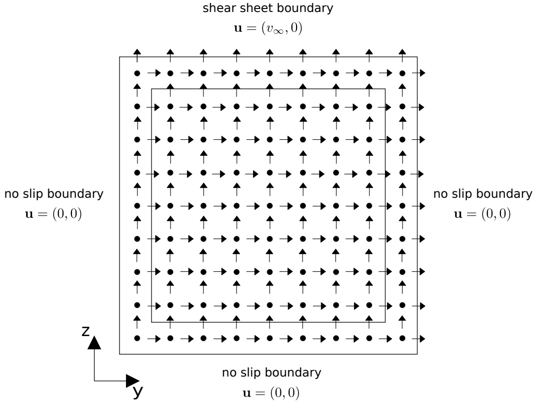

Figura 5: 2D lid-driven cavity configuration.

2D study case: lid-driven cavity

The numerical results obtained at $Re = 1000$ are shown on Figure 6.

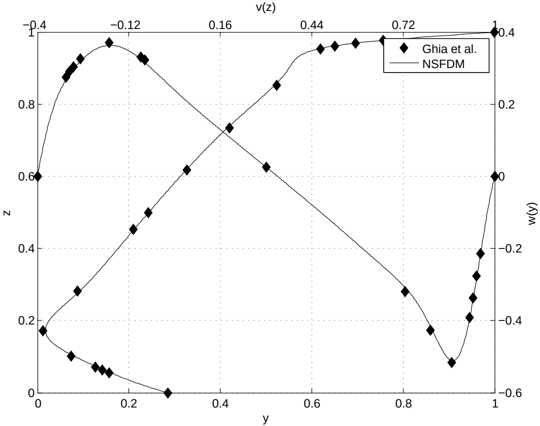

Figura 6: Results obtained at $Re = 1000$ using a grid of 512×512.

2D study case: lid-driven cavity (cont.)

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.

Cuadro 1: 2D lid driven cavity at $Re = 1000$. 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

The main drawback in this study case is the Fourier number, Fo, limiting the time step to a CFL of $\approx 0{,}48$.

2D study case: flow past circular cylinder

This classical external flow test. 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 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.

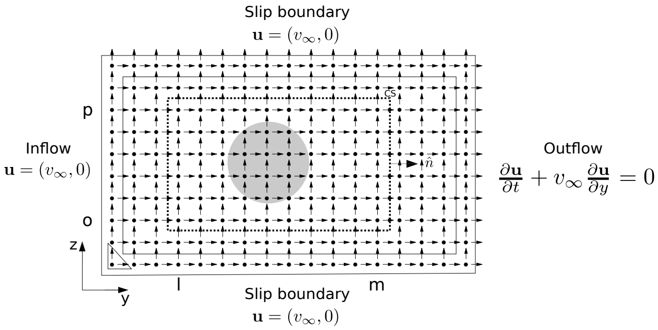

Figura 7: 2D flow past circular cylinder configuration.

2D study case: flow past circular cylinder (cont.)

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 obtained at $Re = 1000$ are shown on Table 2.

Cuadro 2: 2D flow past cylinder at $Re = 1000$.

	C_d	C_l	St
Present formulation	1.56	1.3	0.211
PFEM-2	1.639	1.63	0.2475
FEM	1.48	1.36	0.21

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

2D study case: flow past circular cylinder (cont.)

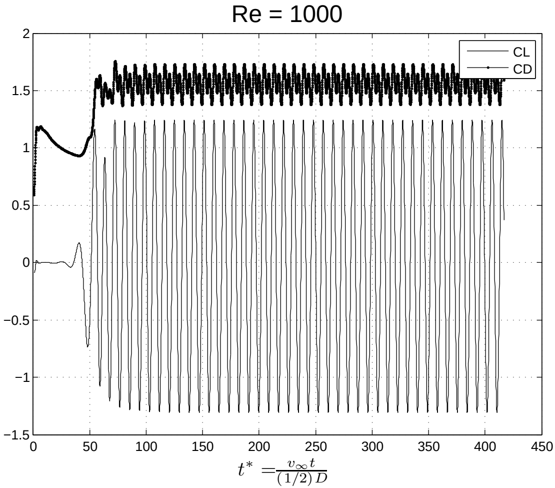

Figura 8: Time evolution of C_d and C_l.

3D study case: lid-driven cavity

The numerical results obtained at $Re = 1000$ is shown on Figure 9.

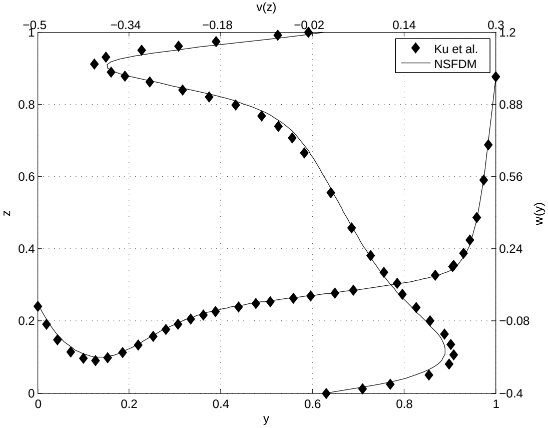

Figura 9: Results obtained at $Re = 1000$ using a grid of $128 \times 128 \times 128$.

3D study case: lid-driven cavity (cont.)

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.

Cuadro 3: 3D lid driven cavity at $Re = 1000$. 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.

3D study case: lid-driven cavity (cont.)

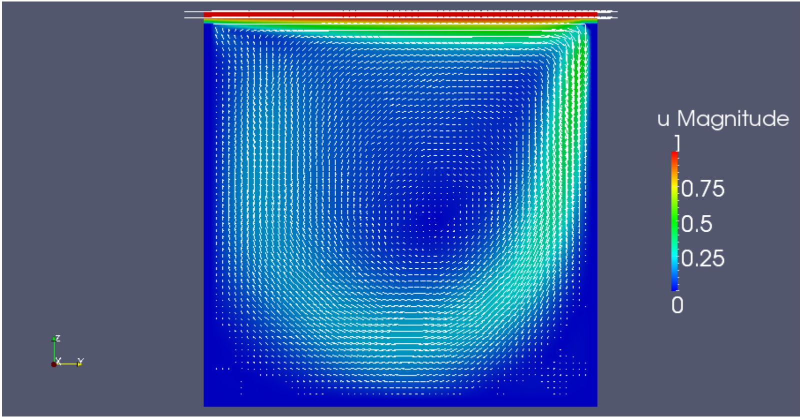

Figura 10: 3D lid-driven cavity.

3D study case: 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 obtained at $Re = 1000$ are shown on Table 4.

Cuadro 4: 3D flow past cylinder at $Re = 1000$.

	C_d	C_l	St
Experimental	1.00		0.21
Present formulation	1.021	0.533	0.183
PFEM-2	1.16	0.2 to 0.3	0.185
OpenFOAM	1.22	0.5	0.195

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.

3D study case: flow past circular cylinder (cont.)

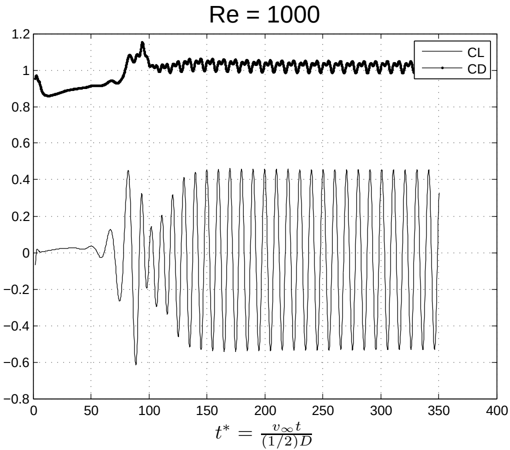

Figura 11: Time evolution of C_d and C_l.

3D study case: flow past circular cylinder (cont.)

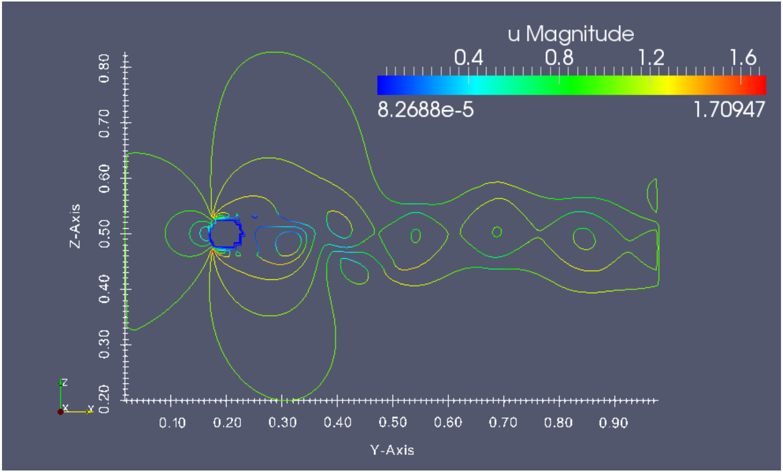

Figura 12: 3D flow past circular cylinder.

QUICK vs MOC+BFECC

Simple	[Mcell/sec]	[Mcell/sec]
Cells	QUICK	BFECC
64^3	29.09	12.38
128^3	75.74	18.00
192^3	78.32	17.81
Double	[Mcell/sec]	[Mcell/sec]
Cells	QUICK	BFECC
64^3	15.9	5.23
128^3	28.6	7.29
192^3	30.3	7.52

Cuadro 5: Computing rates for the whole NS solver (one step) in [Mcell/sec] obtained with the BFECC and QUICK algorithms on a NVIDIA GTX 580. 3 Poisson iterations were used.

QUICK vs MOC+BFECC (cont.)

As a reference, the QUICK algorithm was implemented in CPU obtaining a rate of 3.5 [Mcell/sec] (OpenMP) on an Intel i7-3820@3.47 GHz (Sandy Bridge microarchitecture) for large 3D meshes (above 1 Mcell), i.e. 8.6 times slower with respect to the GPU(QUICK) version. Note that this speedup obtained on the GPU is close to the 8x speedup factor obtained for the FFT. This is normal, because for the QUICK implementation a large part of the computing time is spent in the Poisson step.

The BFECC(GPU) is only 2.15 times faster than the QUICK(CPU) version in Mcells/sec, but taking into account that the CFL is 10 times larger, the overall speedup is 21.5, i.e. BFECC(GPU) is 21.5 times faster than QUICK(CPU) in computing one second of the same physical process.

Conclusions

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

Acknowledge

This work has received financial support of

- Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT, Argentina, grants PICT-1141/2007, PICT-0270/2008),
- Universidad Nacional del Litoral (UNL, Argentina, grants CAI+D 2009-65/334, CAI+D-2009-III-4-2) y
- European Research Council (ERC) Advanced Grant, Real Time Computational Mechanics Techniques for Multi-Fluid Problems (REALTIME, Reference: ERC-2009-AdG, Dir: Dr. Sergio Idelsohn).

Also we use some development tools under Free Software like GNU/Linux OS, GCC/G++ compilers, Octave, and *Open Source* software like VTK, among many others.