Accelerating incompressible fluid flow simulations on hybrid CPU/GPU systems

Yushan Wang ${ }^{1}$, Marc Baboulin ${ }^{1,2}$, Karl Rupp ${ }^{3,4}$,
Yann Fraigneau ${ }^{1,5}$, Olivier Le Maître ${ }^{1,5}$
${ }^{1}$ Université Paris-Sud, France
${ }^{2}$ INRIA, France
${ }^{3}$ Argonne National Laboratory, USA
${ }^{4}$ Vienna University of Technology, Austria
${ }^{5}$ LIMSI-CNRS, France

## Laboratoire de Recherche en Informatique (LRI)



The main research areas addressed by ParSys include high-performance computing, distributed algorithms, compilation and code optimization.

Founded more than 30 years ago, LRI has now over 260 members, including 105 faculty and staff and 90 Ph.D. students.


## Outline

$\square$ Introduction to Navier-Stokes equations

- Helmholtz-like equations
- Poisson equation
$\square$ Hybrid model and performance on a multicore + GPU architecture
$\square$ Conclusion and future work


## Navier-Stokes equations

$\square$ The incompressible NS equations are the fundamental bases of many CFD problems.


Le sommet du mont Rishiri, sur litie japonaise du même nom (flèche) perturbe la circulation des nuages. La turbulence qu'il provoque est visible sous forme d'une "allée tourbillonnaire de Von Karman". (NASA)


A million dollars in cash awaits anyone who can develop a rigorous mathematical model for how fluids flow.
-- Clay Mathematics Institute

## Incompressible Navier-Stokes equations

$\square\left\{\begin{aligned} \frac{\partial \vec{V}}{\partial t}+\nabla \cdot\left(\vec{V} \otimes \vec{V}^{T}\right) & =-\nabla P+\frac{1}{\operatorname{Re}} \Delta \vec{V} \\ \nabla \cdot \vec{V} & =0\end{aligned}\right.$
$\square$ Density is neglected because the problem is supposed to be with constant coefficient.
$\square$ Reynolds number Re indicates the fluid state. Larger Re demands finer mesh discretization.
$\square$ Non-linear convection term $\nabla \cdot\left(\vec{V} \otimes \vec{V}^{T}\right)$ can be simplified as $(\vec{V} \cdot \nabla) \vec{V}$ for incompressible fluid flow.

## Solving NS equations with a prediction-projection method

$\square$ Hodge-Helmholtz decomposition: $\vec{V}^{*}=\vec{V}_{d i v=0}+\nabla \phi$
$\square\left(\vec{V}^{n}, P^{n}\right)$


Poisson equation


Time increments

- Y. Wang, M. Baboulin, J. Dongarra, J. Falcou, Y. Fraigneau, O. Le Maître

A Parallel Solver for Incompressible Fluid Flows. ICCS 2013: 439-448.

## Solving Helmholtz-like equation

$\square\left(I-\frac{2 \Delta t}{3 \operatorname{Re}} \Delta\right)\left(\vec{V}_{i}^{*}-\vec{V}_{i}^{n}\right)=\vec{S}_{i} \quad i \in\{x, y, z\}$
$\square$ Alternating Direction Implicit method:

$$
(I-\alpha \Delta)=\left(I-\alpha \Delta_{x}\right)\left(I-\alpha \Delta_{y}\right)\left(I-\alpha \Delta_{z}\right)+o\left(\alpha^{2}\right)
$$

$\square$ System of 3 Helmholtz-like equations:

$$
\begin{cases}\left(I-\frac{2 \Delta t}{3 \operatorname{Re}} \Delta_{x}\right) T^{\prime} & =\vec{S}_{i} \\ \left(I-\frac{2 \Delta t}{3 \operatorname{Re}} \Delta_{y}\right) T^{\prime \prime} & =T^{\prime} \\ \left(I-\frac{2 \Delta t}{3 \operatorname{Re}} \Delta_{z}\right)\left(\vec{V}_{i}^{*}-\vec{V}_{i}^{n}\right) & =T^{\prime \prime}\end{cases}
$$

$\square$ Thomas algorithm
$\square$ Matrix transpose: reordering

## Solving Helmholtz-like equation

$\square \tilde{B} x=f$.
$\square \tilde{B}$ is a tridiagonal block matrix. $B=\left[a_{i}, b_{i}, c_{i}\right]_{i=1, \ldots, m}$
$\square$ For certain cases, $\tilde{B} x=f$ can by considered as a smaller system with multiple RHS.
$\square$ Two methods available:

- Thomas algorithm: Gaussian elimination without pivoting.
- Explicit inverse of $B \rightarrow x=B^{-1} f$.

$$
\begin{aligned}
& B_{i j}^{-1}=\left\{\begin{array}{cc}
(-1)^{i+j} c_{i} c_{i+1} \ldots c_{j-1} \theta_{i-1} \phi_{j+1} / \theta_{n}, & i>j, \\
\theta_{i-1} \phi_{i+1} / \theta_{n}, & i=j, \\
(-1)^{i+j} a_{j+1} a_{j+2} \ldots a_{i} \theta_{j-1} \phi_{i+1} / \theta_{n}, & i>j,
\end{array}\right. \\
& \theta_{0}=1, \quad \theta_{1}=b_{1}, \quad \theta_{i}=b_{i} \theta_{i-1}-c_{i-1} a_{i} \theta_{i-2,} \quad i=2, \ldots, m, \\
& \phi_{m+1}=1, \quad \phi_{m}=b_{m}, \quad \phi_{i}=b_{i} \phi_{i+1}-c_{i} a_{i+1} \phi_{i+2}, \quad i=m-1, \ldots, 1,
\end{aligned}
$$

## Solving Helmholtz-like equation



$$
\begin{aligned}
& \left\{\begin{array}{cl}
u(x)-\alpha \Delta u(x)=f(x), & x \in \Omega=(0,1)^{3}, \\
u(x)=0, & x \in \partial \Omega
\end{array}\right. \\
& f(x)=\left(1+3 \alpha \pi^{2}\right) u(x), \quad \alpha=10^{7}, \\
& u(x)=\sin \left(\pi x_{1}\right) \sin \left(\pi x_{2}\right) \sin \left(\pi x_{3}\right)
\end{aligned}
$$

Using the explicit inverse of B gains a factor of 4 over the Thomas algorithm, while the accuracy is the same.

However, the application of the explicit inverse is limited: only problems with no immerged body, and with same boundary conditions, etc.

## Solving Poisson equation

$\square \quad \Delta \phi=\frac{3}{2 \Delta t} \nabla \cdot \vec{V}^{*}=S$
$\square$ Partial diagonalization: $\quad \Delta=\Delta_{x}+\Delta_{y}+\Delta_{z}$

$$
\left.\begin{array}{rl}
\Delta_{x} & =Q_{x} \Lambda_{x} Q_{x}^{-1} \\
\Delta_{y} & =Q_{y} \Lambda_{y} Q_{y}^{-1} \\
S^{\prime} & =Q_{x}^{-1} Q_{y}^{-1} S \\
\phi^{\prime} & =Q_{x}^{-1} Q_{y}^{-1} \phi
\end{array}\right\} \longrightarrow\left(\Lambda_{x}+\Lambda_{y}+\Delta_{z}\right) \phi^{\prime}=S^{\prime}
$$

$\square$ Thomas algorithm
$\square$ Matrix transpose: reordering
$\square$ Most time-consuming part is the matrix-matrix multiplication.
$\square$ Using GPU to accelerate.

- MAGMA: Matrix Algebra on GPU and Multicore Architectures.


## CPU vs. GPU on Helmholtz and Poisson problems

| Problem size $=256^{3}$ | Helmholizz <br> (with $\mathrm{B}^{-1}$ ) | Poisson |
| :--- | :---: | :---: |
| Transfer CPU $\rightarrow$ GPU | 102 | 148 |
| Matrix multiplication | 261 | 116 |
| Solution reordering | 162 | 108 |
| Tridiagonal system solve | - | 185 |
| Total GPU solver | 423 | 409 |
| Total CPU solver* (12 MPI processes) | 1813 | 1700 |
| Acceleration | $\times 4.3$ | $\times 4.2$ |

2 Inter Xeon E5645 $\boldsymbol{\rightarrow} 12$ cores in total.
2 NVIDIA Tesla C2075.

## SUNFLUIDH

$\square$ Navier-Stokes solver developed at LIMSI (Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur)
$\square$ 3D simulation of unsteady incompressible flow or low Mach number flow.
$\square$ Forced convection flow

- Thermal convection flow
- Multispecies flow
- Reactive flow
$\square$ The base frame of our current work.
$\square$ More information on http://perso.limsi.fr/yann/.


## Hybrid model of our NS solver


$\square$ Domain is divided equally into subdomains.
$\square$ One subdomain corresponds to one MPI process.
$\square$ Each process is associated to one GPU acceleration.
$\square$ Multi-threading techniques are applied within each subdomain.

## Performance results



- Problem size $=128^{3}$.
- About $50 \%$ of the computational work is done by GPU.
- Multithreading is not yet fully developed.


## Conclusion and future work

$\square$ A hybrid multi-core GPU Navier-Stokes solver which includes the solution of the Helmholtz-like and Poisson equations.
$\square$ Significant acceleration by taking advantage of GPU devices.
$\square$ More computational work to be transferred on GPU.
$\square$ Construction of the tridiagonal systems.

- Computation of convection flux, diffusion flux, etc..
$\square$ Multi-threading implementation to be ameliorated.
$\square$ Using PETSc iterative solver when direct solver is not available.
$\square$ Larger scale simulations.

Thank Youl

