The Poisson equation in projection methods for incompressible flows

Bérengère Podvin, Yann Fraigneau
LIMSI-CNRS

Overview

- Origin of the Poisson equation for pressure
- Resolution methods for the Poisson equation
- Application to the code SUNFLUIDH (Y. Fraigneau, LIMSI)

Conservation Equations for Fluid Mechanics

- Mass $$\frac{\partial \rho}{\partial t} + \nabla.(\rho \underline{u}) = 0$$
- Momentum $$\frac{\partial (\rho \underline{u})}{\partial t} + \nabla.(\rho \underline{uu}) = -\nabla p + \nabla \underline{\underline{\tau}} + \rho \underline{f}$$
- Energy $$\frac{\partial (\rho e)}{\partial t} + \nabla.(\rho e \underline{u}) = -\nabla (p \underline{u}) + \nabla.(\underline{\underline{\tau}}.\underline{u}) + \rho \underline{f}.\underline{u} - \nabla.\dot{q} + r$$

Density ρ

Velocity u

Total energy e

Pressure p

Stress tensor
Newtonian fluid

$$\underline{\underline{\tau}} = \mu[\frac{1}{2}(\nabla \underline{u} + \nabla \underline{u}^T) - (\nabla.\underline{u})\underline{\underline{I}})$$

Heat flux

$$\dot{q} = -k\nabla T$$

\+ equations of state to close the problem

What is an incompressible flow?

Incompressibility: The density of a fluid particle does not change over time

Mass conservation

Incompressibility

$$\left.\begin{aligned}&\frac{\partial\rho}{\partial t}+\nabla.(\rho\underline{u})=0\\&\frac{d\rho}{dt}=\frac{\partial\rho}{\partial t}+\underline{u}.\nabla\rho=0\end{aligned}\right\}\boxed{\Rightarrow\nabla.\underline{u}=0}$$

Characterizes flow
(not fluid)

When can the flow be assumed to be incompressible?

The rate of change of the density of a fluid particle is very small compared to the inverse of the other time scales of the flow

$$\frac{u}{l}<<\frac{1}{\tau_{a\cos tic}}$$ Acoustic time scale small

$$\left(\frac{u}{c}\right)^2<<1$$ Mach number small

The incompressible flow

- Assuming constant-density, Newtonian fluid

$$\nabla.\underline{u} = 0$$

$$\frac{\partial \underline{u}}{\partial t} + \nabla.(\underline{uu}) = -\nabla p + \Delta \underline{u} + \rho \underline{f}$$

- Equations for u are elliptic in space: boundary conditions need to be known over entire physical boundary.

- Pressure has no thermodynamic significance. It is directly related to the zero divergence constraint. There is no boundary conditions for pressure (in general).

How do we solve for pressure?

- Take divergence of equation

$$\frac{\partial \underline{u}}{\partial t} + \underline{u}.\nabla.\underline{u} = -\nabla p + \Delta \underline{u} + \underline{f}$$

$$\Rightarrow \nabla.(\underline{f} - \underline{u}.\nabla.\underline{u}) = \Delta p \quad (\nabla.\underline{u} = 0)$$

- ➔Elliptic equation requires boundary conditions
- ➔ Pressure is defined in a “smaller” space than velocity (dual of velocity divergence): !spurious pressure modes in kernel of discrete gradient!

The direct approach (Uzawa)

All operators are discretized (including boundary conditions)

$$\left.\begin{array}{l} Hu+\nabla p = S \\ \nabla . u = 0 \end{array}\right\} \Rightarrow \left.\begin{array}{l} u = H^{-1}(S-\nabla p) \\ \nabla . H^{-1}\nabla p = \nabla . H^{-1} S \end{array}\right\} \Rightarrow \left.\begin{array}{l} p = \overset{\text{Uzawa operator}}{\boxed{(\nabla . H^{-1}\nabla)^{-1}}}(\nabla . H^{-1} S) \\ u = H^{-1}(S-\nabla p) \end{array}\right\}$$

In some cases, the discretized Poisson operator can be inverted, leading to a simultaneous update of the velocity and the pressure field.

Spectral or finite-element approach, mostly used in 2D.

The projection method (I)

- Separate (fractional) updates of velocity and pressure

$$\nabla . \underline{u} = 0$$

$$\frac{\partial \underline{u}}{\partial t} + \nabla p - \Delta \underline{u} = \underline{f}$$

$$\underline{u}_{bc} = \underline{\gamma}$$

Original equations

$$\frac{(1+\varepsilon)\tilde{\underline{u}}^{n+1} - 2\varepsilon \underline{u}^{n} - (1-\varepsilon)\underline{u}^{n-1}}{\Delta t} + \lambda_1 \nabla p^{n} - \Delta(\theta \underline{u}^{n+1} + (1-\theta)\underline{u}^{n}) = \theta \underline{f}^{n+1} + (1-\theta)\underline{f}^{n}$$

$$\tilde{\underline{u}}^{n+1} = \underline{\gamma}$$

1st step: prediction

$$(1+\varepsilon)\frac{\underline{u}^{n+1} - \tilde{\underline{u}}^{n+1}}{2\Delta t} + \lambda_2 \nabla p^{n+1} + \lambda_3 \nabla p^{n} = 0$$

2nd step: projection or pressure correction

$$\nabla . \underline{u}^{n+1} = 0$$

$$\underline{u}_{bc}{}^{n+1} . \underline{n} = \underline{\gamma}\, \underline{n} \quad \Rightarrow \nabla p_{bc}{}^{n+1} = 0$$

Neuman boundary conditions for pressure

The projection method (II)

- Combined expression

$$\frac{(1+\varepsilon)\underline{u}^{n+1} - 2\varepsilon\underline{u}^{n} - (1-\varepsilon)\underline{u}^{n-1}}{\Delta t} + \lambda_2\nabla p^{n+1} + (\lambda_1+\lambda_3)\nabla p^{n}$$

$$-\frac{2\theta\Delta t}{1+\varepsilon}\Delta(\lambda_2\nabla p^{n+1} + \lambda_3\nabla p^{n}) - \Delta(\theta\underline{u}^{n+1} + (1-\theta)\underline{u}^{n}) = \theta\underline{f}^{n+1} + (1-\theta)\underline{f}^{n}$$

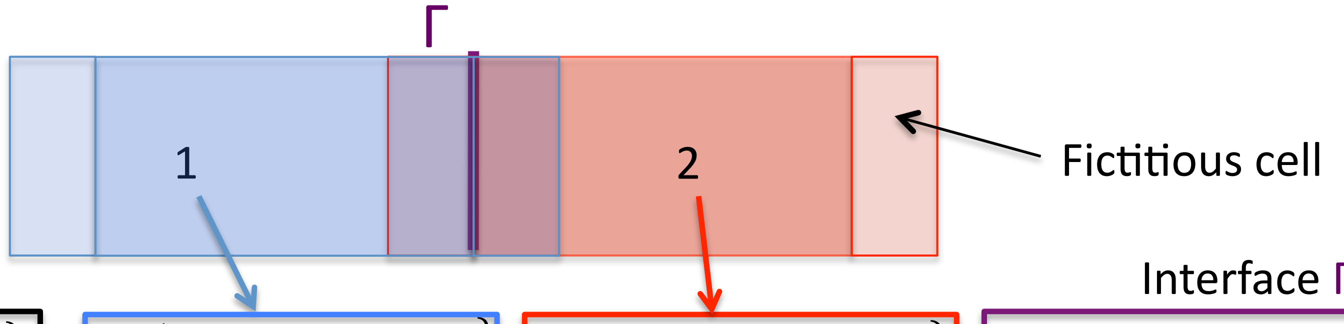
$$\boxed{\lambda_1+\lambda_2+\lambda_3 = 1} \rightarrow O(\Delta t)$$

$$\frac{\varepsilon}{2} = \theta = 1-(\lambda_1+\lambda_3)$$

$$\frac{\theta}{1+\varepsilon}(\lambda_2+\lambda_3) = 0$$

$$\rightarrow O(\Delta t^2)$$

Domain decomposition



Interface Γ

$$\left.\begin{array}{l}\nabla.\underline{u}=0\\ \frac{\partial \underline{u}}{\partial t}+\nabla p-\Delta \underline{u}=\underline{f}\\ \underline{u}_{bc}=\underline{\gamma}\end{array}\right\}=\left.\begin{array}{l}\nabla.\underline{u}^1=0\\ \frac{\partial \underline{u}^1}{\partial t}+\nabla p^1-\Delta \underline{u}^1=\underline{f}\\ \underline{u}^1{}_{bc}=\underline{\gamma}\end{array}\right\}+\left.\begin{array}{l}\nabla.\underline{u}^2=0\\ \frac{\partial \underline{u}^2}{\partial t}+\nabla p^2-\Delta \underline{u}^2=\underline{f}\\ \underline{u}^2{}_{bc}=\underline{\gamma}\end{array}\right\}+\left.\begin{array}{l}\underline{u}_\Gamma^1=\underline{u}_\Gamma^2\\ \frac{\partial \underline{u}_\Gamma^1}{\partial n}-p^1\underline{n}=\frac{\partial \underline{u}_\Gamma^2}{\partial n}-p^2\underline{n}\end{array}\right\}$$

- Schur complement: X=(u,p)

$$\left.\begin{array}{l}A_{11}X_1+A_{1\Gamma}X_\Gamma=S_1\\ A_{22}X_2+A_{2\Gamma}X_\Gamma=S_2\\ A_{\Gamma 1}X_1+A_{\Gamma 2}X_2+A_{\Gamma\Gamma}X_\Gamma=S_\Gamma\end{array}\right\}\Leftrightarrow\left.\begin{array}{l}X_1=A_{11}^{-1}\left(S_1-A_{1\Gamma}X_\Gamma\right)\\ X_2=A_{22}^{-1}\left(S_2-A_{2\Gamma}X_\Gamma\right)\\ \left(A_{\Gamma\Gamma}-A_{\Gamma 1}A_{11}^{-1}A_{1\Gamma}-A_{\Gamma 2}A_{22}^{-1}A_{2\Gamma}\right)X_\Gamma=S_\Gamma-A_{\Gamma 1}A_{11}^{-1}S_1-A_{\Gamma 2}A_{22}^{-1}S_2\end{array}\right\}$$

Resolution of Navier-Stokes equations in SUNFLUIDH

- Incompressible flow (constant-property fluid, ν=1)

$$\frac{\partial \vec{\mathbf{V}}}{\partial \mathbf{t}} + \nabla.({}^{t}\vec{\mathbf{V}} \otimes \vec{\mathbf{V}}) = -\nabla \mathbf{P} + \nu \nabla^{2} \vec{\mathbf{V}}$$

- Second-order accurate temporal discretization ➔ BDF2

$$\frac{\partial \vec{\mathbf{V}}}{\partial \mathbf{t}} \equiv \frac{3\vec{\mathbf{V}}^{n+1} - 4\vec{\mathbf{V}}^{n} + \vec{\mathbf{V}}^{n-1}}{2\Delta \mathbf{t}}$$

- Spatial Discretization
  - Finite-volume approach
  - Second-order centered scheme
- Viscous terms treated implicitly for stability reasons

Application of the prediction-projection method

- **Incompressible, constant-property , fluid**

$$\frac{\partial \vec{V}}{\partial t} + \nabla.({}^{t}\vec{V} \otimes \vec{V}) = -\nabla P + \nabla^2 \vec{V}$$

$$\nabla.\vec{V} = 0$$

- **Incremental prediction –projection method**

Prediction : Resolution N-S eqs

$$\frac{3V_i^* - 4V_i^n + V_i^{n-1}}{2\Delta t} + \frac{\partial.(V_i^n.V_j^n)}{\partial x_j} = -\nabla P^n + \nabla^2 V_i^*$$

$$\nabla.\vec{\mathbf{V}}^* \neq \mathbf{0}$$

Projection : Resolution Poisson

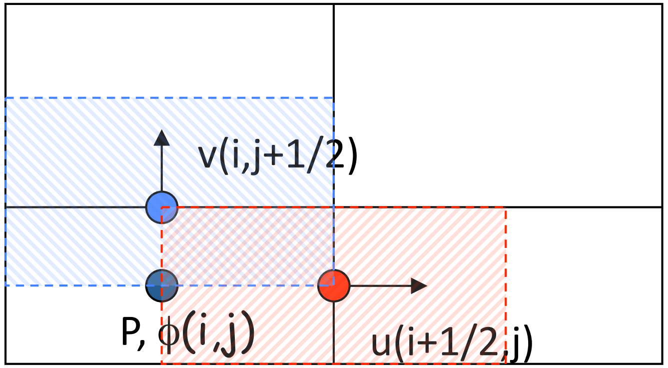
$$\nabla^2 \phi = \frac{\nabla.\vec{V}^*}{\Delta t}$$

Update V et P

$$P^{n+1} = P^n + \frac{3}{2}\phi + \nabla.\vec{V}^*$$

$$\vec{V}^{n+1} = \vec{V}^* - \Delta t \nabla \phi \qquad \Longrightarrow \qquad \nabla.\vec{\mathbf{V}}^{\mathbf{n+1}} = \mathbf{0}$$

Staggered mesh (MAC scheme)



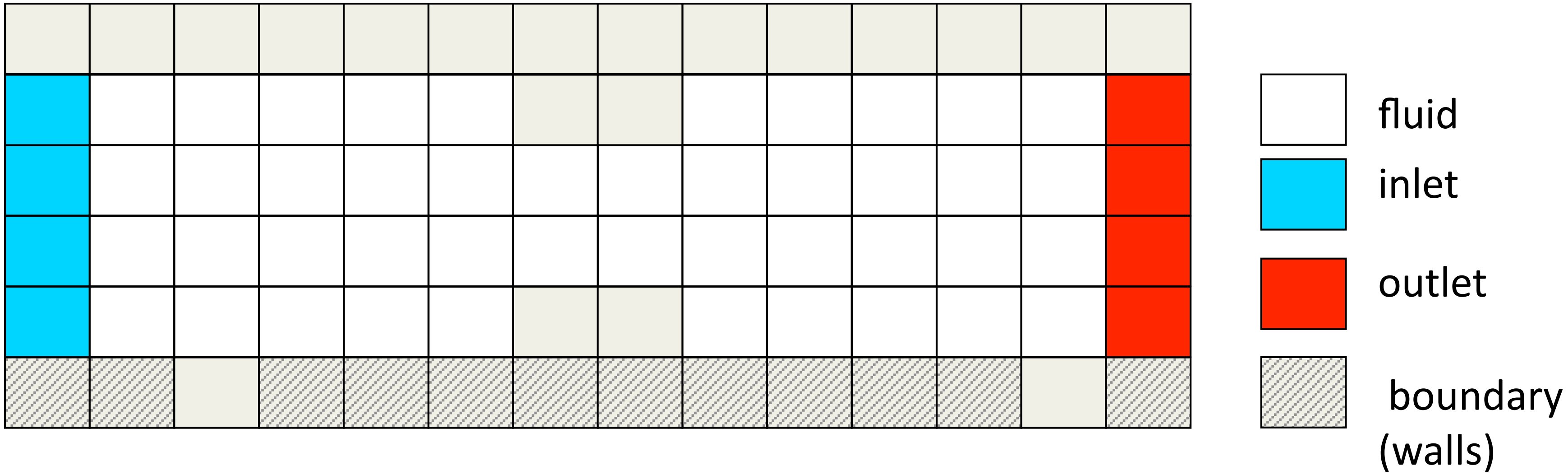
Define metric:

$$\boxed{\Delta X_{i+1/2} = X_{i+1} - X_i} \quad \Delta X_i = X_{i+1/2} - X_{i-1/2}$$

$$\boxed{\Delta Y_{j+1/2} = Y_{j+1} - Y_j} \quad \Delta Y_j = Y_{j+1/2} - Y_{j-1/2}$$

- **Structured staggered meshes with a Cartesian topology**
  - Center of cells I (i,j) : definition of scalar(P,ϕ, T)
  - Staggered meshes (i+1/2,j) et (i,j+1/2) en 2D: definition of the velocity compoments
- Consistence in the definition of the different discrete (2nd-order) operators
  - ➔ div(grad) = laplacian
  - Discretization of the Poisson equation:
    - Avoid generation of spurious pressure modes
    - Div(V) = 0 is enfocred without roundoff error
- Need to define operators in the center and on the faces of the cells
  - Discretization of operators depends on the variable (irregular mesh)

Computational Domain



- **Computation Domain defined with respect to main cartesian mesh (i,j)**
  - Fluid Domain
  - Fictitious cells around the fluid domain
    - Handling boundary conditions on the frontier of the domain
  - Definition of a phase function
    - Localization of the immersed boundary (masks on mesh)
    - Definition of the boundary conditions for the simulation

Prediction step (I)

- Resolution of Navier-Stokes equations➔ V* velocity field at t$_{n+1}$
  - Second-order time and space discretization of equations
  - Viscous terms treated implicity (stability wrt time step)

$$\frac{3V_i^* - 4V_i^n + V_i^{n-1}}{2.\Delta t} + \left(2\frac{\partial.(V_i^n.V_j^n)}{\partial x_j} - \frac{\partial.(V_i^{n-1}.V_j^{n-1})}{\partial x_j}\right) = -\nabla P^n + \nabla^2 V_i^*$$

➔Helmholtz Equations

$$(\mathrm{I} - \frac{2\Delta t}{3}\nabla^2)(V_i^* - V_i^n) = S$$

- Resolution using ADI (Alternating Direction Implicit method)

$$(\mathrm{I} - \frac{2\Delta t}{3}\nabla^2) \approx (\mathrm{I} - \frac{2\Delta t}{3}\nabla_x{}^2)(\mathrm{I} - \frac{2\Delta t}{3}\nabla_y{}^2)(\mathrm{I} - \frac{2\Delta t}{3}\nabla_z{}^2) + O\left(\Delta t^2\right)$$

System 3D ➔ 3 tridiagonal 1-D systems (direct resolution using Thomas algorithm)

Prediction step (II)

- Approximation of the 3D operator using 1-D operators

$$(I-\frac{2\Delta t}{3}\nu\nabla^{2}) \approx (I-\frac{2\Delta t}{3}\nu\nabla_{x}^{2})(I-\frac{2\Delta t}{3}\nu\nabla_{y}^{2})(I-\frac{2\Delta t}{3}\nu\nabla_{z}^{2})$$

➔ **Successive resolution of 3 1-D systems**

$$(I-\frac{2\Delta t}{3}\nu\nabla_{x}^{2})V_{1} = S$$

$$(I-\frac{2\Delta t}{3}\nu\nabla_{y}^{2})V_{2} = V_{1}$$

$$(I-\frac{2\Delta t}{3}\nu\nabla_{z}^{2})(V_{i}^{*} - V_{i}^{n}) = V_{2}$$

Spatial discretization of viscous fluxes ➔ Tridiagonal systems

Projection Step: Poisson equation (I)

- **Direct Method: Partial diagonalization of Laplacian L**
  - Principle

$$L\phi = \nabla.V^* = S \Leftrightarrow \left(L_x + L_y + L_z\right)\phi = S \Leftrightarrow \left(\Lambda_x + \Lambda_y + L_z\right)\phi' = S'$$

$$\Lambda_x = P_x^{-1}L_xP_x \text{ (diag. w.r.t x) ; } \Lambda_y = P_y^{-1}L_yP_y \text{ (diag. w.r.t y)}$$

$$S' = P_x^{-1}P_y^{-1}\mathrm{S} \text{ ; } \phi' = P_x^{-1}P_y^{-1}\phi$$

  - Solving for ϕ :
    - Project source term onto the eigenspaces of operators in X and Y
    - Integrate 1-D Helmholtz-like equation (tridiagonal system)
      - Use Thomas algorithm for direct solution
    - Inverse Projection ϕ’ ➔ ϕ
  - Need problem to be separable i.e:
    - Homogeneous boundary conditions
    - Discretization of operator can be done in one direction, independently from other directions (tensorization possible)

Parallelization on distributed memory architecture: Resolution of tridiagonal systems with direct methods

- Parallelization on distributed memory architecture:
  - Domain decomposition method: each processor corresponds to 1 sub-domain

- Direct methods for domain decomposition
  - Resolution of tridiagonal systems (Helmholtz 1D)

➔ **Schur complement method:**

Consider the tridiagonal system $$\mathbf{AX} = \mathbf{S} \Leftrightarrow \begin{pmatrix} \mathbf{A_k} & \mathbf{A_{kI}} \\ \mathbf{A_{Ik}} & \mathbf{A_I} \end{pmatrix} \begin{pmatrix} \mathbf{X_k} \\ \mathbf{X_I} \end{pmatrix} = \begin{pmatrix} \mathbf{S_k} \\ \mathbf{S_I} \end{pmatrix}$$

1. Solve $\mathbf{A_k X_{k,o} = S_k}$
2. Compute X_{iu} using Schur complement (tridiagonal system)

$$\left(\mathbf{A_I} - \mathbf{A_{Ik} A_k^{-1} A_{kI}}\right)\mathbf{X_I} = \left(\mathbf{S_I} - \mathbf{A_{Ik} X_{k,0}}\right)$$

3. Solve on each proc. $\mathbf{A_k X_k = S_k - A_{kI} X_I}$; $\mathbf{X_I}$ conditions at interfaces

➔ At each step direct resolution (Thomas algorithm)

Projection Step: Poisson equation (II)

- **Iterative Method: Relaxed Gauss-Seidel (SOR) + Multigrid**
  - SOR algorithm applied to the system $\mathrm{LX} = \mathrm{S}$
    - At k-th iteration

$$\mathbf{x}_{\mathbf{i}}^{(\mathbf{k}+1)} = \omega.\frac{\mathbf{1}}{\mathbf{L}_{\mathbf{ii}}}\left(\mathbf{S}_{\mathbf{i}} - \sum_{\mathbf{j}=1}^{\mathbf{i}-1}\mathbf{L}_{\mathbf{ij}}\mathbf{x}_{\mathbf{j}}^{(\mathbf{k}+1)} - \sum_{\mathbf{j}=\mathbf{i}+1}^{\mathbf{n}}\mathbf{L}_{\mathbf{ij}}\mathbf{x}_{\mathbf{j}}^{(\mathbf{k})}\right) + (\mathbf{1}-\omega).\mathbf{x}_{\mathbf{i}}^{(\mathbf{k})} \ ; \mathbf{0} \leq \omega \leq \mathbf{2}$$

$$\forall \mathbf{i}, \mathbf{i} = \mathbf{1}, \mathbf{2} \ldots, \mathbf{n}$$

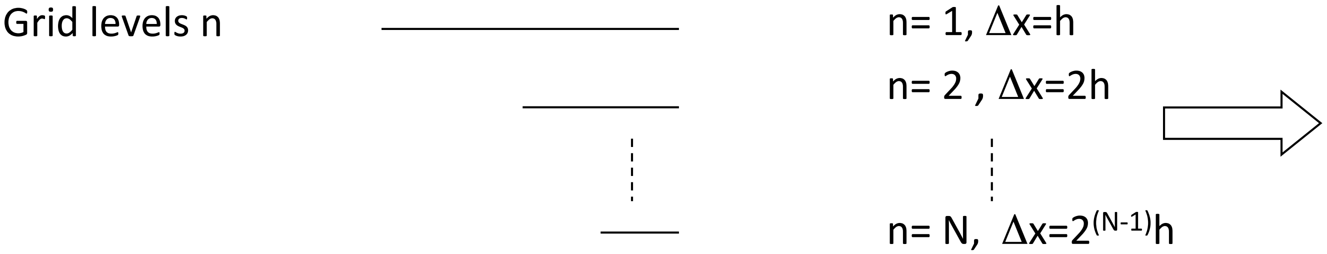
    - Convergence obtained at k-th iteration if $\| \mathrm{LX}^{k+1} - \mathrm{S} \| < \varepsilon$
  - Domain decomposition:
    - Boundary conditions handled using overlaps between subdomains
    - Boundary conditions update at each iteration

Projection Step: Poisson equation (III)

- Multigrid method: motivation and principle
  - **SOR**

    Slow convergence of low-frequency eror
  - **SOR coupled with a V-cycle multigrid method**

    Increase convergence
  - **Multigrid method**
    - N grid levels – with factor of 2 between grid resolutions
    - On each grid level n > 1
      - Error estimated on level n-1
      - Low-frequency error decreases faster over coarser grids



Grid levels n

n= 1, $\Delta x=h$

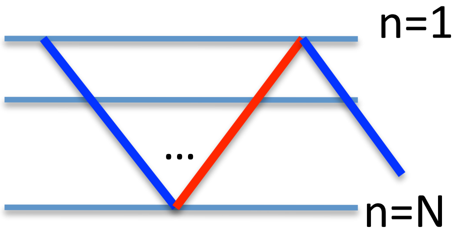
n= 2 , $\Delta x=2h$

n= N, $\Delta x=2^{(N-1)}h$

$$\mathbf{X}^{(n)} \approx \mathbf{X}^{(n-1)}_{\text{exact}} - \mathbf{X}^{(n-1)}_{\text{calculé}}$$

SOR Algorithm + Multigrid

- Resolution procedure for nV-cycle multigrid
  - **Restricting from the finest grid (n=1) to the coarsest grid (n= N)**
    - Solve $LX^{(1)}=S$ on fine grid (n=1) => Approximate solution $X^{(1)}$
    - Compute residue $R^{(1)}= S- LX^{(1)}$
    - Restrict residue to coarser grid : $R^{(2)}= f(R^{(1)})$
    - Solve on grid n=2 $LX^{(2)}=R^{(2)}$ …. Repeat procedure down to n=N
    - For n > 1 , each solution $X^{(n)}$ ➔ error estimate on the solution $X^{(n-1)}$



  - **Interpolating from coarsest grid to the finest grids**
    - Extension : Estimate error $E^{(n-1)}$ from $X^{(n)}$: $E^{(n-1)}= f(X^{(n)})$
    - Estimate corrected solution on the grid at level n-1 : $X^{(n-1)}=X^{(n-1)}+E^{(n-1)}$
    - Eliminate errors by extending correction on grid n-1
      - Additional iterations of the SOR algorithm from the initial $X^{(n-1)}$ $LX^{(n-1)}= R^{(n-1)}$; Si n= 1 $LX^{(1)}= S$
    - Recursive procedure up to n= 1.
  - **If the convergence criterion ($||R^{(1)}|| < \varepsilon$) is not satisfied ➔New cycle**

Discretization of the Poisson equation

- **Spatial discretization: 2nd order centered scheme**
  - **2D Laplacian**

$$\text{In x :}\quad \nabla_{\mathrm{x}}^2\phi \equiv \alpha\phi_{i+1,j} - (\alpha+\beta)\phi_{i,j} + \beta\phi_{i-1,j} \text{ avec } \alpha = \frac{\nu}{\Delta X_{i+1/2}.\Delta X_i} \text{ et } \beta = \frac{\nu}{\Delta X_{i-1/2}.\Delta X_i}$$

$$\text{In y :}\quad \nabla_{\mathrm{y}}^2\phi \equiv \alpha\phi_{i,j+1} - (\alpha+\beta)\phi_{i,j} + \beta\phi_{i,j-1} \text{ avec } \alpha = \frac{\nu}{\Delta Y_{j+1/2}.\Delta Y_j} \text{ et } \beta = \frac{\nu}{\Delta Y_{ji-1/2}.\Delta Y_j}$$

  - **Source term : div(V*)**

$$\nabla.\vec{\mathbf{V}}^* \equiv \frac{(\mathbf{u}_{\mathbf{i+1/2,j}} - \mathbf{u}_{\mathbf{i-1/2,j}})}{\Delta\mathbf{X}_{\mathbf{i}}} + \frac{(\mathbf{v}_{\mathbf{i,j+1/2}} - \mathbf{v}_{\mathbf{i,j-1/2}})}{\Delta\mathbf{Y}_{\mathbf{j}}}$$

- **Pressure Gradient (2D)**

$$\text{In x :}\quad \frac{\partial P}{\partial x}_{i+1/2,j} \equiv \frac{P_{i+1,j} - P_{i,j}}{\Delta X_{i+1/2}}$$

$$\text{In y :}\quad \frac{\partial P}{\partial y}_{i,j+1/2} \equiv \frac{P_{i,j+1} - P_{i,j}}{\Delta Y_{i+1/2}}$$

Staggered mesh: Pressure gradient is defined at same location as velocity components

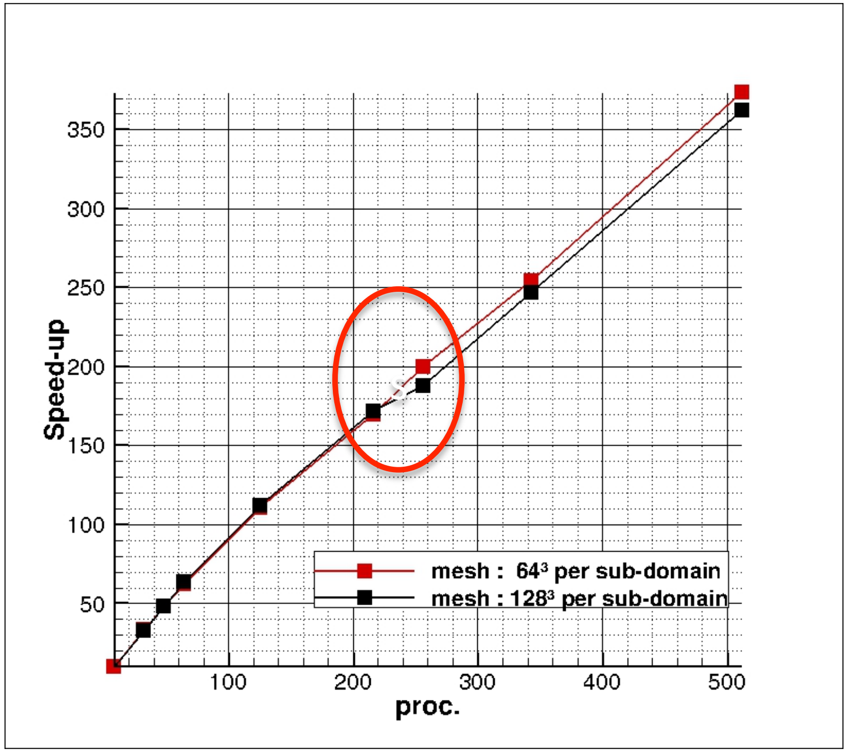
Essentials of SUNFLUIDH

- **Incompressible flows or Low Mach approximation**
  - Projection method for the resolution of Navier-Stokes
    - Second-order accuracy in time and space: Finite-Volume approach with MAC scheme
    - Viscous terms treated implicitly => Helmholtz-type system solved with ADI method
    - Projection method => Resolution of a Poisson equation for pressure
      - Direct method: Partial diagonalization of Laplacian (separable problem)
      - Iterative method: SOR + Multigrid (Next step: HYPRE)

- **Parallelization**
  - Multithreading « fine grain » (OpenMP)
  - Domain decomposition (MPI implementation)
  - Domain Hybrid parallelization (ongoing work)
    - MPI+ GPU : thèse LRI/LIMSI (project Digitéo CALIPHA)

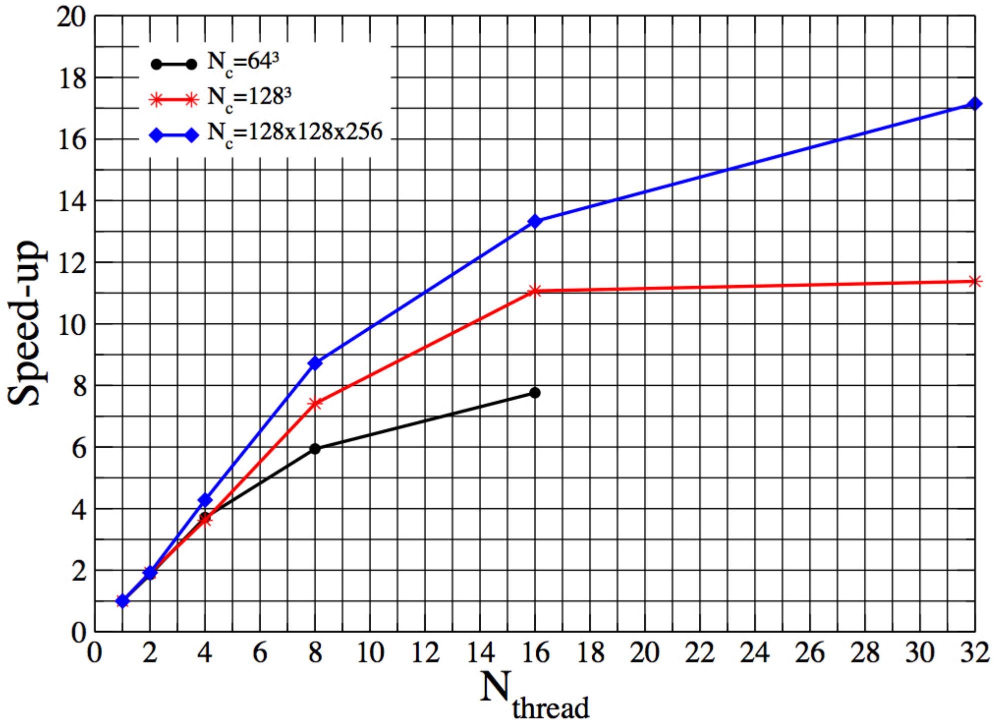
Code performance (IBM SP6 - IDRIS)

- Parallelization MPI



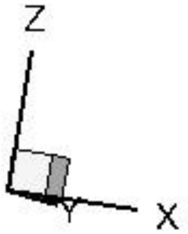
Bottleneck: communication between processors

- Parallelization OpenMP

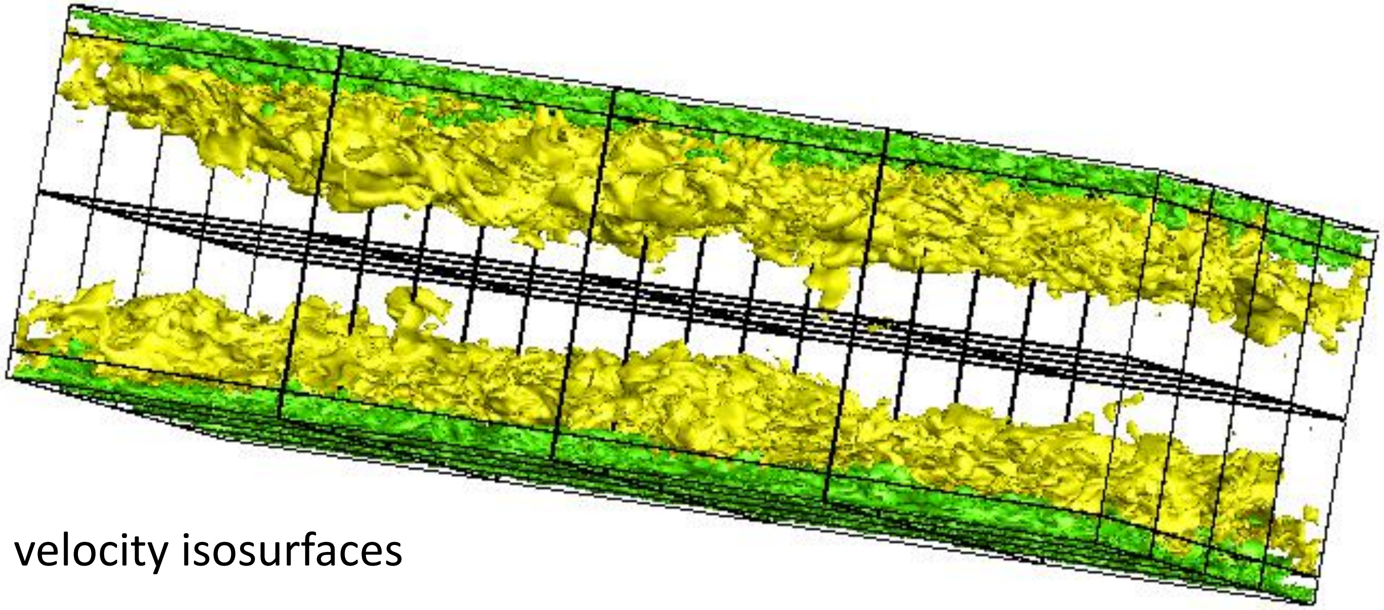


Example: Channel flow

$$R_\tau = \frac{u_\tau h}{\nu} = 980$$



$(512)^3$ cells
64 processors: 4x4x4
Cost: 6.0 10^{-8} s/step/node
Fraction spent in Poisson resolution: 30-60%



Streamwise velocity isosurfaces
u=10, u=15