A Godunov-Type Solver for the Numerical Approximation of Gravitational Flows

Edouard Audit, Micolaj Szydlarski,
Serge Van Criekingen and Jeaniffer Vides

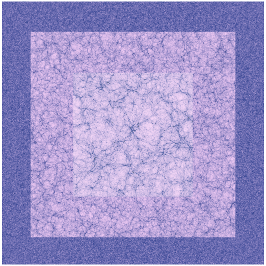
July 4, 2013

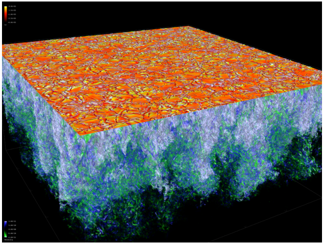
Outline

1. Introduction
2. Explicit
   - Model
   - Relaxation Scheme
   - Numerical Results
3. Implicit
4. Conclusions

Gravitational flows

- Gravitational flows are ubiquitous in Astrophysics
- Sometime gravity is largely dominant over other forces likes pressure gradients (i.e. cosmology,...)
- In many other cases we have a balance between gravity and other forces and are close to steady state. (i.e. stellar physics,...)





Euler-Poisson Model

- Study the numerical approximation of Euler equations when gravitational effects are taken into account
- System of partial differential equations (PDEs):

$$\begin{cases} \partial_t \rho & + & \nabla \cdot (\rho \mathbf{u}) & = & 0 \\ \partial_t (\rho \mathbf{u}) & + & \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + p) & = & -\rho \nabla \phi \\ \partial_t (\rho E) & + & \nabla \cdot ((\rho E + p) \mathbf{u}) & = & -\rho \mathbf{u} \cdot \nabla \phi \\ \Delta \phi = 4\pi G \rho \end{cases} \tag{1}$$

  - Gravitational potential ϕ
  - Pressure p governed by an equation of state $p := p(\rho, \epsilon)$
  - Specific internal energy $\epsilon = E - |\mathbf{u}|^2/2$
- Need of preserving the asymptotic regime of self-gravitational fluid flows in numerical simulations

Standard Fractional Step Splitting Method

1. Solve the Euler equations without source terms

$$\left\{\begin{array}{lllll} \partial_t \rho & + & \nabla \cdot (\rho \mathbf{u}) & = & 0 \\ \partial_t (\rho \mathbf{u}) & + & \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + p) & = & 0 \\ \partial_t (\rho E) & + & \nabla \cdot ((\rho E + p) \mathbf{u}) & = & 0 \end{array}\right. \quad (2)$$

2. Solve the ordinary differential equation (ODE)

$$\left\{\begin{array}{lll} \partial_t \rho & = & 0 \\ \partial_t (\rho \mathbf{u}) & = & -\rho \nabla \phi \\ \partial_t (\rho E) & = & -\rho \mathbf{u} \cdot \nabla \phi \end{array}\right. \quad (3)$$

3. Solve the Poisson equation $\Delta \phi = 4\pi G \rho$

A fully conservative approach

- Use a fully conservative reformulation of the Euler-Poisson system :

Balbus & Papaloizou 99, Chièze 98

$$\begin{cases} \partial_t(\rho) & + \quad \nabla\cdot(\rho\mathbf{u}) & = \quad 0, \\ \partial_t(\rho\mathbf{u}) & + \quad \nabla\cdot\left(\rho\mathbf{u}\otimes\mathbf{u}+p+\frac{\nabla\phi\otimes\nabla\phi}{8\pi G}\right) & = \quad 0, \\ \partial_t\left(\rho E_\phi+\frac{|\nabla\phi|^2}{8\pi G}\right) & + \quad \nabla\cdot\left(\rho E_\phi\mathbf{u}-\frac{\nabla\phi\partial_t\phi}{4\pi G}\right) & = \quad 0. \end{cases}$$

A fully conservative approach

- Use a fully conservative reformulation of the Euler-Poisson system :

Balbus & Papaloizou 99, Chièze 98

$$\begin{cases} \partial_t(\rho) & + \quad \nabla\cdot(\rho\mathbf{u}) & = \quad 0, \\ \partial_t(\rho\mathbf{u}) & + \quad \nabla\cdot\left(\rho\mathbf{u}\otimes\mathbf{u}+p+\frac{\nabla\phi\otimes\nabla\phi}{8\pi G}\right) & = \quad 0, \\ \partial_t\left(\rho E_\phi+\frac{|\nabla\phi|^2}{8\pi G}\right) & + \quad \nabla\cdot\left(\rho E_\phi\mathbf{u}-\frac{\nabla\phi\partial_t\phi}{4\pi G}\right) & = \quad 0. \end{cases}$$

New Finite-Volume Numerical Method [1]

- Derive a Godunov-type solver for the Euler-Poisson system and demonstrate its performace
  - Discretization of gravity introduced into the approximate Riemann solver used for the Euler equations
  - Solver based on a relaxation system
  - Implented in the software HERACLES
- Joint work with J. Vides, B. Braconnier, C. Berthon and B. Nkonga

[1] J. Vides, B. Braconnier, E. Audit, C. Berthon, and B. Nkonga. A Godunov-type solver for the numerical approximation of gravitational flows, *CiCP in press*

1-D Relaxation Model

- Relaxation
  - Thermodynamic pressure p
    $\partial_t p + u\partial_x p + \rho c^2 \partial_x u = 0 \quad \rightarrow \quad \partial_t \pi + u\partial_x \pi + \frac{a^2}{\rho}\partial_x u = \frac{1}{\delta}(p - \pi)$
  - Gravitational potential ϕ
    $\partial_t \psi = \frac{1}{\delta}(\phi - \psi)$
- Model

$$\begin{cases} \partial_t \rho + \partial_x (\rho u) = 0 \\ \partial_t (\rho u) + \partial_x (\rho u^2 + \pi) + \rho \partial_x \psi = 0 \\ \partial_t (\rho E) + \partial_x ((\rho E + \pi) u) + \rho u \partial_x \psi = 0 \\ \partial_t \pi + u\partial_x \pi + \frac{a^2}{\rho}\partial_x u = \frac{1}{\delta}(p - \pi) \\ \partial_t \psi = \frac{1}{\delta}(\phi - \psi) \end{cases} \tag{4}$$

- Compact form

$$\partial_t \mathbf{W}_\delta + \partial_x \mathbf{F}_\delta(\mathbf{W}_\delta) + \mathbf{B}_\delta(\mathbf{W}_\delta)\partial_x \psi = \frac{1}{\delta}\mathbf{R}_\delta(\mathbf{W}_\delta)$$

1-D Relaxation Scheme

First-order operator splitting approach to decompose (1) into two parts:

1. **Euler equations with gravity source terms**
   $\partial_t \mathbf{W} + \partial_x \mathbf{F}(\mathbf{W}) + \mathbf{B}(\mathbf{W})\partial_x \phi = 0$

   $\mathbf{W}$ is the unknown vector and ϕ is an *a priori* given function. We write the relaxation model (4) as $\partial_t \mathbf{W}_\delta + \partial_x \mathbf{F}_\delta(\mathbf{W}_\delta) + \mathbf{B}_\delta(\mathbf{W}_\delta)\partial_x \psi = \frac{1}{\delta}\mathbf{R}_\delta(\mathbf{W}_\delta)$.

   a. Evolution in time ($\delta = \infty$, $\partial_t \mathbf{W}_\delta + \partial_x \mathbf{F}_\delta(\mathbf{W}_\delta) + \mathbf{B}_\delta(\mathbf{W}_\delta)\partial_x \psi = 0$)

   $$\mathbf{W}_i^n \rightarrow (\mathbf{W}_\delta)_i^n \Rightarrow (\mathbf{W}_\delta)_i^{n+1,-}$$

   b. Relaxation ($\delta = 0$, $\partial_t \mathbf{W}_\delta = \frac{1}{\delta}\mathbf{R}_\delta(\mathbf{W}_\delta)$)

   $$(\mathbf{W}_\delta)_i^{n+1,-} \Rightarrow \mathbf{W}_i^{n+1}$$

2. **Poisson equation** $\partial_{xx}\phi = 4\pi G\rho$
   Use ρ_i^{n+1}, to solve the Poisson equation and obtain ϕ_i^{n+1} by means of a second-order finite difference approach which yields a tridiagonal matrix.

1-D Relaxation Scheme

1. $\partial_t \mathbf{W}_\delta + \partial_x \mathbf{F}_\delta(\mathbf{W}_\delta) + \mathbf{B}_\delta(\mathbf{W}_\delta)\partial_x \psi = \frac{1}{\delta}\mathbf{R}_\delta(\mathbf{W}_\delta)$
2. change of variables : $\mathbf{W}_\delta \to \mathbf{V}_\delta = (\rho, u, \epsilon, \pi, \psi)^T$.

Then, omitting the relaxation source term, the previous systeme can be writen as :

$$\partial_t \mathbf{V}_\delta + \mathbf{A}_\delta(\mathbf{V}_\delta)\partial_x \mathbf{V}_\delta = 0, \tag{5}$$

3. Compute the wave patern of $\mathbf{A}_\delta$ and solve the Riemann problem
4. Average over the solution
5. Apply the Scheme of the previous slide

$$\mathbf{W}_i^{n+1} = \mathbf{W}_i^n - \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i+\frac{1}{2}}^{\mathbf{L},n} - \mathbf{F}_{i-\frac{1}{2}}^{\mathbf{R},n}\right), \tag{6}$$

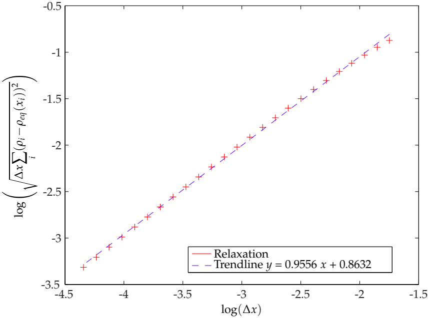
where

$$\mathbf{F}_{i+\frac{1}{2}}^{L,n} = \mathbf{F}_{i+\frac{1}{2}}^{L,n}(\rho_i^n,\ \mathbf{u}_i^n,\ (\rho E)_i^n,\ \phi_i^n,\ \rho_{i+1}^n,\ \mathbf{u}_{i+1}^n,\ (\rho E)_{i+1}^n,\ \phi_{i+1}^n), \tag{7}$$

$$\mathbf{F}_{i+\frac{1}{2}}^{R,n} = \mathbf{F}_{i+\frac{1}{2}}^{R,n}(\rho_i^n,\ \mathbf{u}_i^n,\ (\rho E)_i^n,\ \phi_i^n,\ \rho_{i+1}^n,\ \mathbf{u}_{i+1}^n,\ (\rho E)_{i+1}^n,\ \phi_{i+1}^n). \tag{8}$$

1-D Equilibrium Flow : hydrostatic atmosphere

Hydrostatic atmosphere : $\rho_{eq}(x) = \rho(0)e^{-gx/c^2}$



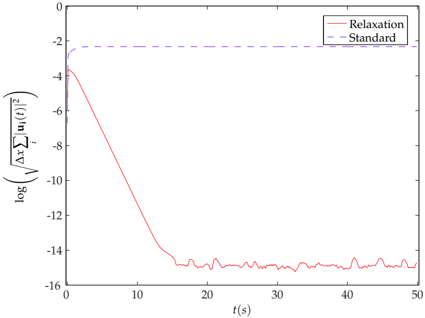


Figure : Accuracy of the proposed relaxation method (left); L^2 norm of the velocity in logarithmic scale as a function of time t with 1000 grid points (right)

$$T_{ref} = \sqrt{L_{ref}/g} \simeq 0.5$$

MAISON DE LA SIMULATION

Self-gravitating *star* - Lane-Emden Equation

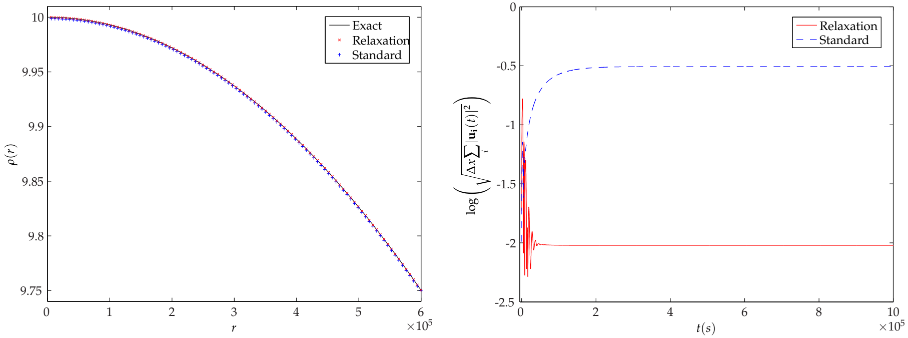
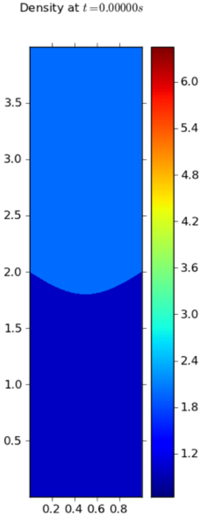


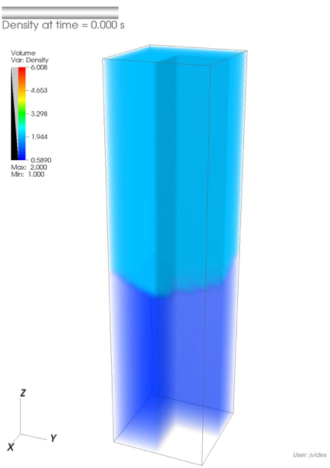
Figure : Densities compared to the exact solution $\rho(r) = 10 * \sin(z)/z$ with $z = A\, r$ (left); L^2 norm of the velocity in logarithmic scale as a function of time t with 100 grid points (right)

Rayleigh Taylor Instability

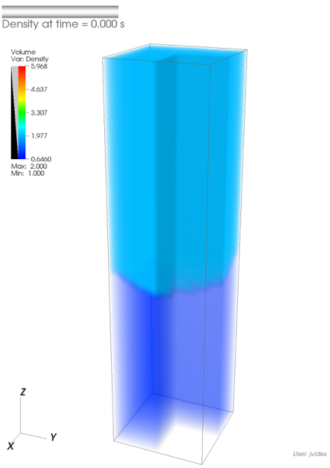


- Instability
  - Heavy fluid driven into lighter under the acceleration of a gravitational field
  - Initially, unstable interface separates the fluids with different densities
- Simulation
  - Positivity preserving limiter
  - 2000 × 4000 points
  - 256 processors

3-D Rayleigh Taylor Instability



First order



Positivity preserving

Implicit Formulation

1-D homogeneous case:

$$\partial_t \mathbf{W} + \nabla \cdot \mathbf{F}(\mathbf{W}) = 0$$

$$\begin{cases} \text{Finite volumes (spatial grid index } i\text{)} \\ \text{Explicit in time (time step index } n\text{)} \end{cases}$$

$$\Rightarrow \qquad \mathbf{W}_i^{n+1} = \mathbf{W}_i^n - \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i+\frac{1}{2}}^{L,n} - \mathbf{F}_{i-\frac{1}{2}}^{R,n}\right)$$

where the numerical flux $\mathbf{F}_{i\pm\frac{1}{2}}^n$ are obtained by Godunov's method, i.e., by solving Riemann problems: $\mathbf{F}_{i\pm\frac{1}{2}}^n(\mathbf{W}_i^n, \mathbf{W}_{i\pm1}^n)$.

Implicit Formulation

1-D homogeneous case:

$$\partial_t \mathbf{W} + \nabla \cdot \mathbf{F}(\mathbf{W}) = 0$$

$$\begin{cases} \text{Finite volumes (spatial grid index } i\,) \\ \text{Explicit in time (time step index } n\,) \end{cases}$$

$$\Rightarrow \qquad \mathbf{W}_i^{n+1} = \mathbf{W}_i^n - \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i+\frac{1}{2}}^{L,n+1} - \mathbf{F}_{i-\frac{1}{2}}^{R,n+1}\right)$$

where the numerical flux $\mathbf{F}_{i\pm\frac{1}{2}}^n$ are obtained by Godunov's method, i.e., by solving Riemann problems: $\mathbf{F}_{i\pm\frac{1}{2}}^n(\mathbf{W}_i^n, \mathbf{W}_{i\pm1}^n)$.

To avoid restrictions on Δt from CFL condition : implicit method.

More on Implicit Solving of Euler Equations

$$\mathbf{W}_i^{n+1} = \mathbf{W}_i^n - \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i+\frac{1}{2}}^{n+1} - \mathbf{F}_{i-\frac{1}{2}}^{n+1}\right)$$

Define

$$\mathcal{F}\left(\mathbf{W}_i^{n+1}, \mathbf{W}_{i\pm1}^{n+1}\right) = \frac{1}{\Delta x}\left(\mathbf{F}_{i+\frac{1}{2}}^{n+1} - \mathbf{F}_{i-\frac{1}{2}}^{n+1}\right)$$

so that

$$\frac{\mathbf{W}_i^{n+1} - \mathbf{W}_i^n}{\Delta t} = -\mathcal{F}\left(\mathbf{W}_i^{n+1}, \mathbf{W}_{i\pm1}^{n+1}\right)$$

More on Implicit Solving of Euler Equations

$$\mathbf{W}_i^{n+1} = \mathbf{W}_i^n - \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i+\frac{1}{2}}^{n+1} - \mathbf{F}_{i-\frac{1}{2}}^{n+1}\right)$$

Define

$$\mathcal{F}\left(\mathbf{W}_i^{n+1}, \mathbf{W}_{i\pm1}^{n+1}\right) = \frac{1}{\Delta x}\left(\mathbf{F}_{i+\frac{1}{2}}^{n+1} - \mathbf{F}_{i-\frac{1}{2}}^{n+1}\right)$$

so that

$$\frac{\mathbf{W}_i^{n+1} - \mathbf{W}_i^n}{\Delta t} = -\mathcal{F}\left(\mathbf{W}_i^{n+1}, \mathbf{W}_{i\pm1}^{n+1}\right)$$

For the whole mesh:

$$\frac{\mathbf{W}^{n+1} - \mathbf{W}^n}{\Delta t} = -\mathcal{F}\left(\mathbf{W}^{n+1}\right) \approx -\mathcal{F}(\mathbf{W}^n) - \frac{\partial \mathcal{F}}{\partial \mathbf{W}}\left(\mathbf{W}^{n+1} - \mathbf{W}^n\right)$$

$$\Rightarrow \underbrace{\left[\frac{\mathcal{I}}{\Delta t} + \frac{\partial \mathcal{F}}{\partial \mathbf{W}}\right]}_{\text{Jacobian}\,\mathcal{J}}\left(\mathbf{W}^{n+1} - \mathbf{W}^n\right) = -\mathcal{F}(\mathbf{W}^n)$$

More on Implicit Solving of Euler Equations

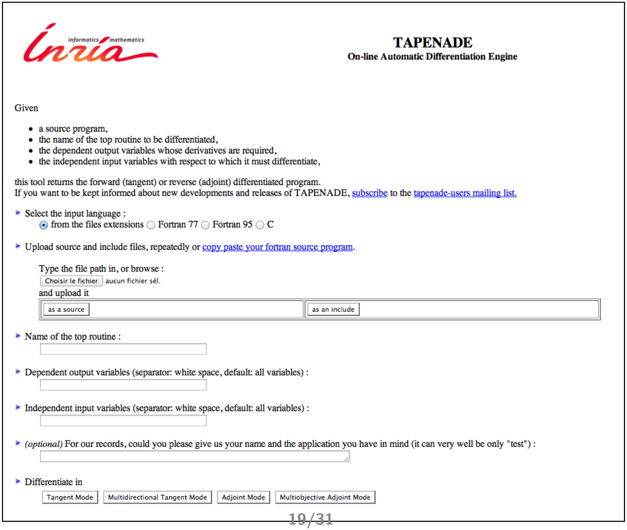
At each time step, Jacobian system solved using PETSc:

$$\mathcal{J}\left(\mathbf{W}^{n+1} - \mathbf{W}^{n}\right) = -\mathcal{F}(\mathbf{W}^{n})$$

Jacobian $\mathcal{J}$ not symmetric, but block symmetric.

The Jacobian is computed using Tapenade

tapenade.inria.fr



Tapenade example (1/3)

Input function:

```
subroutine ff(X,f)
  implicit none
  real :: x,f
  f = x*cos(abs(x))
  return
end subroutine ff
:
```

Tapenade example (2/3)

Input function re-written by Tapenade:

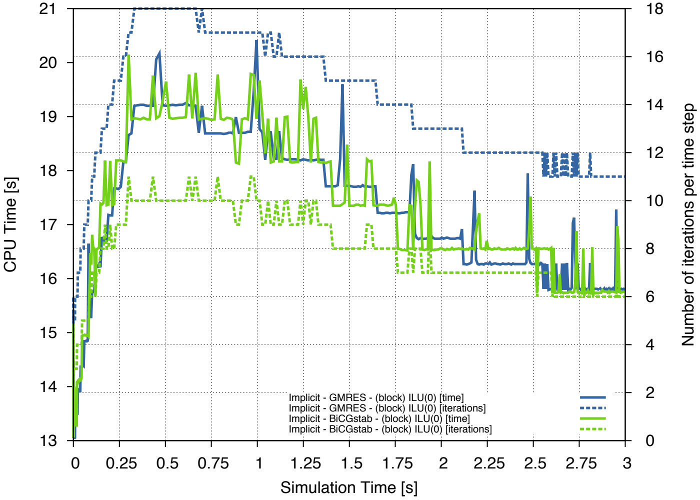
```
!        Generated by TAPENADE     (INRIA, Tropics team)
!  Tapenade 3.7 (r4888) - 28 May 2013 10:47
!
SUBROUTINE FF(x, f)
  IMPLICIT NONE
  REAL :: x, f
  INTRINSIC COS
  INTRINSIC ABS
  REAL :: abs0
  IF (x .GE. 0.) THEN
    abs0 = x
  ELSE
    abs0 = -x
  END IF
  f = x*COS(abs0)
  RETURN
END SUBROUTINE FF
```

Tapenade example (3/3)

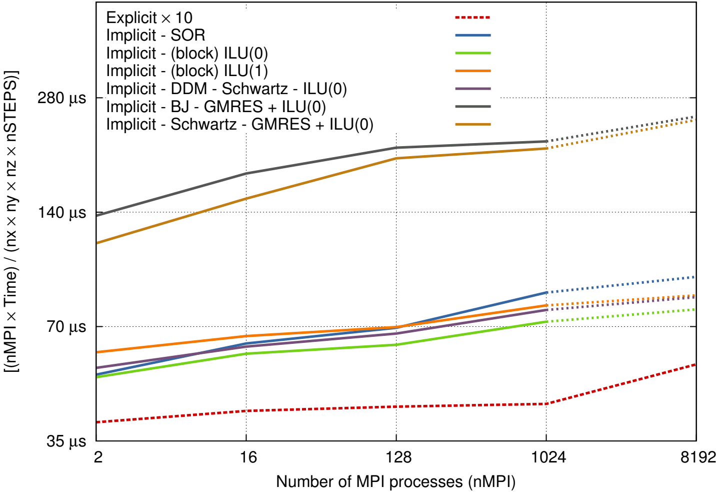
Output function by Tapenade:

```
!        Generated by TAPENADE     (INRIA, Tropics team)
!  Tapenade 3.7 (r4888) - 28 May 2013 10:47
!
!  Differentiation of ff in forward (tangent) mode:
!   variations   of useful results: f
!   with respect to varying inputs: x
!   RW status of diff variables: f:out x:in
SUBROUTINE FF_D(x, xd, f, fd)
  IMPLICIT NONE
  REAL :: x, f
  REAL :: xd, fd
  INTRINSIC COS
  INTRINSIC ABS
  REAL :: abs0d
  REAL :: abs0
  IF (x .GE. 0.) THEN
    abs0d = xd
    abs0 = x
  ELSE
    abs0d = -xd
    abs0 = -x
  END IF
  fd = xd*COS(abs0) - x*abs0d*SIN(abs0)
  f = x*COS(abs0)
  RETURN
END SUBROUTINE FF_D
```

Choice of Solver : BiCGstab vs GMRES



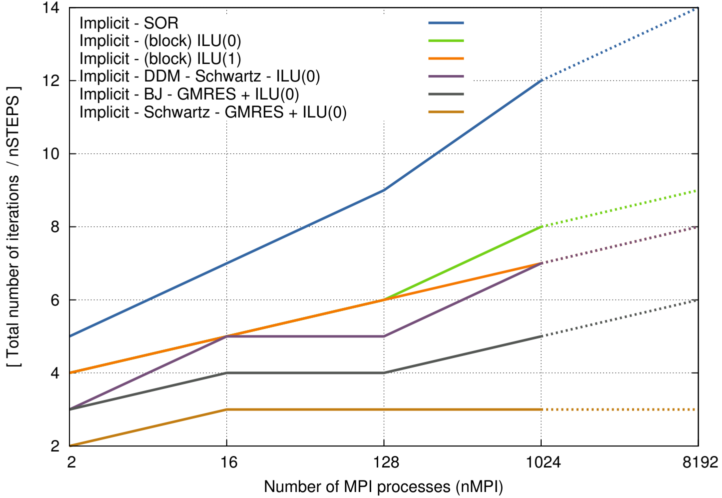
Weak scaling (time) - $64^3 \times nCPU$



---

dot line = extrapolation

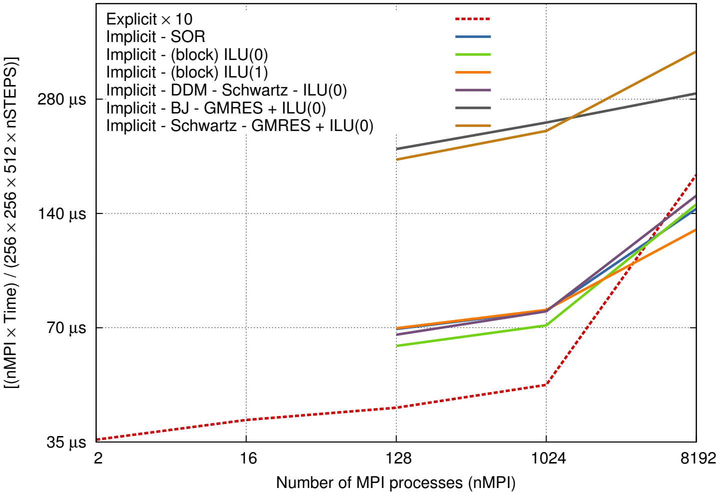
Weak scaling (iterations) - $64^3 \times nCPU$



dot line = extrapolation

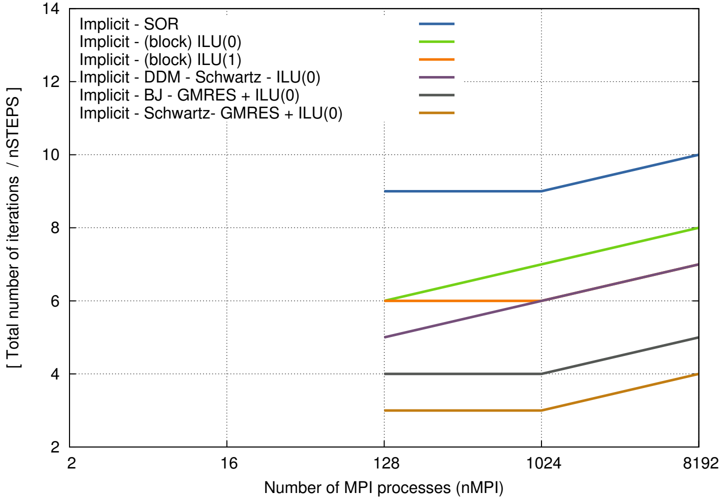
MAISON DE LA SIMULATION

Strong scaling (time) - ($nx = ny = 256, nz = 512$)



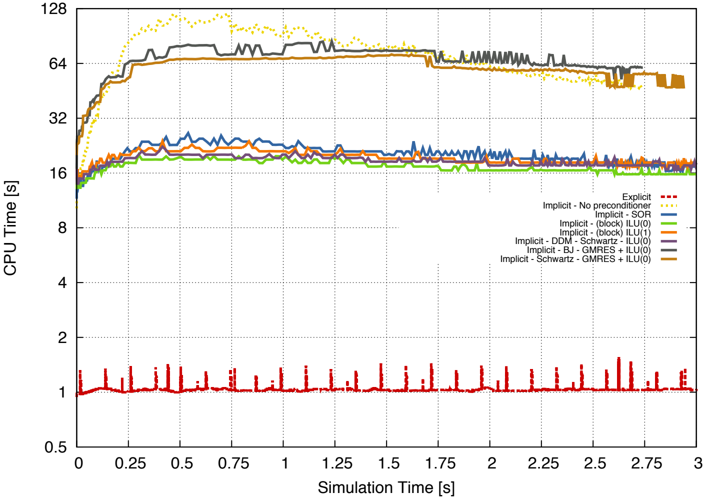
Not enough memory for $\text{nMPI} = [2, 16]$. 82GB is required if $\text{nMPI} = 128$.

Strong scaling (iterations) - ($nx = ny = 256, nz = 512$)

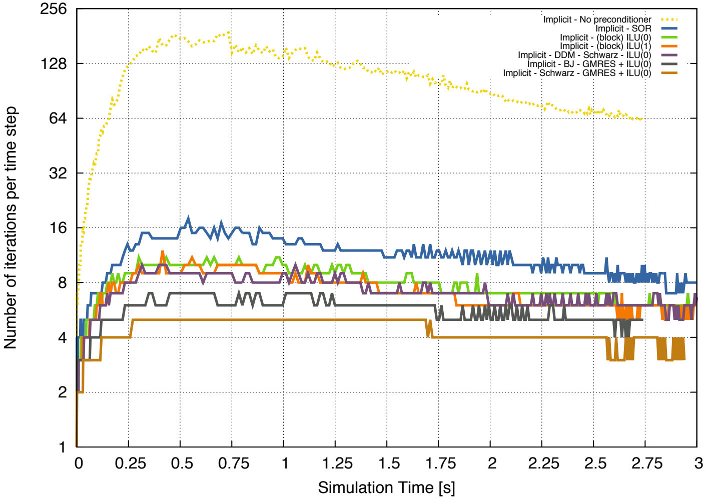


Not enough memory for nMPI $= [2, 16]$. 82GB is required if nMPI $= 128$.

Time evolution - $(nx = ny = 256, nz = 512, nMPI = 128)$



Iterations evolution - ($nx = ny = 256, nz = 512, nMPI = 128$)



Implicit : summary

- About **20** times slower than explicit
- Memory footprint is about **4** times larger
- Scaling is more difficult to achieve - the "best" method probably depends on the number of cores
- Dependance on the test case....

- Relaxation solver to integrate the Euler-Poisson systeme
- Accurate resolution of steady-state flows
- Implicit formulation using "automated" jacobian