Time-Implicit Hydrodynamics for Gravitational Flows

S. Van Criekingen[1], E. Audit[1], M. Szydlarski[1], B. Braconnier[2] and J. Vides[1]

[1] Maison de la Simulation, CEA Saclay, Gif-sur-Yvette, France
[2] IFP-Energie Nouvelles, Rueil-Malmaison, France

Efficient solution of large systems of non-linear PDEs in Science
7-9 Oct 2013 - Lyon, France

Contents

Context

Flows with **gravitation** (self- or not) in astrophysics

$\Rightarrow$ Euler-Poisson Equations

Context

Flows with **gravitation** (self- or not) in astrophysics

$\Rightarrow$ Euler-Poisson Equations

Parallel 3-D code **HERACLES** by Audit et al.
(CEA-Saclay, DSM/Service d'astrophysique)

hydrodynamics + MHD + radiative transfer + gravity + conduction

Euler-Poisson Equations

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

where

- fluid density ρ
- fluid velocity $\mathbf{u} \in \mathbf{R}^d$
- fluid specific Energy E
- fluid pressure $p = p(\rho, \epsilon) \leftarrow$ equation of state
  with the specific internal energy $\epsilon = E - |\mathbf{u}|^2/2$
- gravity potential ϕ (self or external)
- universal gravitational constant $G \approx 6.67\,10^{-11}\,m^3\,kg^{-1}\,s^{-2}$

Euler-Poisson Equations

$$\begin{cases} \partial_t \mathbf{W} + \nabla \cdot \mathbf{F}(\mathbf{W}) = -\mathbf{B}(\mathbf{W}) \nabla \phi \\ \Delta \phi = 4\pi G \rho \end{cases}$$

where

$$\mathbf{W} = \begin{pmatrix} \rho \\ \rho \mathbf{u} \\ \rho E \end{pmatrix} \qquad \mathbf{F}(\mathbf{W}) = \begin{pmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \otimes \mathbf{u} + p \\ (\rho E + p) \mathbf{u} \end{pmatrix} \qquad \mathbf{B}(\mathbf{W}) = \rho \begin{pmatrix} \mathbf{0}_d^T \\ \mathbf{e}_1^T \\ \vdots \\ \mathbf{e}_d^T \\ \mathbf{u}^T \end{pmatrix}$$

with

- $\mathbf{0}_d$ the null vector in $\mathbf{R}^d$
- $\mathbf{e}_i$ the i$^{\text{th}}$ canonical vector in $\mathbf{R}^d$.

Euler-Poisson Equations

$$\begin{cases} \partial_t \mathbf{W} + \nabla \cdot \mathbf{F}(\mathbf{W}) = -\mathbf{B}(\mathbf{W})\nabla\phi & (1) \\ \Delta\phi = 4\pi G\rho & (2) \end{cases}$$

Steps:

- With initial density ρ^0 compute ϕ^0 using Poisson Eq. (2)
- Solve Euler Eq. (1) using ϕ^0, yielding $\mathbf{W}^1$ at first time step
- Extract ρ^1 from $\mathbf{W}^1$, and compute ϕ^1 using Poisson Eq. (2)
- And so on...

Euler-Poisson Equations

$$\begin{cases} \partial_t \mathbf{W} + \nabla \cdot \mathbf{F}(\mathbf{W}) = -\mathbf{B}(\mathbf{W})\nabla\phi & (1) \\ \Delta\phi = 4\pi G\rho & (2) \end{cases}$$

Steps:

- With initial density ρ^0 compute ϕ^0 using Poisson Eq. (2)
- Solve Euler Eq. (1) using ϕ^0, yielding $\mathbf{W}^1$ at first time step
- Extract ρ^1 from $\mathbf{W}^1$, and compute ϕ^1 using Poisson Eq. (2)
- And so on...

To solve (1): finite volumes + Godunov (with relaxation of p and ϕ)
See *J. Vides et al., Comm. in Comp. Physics, 15(1), 2014*

Euler-Poisson Equations

$$\begin{cases} \partial_t \mathbf{W} + \nabla \cdot \mathbf{F}(\mathbf{W}) = -\mathbf{B}(\mathbf{W}) \nabla \phi & (1) \\ \Delta \phi = 4\pi G \rho & (2) \end{cases}$$

Steps:

- With initial density ρ^0 compute ϕ^0 using Poisson Eq. (2)
- Solve Euler Eq. (1) using ϕ^0, yielding $\mathbf{W}^1$ at first time step
- Extract ρ^1 from $\mathbf{W}^1$, and compute ϕ^1 using Poisson Eq. (2)
- And so on...

To solve (1): finite volumes + Godunov (with relaxation of p and ϕ)
See *J. Vides et al., Comm. in Comp. Physics, 15(1), 2014*

To solve (2): finite differences + CG

This contribution

$$\begin{cases} \partial_t \mathbf{W} + \nabla \cdot \mathbf{F}(\mathbf{W}) = -\mathbf{B}(\mathbf{W}) \nabla \phi & (1) \\ \Delta \phi = 4\pi G \rho & (2) \end{cases}$$

This contribution: implicit version of the explicit one, by implicitly solving the Euler equations (1)

This contribution

$$\begin{cases} \partial_t \mathbf{W} + \nabla \cdot \mathbf{F}(\mathbf{W}) = -\mathbf{B}(\mathbf{W}) \nabla \phi & (1) \\ \Delta \phi = 4\pi G \rho & (2) \end{cases}$$

This contribution: implicit version of the explicit one, by implicitly solving the Euler equations (1)

- **Jacobian** computed symbolically using the Automatic Differentiation tool **TAPENADE** (INRIA)

This contribution

$$\begin{cases} \partial_t \mathbf{W} + \nabla \cdot \mathbf{F}(\mathbf{W}) = -\mathbf{B}(\mathbf{W}) \nabla \phi & (1) \\ \Delta \phi = 4\pi G \rho & (2) \end{cases}$$

This contribution: implicit version of the explicit one, by implicitly solving the Euler equations (1)

- **Jacobian** computed symbolically using the Automatic Differentiation tool **TAPENADE** (INRIA)
- Coupling to **PETSc** to solve the Jacobian system (BiCGStab and GMRES + preconditioning)

Contents

More on Solving Euler Equations (1/4)

1-D homogeneous case:

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

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

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

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

More on Solving Euler Equations (1/4)

1-D homogeneous case:

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

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

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

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

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

More on Implicit Solving of Euler Equations (2/4)

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

Define

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

so that

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

More on Implicit Solving of Euler Equations (2/4)

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

Define

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

so that

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

For the whole mesh:

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

More on Implicit Solving of Euler Equations (3/4)

$$
\begin{aligned}
\frac{\mathbf{W}^{n+1} - \mathbf{W}^{n}}{\Delta t} &= -\mathscr{F}\left(\mathbf{W}^{n+1}\right) \\
&\approx -\mathscr{F}(\mathbf{W}^{n}) - \frac{\partial \mathscr{F}}{\partial \mathbf{W}}\left(\mathbf{W}^{n+1} - \mathbf{W}^{n}\right)
\end{aligned}
$$

linearly implicit

More on Implicit Solving of Euler Equations (3/4)

$$\begin{aligned} \frac{\mathbf{W}^{n+1}-\mathbf{W}^{n}}{\Delta t} &= -\mathscr{F}\left(\mathbf{W}^{n+1}\right) \\ &\approx -\mathscr{F}\left(\mathbf{W}^{n}\right)-\frac{\partial \mathscr{F}}{\partial \mathbf{W}}\left(\mathbf{W}^{n+1}-\mathbf{W}^{n}\right) \end{aligned}$$

linearly implicit

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

More on Implicit Solving of Euler Equations (4/4)

At each time step, Jacobian system solved using PETSc:

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

Jacobian $\mathscr{J}$:

- not symmetric, but block symmetric.
- computed symbolically by TAPENADE .

Contents

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
```

Contents

Heracles code ported on

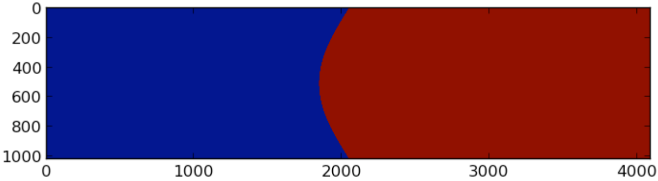
- *Poincaré* at Maison de la Simulation (1472 CPU cores)
- *Jade* at CINES (75 000 scalar hours from GENCI)
  Calculations (2-D) up to 4096 CPU cores
- *Curie* at TGCC
  Calculations (3-D) up to 8192 CPU cores

Heracles code ported on

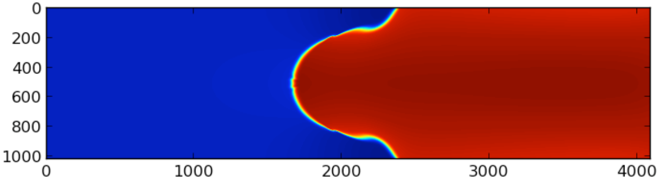
- *Poincaré* at Maison de la Simulation (1472 CPU cores)
- *Jade* at CINES (75 000 scalar hours from GENCI)
  Calculations (2-D) up to 4096 CPU cores
- *Curie* at TGCC
  Calculations (3-D) up to 8192 CPU cores

Test case: Rayleigh-Taylor instability

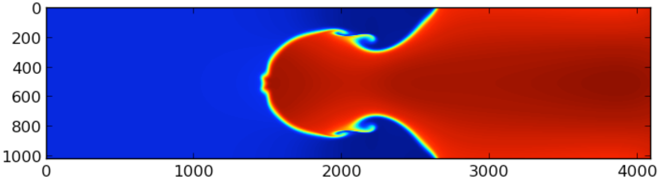
Rayleigh-Taylor Instability (T = 0.0s.)



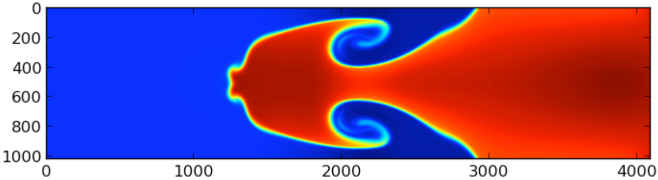
Rayleigh-Taylor Instability (T = 1.6s.)



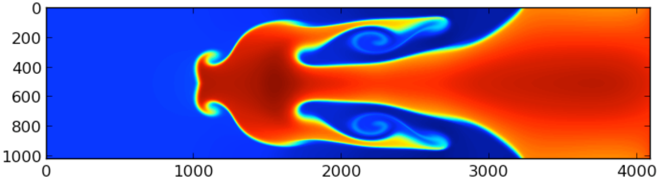
Rayleigh-Taylor Instability (T = 2.4s.)



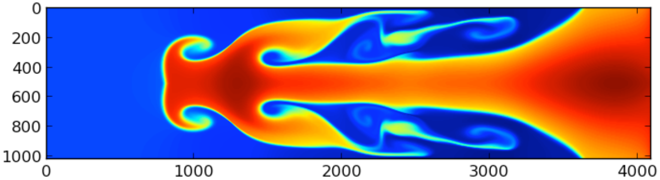
Rayleigh-Taylor Instability (T = 3.2s.)



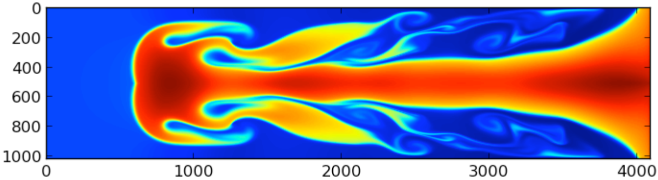
Rayleigh-Taylor Instability (T = 4.0s.)



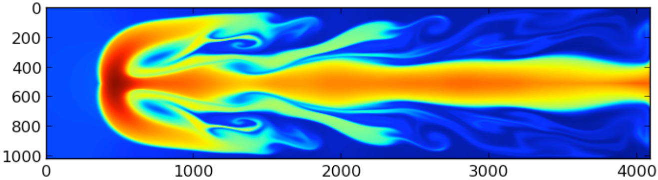
Rayleigh-Taylor Instability (T = 4.8s.)



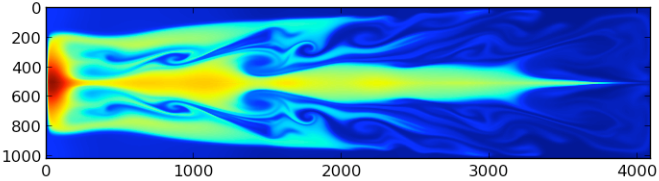
Rayleigh-Taylor Instability (T = 5.6s.)



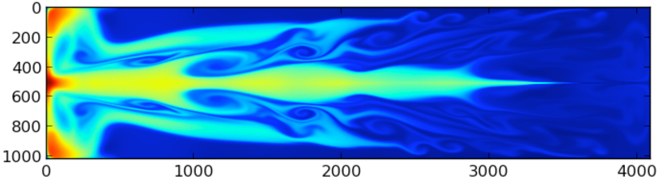
Rayleigh-Taylor Instability (T = 6.4s.)



Rayleigh-Taylor Instability (T = 7.2s.)



Rayleigh-Taylor Instability (T = 8.0s.)



Contents

Qualitative numerical results at $t = 4s$

EXPLICIT | IMPLICIT

1024×256 mesh

2048×512 mesh

4096×1024 mesh

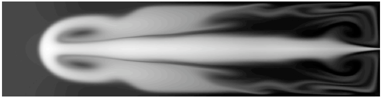
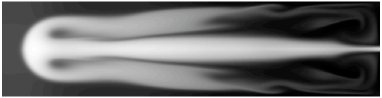
Time step: $\Delta t_{impl} \approx \Delta t_{expl} \times 60$

Total computing time: $T_{impl} \approx T_{expl}/3$

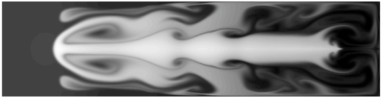
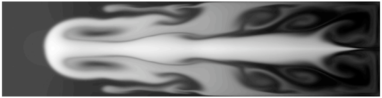
Qualitative numerical results at $t = 7s$

EXPLICIT IMPLICIT

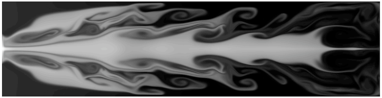
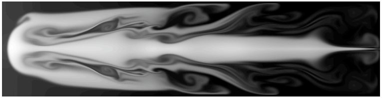
1024×256 mesh

2048×512 mesh

4096×1024 mesh

Time step: $\Delta t_{impl} \approx \Delta t_{expl} \times 60$

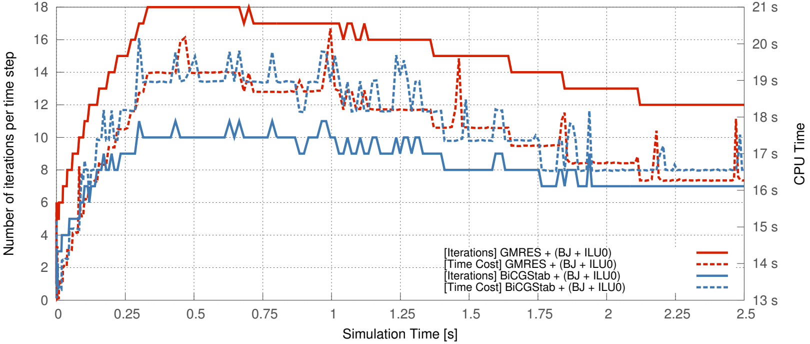
Total computing time: $T_{impl} \approx T_{expl}/3$

Qualitative discussion

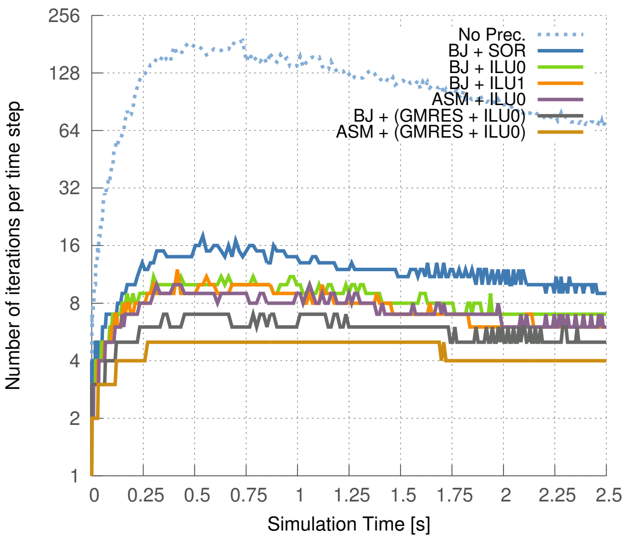
- Implicit more diffusive than explicit
- Discrepancies grow along with time evolution
- Fair quantitative comparison hardly possible without clear target result(s)

Contents

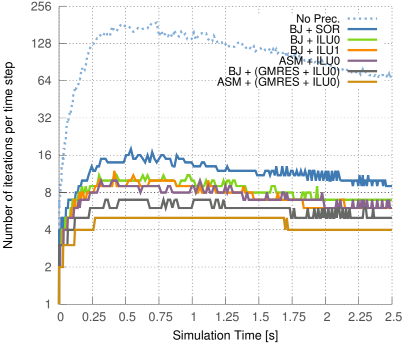
BiCGStab vs. GMRES (256 × 256 × 512 mesh ; 128 CPU)

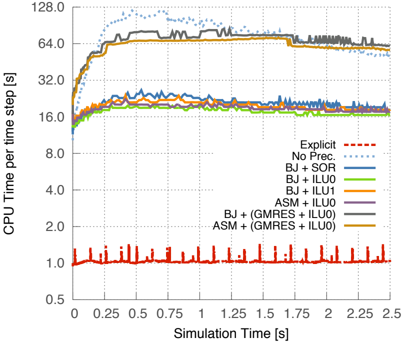


Comparing preconditioners (256 × 256 × 512 mesh ; 128 CPU)

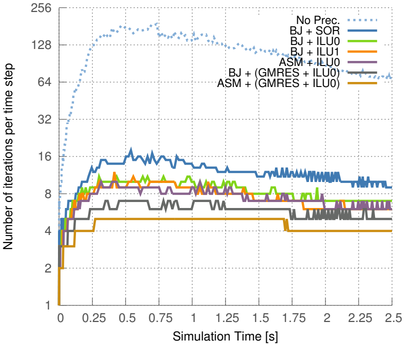


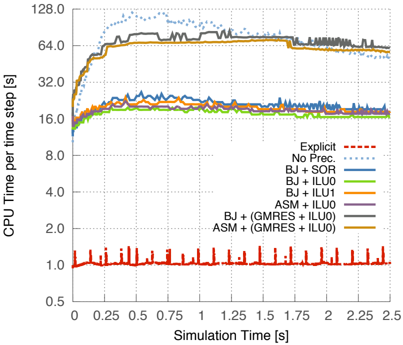
Comparing preconditioners (256 × 256 × 512 mesh ; 128 CPU)





Comparing preconditioners ($256 \times 256 \times 512$ mesh ; 128 CPU)

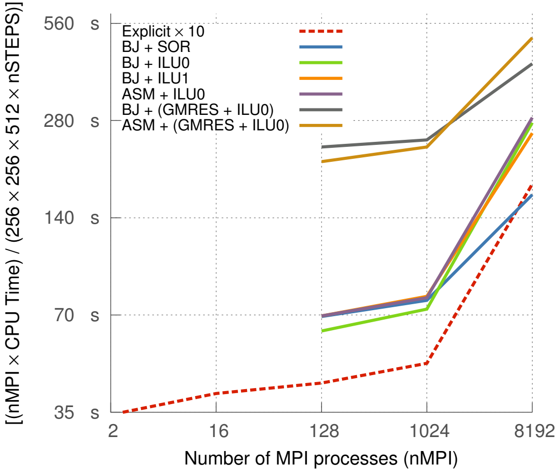




Time step: $\Delta t_{impl} \approx \Delta t_{expl} \times 60$

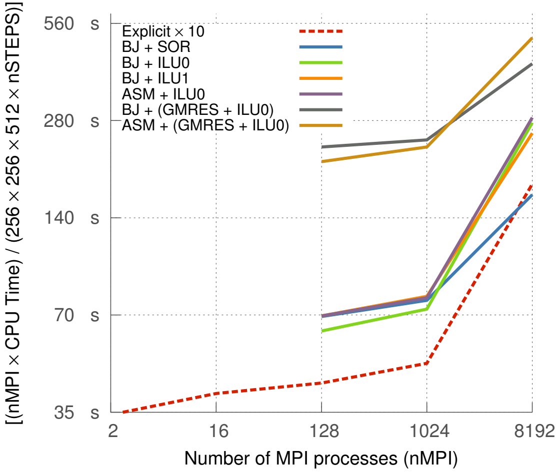
Total computing time: $T_{impl} \approx T_{expl}/3$

Strong scaling (256 × 256 × 512 mesh ; up to 8192 CPU)



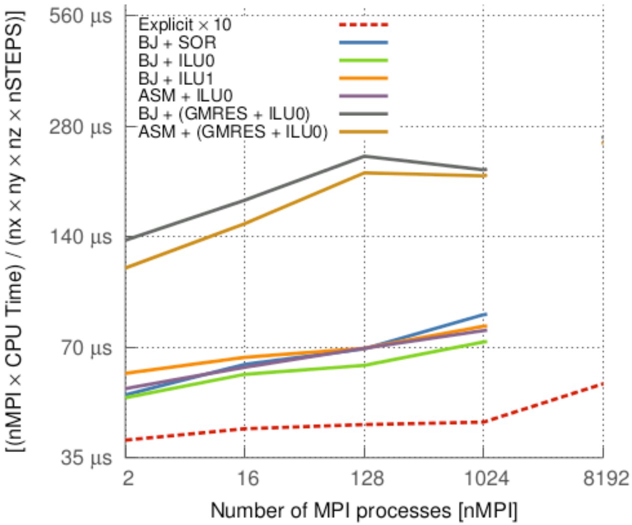
Not enough memory for nMPI=[2,16]

Strong scaling (256 × 256 × 512 mesh ; up to 8192 CPU)



Not enough memory for nMPI=[2,16] !!! Explicit × 10 !!!

Weak scaling ($64 \times 64 \times 64$ per nMPI)



Quantitative discussion

- Memory footprint 3 to 4 times larger in implicit.
- So far no better preconditioning than "simple" BJ+ILU(0) or BJ+SOR .
- Scaling difficult to achieve above 1024 cores.

Conclusions and Perspectives

- Implicit formulation using “automated” Jacobian: *feasibility study OK*
- Fair implict vs. explicit comparison requires target result.
- Test case with “hydro + **self-**gravity” under investigation.

Contact

serge.van-criekingen@cea.fr
edouard.audit@cea.fr

Choice of Δt

$$\text{cfl\_limit} = \min_{(i,j)}\left(\frac{\Delta x}{c_s + |u_x|_{(i,j)}} + \frac{\Delta y}{c_s + \left|u_y\right|_{(i,j)}}\right)$$

$$\Delta t_{\text{expl}} = \frac{1}{2} \times \text{cfl\_limit} \quad (1)$$

$$\Delta t_{\text{impl}} = \min\left(K_\rho, K_E\right) \times \text{cfl\_limit} \quad (2)$$

where (similarly for E):

$$K_\rho = \frac{\delta_{dt}\,\delta_{rel}}{\max\left(\delta_{dt}\max_{(i,j)}\left|\frac{\Delta\rho_{(i,j)}}{\rho_{(i,j)}}\right|,\ \delta_{rel}\right)} \qquad \Delta\rho_{(i,j)} = \rho^{n}_{(i,j)} - \rho^{n-1}_{(i,j)}$$

$\Delta\rho$ small: $\delta_{dt} = 1.05$ = time step length increase.
$\Delta\rho$ large: $\delta_{rel} = 0.05$ = relative variation of ρ