Splitting based Implicit solvers for compressible fluid models

D. Coulette[3], E. Franck[1], M. Gaja[2], P. Helluy[3], J. Lakhlili [2], M. Mazza[2], M. Mehrenberger[3], A. Ratnani[2], S. Serra-Capizzano[4] , E. Sonnendrücker[2]

NMPP Seminar, IPP, December 2016

[1]Inria Nancy Grand Est and IRMA Strasbourg, France
[2]Max-Planck-Institut für Plasmaphysik, Garching, Germany
[3]University of Strasbourg, France
[4]University of Insubria, Como, Italy

Outline

Mathematical and physical problems

Hyperbolic systems and explicit scheme

- We consider the general problem

$$\partial_t \boldsymbol{U} + \partial_x(\boldsymbol{F}(\boldsymbol{U})) = \nu \partial_x (D(\boldsymbol{U})\partial_x \boldsymbol{U})$$

- with $\boldsymbol{U} : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ (idem for $\boldsymbol{F}(\boldsymbol{U})$) and D a matrix.
- This system is parabolic and derivate on hyperbolic system when $\nu << 1$.
- In the following we consider the limit $\nu << 1$.
- Wave structure :

$$A(\boldsymbol{U}) = \frac{\partial F}{\partial \boldsymbol{U}} \text{ and } A = P(\boldsymbol{U})\Lambda(\boldsymbol{U})P^{-1}(\boldsymbol{U})$$

- The Riemann invariants given by $P(\boldsymbol{U})\boldsymbol{U}$ are propagated at the speed velocities (eigenvalues of A) contained in the matrix $\Lambda(\boldsymbol{U})$.

Explicit scheme

- CFL for explicit scheme: $\Delta t < \min\left(\frac{\Delta x}{\lambda_{max}}, \frac{\Delta x^2}{\nu}\right)$.

Problem of Explicit scheme

- **Problem**: if $V << \lambda_{max}$ (with V the characteristic velocity of the phenomena studied), the CFL is too restrictive.

Hyperbolic systems and explicit scheme

Implicit scheme

- **Implicit scheme**: allows to avoid the CFL condition filtering the fast phenomena.
- **Problem of implicit scheme**: need to invert large matrix. Direct solver not useful in 3D, we need iterative solvers.
- Conditioning of the implicit matrix: given by the ratio of the maximal and minimal eigenvalues.

- Implicit scheme :

$$\boldsymbol{U} + \Delta t \partial_x(\boldsymbol{F}(\boldsymbol{U})) - \Delta t \nu \partial_x(D(\boldsymbol{U})\partial_x \boldsymbol{U}) = \boldsymbol{U}^n$$

- At the limit $\nu << 1$ and $\Delta t >> 1$ (large time step) we solve $\partial_x \boldsymbol{F}(\boldsymbol{U}) = 0$

Problem of implicit scheme

- **Conclusion**: for $\nu << 1$ and $\Delta t >> 1$ the conditioning of the full system is closed to conditioning of the steady system given by the ratio of the speed waves to the hyperbolic system:

$$\text{condi} \approx \frac{\lambda_{max}}{\lambda_{min}}$$

.

Example of ill-conditioning systems

- Euler equation

$$\begin{cases} \partial_t\rho + \nabla\cdot(\rho\boldsymbol{u}) = 0, \\ \partial_t(\rho\boldsymbol{u}) + \nabla\cdot(\rho\boldsymbol{u}\otimes\boldsymbol{u} + pI_d) = 0, \\ \partial_t(\rho e) + \nabla\cdot(\rho\boldsymbol{u}e + \boldsymbol{u}p) = 0 \end{cases}$$

- Eigenvalues : $(\boldsymbol{u},\boldsymbol{n}) \pm c$ and $(\boldsymbol{u},\boldsymbol{n})$ with c the sound speed.
- Mach number : $M = \frac{|\boldsymbol{u}|}{c}$
- Nondimensional eigenvalues :

$$M-1, M, M+1$$

- **Conclusion**: ill-conditioned system for

$$M << 1 \text{ and } M = 1$$

- Same type of problem : Shallow - Water with sedimentation transport.

- Ideal MHD

$$\begin{cases} \partial_t\rho + \nabla\cdot(\rho\boldsymbol{u}) = 0, \\ \rho\partial_t\boldsymbol{u} + \rho\boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nabla p = \boldsymbol{J}\times\boldsymbol{B}, \\ \partial_t p + \boldsymbol{u}\cdot\nabla p + p\nabla\cdot\boldsymbol{u} = 0 \\ \partial_t\boldsymbol{B} = -\nabla\times(-\boldsymbol{u}\times\boldsymbol{B}), \\ \nabla\cdot\boldsymbol{B} = 0, \quad \nabla\times\boldsymbol{B} = \boldsymbol{J}. \end{cases}$$

- Eigenvalues : $(\boldsymbol{u},\boldsymbol{n})$, $(\boldsymbol{u},\boldsymbol{n}) \pm V_a$, $(\boldsymbol{u},\boldsymbol{n}) \pm \phi(c, V_a, \theta)$ with c the sound speed, V_a the Alfven speed and θ the angle between $\boldsymbol{n}$ and the $\boldsymbol{B}$.
- Mach number : $M = \frac{|\boldsymbol{u}|}{c}$ and β-number : $\beta = \frac{c}{V_a}$
- Approximated Nondimensional eigenvalues for $\beta << 1$ (Tokamak)

$$\beta M, \quad \beta M \pm 1, \quad M\beta \pm (\beta + 1)$$

in the parallel direction of the magnetic field (different in the perpendicular region).
- **Conclusion**: for example we have an ill-conditioned system for

$$M << 1, \quad \beta << 1$$

Other problems of conditioning

- Simple model

$$\nu u - \Delta u = f$$

- We define $\hat{u}(\boldsymbol{\theta})$ with $\boldsymbol{\theta} \in [-\pi, \pi]^2$ the Fourier transform of u.
- Applying the Fourier transform $\mathcal{F}$ we obtain

$$(\nu + \parallel \boldsymbol{\theta} \parallel^2)\hat{u} = \hat{f}$$

- After discretization more the mesh is fine more we have discrete low frequencies ($\boldsymbol{\theta} \approx 0$) $\longrightarrow$ ill conditioned discrete system.

- For fluids models (for $\nu << 1$ and $\Delta t >> 1$) the solutions are given by $\partial_x(\boldsymbol{F}_x(\boldsymbol{U})) + \partial_y(F_y(\boldsymbol{U})) = 0$.
- Linearizing around a constant state we obtain $A(\boldsymbol{U}_0)\partial_x \delta \boldsymbol{U} + B(\boldsymbol{U}_0)\partial_y \delta \boldsymbol{U} = 0$. Applying $\mathcal{F}$ we obtain

$$(A(\boldsymbol{U}_0, \boldsymbol{\theta}) + B(\boldsymbol{U}_0, \boldsymbol{\theta}))\, \hat{\boldsymbol{U}} = 0 \longleftrightarrow \Lambda(\boldsymbol{U}_0, \boldsymbol{\theta})(P^{-1}(\boldsymbol{U}_0, \boldsymbol{\theta})\hat{\boldsymbol{U}}) = 0$$

- Example: eigenvalues of linearized Euler equation in Fourier space

$$(\boldsymbol{u}, \boldsymbol{\theta}) - c, \quad (\boldsymbol{u}, \boldsymbol{\theta}), \quad (\boldsymbol{u}, \boldsymbol{\theta}) + c$$

- The Euler equations are ill-conditioned for the frequencies perp to the velocity.
- This type of problem existes for lot of fluid models and generate ill-conditioned matrices at the discrete level.

Idea

Limit of the classical method

- High memory consumption to store Jacobian and perhaps preconditioning.
- CPU time does not increase linearly comparing to the size problem (effect of the ill-condiitoning link to the physic).

Future of scientific computing

- Machines able to make lot of parallel computing.
- Small memory by node.

Idea: Divise and Conquer

- Propose algorithm with approximate the full problems by a collection of more simple one.
- Perform the resolution of the simple problems.
- Avoid memory consumption using matrix-free.

Physic-Based preconditioning and semi-implicit scheme

Linearized Euler equation

- We consider the 2D Euler equation in the conservative form,

$$\left\{\begin{array}{l}\partial_t\rho+\nabla\cdot(\rho\boldsymbol{u})=0\\ \rho\partial_t\boldsymbol{u}+\rho\boldsymbol{u}\cdot\nabla\boldsymbol{u}+\nabla p=0\\ \rho\partial_t T+\rho\boldsymbol{u}\cdot\nabla T+\gamma\rho T\nabla\cdot\boldsymbol{u}=0\end{array}\right.$$

- **Linearization**: $\boldsymbol{u}=\boldsymbol{u}_0+\delta\boldsymbol{u}$, $\rho=\rho_0+\delta\rho$, $T=T_0+\delta T$ and $\sqrt{\gamma T_0}$

$$\left\{\begin{array}{l}\partial_t\delta\rho+\boldsymbol{u}_0\cdot\nabla\delta\rho+\rho_0\nabla\cdot\delta\boldsymbol{u}=0\\ \rho_0\partial_t\delta\boldsymbol{u}+\rho_0\boldsymbol{u}_0\cdot\nabla\delta\boldsymbol{u}+\rho_0\nabla\delta T+T_0\nabla\delta\rho=0\\ \rho_0\partial_t\delta T+\rho_0\boldsymbol{u}_0\cdot\nabla\delta T+\gamma\rho_0 T_0\nabla\cdot\delta\boldsymbol{u}=0\end{array}\right.$$

- We multiply the first equation by T_0 and sum the first and third equations. After that we define $\delta p=\rho_0\delta T+T_0\delta\rho$

$$\left\{\begin{array}{l}\partial_t\delta p+\boldsymbol{u}_0\cdot\nabla\delta p+\rho_0 c^2\nabla\cdot\delta\boldsymbol{u}=0\\ \partial_t\delta\boldsymbol{u}+\boldsymbol{u}_0\cdot\nabla\delta\boldsymbol{u}+\frac{1}{\rho_0}\nabla\delta p=0\end{array}\right.$$

- After normalization we obtain the final model.

Final model

$$\left\{\begin{array}{l}\partial_t\boldsymbol{u}+M\boldsymbol{a}\cdot\nabla\boldsymbol{u}+\nabla p=0\\ \partial_t p+M\boldsymbol{a}\cdot\nabla p+\nabla\cdot\boldsymbol{u}=0\end{array}\right.$$

with $M\in]0,1]$, and $\parallel\boldsymbol{a}\parallel=1$.

Schur preconditioning method

- Implicit problem after time discretization:

$$\begin{pmatrix} I_d + M\lambda \boldsymbol{a}\cdot\nabla & \lambda\nabla\cdot \\ \lambda\nabla & I_d + M\lambda \boldsymbol{a}\cdot\nabla \end{pmatrix}\begin{pmatrix} p^{n+1} \\ \boldsymbol{u}^{n+1} \end{pmatrix} = \begin{pmatrix} I_d - M\lambda \boldsymbol{a}\cdot\nabla & \lambda_e\nabla\cdot \\ \lambda\nabla & I_d - M\lambda_e \boldsymbol{a}\cdot\nabla \end{pmatrix}\begin{pmatrix} p^{n} \\ \boldsymbol{u}^{n} \end{pmatrix}$$

- with $\lambda = \theta\Delta t$ and $\lambda_e = (1-\theta)\Delta t$.
- The implicit system after linearization is given by

$$\begin{pmatrix} p^{n+1} \\ \boldsymbol{u}^{n+1} \end{pmatrix} = \begin{pmatrix} A & \lambda\nabla\cdot \\ \lambda\nabla & A \end{pmatrix}^{-1}\begin{pmatrix} R_p \\ R_u \end{pmatrix}, \quad \text{with } A = I_d + M\lambda\boldsymbol{a}\cdot\nabla.$$

- Applying the Schur decomposition we obtain

$$\begin{pmatrix} p^{n+1} \\ \boldsymbol{u}^{n+1} \end{pmatrix} = \begin{pmatrix} I_d & A^{-1}\lambda\nabla\cdot \\ 0 & I_d \end{pmatrix}\begin{pmatrix} A^{-1} & 0 \\ 0 & P_{schur}^{-1} \end{pmatrix}\begin{pmatrix} I_d & 0 \\ -\lambda\nabla A^{-1} & I_d \end{pmatrix}\begin{pmatrix} R_p \\ R_u \end{pmatrix}$$

- Using the previous Schur decomposition, we obtain the following algorithm:

$$\begin{cases} \text{Predictor :} \quad Ap^{*} = R_p \\ \text{Velocity evolution :} \quad P_{schur}\boldsymbol{u}^{n+1} = (-\lambda\nabla p^{n+1} + R_u) \\ \text{Corrector :} \quad Ap^{n+1} = Ap^{*} - \lambda\nabla\cdot\boldsymbol{u}^{n+1} \end{cases}$$

Approximation (PC)

- $P_{schur} = A - \lambda^2\nabla((A^{-1})\nabla\cdot \approx A - \lambda^2\nabla(\nabla\cdot)$ and $A \approx I_d$ in the third equation. The approximation is valid in the low Mach regime.

Results on PC

- Firstly we consider the low Mach regime($M \approx 0$) with $\Delta t = 0.1$. We study the efficiency depending of the mesh.

PC n cells	$16*16$	$32*32$	$64*64$	$128*128$
no pc	250	90	20	25
PC_u	5	5	2	1
PC_p	7	6	2	2

- We call PC_p (resp PC_u) the case where the elliptic operator in on p (resp $\boldsymbol{u}$).
- Secondly, we consider the low Mach regime $M \approx 0$ with $h = 1/64$. We study the efficiency depending of the time step.

Preconditioning Δt	$\Delta t = 0.1$	$\Delta t = 0.2$	$\Delta t = 0.5$	$\Delta t = 1$	$\Delta t = 2$
no pc	20	35	70	130	230
PC_u	2	2	2	2	3
PC_p	2	2	2	3	3

Conclusion

- In the low Mach regime more the mesh is fine and the time step large more the PC is efficient.
- For Mach between 0.1 and 1 the efficiency for large time step is bad.

Interpretation of PB-PC as splitting scheme

- Splitting scheme:

$$\left\{\begin{array}{l}\partial_t p + Ma\,\boldsymbol{a}\cdot\nabla p = 0\\ \partial_t \boldsymbol{u} = 0\end{array}\right. ,\quad \left\{\begin{array}{l}\partial_t p + \nabla\cdot\boldsymbol{u} = 0\\ \partial_t \boldsymbol{u} + Ma\,\boldsymbol{a}\cdot\nabla\boldsymbol{u} + \nabla p = 0\end{array}\right. \tag{1}$$

- Discretization each subsystem with a θ scheme and using a Lie Splitting we obtain

$$(I_d + A_p)(I_d + A_u + C)\begin{pmatrix} p^{n+1}\\ \boldsymbol{u}^{n+1}\end{pmatrix} = \begin{pmatrix} R_p\\ R_{\boldsymbol{u}}\end{pmatrix} \tag{2}$$

- with

$$A_p = \begin{pmatrix} I_d + M\lambda\boldsymbol{a}\cdot\nabla & 0\\ 0 & 0\end{pmatrix}, A_u = \begin{pmatrix} 0 & 0\\ 0 & I_d + M\lambda\boldsymbol{a}\cdot\nabla\end{pmatrix}, C = \begin{pmatrix} 0 & \lambda\nabla\cdot\\ \lambda\nabla & I_d\end{pmatrix}$$

- The first step correspond to the predictor step

$$(I_d + A_p)\begin{pmatrix} p^*\\ \boldsymbol{u}^*\end{pmatrix} = \begin{pmatrix} R_p\\ R_{\boldsymbol{u}}\end{pmatrix}$$

- The second step can be rewritten (which correspond to update-corrector step of PBPC)

$$(I_d + A_u + C)\begin{pmatrix} p^{n+1}\\ \boldsymbol{u}^{n+1}\end{pmatrix} = \begin{pmatrix} p^*\\ \boldsymbol{u}^*\end{pmatrix} \Longleftrightarrow \left\{\begin{array}{l} P_{schur}\boldsymbol{u}^{n+1} = (-\lambda\nabla p^{n+1} + \boldsymbol{u}^*)\\ p^{n+1} = p^* - \lambda\nabla\cdot\boldsymbol{u}^{n+1}\end{array}\right.$$

- **Conclusion**: The PB-PC is equivalent to a first order implicit splitting scheme.

Splitting schemes and numerical results

- **Problem of PC** :
  - Less accurate for Mach closed to one.
  - Discretization effect which limited the extension of the classical PC.
- **Proposition** : use directly splitting schemes.
- **Different splitting schemes** (first or second order version can be used):

Schemes	Formula
Ap-AuC	$(Id + A_p)(Id + A_u + C)$
A-C	$(Id + A_p + A_u)(Id + C)$
Au-ApC	$(Id + A_u)(Id + A_p + C)$

- **Splitting error**: Splitting error E= O(Mach).
- Numerical results (for Mach=0.5) :

	Ap-AuC		A-C		Au-ApC	
	Order 1	Order 2	Order 1	Order 2	Order 1	Order 2
$\Delta t = 0.5$	0.9	1.1	0.9	$9E^{-2}$	1.4	1.1
$\Delta t = 0.25$	0.5	0.5	0.4	0.18	0.8	0.21
$\Delta t = 0.125$	0.3	$1.2E^{-1}$	0.45	$5.9E^{-2}$	0.55	$6.7E^{-2}$
$\Delta t = 0.0625$	0.15	$3.3E^{-2}$	0.18	$1.5E^{-2}$	0.28	$1.7E^{-2}$
$\Delta t = 0.03125$	$7.2E^{-2}$	$8.5E^{-3}$	$8.2E^{-2}$	$3.6E^{-3}$	0.14	$4.5E^{-3}$
$\Delta t = 0.015625$	$3.5E^{-2}$	$2.1E^{-3}$	$4.0E^{-2}$	$9.0E^{-4}$	$7.0E^{-2}$	$1.1E^{-3}$

- **Results**: expected order for the different splitting.

Numerical results

- We compare the CPU time for different simulation, changing the Mach number. **Test**: acoustic wave.

	$M = 10^{-4}$	$M = 10^{-2}$	$M = 10^{-1}$	$M = 0.5$
PC 1	101.6	145	240	5200
PC 2	98.9	125.8	208	5000
Sp $A_p - A_u C$	101.7	102.8	103	115.2
Sp $A_u - A_p C$	98.2	99.6	99.6	111.4
Sp $A - C_u$	90.4	92.1	92.7	102.3
Sp $A - C_p C$	93	94.3	95	104.5

- Comparison of the numerical solution (pressure). **Test**: acoustic wave with M=0.5.
- **Implicit time step** : $\Delta t = 0.01$ (2 CFL time step)

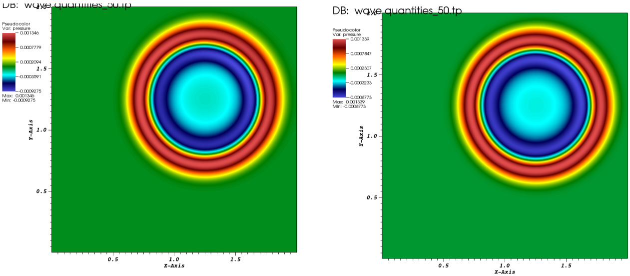


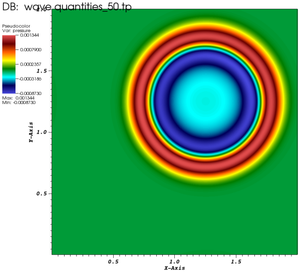
Figure: Left: solution for implicit scheme, Right: solution for Sp scheme $A_u - A_p C$

Numerical results

- We compare the CPU time for different simulation, changing the Mach number. **Test**: acoustic wave.

	$M = 10^{-4}$	$M = 10^{-2}$	$M = 10^{-1}$	$M = 0.5$
PC 1	101.6	145	240	5200
PC 2	98.9	125.8	208	5000
Sp $A_p - A_u C$	101.7	102.8	103	115.2
Sp $A_u - A_p C$	98.2	99.6	99.6	111.4
Sp $A - C_u$	90.4	92.1	92.7	102.3
Sp $A - C_p C$	93	94.3	95	104.5

- Comparison of the numerical solution (pressure). **Test**: acoustic wave with M=0.5.
- **Implicit time step** : $\Delta t = 0.01$ (2 CFL time step)



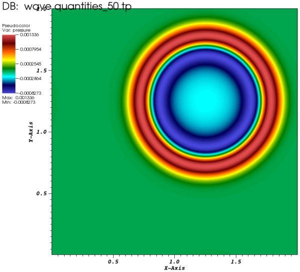


Figure: Left: solution for Sp scheme $A_p - A_u C$, Right: solution for Sp scheme $A - C$

Numerical results

- We compare the CPU time for different simulation, changing the Mach number. **Test**: acoustic wave.

	$M = 10^{-4}$	$M = 10^{-2}$	$M = 10^{-1}$	$M = 0.5$
PC 1	101.6	145	240	5200
PC 2	98.9	125.8	208	5000
Sp $A_p - A_u C$	101.7	102.8	103	115.2
Sp $A_u - A_p C$	98.2	99.6	99.6	111.4
Sp $A - C_u$	90.4	92.1	92.7	102.3
Sp $A - C_p C$	93	94.3	95	104.5

- Comparison of the numerical solution (pressure). **Test**: acoustic wave with M=0.5.
- **Implicit time step** : $\Delta t = 0.05$ (10 CFL time step)

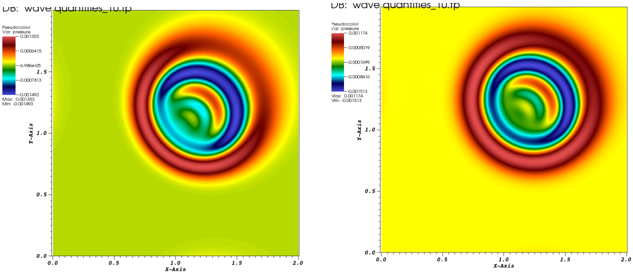


Figure: Left: solution for implicit scheme, Right: solution for Sp scheme $A_u - A_p C$

Numerical results

- We compare the CPU time for different simulation, changing the Mach number. **Test**: acoustic wave.

	$M = 10^{-4}$	$M = 10^{-2}$	$M = 10^{-1}$	$M = 0.5$
PC 1	101.6	145	240	5200
PC 2	98.9	125.8	208	5000
Sp $A_p - A_u C$	101.7	102.8	103	115.2
Sp $A_u - A_p C$	98.2	99.6	99.6	111.4
Sp $A - C_u$	90.4	92.1	92.7	102.3
Sp $A - C_p C$	93	94.3	95	104.5

- Comparison of the numerical solution (pressure). **Test**: acoustic wave with M=0.5.
- **Implicit time step** : $\Delta t = 0.05$ (10 CFL time step)

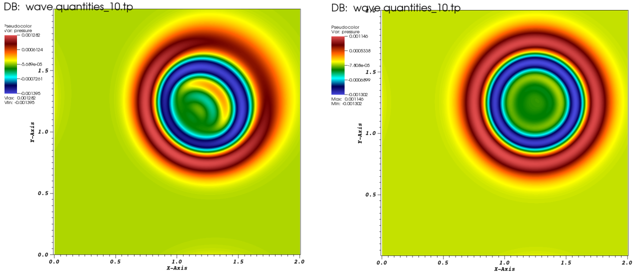


Figure: Left: solution for Sp scheme $A_p - A_u C$, Right: solution for Sp scheme $A - C$

Compressible Navier-Stokes equation splitting

- Compressible Navier-Stokes equation. Extension of previous method: three-step splitting:

$$\left\{\begin{array}{l}\partial_t\rho+\nabla\cdot(\rho\boldsymbol{u})=0\\ \rho\partial_t\boldsymbol{u}+\rho\boldsymbol{u}\cdot\nabla\boldsymbol{u}+\nabla p=\nu\Delta\boldsymbol{u}+(\nu+\lambda)\nabla(\nabla\cdot\boldsymbol{u})-\rho\boldsymbol{g}\\ \rho\partial_t T+\rho\boldsymbol{u}\cdot\nabla T+\gamma\rho T\nabla\cdot\boldsymbol{u}=\nu(\nabla\boldsymbol{u})^2+(\nu+\lambda)(\nabla\cdot\boldsymbol{u})^2+\nabla\cdot(\eta\nabla T)\end{array}\right. \quad (3)$$

- First solution:
  - Step 1:

$$\left\{\begin{array}{l}\partial_t\rho=0\\ \rho\partial_t\boldsymbol{u}=\nu\Delta\boldsymbol{u}+(\nu+\lambda)\nabla(\nabla\cdot\boldsymbol{u})\\ \rho\partial_t T=\nu(\nabla\boldsymbol{u})^2+(\nu+\lambda)(\nabla\cdot\boldsymbol{u})^2+\nabla\cdot(\eta\nabla T)\end{array}\right\}\text{Diffusion}\longrightarrow\text{CN}+\text{finit element}$$

  - Step 2:

$$\left\{\begin{array}{l}\partial_t\rho+\boldsymbol{u}\cdot\nabla\rho=0\\ \rho\partial_t\boldsymbol{u}+\rho\boldsymbol{u}\cdot\nabla\boldsymbol{u}=0\\ \rho\partial_t T+\rho\boldsymbol{u}\cdot\nabla T=0\end{array}\right\}\text{Transport}\longrightarrow\text{Semi Lagrangian}$$

  - Step 3:

$$\left\{\begin{array}{l}\partial_t\rho+\rho\nabla\cdot\boldsymbol{u}=0\\ \rho\partial_t\boldsymbol{u}+\nabla p=-\rho\boldsymbol{g}\\ \rho\partial_t T+\gamma\rho T\nabla\cdot\boldsymbol{u}=0\end{array}\right. \quad \text{Acoustic}+\text{gravity}\longrightarrow\text{CN}+\text{parabolization}+\text{FE}$$

- **Splitting Error**: O(Mach + Diffusion)

Compressible Navier-Stokes equation splitting

- Compressible Navier-Stokes equation. Extension of previous method: three-step splitting:

$$\left\{\begin{array}{l}\partial_t\rho+\nabla\cdot(\rho\boldsymbol{u})=0\\ \rho\partial_t\boldsymbol{u}+\rho\boldsymbol{u}\cdot\nabla\boldsymbol{u}+\nabla p=\nu\Delta\boldsymbol{u}+(\nu+\lambda)\nabla(\nabla\cdot\boldsymbol{u})-\rho\boldsymbol{g}\\ \rho\partial_t T+\rho\boldsymbol{u}\cdot\nabla T+\gamma\rho T\nabla\cdot\boldsymbol{u}=\nu(\nabla\boldsymbol{u})^2+(\nu+\lambda)(\nabla\cdot\boldsymbol{u})^2+\nabla\cdot(\eta\nabla T)\end{array}\right. \tag{3}$$

- Second solution:
  - Step 1:

$$\left\{\begin{array}{l}\partial_t\rho=0\\ \rho\partial_t\boldsymbol{u}+\rho\boldsymbol{u}\cdot\nabla\boldsymbol{u}=\nu\Delta\boldsymbol{u}+(\nu+\lambda)\nabla(\nabla\cdot\boldsymbol{u})\\ \rho\partial_t T=0\end{array}\right\}\text{Burgers}\longrightarrow\ \text{CN + FE or ?? (next part))}$$

  - Step 2:

$$\left\{\begin{array}{l}\partial_t\rho+\boldsymbol{u}\cdot\nabla\rho=0\\ \rho\partial_t\boldsymbol{u}=0\\ \rho\partial_t T+\rho\boldsymbol{u}\cdot\nabla T=\nu(\nabla\boldsymbol{u})^2+(\nu+\lambda)(\nabla\cdot\boldsymbol{u})^2+\nabla\cdot(\eta\nabla T)\end{array}\right\}\text{Convection diffusion}\longrightarrow\ \text{CN}$$

  - Step 3:

$$\left\{\begin{array}{l}\partial_t\rho+\rho\nabla\cdot\boldsymbol{u}=0\\ \rho\partial_t\boldsymbol{u}+\nabla p=-\rho\boldsymbol{g}\\ \rho\partial_t T+\gamma\rho T\nabla\cdot\boldsymbol{u}=0\end{array}\right.\quad\text{Acoustic + gravity}\longrightarrow\text{CN + parabolization + FE}$$

- **Splitting Error**: O(Mach + Diffusion)
- **Assumption**: First solution better for low diffusion (opposite for large diffusion).

Implicit scheme for linear MHD equation

Final model

$$\begin{cases} \partial_t \boldsymbol{u} + (M\sqrt{\beta}V_a)\boldsymbol{a}\cdot\nabla\boldsymbol{u} + \nabla p & = \frac{V_a^2}{|\boldsymbol{B}_0|}\left((\nabla\times\boldsymbol{B})\times\boldsymbol{b}_0\right) \\ \partial_t p + (M\sqrt{\beta}V_a)\boldsymbol{a}\cdot\nabla p + \beta V_a^2\nabla\cdot\boldsymbol{u} & = 0 \\ \partial_t \boldsymbol{B} + (M\sqrt{\beta}V_a)\boldsymbol{a}\cdot\nabla\boldsymbol{B} + \mid \boldsymbol{B}_0 \mid \nabla\times(\boldsymbol{b}_0\times\boldsymbol{u}) & = \frac{M\sqrt{\beta}V_a}{R_m}\nabla\times(\nabla\times\boldsymbol{B}) \end{cases}$$

with $M \in]0,1]$, $\beta \in \left]10^{-6}, 10^{-1}\right]$, $\mid \boldsymbol{a} \mid = \mid \boldsymbol{b}_0 \mid = 1$.

- We use a implicit scheme.
- We propose to apply PB-PC or splitting $A_p - A_u C$ method. At the end we must invert three operators

Operators of the PB-PC

$$I_d + (M\sqrt{\beta}\lambda)\boldsymbol{a}\cdot\nabla I_d - \frac{M\sqrt{\beta}\lambda}{R_m}\Delta I_d, \quad I_d + (M\sqrt{\beta}\lambda)\boldsymbol{a}\cdot\nabla I_d$$

$$P = \left(I_d + M\sqrt{\beta}\lambda\boldsymbol{a}\cdot\nabla I_d - \beta\lambda^2\nabla(\nabla\cdot I_d) - \lambda^2\left(\boldsymbol{b}_0\times(\nabla\times\nabla\times(\boldsymbol{b}_0\times I_d)\right)\right)$$

with $\mid \boldsymbol{a} \mid = 1$, $M << 1$, $\beta \in \left]10^{-4}, 10^{-1}\right]$ and $\lambda = V_a\Delta t$.

Relaxation methods

General principle

- We consider the following nonlinear system

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}(\boldsymbol{U}) = \nu \partial_x (D(\boldsymbol{U}) \partial_x \boldsymbol{U}) + \boldsymbol{G}(\boldsymbol{U})$$

- **Aim**: Find a way to approximate this systemwith a suite of simple systems.
- **Idea**: Xin-Jin relaxation method (finite volume method).

$$\begin{cases} \partial_t \boldsymbol{U} + \partial_x \boldsymbol{V} = \boldsymbol{G}(\boldsymbol{U}) \\ \partial_t \boldsymbol{V} + \alpha^2 \partial_x \boldsymbol{U} = \dfrac{1}{\varepsilon}(\boldsymbol{F}(\boldsymbol{U}) - \boldsymbol{V}) + \boldsymbol{H}(\boldsymbol{U}) \end{cases}$$

Limit of relaxation scheme

- The limit scheme of the relaxation system is

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{G}(\boldsymbol{U}) + \varepsilon \partial_x ((\alpha^2 - \mid A(\boldsymbol{U}) \mid^2) \partial_x \boldsymbol{U}) + \varepsilon \partial_x \boldsymbol{G}(\boldsymbol{U}) - \varepsilon \partial_x \boldsymbol{H}(\boldsymbol{U}) + o(\varepsilon^2)$$

- with $A(\boldsymbol{U})$ the Jacobian of $\boldsymbol{F}(\boldsymbol{U})$.

- **Conclusion**: the relaxation system is an approximation of the hyperbolic original system (error in ε).
- **Stability**: the limit system is dissipative if $(\alpha^2 - \mid \rho \mid^2) > 0$.

General principle II

Generalization

- Replacing $\frac{1}{\varepsilon} I_d$ by $\mathcal{E}^{-1}$ with

$$\mathcal{E} = \nu D(\boldsymbol{U})(\alpha^2 - \mid \rho \mid^2)^{-1}$$

- and taking $\boldsymbol{H}(\boldsymbol{U}) = A(\boldsymbol{U})\boldsymbol{G}(\boldsymbol{U})$: we obtain the following limit system

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{G}(\boldsymbol{U}) + \nu \partial_x (D\boldsymbol{U}\partial_x \boldsymbol{U}) + o(\nu^2)$$

- **Relaxation system**: "the nonlinearity is local and the non locality is linear".
- **Key method**: Splitting between source and linear hyperbolic part.

Solver for linear part

- The system

$$\left\{ \begin{array}{l} \partial_t \boldsymbol{U} + \partial_x \boldsymbol{V} = 0 \\ \partial_t \boldsymbol{V} + \alpha^2 \partial_x \boldsymbol{U} = 0 \end{array} \right.$$

- can be rewritten as N independent wave systems.
- **Wave solver**: Schur complement. We solve two mass matrices and one Laplacian to obtain the solution of the implicit wave problem.

Exemple 1 : 1D Burgers equation

- **Model** : Viscous Burgers equation

$$\partial_t\rho + \partial_x\left(\frac{1}{2}\rho^2\right) = \partial_x(\nu\partial_x\rho) + f$$

- Classical implicit scheme : Cranck-Nicholson + linearization + Newton.
- Relaxation system:

$$\begin{cases} \partial_t\rho + \partial_x u = f \\ \partial_t u + \alpha^2\partial_x\rho = \dfrac{1}{\varepsilon}\left(\dfrac{\rho^2}{2} - u\right) \end{cases}$$

Limit of relaxation scheme

- The limit scheme is given by

$$\partial_t\rho + \partial_x\left(\frac{1}{2}\rho^2\right) = \varepsilon\partial_x((\alpha^2 - \mid\rho\mid^2)\partial_x\rho) + f + o(\varepsilon^2)$$

- taking $\varepsilon = \frac{\nu}{\alpha^2 - |\rho|^2}$ we recover the initial equation.
- **Stability condition**: $\alpha > \mid u \mid$.

Exemple 1 : Time scheme for Burgers

Step:

- **Transport step** $(T(\Delta t))$:

$$\begin{pmatrix} I_d & \theta\Delta t\partial_x \\ \alpha^2\theta\Delta t\partial_x & I_d \end{pmatrix}\begin{pmatrix} \rho^* \\ u^* \end{pmatrix} = \begin{pmatrix} I_d & -(1-\theta)\Delta t\partial_x \\ -\alpha^2(1-\theta)\Delta t\partial_x & I_d \end{pmatrix}\begin{pmatrix} \rho^n \\ u^n \end{pmatrix}$$

- **Relaxation step** $(R(\Delta t))$:

$$\begin{cases} \rho^* = \rho^n + \Delta t f \\ u^* = \dfrac{\Delta t}{\varepsilon+\theta\Delta t}\dfrac{\rho^2}{2} + \dfrac{\varepsilon-(1-\theta)\Delta t}{\varepsilon+\theta\Delta t}u \end{cases}$$

- First order time scheme: $T(\Delta t)\circ R(\Delta t)$ with $\theta = 1$
- Second order time scheme: $T\left(\frac{\Delta t}{2}\right)\circ R(\Delta t)\circ T\left(\frac{\Delta t}{2}\right)$ or inverse with $\theta = 0.5$.

Consistency at the limit

- The first order scheme at the limit is consistent with

$$\partial_t\rho + \partial_x\left(\frac{1}{2}\rho^2\right) = (\varepsilon + \frac{\Delta t}{2})\partial_x((\alpha^2 - \mid\rho\mid^2)\partial_x\rho) + \frac{\Delta t}{2}\partial_x(\alpha^2\partial_x u) + f + o(\varepsilon^2 + \Delta t^2 + \varepsilon\Delta t)$$

Results I

- **Model** : We consider the Burgers equation without viscosity with source term.
- We choose as source term $f = g\rho$ to obtain a steady solution given by

$$\rho(t,x) = 1.0 + 0.1 e^{-\frac{x^2}{\sigma}}, \quad g(t,x) = -\frac{2x}{\sigma} e^{-\frac{x^2}{\sigma}}$$

- We consider the final time $T = 0.1$ and a fine mesh (10000 cells with third order polynomials). The first and second order schemes are compared for different time step.

	Order 1		Order 2	
	Error	Order	Error	Order
$\Delta t = 0.02$	$1.58E^{-2}$	-	$3.1E^{-4}$	-
$\Delta t = 0.01$	$9.47E^{-3}$	0.74	$7.75E^{-5}$	2.0
$\Delta t = 0.005$	$5.18E^{-3}$	0.87	$1.95E^{-5}$	2.0
$\Delta t = 0.0025$	$2.7E^{-3}$	0.94	$4.86E^{-6}$	2.0
$\Delta t = 0.00125$	$1.38E^{-3}$	0.97	$1.21E^{-6}$	2.0

Table: Error and order for the test 1 one with the relaxation scheme.

- The splitting scheme allows to obtain first and second order scheme without CFL condition.

- **Model** : Viscous - Burgers model.
- Spatial discretization: $N_{cell} = 10000$, $order = 3$. Initial condition : Gaussian.
- **Explicit time step** : stable if for $\Delta t = 1.0E^{-5}$.
- **Implicit time step** : $\Delta t = 1.0E^{-3}$

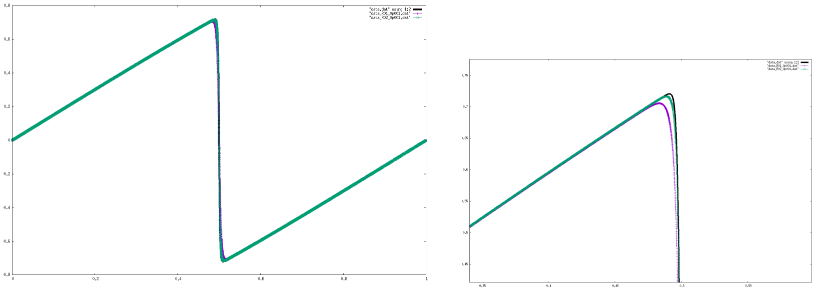


Figure: Left: numerical solution for first order and second order schemes for $\Delta t = 0.001$, Right: Zoom

- **Remark**: for discontinuous solution (or strong gradient solution) the scheme admits high numerical dispersion and instabilities.
- **Instability**: oscillations $\longrightarrow$ α increase and α increase $\longrightarrow$ oscillations increase.

Results II

- **Model** : Viscous - Burgers model.
- Spatial discretization: $N_{cell} = 10000$, $order = 3$. Initial condition : Gaussian.
- **Explicit time step** : stable if for $\Delta t = 1.0E^{-5}$.
- **Implicit time step** : $\Delta t = 1.0E^{-3}$, $\Delta t = 5.0E^{-3}$ and $\Delta t = 1.0E^{-2}$ (only for first order).

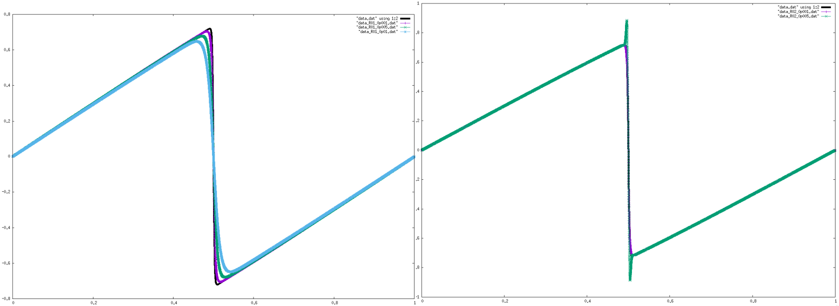


Figure: Left: numerical solution for first order scheme, Right: numerical solution for second order scheme. $\nu = 10^{-3}$

- **Remark**: for discontinuous solution (or strong gradient solution) the scheme admits high numerical dispersion and instabilities.
- **Instability**: oscillations $\longrightarrow$ α increase and α increase $\longrightarrow$ oscillations increase.

Results II

- **Model** : Viscous - Burgers model.
- Spatial discretization: $N_{cell} = 10000$, $order = 3$. Initial condition : Gaussian.
- **Explicit time step** : stable if for $\Delta t = 1.0E^{-5}$.
- **Implicit time step** : $\Delta t = 1.0E^{-3}$, $\Delta t = 5.0E^{-3}$ and $\Delta t = 1.0E^{-2}$.

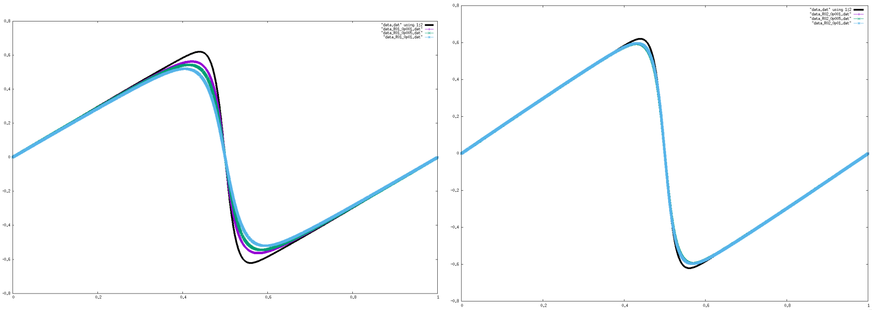


Figure: Left: numerical solution for first order scheme, Right: numerical solution for second order scheme. $\nu = 10^{-2}$

- **Remark**: for discontinuous solution (or strong gradient solution) the scheme admits high numerical dispersion and instabilities.
- **Instability**: oscillations $\longrightarrow$ α increase and α increase $\longrightarrow$ oscillations increase.

Results II

- **Model** : Viscous - Burgers model.
- **Conditioning** : well-conditioning system in 1D.
- Spatial discretization: $N_{cell} = 10000$, $order = 3$. Initial condition : Gaussian.
- **Explicit time step** : stable if for $\Delta t = 1.0E^{-5}$

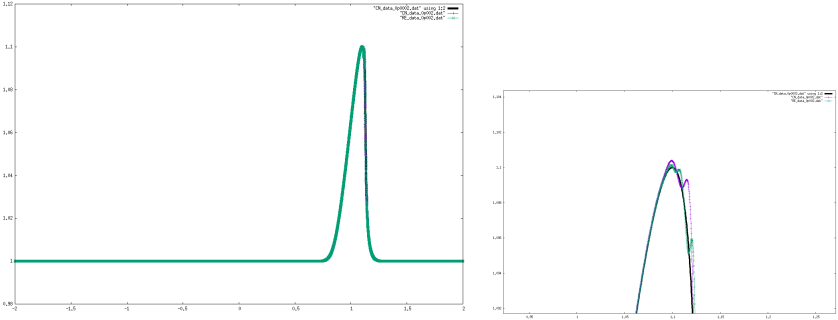


Figure: Left: Comparison between fine solution, CN and relaxation numerical solutions. Right: zoom. $\nu = 10^{-10}$, $\Delta t = 0.002$

- **Conclusion**: the Relaxation method is a little more dispersive that the Cranck-Nicholson method.

Results II

- **Model** : Viscous - Burgers model.
- **Conditioning** : well-conditioning system in 1D.
- Spatial discretization: $N_{cell} = 10000$, $order = 3$. Initial condition : Gaussian.
- **Explicit time step** : stable if for $\Delta t = 1.0E^{-5}$

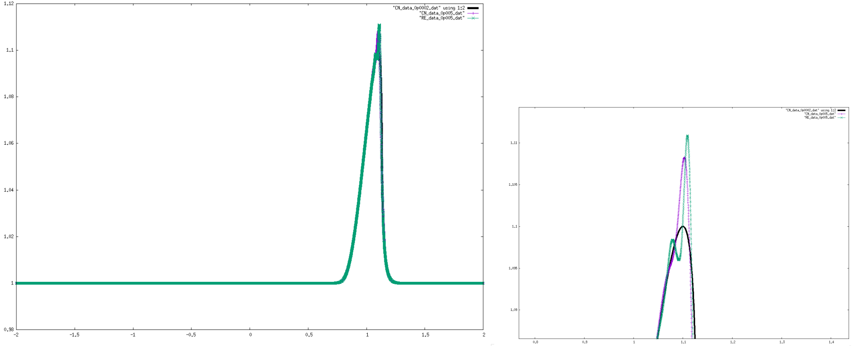
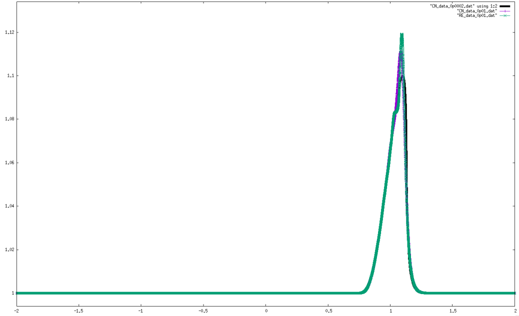


Figure: Left: Comparison between fine solution, CN and relaxation numerical solutions. Right: zoom. $\nu = 10^{-10}$, $\Delta t = 0.005$

- **Conclusion**: the Relaxation method is a little more dispersive that the Cranck-Nicholson method.

Results II

- **Model** : Viscous - Burgers model.
- **Conditioning** : well-conditioning system in 1D.
- Spatial discretization: $N_{cell} = 10000$, $order = 3$. Initial condition : Gaussian.
- **Explicit time step** : stable if for $\Delta t = 1.0E^{-5}$



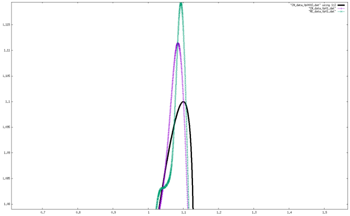


Figure: Left: Comparison between fine solution, CN and relaxation numerical solutions. Right: zoom. $\nu = 10^{-10}$, $\Delta t = 0.01$

- **Conclusion**: the Relaxation method is a little more dispersive that the Cranck-Nicholson method.

Results II

- **Model** : Viscous - Burgers model with $\nu = 10^{-12}$.
- Comparison of CPU time between two methods.

	CN method			Relaxation method		
Δt cells	5.10^3	10^4	2.10^4	5.10^3	10^4	2.10^4
$\Delta t = 0.005$	67	217.5	980	75.5	240	1100
$\Delta t = 0.01$	35	114	518	41	122.5	561
$\Delta t = 0.02$	18	61	280	20	63	294
$\Delta t = 0.05$	9.5	32.5	144	8	29	126

Remark

- The Relaxation method is competitive when the solver converges slowly for the classical method (high time step in this case).
- The assembly time is negligible in 1D not in 2D and 3D. The 1D burgers equation is not an ill-posed problem contrary multi-D hyperbolic systems or low Mach Euler equations.
- Therefore for complex models or in multi-D.

Future optimization:

- CN scheme does not use a PC and the relaxation scheme solves sequentially the independent subsystems.

Exemple II : 1D Navier-Stokes equation

- **Model** : Viscous Burgers equation

$$\left\{\begin{array}{l}\partial_t\rho+\partial_x(\rho u)=0\\ \partial_t\rho u+\partial_x(\rho u^2+p)=\partial_x(\nu(\rho)\partial_x u)-\rho g\\ \partial_t E+\partial_x(Eu+pu)=\partial_x(\nu(\rho)\partial_x\frac{u^2}{2})+\partial_x(\eta\partial_x T)-\rho v g\end{array}\right.$$

- We apply the relaxation method: three additional variables.

Stability

- The relaxation scheme is stable if $\alpha^2-\mid A\mid^2>0$ with A the Jacobian.
- **Classical choice**: $\alpha>u+c$.

Diffusion

- To obtain the physical diffusion matrix:

$$\mathcal{E}=\begin{pmatrix}0 & 0 & 0\\ -\frac{\nu(\rho)u}{\rho} & \frac{\nu(\rho)}{\rho} & 0\\ -\eta\frac{3}{2}\eta(\gamma-1)E-\nu(\rho)u^2 & \nu(\rho)-(\gamma-1)\rho\eta & (\gamma-1)\rho\eta\end{pmatrix}(\alpha^2-\mid A\mid^2)^{-1}$$

- Simple test case: $\rho(t,x) = 1 + G(x - ut)$, $u(t,x) = 2$ and $T(t,x) = 0$.

Scheme Δt	$\Delta t = 1.0E^{-2}$	$\Delta t = 5.0E^{-3}$	$\Delta t = 2.5E^{-3}$	$\Delta t = 1.25E^{-3}$
CN scheme	$8.8E^{-3}$	$2.25E^{-3}$	$5.7E^{-3}$	$1.4E^{-3}$
Relaxation scheme	$2.25E^{-3}$	$5.7E^{-4}$	$1.4E^{-4}$	$3.6E^{-5}$

- **Conclusion**: the relaxation scheme converges with the second order as expected.
- Acoustic wave test case:

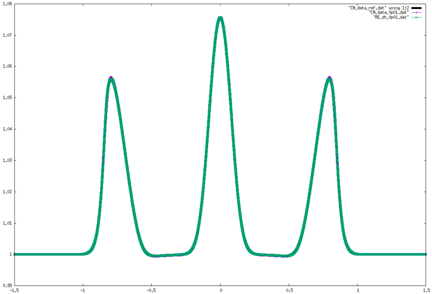


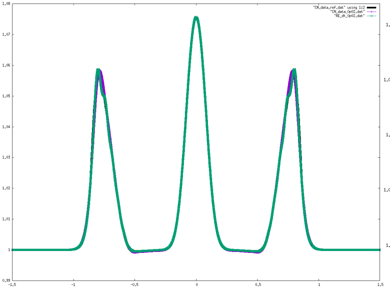
Figure: Fine solution (black). CN solution (violet) and Relaxation solution(green) $\Delta t = 0.01$

Results for Navier-Stokes equation I

- Simple test case: $\rho(t,x) = 1 + G(x - ut)$, $u(t,x) = 2$ and $T(t,x) = 0$.

Scheme Δt	$\Delta t = 1.0E^{-2}$	$\Delta t = 5.0E^{-3}$	$\Delta t = 2.5E^{-3}$	$\Delta t = 1.25E^{-3}$
CN scheme	$8.8E^{-3}$	$2.25E^{-3}$	$5.7E^{-3}$	$1.4E^{-3}$
Relaxation scheme	$2.25E^{-3}$	$5.7E^{-4}$	$1.4E^{-4}$	$3.6E^{-5}$

- **Conclusion**: the relaxation scheme converges with the second order as expected.
- Acoustic wave test case:



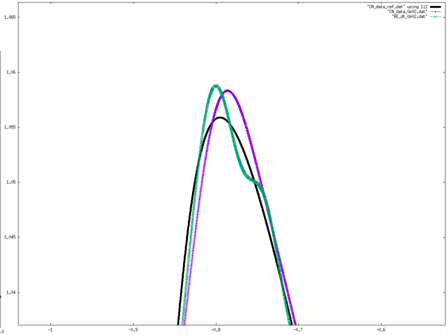


Figure: Fine solution (black). CN solution (violet) and Relaxation solution(green) $\Delta t = 0.02$

Results for Navier-Stokes equation I

- Simple test case: $\rho(t,x) = 1 + G(x - ut)$, $u(t,x) = 2$ and $T(t,x) = 0$.

Scheme Δt	$\Delta t = 1.0E^{-2}$	$\Delta t = 5.0E^{-3}$	$\Delta t = 2.5E^{-3}$	$\Delta t = 1.25E^{-3}$
CN scheme	$8.8E^{-3}$	$2.25E^{-3}$	$5.7E^{-3}$	$1.4E^{-3}$
Relaxation scheme	$2.25E^{-3}$	$5.7E^{-4}$	$1.4E^{-4}$	$3.6E^{-5}$

- **Conclusion**: the relaxation scheme converges with the second order as expected.
- Acoustic wave test case:

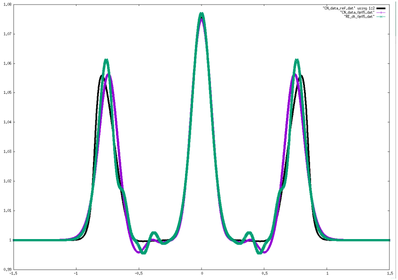


Figure: Fine solution (black). CN solution (violet) and Relaxation solution (green) $\Delta t = 0.05$

- The two methods (CN and relaxation) capture well the fine solution.

Results II

- **Model** : Compressible Navier-Stokes equation model with $\varepsilon = 10^{-10}$.
- **Initial data**: Constant pressure with a perturbation of density. Initial velocity null.
- **Test**: Propagation of acoustic wave.

	CN method			Relaxation method		
Δt / cells	5.10^3	10^4	2.10^4	5.10^3	10^4	2.10^4
$\Delta t = 0.005$	170	580	2550	135	420	1890
$\Delta t = 0.01$	100	345	1500	70	215	980
$\Delta t = 0.02$	60	205	920	40	120	525
$\Delta t = 0.05$	30	120	525	20	65	270

Conclusion:

- The 1D Navier-Stokes problem is ill-conditioned comparing to Burgers. In this case the efficiency of Relaxation comparing to Cranck-Nicholson is better.
- In this case the Relaxation method is competitive with the classical scheme without important optimization (no parallelization of the problem, etc).

Problem of relaxation solvers

- **Problem for Relaxation solver I**: high diffusion

$$\partial_t \boldsymbol{U} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U}) = \nabla \cdot (D(\boldsymbol{U}) \nabla \boldsymbol{U}) + \boldsymbol{G}(\boldsymbol{U}) + O(\mid D(\boldsymbol{U}) \mid^2)$$

- **Conclusion**: For $\mid D(\boldsymbol{U}) \mid << 1$ the relaxation system is valid.
- **Tokamak MHD context**: the anisotropic diffusion in the parallel direction is in O(1) for Tokamak. We must adapt the method.
- **Toy model**:

$$\left\{ \partial_t T + \nabla \cdot (\boldsymbol{u} T) = \nabla \cdot (D(\boldsymbol{b}) \nabla T), \quad D(\boldsymbol{b}) \nabla T = (\boldsymbol{b} \otimes \boldsymbol{b}) \nabla T + \kappa \nabla T \right.$$

- There exists different relaxation schemes for the diffusion.
- The first results (we need more results) show difficulty to treat large time steps if we use implicit schemes.
- **Possible solution** : modification of the relaxation method (keeping a part of relaxation step in the transport step) to treat high time step.

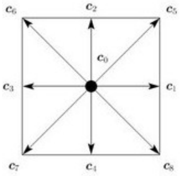
- **Problem for Relaxation solver II**: more numerical and physical dispersion (more critical problem)
- **Possible solution** : adaptive time scheme ? limiter or other treatment for discontinuities, high order scheme in time ?

Lattice Boltzmann schemes

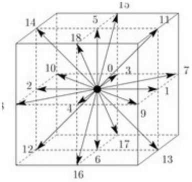
- **Lattice Boltzmann schemes**: use a kinetic interpretation of the Fluid mechanics model.

Lattice Scheme

- For N velocities $\rightarrow$ compute equilibrium: $f_i = w_i \rho \left(1 + 3(\boldsymbol{u}_i \cdot \boldsymbol{u}) + \frac{9}{2}(\boldsymbol{u}_i \boldsymbol{u}_i - \frac{1}{2} I_d) : \boldsymbol{uu}\right)$
- For N velocities $\rightarrow$ relaxation to the equilibrium: $\partial_t f_i = \frac{1}{\tau}(f_i^{eq} - f_i)$
- For N velocities $\rightarrow$ transport : $\partial_t f_i + v_i \cdot \nabla f_i = 0$
- We compute the moments $\rho = \sum_i f_i$, $\rho \boldsymbol{u} = \sum_i \boldsymbol{u}_i f_i$ etc



D2Q9



D3Q19

- **Advantage**: In DG context the transport matrices are triangular by block and can be solved by a up-down algorithm without stocking
- **Problem**: physical limitation. Example D2Q9 is consistent with isothermal Navier-Stokes + a destabilizing diffusion homogeneous to $O(Mach^3)$.
- **Solution**: use $DdQ(d+1)^n$ lattice we obtain a relaxation system where the transport is diagonal with properties closed to the Jin-Xin relaxation.

Elliptic problems

Elliptic problems for "Splitting" implicit schemes

Resume :

- All the methods proposed before split the complex systems between some simple systems.

- **Simples systems**:
  - Laplacian : $\nu u - \lambda \Delta u = f$
  - Advection: $\nu u + \lambda \boldsymbol{a} \cdot \nabla u = f$
  - Div-Div and Curl-Curl: $\nu \boldsymbol{u} - \lambda \nabla(\nabla \cdot \boldsymbol{u}) = \boldsymbol{f}, \quad \nu \boldsymbol{u} - \lambda \nabla \times (\nabla \times \boldsymbol{u}) = \boldsymbol{f}$
  - Alfven Curl-Curl: $\nu \boldsymbol{u} - \beta\lambda \nabla(\nabla \cdot \boldsymbol{u}) - \lambda \left(\boldsymbol{b}_0 \times (\nabla \times \nabla \times (\boldsymbol{b}_0 \times \boldsymbol{u}))\right) = \boldsymbol{f}$
  - For the last operator, we have additional complexity, but the scale can be probably separate using a formulation parallel-perp of the MHD and PC.
- **Conclusion**: to obtain efficient methods in time we need efficient methods for all these systems.
- **Efficient solvers**: solvers with an accuracy independent of λ , the order and the size of the mesh. Parallelized solvers.

GLT principle

- PDE : $Lu = g$ after discretization gives $L_n u_n = g_n$ with $\{L_n\}_n$ a sequence of matrices.

- It is often the case that the matrix L_n is a **linear combination, product, inversion or conjugation** of these two simple kinds of matrices
  - $T_n(f)$, i.e., a Toeplitz matrix obtained from the Fourier coefficient of $f : [-\pi, \pi] \to \mathbb{C}$, with $f \in L^1([-\pi, \pi])$.
  - $D(a)$, i.e., a diagonal matrix such that $(D_n(a))_{ii} = a(\frac{i}{n})$ with $a : [0, 1] \to \mathbb{C}$ Riemann integrable function.

  In such a case $\{L_n\}_n$ is called a **GLT sequence**.

Fundamental property

- Each GLT sequence $\{L_n\}_n$ is equipped with a "symbol", a function $\chi : [0, 1] \times [-\pi, \pi] \to \mathbb{C}$ which describes the asymptotic spectral behaviour of $\{L_n\}_n$:

$$\{L_n\}_n \sim \chi$$

E.g.: if $L_n = D_n(a) T_n(f)$, then $\{L_n\}_n \sim \chi = a \cdot f$

- **Advantage of this tool**: studying the symbol we retrieve information on the conditioning and propose new preconditioning based on this symbol.

GLT for stiffness matrix

- **Application**: B-Splines discretization of the model

$$-\Delta u = f, \quad \text{in } [0,1]^d.$$

- The basis functions are given by $\phi_i(x)$ a tensor product of 1D B-Splines functions.

Symbol of the problem

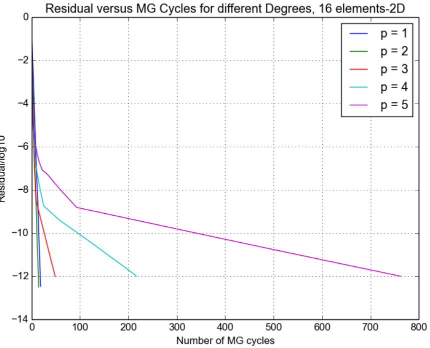
$$\left\{n^{d-2}L_n\right\}_n \sim \frac{1}{n}\left(\mathbf{\Pi}_{k=1}^d m_{p_k-1}(\theta_k)\right)\left(\sum_{k=1}^d \mu_k^2(2-2\cos(\theta_k))\mathbf{\Pi}_{j=1,j\neq k}^d w_{p_j}(\theta_j)\right)$$

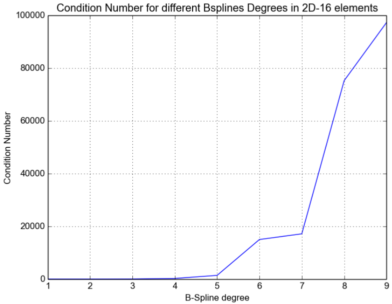
with $\theta_k \in [-\pi,\pi]$ and $w_p(\theta) := m_p(\theta)/m_{p-1}(\theta)$.

- $\left(\frac{4}{\pi^2}\right)^p \leq m_{p-1}(\theta) \leq m_{p-1}(0) = 1.$
- **Remark 1**: The symbol has a zero in $\theta = (0,\dots,0) \Rightarrow n^{d-2}L_n$ is ill-conditioned in the low frequencies. Classical problem solved by MG preconditioning.
- **Remark 2**: The symbol has infinitely many exponential zeros at the points θ with $\theta_j = \pi$ for some j when $p_j \rightarrow \infty \Rightarrow n^{d-2}L_n$ is ill-conditioned in the high frequencies. Non-canonical problem solvable by GLT theory.
- **Preconditioning**: Using the symbol we can construct a smoother for MG valid for high-frequencies. (i.e. CG preconditioned with a Kronecker product whose jth factor is $T_{\mu_j n+p_j-2}(m_{p_j-1})$).
- **Extension**: the method can be extended to the case with mapping (general geometries) and more general operators.

Numerical results

- **Solver**: Comparison between classical multi-grid solver and MG with CG + GLT preconditioning smoother.
- **Model**: 2D Laplacian with Homogeneous Dirichlet BC
- Efficiency of the multi-grid method depending to the polynomial degree.



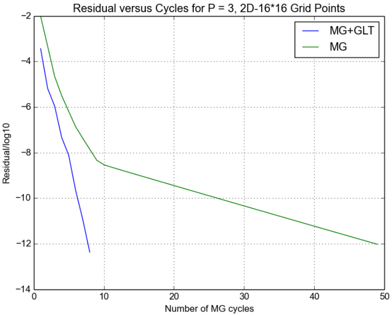


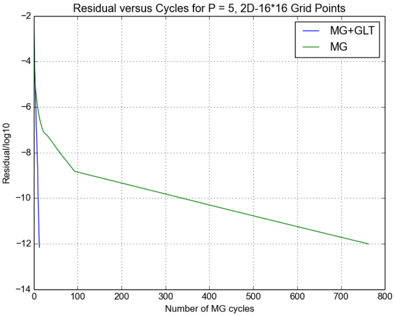
- **Conclusion**: the MG (as expected) is not efficient for high-order polynomial degrees.

Numerical results

- **Solver**: Comparison between classical multi-grid solver and MG with CG + GLT preconditioning smoother.
- **Model**: 2D Laplacian with Homogeneous Dirichlet BC

- **Conclusion**: the MG (as expected) is not efficient for high-order polynomial degrees.
- The efficiency of the multi-grid method + GLT PC method depending on the polynomial degree.





- **Conclusion**: the MG + CG-GLT is efficient for all high-order polynomial degrees.

Numerical results

- **Solver**: Comparison between classical multi-grid solver and MG with CG + GLT preconditioning smoother.
- **Model**: 2D Laplacian with Homogeneous Dirichlet BC
- **Conclusion**: the MG (as expected) is not efficient for high-order polynomial degrees.

Degree/Scheme	MG + GLT	MG
1	1.32	1.76
2	2.56	2.75
3	2.58	4.42
4	3.42	21.62
5	6.35	170.48
6	15.71	677.17*
7	25.99	825.56*
8	27.89	800.72*
9	58.03	1098.94*

Table: Computational cost comparison for the Laplacian operator -2D 64*64 elements

Conclusion

- The GLT preconditioning allows to avoid the problem of conditioning for high degree polynomial and limit CPU time.

Numerical results

- The GLT preconditioning is based on the "symbol" which describe the eigenvalues linked to the mass matrix.
- **Conclusion**: it can be also used as a PC for the mass matrix (closed to Kronecker product preconditioning).
- Result inverting the mass matrix with CG + GLT.

Degree	PCG	CG	Degree	PCG	CG
3	10	111	3	10	117
5	25	449	5	23	533
7	40	1777	7	38	2166

Table: Left: Number of iterations-mass matrix on a square 32*32. Right on a square 64*64

Degree	PCG	CG	Degree	PCG	CG
3	50	210	3	71	340
5	83	796	5	118	1711
7	125	2639	7	186	>3000

Table: Left: Number of iterations-mass matrix on a circle 32*32. Right on a circle 64*64

- **Conclusion**: the GLT PC is also a good PC for the mass matrix.

Vectoriel elliptic problems and advection

- Study of the conditioning problem using Fourier analysis.
- Fourier transform for Advection

$$\left[\nu + i(\boldsymbol{a}\cdot\boldsymbol{\theta})\right]\hat{u} = 0$$

- For $\nu << 1$ the system is ill-conditioning to the orthogonal frequencies to the velocity $\boldsymbol{a}$.
- Fourier transform for vectorial elliptic problems (ex grad div problem):

$$\left[\nu I_d + \begin{pmatrix} \theta_1^2 & \theta_1\theta_2 \\ \theta_1\theta_2 & \theta_2^2 \end{pmatrix}\right]\hat{\boldsymbol{u}} = 0$$

$$\left[\nu I_d + \begin{pmatrix} 0 & 0 \\ 0 & \parallel \boldsymbol{\theta} \parallel^2 \end{pmatrix}\right] P^{-1}\hat{\boldsymbol{u}} = 0$$

- For small ν the vectorial problems are ill-conditioning.

- **In the future**: GLT analysis to find additional problems due to the numerical discretization.
- **Aim**: find preconditioning for these problems. Open problem for advection. Auxiliary space or GLT with diagonalization for vectorial problems.

Conclusion

Conclusion

- **First way: Splitting method**. M. Gaja Phd and NMPP group.

Physic-based method

- **Advantages**:
  - Efficient method for low Mach method.
  - Compatible with equilibrium conservation.
  - Few memory consumption if coupled with Jacobian free.
- **Defaults**:
  - Nonlinear matrices (important cost)
  - Less efficient is the regime Mach closed to one.
  - Efficiency of PC depend also to the mesh, discretization etc (not clear)
  - Need Preconditioning for advection ?

Semi Implicit

- **Advantages**:
  - Probably efficient for all Mach regimes between zero and one.
  - Compatible with equilibrium conservation.
  - Few memory consumption if coupled with Jacobian free
- **Defaults**:
  - Nonlinear matrices (important cost)
  - Efficiency of PC depend also to the mesh, discretization etc (not clear)
  - Need Preconditioning for advection ?

Conclusion

Second way: Relaxation method. INRIA Tonus team and NPP group.

Relaxation

- **Advantages**:
  - Few memory consumption (derivates matrices and perhaps mass).
  - Good parallelization (models + domain decomposition).
  - Able to treat lots of regimes.
- **Defaults**:
  - Not directly able to treat high diffusion (on going work).
  - Lose of parallelization for complex BC.
  - A little bit more numerical dispersion.
  - not compatible with equilibrium conservation.

Remark

- All the methods needs preconditioning for mass, Laplacian and vectorial elliptic problems.
- All the methods needs stabilization or other treatment in the nonlinear phase for the numerical dispersion.
- Find 4th order schemes for the two methods could be possible and useful (ongoing work in TONUS team)