# Preconditioning and Reduced MHD solvers in DJANGO

E. Franck[1], M. Hölzl[2], J. Lakhlili [4],
A. Ratnani[3], E. Sonnendrücker[2]

Jorek Meeting, INRIA Sophia, April 14th

[1]Inria Nancy Grand Est and IRMA Strasbourg, France
[2]Max-Planck-Institut für Plasmaphysik, Garching, Germany

# Outline

## Preconditioning and Physic-Based PC

# Linear Solvers and preconditioning

- We solve a nonlinear problem $G(\boldsymbol{U}^{n+1}) = b(\boldsymbol{U}^n, \boldsymbol{U}^{n-1})$. First order linearization

$$\left(\frac{\partial G(\boldsymbol{U}^n)}{\partial \boldsymbol{U}^n}\right)\delta \boldsymbol{U}^n = -G(\boldsymbol{U}^n) + b(\boldsymbol{U}^n, \boldsymbol{U}^{n-1}) = R(\boldsymbol{U}^n),$$

  with $\delta \boldsymbol{U}^n = \boldsymbol{U}^{n+1} - \boldsymbol{U}^n$, and $J_n = \frac{\partial G(\boldsymbol{U}^n)}{\partial \boldsymbol{U}^n}$ the Jacobian matrix of $G(\boldsymbol{U}^n)$.

- Principle of the preconditioning step:
  - Replace the problem $J_k \delta \boldsymbol{U}_k = R(\boldsymbol{U}^n)$ by $P_k(P_k^{-1} J_k)\delta \boldsymbol{U}_k = R(\boldsymbol{U}^n)$.
  - Solve the new system with two steps $P_k \delta \boldsymbol{U}_k^* = R(\boldsymbol{U}^n)$ and $(P_k^{-1} J_k)\delta \boldsymbol{U}_k = \delta \boldsymbol{U}_k^*$
- If $P_k$ is easier to invert than $J_k$ and $P_k \approx J_k$ the solving step is more robust and efficient.

## Physic-based Preconditioning

- In the GMRES context if we have a algorithm to solve $P_k \mathbf{U} = \mathbf{b}$ we have a Preconditioning.
- **Principle**: construct an algorithm to solve $P_k \mathbf{U} = \mathbf{b}$ approximating and splitting the equations and approximating the discretizations.

## Application: Linearized Euler equation

# Implicit scheme for wave equation

## Linearized Euler equation:

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

- The hyperbolic part of the system admits the following eigenvalues and eigenvectors

$$\lambda_- = (\mathbf{a},\mathbf{n}) - c \quad \lambda_0 = (\mathbf{a},\mathbf{n}) \quad \lambda_+ = (\mathbf{a},\mathbf{n}) + c$$

  with $\mathbf{n}$ the direction of the wave.
- The implicit system is given by

$$\begin{pmatrix} p^{n+1}\\ \mathbf{u}^{n+1}\end{pmatrix} = \begin{pmatrix} AD_p & Div\\ Grad & AD_u\end{pmatrix}^{-1}\begin{pmatrix} R_p\\ R_{\mathbf{u}}\end{pmatrix}$$

- The solution of the system is given by

$$\begin{pmatrix} p^{n+1}\\ \mathbf{u}^{n+1}\end{pmatrix} = \begin{pmatrix} I & AD_p^{-1}Div\\ 0 & I\end{pmatrix}\begin{pmatrix} AD_p^{-1} & 0\\ 0 & P_{schur}^{-1}\end{pmatrix}\begin{pmatrix} I & 0\\ -GradAD_p^{-1} & I\end{pmatrix}\begin{pmatrix} R_p\\ R_{\mathbf{u}}\end{pmatrix}$$

  with $P_{schur} = AD_u - Grad(AD_p^{-1})Div$.
- Using the previous Schur decomposition, we can solve the implicit wave equation with the following algorithm:

$$\left\{\begin{array}{l}\text{Predictor}: \quad AD_p p^* = R_p\\ \text{Velocity evolution}: \quad P\mathbf{u}^{n+1} = (-Grad\, p^* + R_{\mathbf{u}})\\ \text{Corrector}: \quad AD_p p^{n+1} = AD_p p^* - Div\,\mathbf{u}_{n+1}\end{array}\right.$$

# PC for linearized Euler equations

- The preconditioning is given by the previous algorithm [3]-[4].

## Low Mach approximation:

- We assume that $a << c$ therefore we use the approximation

$$AD_p^{-1} = (I_d + \Delta t \mathbf{a} \cdot \nabla)^{-1} \approx I_d$$

in the second and third step.

- We obtain

$$\begin{cases} \text{Predictor :} \quad AD_p p^* = R_p \\ \text{Velocity evolution :} \quad P\mathbf{u}^{n+1} = (-Grad\, p^* + R_{\mathbf{u}}) \\ \text{Corrector :} \quad p^{n+1} = p^* - Div \mathbf{u}_{n+1} \end{cases}$$

with

$$AD_p = I_d + \Delta t \mathbf{a} \cdot \nabla, \text{ and } \quad P_{schur} = I_d + \Delta t \mathbf{a} \cdot \nabla - c^2 \Delta t^2 \nabla(\nabla \cdot)$$

## Remarks :

- $AD_p$ for $\mid \mathbf{a} \mid \Delta t >> 1$ is not easy to invert. **Solution**: specific PC or stabilization.
- $P_{schur}$ for $\mid \mathbf{a} \mid \Delta t >> 1$ or $c \Delta t >> 1$ is not easy to invert. Indeed $\text{Ker}(\nabla(\nabla \cdot \mathbf{u})) = \text{Span}\left\{\mathbf{u}, \nabla \times \mathbf{u} = 0\right\}$.
- **Solution**: specific preconditioning and/or specific finite element methods.

# Results I

- To validate the PC we compare the number of iterations to converge when we change the time step and the mesh.
- Steady test case for wave ( with $c = 1$ and $\mathbf{a} = 0$).

| h / $\Delta t$ | $\Delta t = 0.01$ | $\Delta t = 0.1$ | $\Delta t = 1$ | $\Delta t = 5$ | $\Delta t = 50$ |
|---|---|---|---|---|---|
| 16*16 | 1 | 1 | 1.2 | 5 | 9 |
| 32*32 | 1 | 1 | 1 | 1.2 | 6 |
| 64*64 | 1 | 1 | 1 | 1 | 1 |

- Unsteady test case for wave ( with $c = 1$ and $\mathbf{a} = 0$).

| h / $\Delta t$ | $\Delta t = 0.01$ | $\Delta t = 0.1$ | $\Delta t = 1$ | $\Delta t = 5$ | $\Delta t = 50$ |
|---|---|---|---|---|---|
| 16*16 | 1 | 1.2 | 2 | 2 | 2 |
| 32*32 | 1 | 1 | 1 | 1 | 1 |
| 64*64 | 1 | 1 | 1 | 1 | 1 |

- Comparison of different PC.
- Number of iterations for different PC with Mesh $32 \times 32$.

| $\Delta t / PC$ | Jacobi | ILU(0) | ILU(4) | Pb-PC |
|---|---|---|---|---|
| $\Delta t = 0.1$ | x | 70 | 20 | 1 |
| $\Delta t = 1$ | x | x | x | 1 |

# Result II

- Now we propose to compare the efficiency of the Physic based preconditioning for different value of the Mach number

$$M = \frac{|\,\mathbf{a}\,|}{c}$$

- **Test**: Propagation of a perturbation of the pressure ($\varepsilon = 10^{-9}$ for GMRES).

| h / M= | 0 | $10^{-4}$ | $10^{-2}$ | 0.1 | 1 | 10 | 100 |
|---|---|---|---|---|---|---|---|
| 16*16 | 16 | 16 | 22 | 65 | $> 100$ | $> 100$ | 19 |
| 32*32 | 8 | 9 | 17 | 53 | $> 100$ | $> 100$ | 16 |

- **Test**: Sinuosidal velocity and pressure ($\varepsilon = 10^{-9}$ for GMRES).

| h / M= | 0 | $10^{-4}$ | $10^{-2}$ | 0.1 | 1 | 10 | 100 |
|---|---|---|---|---|---|---|---|
| 16*16 | 2 | 3.5 | 5 | 7 | 7 | 10 | 7 |
| 32*32 | 1 | 2 | 3 | 3 | 5 | 4 | 4 |

# Conclusion on PC for linearized Euler equations

## Remark on global convergence :

- The global convergence is lower as $\Delta t$ increases. Indeed the PB-PC can be partially interpreted as a splitting method (error depend of $\Delta t$).
- The global convergence is faster as $h$ decreases. When $h$ is smaller the error comes only from the splitting. We kill the error like

$$\mid (\nabla \, \mathrm{Div})_h - \nabla_h \, \mathrm{Div}_h \mid = O(h^p)$$

- Generally the GMRES residue decreases a lot at the beginning and less after.

## Remark on sub-systems :

- We solve the sub-systems with an accuracy a little bit smaller that for the full systems.
- Finding a good solver for each sub-system is essential [5]-[6] .

## Remark on physical approximation:

- As expected the method is less efficient when the Mach number increase. Have a high accuracy for high mach number is an open question.
- The method is perhaps more efficient using an inexact Newton method and with diffusion or stabilization.

## Application: Current Hole

# Current Hole and preconditioning associated

- Current Hole : reduced problem in cartesian coordinates.
- The model

$$\left\{\begin{array}{l}\partial_t\psi=[\psi,u]+\eta(\Delta\psi-j_e)\\ \\ \partial_t\Delta u=[\Delta u,u]+[\psi,\Delta\psi]+\nu\Delta^2 u\end{array}\right.$$

with $w=\Delta u$ and $j=\Delta\psi$.

- In this formulation we split evolution and elliptic equations.
- For the time discretization we use a Cranck-Nicholson scheme and linearized the nonlinear system to obtain

$$\begin{pmatrix} M & U\\ L & D\end{pmatrix}\begin{pmatrix}\Delta\psi^n\\ \Delta u^n\end{pmatrix}=\begin{pmatrix}R_\psi\\ R_u\end{pmatrix}$$

or

$$\begin{pmatrix} I_d-\Delta t\theta[\cdot,u^n]-\Delta t\theta\Delta & -\Delta\theta[\psi^n,\cdot]\\ \\ -\Delta t\theta[\psi^n,\Delta\cdot]-\Delta t\theta[\cdot,\Delta\psi^n] & \Delta-\Delta t\theta([\Delta\cdot,u^n]+[\cdot,\Delta u^n]+\Delta^2)\end{pmatrix}\begin{pmatrix}\delta\psi^n\\ \delta u^n\end{pmatrix}=\begin{pmatrix}R_\psi\\ R_u\end{pmatrix}$$

## Low Mach PB-PC for Current Hole

$$\left\{\begin{array}{l}\text{Predictor :}\quad M\delta\psi_p^n=R_\psi\\ \text{potential update :}\quad P_{schur}\delta u^n=(-L\delta\psi_p^n+R_u))\\ \text{Corrector :}\quad \delta\psi^n=\delta\psi_p^n-U\delta u^n\\ \text{Elliptic update :}\quad \delta z_j^n=\Delta\delta\psi^n,\quad \delta w^n=\Delta\delta u^n\end{array}\right.$$

# Approximation of the Schur complement

- Computation of Schur complement (slow flow approximation $M^{-1} \approx \Delta t$)

$$P_{schur} = \frac{\Delta \delta u}{\Delta t} + \boldsymbol{u}^n \cdot \nabla(\Delta \delta u) + \delta \boldsymbol{u} \cdot \nabla(\Delta u^n) - \theta \nu \Delta^2 \delta u - \theta^2 \Delta t LU$$

- Operator $LU = \boldsymbol{B}^n \cdot \nabla(\Delta^*(\boldsymbol{B}^n \cdot \nabla \delta u)) + \frac{\partial j^n}{\partial \psi^n} \boldsymbol{B}^n_{pol} \cdot \nabla(\boldsymbol{B}^n \cdot \nabla \delta u)$.
- $\boldsymbol{B}^n \cdot \nabla \delta u = -[\psi^n, \delta u]$ and $\boldsymbol{u}^n \cdot \nabla \delta u = -[\delta u, u^n]$ et $\delta \boldsymbol{u} \cdot \nabla u^n = -[u^n, \delta u]$.
- **Remark**: the $LU$ operator is the parabolization of coupling hyperbolic terms which contains only the Alfvén waves.

## Properties of $LU$ operator

- We consider the $L^2$ space. The operator $LU$ is not self adjoint and not positive for all $\delta u$

$$< LU\delta u, \delta u >_{L^2} = \int |\nabla_{pol}(\boldsymbol{B}^n \cdot \nabla \delta u)|^2 - \int \frac{\partial j^n}{\partial \psi^n} (\boldsymbol{B}^n_{pol} \cdot \nabla \delta u)(\boldsymbol{B}^n \cdot \nabla \delta u)$$

- We propose the following approximation $LU^{approx} = \mathbf{B}^n \cdot \nabla(\Delta^*(\mathbf{B}^n \cdot \nabla \delta u))$.
- The operator $LU^{approx}$ is positive and self-adjoint.

- There are different methods to solve the Schur complement using splitting to solve smaller and more simple operators.

# Results Current Hole

- We give some results on the Physic-Based PC for the resistive kink instability.
- We use a small tolerance for the GMRES to avoid numerical instability linked to the mesh.
- We give results for different tolerances of the GMRES. Total run (linear and nonlinear phase).

| $\Delta t$ / $\varepsilon_{gmres}$ | $\varepsilon = 10^{-8}$ | $\varepsilon = 10^{-9}$ | $\varepsilon = 10^{-10}$ | $\varepsilon = 10^{-11}$ |
|---|---|---|---|---|
| $\Delta t = 1$ Mesh=32*32 | 1 | 1-3 | 3-5 | 4-10 |
| $\Delta t = 10$ Mesh=32*32 | 2-8 | 4-25 | 10-45 | 15-60 |
| $\Delta t = 10$ Mesh=64*64 | 1-10 | 1-20 | 10-55 | 20-70 |

- Worst phase for convergence: the beginning.
- In general the GMRES begin with a very good error and this error decreases slowly.
- Good behavior for the coupling with inexact Newton method.
- **Remark**: solve the mesh problem to get fully pertinent results.

## Application: Reduced MHD without parallel velocity

# Current work: model 199

- Algorithm:
  - **Step 1**: Solve Grad-Shafranov on circular mesh using Picard method.
  - **Step 2**: Construction of initial data using $\psi_{eq}$ for the model 199.
  - **Step 3**: Loop in time (same model as JOREK).
- **Remark**: Currently no aligned grid (external grid).
- New matrix for the Preconditioning:

$$M = \begin{pmatrix} I_d - \Delta([\cdot, u^n] + \eta\Delta^*) & 0 & 0 \\ 0 & I_d - \Delta t([\cdot, u^n] + \nabla\cdot(D\nabla\cdot)) & 0 \\ 0 & 0 & I_d - \Delta t([\cdot, u^n] + \nabla\cdot(K\nabla\cdot)) \end{pmatrix}$$

and

$$P_{schur} \approx \frac{\nabla_{pol}\cdot(R^2\rho\nabla\delta u)}{\Delta t} + \frac{1}{R^2}\boldsymbol{u}^n\cdot\nabla(R^2\rho\Delta\delta u) + \frac{1}{R^2}\delta\boldsymbol{u}\cdot\nabla(R^2\rho\Delta u^n) - \theta\nu\Delta^2_{pol}\delta u - \theta^2\Delta t L U$$

- Operator

$$LU \approx \boldsymbol{B}^n\cdot\nabla(\Delta^*(\boldsymbol{B}^n\cdot\nabla\delta u)) + \frac{1}{R}\left([R^2, \delta\boldsymbol{u}\cdot\nabla p^n + \gamma p^n\nabla\cdot\delta\boldsymbol{u}]\right)$$

with $\mathbf{B}^n\cdot\nabla = -\frac{1}{R}[\psi^n,\cdot] + \frac{F_0}{R^2}\partial_\phi$, $\delta\boldsymbol{u}\cdot\nabla p^n = -R[p^n, \delta u]$ and $p^n\nabla\cdot\delta\boldsymbol{u} = -2p^n\partial_Z\delta u$

## Current situation

- Stability of equilibrium started working two days ago. **Next**: Internal Kink instability

## Lattice Boltzmann schemes for MHD

**New work with: P. Helluy, M. Mehrenberger, D. Coulette**

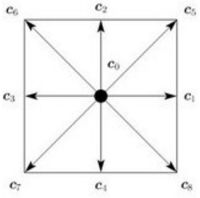
# Lattice Boltzmann schemes

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

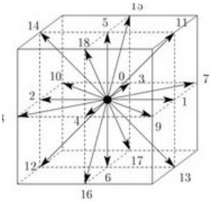
## Lattice Scheme

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

- **Advantage**: local computation relaxation and computation of moments. The matrices computed are linear and sparse.
- **Problem**: physical limitation like small Mach number.




D2Q9




D3Q19

# Lattice Boltzmann schemes for MHD

## Idea

- **Lattice Boltzmann schemes** could be used as a solver or as a preconditioning in the Tokamak context.

## Application to the MHD

- Additional moment: the energy.
- one kinetic equation for the fluid, three kinetic equations for magnetic field.
- More complex or different Lattice to increase the maximum Mach or Reynolds Number.

## Braginskii closure

- The choice of the **relaxation coefficient** allows to choose the viscosity and resistivity coefficients.
- Multiple relaxation method: allows to obtain anisotropic viscosity.

## Future extension

- To simulate instabilities like ELM's with Lattice it is important to extend the scheme for [6] for more general tensor (with gyro-viscous effect) and generalized Ohm law.

# Conclusion and future work

## Future work for physics

- **Short time**: validate the model 199 with kink instability, tearing and ballooning modes.
- **More long time**: Model 303 and x-point geometry.

## Future work for informatics

- **Short time**:
  - Parallelization Open MP-MPI and cleaning
  - New construction of matrices (faster method),
  - Construction of the matrices in the same time.

## Future work for numerics

- **Short time**:
  - Jacobian-free matrices
  - Specific preconditioning using GLT [5]-[6] for advection, diffusion and high-order operators

## Other work

- Improve the Lattice Boltzmann approach (project EXAMAG and INRIA Nancy).

# References


1 O. Czarny and G. Huysmans, *Bézier surface and finite element for MHD simulations*, JCP 2008.

2 L. Chacon *An optimal, parallel, fully implicit Newton Krylov solver for three-dimensional viscoresistive magnetohydrodynamics*, Physic of Plasma (2007).

3 L. Chacon, D. A. Knoll *A 2D high-$\beta$ Hall MHD implicit nonlinear solver*, JCP 2003.

4 S. Serra-Capizzano, *The GLT class as a generalized Fourier analysis and applications*, Linear Algebra Appl. 419 (2006).

5 M. Donatelli, C. Garoni, C. Manni, S. Serra-Capizzano, H. Speleers, *Robust and optimal multi-iterative techniques for IgA Galerkin linear systems*, Comput. Methods Appl. Mech. Engrg. (2015).

6 J. P Dellar, *Lattice Boltzmann formulation for Braginskii magnetohydrodynamics*, Computers and Fluids 46 201-205


EUROfusion


This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement number 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.