Physics-based preconditioning for stiff hyperbolic PDEs

L. Chacón

Oak Ridge National Laboratory

Oak Ridge, TN 37830

BOUT++ Workshop

September 14-16, 2011

Livermore, CA

Outline

- Motivation: the tyranny of scales
- Parabolization: key for SCALABILITY
- 2D reduced resistive MHD
- 2D extended reduced MHD
- BOUT++ physics-based-preconditioning example

Luis Chacón, chaconl@ornl.gov

"The tyranny of scales"
(SBES report, 2006)

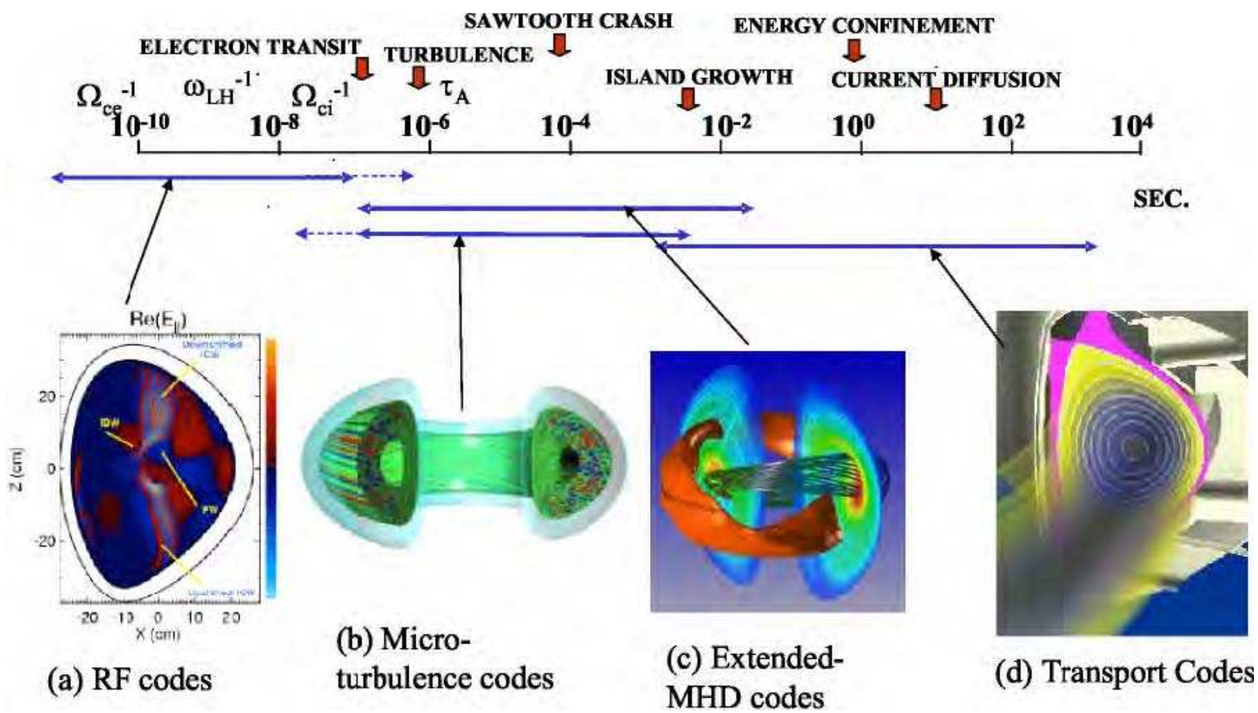


Figure 1: Time scales in fusion plasmas (FSP report)

> "The tyranny of scales will not be simply defeated by building bigger and faster computers" (SBES report, p. 30)

Algorithmic challenges in temporal scale-bridging

- PDE systems of interest typically have mixed character, with strongly hyperbolic and parabolic components.
  - Hyperbolic stiffness (linear and dispersive waves): $\kappa(J) \sim \Delta t\, \omega_{fast} \sim \frac{\Delta t}{\Delta t_{CFL}} \gg 1$
  - Parabolic stiffness (diffusion): $\kappa(J) \sim \frac{\Delta t\, D}{\Delta x^2} \gg 1$
- Bridging the time-scale disparity requires a combination of approaches:
  - Analytical elimination (e.g., reduced models).
  - Well-posed numerical discretization (e.g., asymptotic preserving methods)
  - Some level of implicitness in the temporal formulation (for stability; accuracy requires care).
- Key algorithmic requirement: SCALABILITY

$$\boxed{CPU \sim \mathcal{O}\left(\frac{N}{n_p}\right)}$$

Luis Chacón, chaconl@ornl.gov

Algorithmic scalability vs. parallel scalability

"The tyranny of scales will not be simply defeated by building bigger and faster computers"
(NSF SBES 2006 report, p. 30)

- Optimal algorithm: $\boxed{CPU \sim N/n_p}$.

$$CPU \sim \frac{N^{1+\alpha}}{n_p^{1-\beta}} \; ; \; N = \left(\frac{L}{\delta}\right)^d \begin{cases} \alpha \geq 0, \text{ algorithmic scalability} \\ \beta \geq 0, \text{ parallel scalability} \end{cases}$$

- Much emphasis has been placed on parallel scalability (β).
- However, parallel scalability is limited by the lack of algorithmic scalability:
  - **Weak scaling:** $N \propto n_p \Rightarrow CPU \sim n_p^{\alpha+\beta} \Rightarrow$ **requires** $\alpha = \beta = 0$!

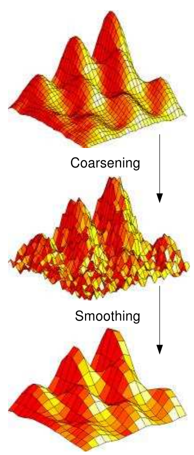
Explicit	Implicit (direct)	Implicit (Krylov iterative)	Implicit (multilevel)
$\alpha = 1/d$	$\alpha = 2 - 2/d$	$\alpha > 1$ (varies)	$\alpha \approx 0$

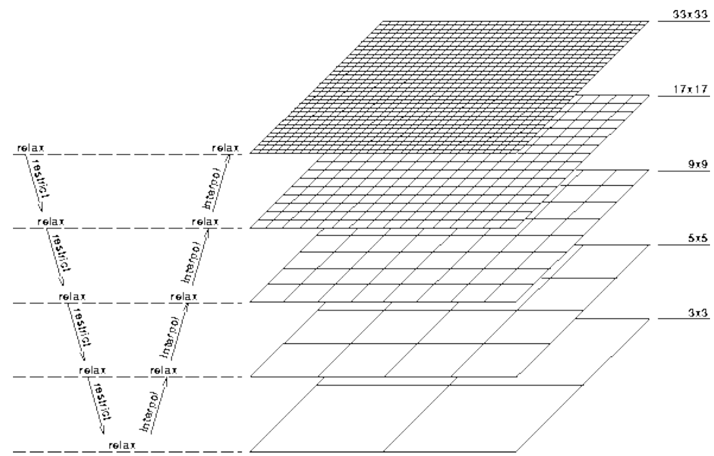
Luis Chacón, chaconl@ornl.gov

How do multilevel (multigrid) methods work?

- MG employs a divide-and-conquer approach to attack error components in the solution.
  - Oscillatory components of the error are “EASY” to deal with (if a SMOOTHER exists)
  - Smooth components are DIFFICULT.

> Idea: coarsen grid to make "smooth" components appear oscillatory, and proceed recursively





- SMOOTHER is make or break of MG!
- Smoothers are hard to find for hyperbolic systems, but fairly easy for parabolic ones:

CAN ONE PARABOLIZE HYPERBOLIC PDEs?

Parabolization: a simple example

- Linear, coupled wave equation:

$$\partial_t u = \partial_x v \ , \ \partial_t v = \partial_x u.$$

- Discretize implicitly in time (but keep in the spatial continuum for now):

$$u^{n+1} = u^n + \Delta t \partial_x v^{n+1}, v^{n+1} = v^n + \Delta t \partial_x u^{n+1}.$$

- Combine equations:

$$\boxed{(I - \Delta t^2 \partial_x^2) u^{n+1} = u^n + \Delta t \partial_x v^n}$$

- Equation is now parabolic!
  - Achieved by implicit discretization ALONE.
  - Implicit discretization automatically parabolizes *unresolved* hyperbolic time scales.

Luis Chacón, chaconl@ornl.gov

Parabolization: a Schur complement perspective

$$u^{n+1} = u^n + \Delta t \partial_x v^{n+1} \; , \; v^{n+1} = v^n + \Delta t \partial_x u^{n+1}$$

- Coupling structure:

$$\begin{bmatrix} \mathrm{I} & -\Delta t \partial_x \\ -\Delta t \partial_x & \mathrm{I} \end{bmatrix} \begin{pmatrix} u^{n+1} \\ v^{n+1} \end{pmatrix} = \begin{pmatrix} u^n \\ v^n \end{pmatrix}$$

- 2×2 block can be formally inverted via block factorization:

$$\begin{bmatrix} D_1 & U \\ L & D_2 \end{bmatrix} = \begin{bmatrix} I & UD_2^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} D_1 - UD_2^{-1}L & 0 \\ 0 & D_2 \end{bmatrix} \begin{bmatrix} I & 0 \\ D_2^{-1}L & I \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} I & U \\ L & I \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ -L & I \end{bmatrix} \begin{bmatrix} (I - UL)^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & -U \\ 0 & I \end{bmatrix}$$

- Only inverse of $I - UL$ (Schur complement) is required! The system has been "PARABOLIZED."

$$\boxed{I - UL = (I - \Delta t^2 \partial_x^2)}$$

Generalization to nonlinear systems: Jacobian-Free Newton-Krylov Methods

- Objective: solve nonlinear system $\vec{G}(\vec{x}^{n+1}) = \vec{0}$ efficiently (scalably).
- Converge nonlinear couplings using Newton-Raphson method: $\left.\frac{\partial \vec{G}}{\partial \vec{x}}\right|_k \delta \vec{x}_k = -\vec{G}(\vec{x}_k)$.
- Jacobian-free implementation: $\left(\frac{\partial \vec{G}}{\partial \vec{x}}\right)_k \vec{y} = J_k \vec{y} = \lim_{\epsilon \to 0} \frac{\vec{G}(\vec{x}_k + \epsilon \vec{y}) - \vec{G}(\vec{x}_k)}{\epsilon}$
- Krylov method of choice: GMRES (nonsymmetric systems).
- Right preconditioning: solve equivalent Jacobian system for $\delta y = P_k \delta \vec{x}$:

$$J_k P_k^{-1} \underbrace{P_k \delta \vec{x}}_{\delta \vec{y}} = -\vec{G}_k$$

- Approximations in preconditioner do not affect accuracy of converged solution; only efficiency!
- **Parabolization+MG is employed ONLY in preconditioning step.**

Oak Ridge National Laboratory

Luis Chacón, chaconl@ornl.gov

Physics-based preconditioner development: a template

- Conceptual formulation:
  1. Understand stiff time scales in your PDE system: normal mode analysis
  2. Identify couplings in PDE system that are responsible for fast time scales
  3. Analyze block coupling structure of Jacobian to identify and target suitable 2×2 blocks
  4. Perform parabolization (Schur factorization) in the linearized, semi-discrete (discrete time, spatial continuum)
- Implementation:
  1. Code full nonlinear residual (based on original nonlinear set of PDEs) to drive JFNK
  2. Code linear, parabolized (approximate) set of PDEs in the preconditioner stage
  3. ESSENTIAL FOR EFFICIENCY: Invert Schur complement using scalable PDE solver, e.g., multigrid.
  4. Done!

Luis Chacón, chaconl@ornl.gov

Limitations of physics-based preconditioning

- The parabolization step requires spatial rediscretization of parabolized PDE.
- As a consequence, there will be spatial truncation error differences between preconditioner and nonlinear residual.
- This fact limits practical nonlinear convergence tolerances:
  - Convergence rate of PBP stalls when nonlinear residual is comparable to the truncation error.
  - It is not of practical interest to converge the nonlinear system any further!
  - Nonlinear *relative* tolerances of 10^{-4} are typical.

Luis Chacón, chaconl@ornl.gov

Application #1: 2D Resistive reduced MHD

L. Chacón, D. A. Knoll and J. M.Finn, *J. Comput. Phys.*, 178, 15-36 (2002)

Equations

- The 2D reduced MHD equations (incompressible flow) are (Alfvénic units):

$$\nabla^2\Phi = \omega \quad (1)$$

$$(\partial_t + \vec{v}\cdot\nabla - \eta\nabla^2)\Psi + E_0 = 0 \quad (2)$$

$$(\partial_t + \vec{v}\cdot\nabla - \nu\nabla^2)\omega + S_\omega = \vec{B}\cdot\nabla(\nabla^2\Psi) \quad (3)$$

  where $\vec{v} = \vec{z}\times\nabla\Phi$; $\vec{B} = \vec{z}\times\nabla\Psi$, η is the resistivity and ν the viscosity.

- This system supports the Alfvén wave, which is a fast normal mode of the system and limits the time step in explicit implementations.
- The domain is a rectangle of size $L_x \times L_y$.
- Differencing: second-order accurate in space and in time.

Physics-based preconditioner for RMHD

- We linearize the RMHD system and eliminate $\delta\omega$:

$$\begin{aligned}
L_\eta \delta\Psi &= \theta \vec{B}_0 \cdot \nabla \delta\Phi - G_\Psi, \\
\left[L_\nu \nabla^2 - \theta(\vec{z} \times \nabla\omega_0) \cdot \nabla\right] \delta\Phi &= \theta[(\vec{B}_0 \cdot \nabla)\nabla^2 - \vec{z} \times \nabla J_0 \cdot \nabla]\delta\Psi - G_\omega + L_\nu(G_\Phi), \\
L_\chi &= \frac{1}{\Delta t} + \theta(\vec{v}_0 \cdot \nabla - \chi\nabla^2)
\end{aligned}$$

- Terms propagating Alfvén wave:

$$\begin{aligned}
\delta\Psi &= \Delta t \vec{B}_0 \cdot \nabla\delta\Phi \\
\nabla^2 \delta\Phi &= \Delta t \vec{B}_0 \cdot \nabla(\nabla^2 \delta\Psi)
\end{aligned}$$

∇^2 complicates matters, because a direct substitution is not possible.

- However, a simplification is possible, and yields the approximation:

$$\begin{aligned}
L_\eta \delta\Psi &= \theta \vec{B}_0 \cdot \nabla \delta\Phi - G_\Psi, \\
L_\nu \delta\Phi &\approx \theta \vec{B}_0 \cdot \nabla \delta\Psi + \underbrace{\nabla^{-2}\left[-G_\omega - L_\nu(-G_\Phi)\right]}_{rhs_\Phi}.
\end{aligned}$$

OAK RIDGE National Laboratory

Physics-based preconditioner for RMHD: Parabolization

- System is simplified further by:

1. Stationary iteration in $\delta\Phi$ (Jacobi).

$$\begin{aligned} L_\eta \delta\Psi^{m+1} &= \theta \vec{B}_0 \cdot \nabla \delta\Phi^{m+1} - G_\Psi, \\ \delta\Phi^{m+1} &\approx \theta D_\nu^{-1} \vec{B}_0 \cdot \nabla \delta\Psi^{m+1} + rhs_\Phi^m. \end{aligned}$$

2. Parabolization:

$$\begin{pmatrix} I & -\theta D_\nu^{-1} \vec{B}_0 \cdot \nabla \\ 0 & P_{SI} \end{pmatrix} \begin{pmatrix} \delta\Phi^{m+1} \\ \delta\Psi^{m+1} \end{pmatrix} = \begin{pmatrix} rhs_\Phi^m \\ -G_\Psi + \theta(\vec{B}_0 \cdot \nabla) rhs_\Phi^m \end{pmatrix}$$

$$P_{SI} = L_\eta - \theta^2 (\vec{B}_0 \cdot \nabla) D_\nu^{-1} (\vec{B}_0 \cdot \nabla)$$

- Only P_{SI} requires nontrivial inversion $\longrightarrow$ MG:
  - Piece-wise constant restriction.
  - Bilinear prolongation.
  - Matrix-free matrix-vector products.
  - Matrix-light point Jacobi smoothing: $u^{s+1} = u^s + D^{-1}(b - Au^s)$

Luis Chacón, chaconl@ornl.gov

Efficiency performance

Iteration count

$$\Delta t = 160\Delta t_{CFL},\ T_{max} = 30\tau_A$$

Grid	$\Delta t(\tau_A)$	n_{Nt}	n_{GM}	$\frac{\text{GMRES}}{\text{time step}}$	CPU time (s)	$\widehat{CPU}$
32x32	15	3.5	6.3	22	3.5	0.16
64x64	7.5	3.25	6	19	31	1.6
128x128	3.75	3	6	18	277	15
256x256	1.875	3	10.3	31	4367	141

CPU time comparison

$$\Delta t = 5\tau_A$$

Grid	CPU_{exp}/CPU	$\Delta t/\Delta t_{exp}$
64x64	4.3	142
128x128	6	294
256x256	7.8	578

Luis Chacón, chaconl@ornl.gov

App. #2: 2D reduced extended MHD

L. Chacón and D. A. Knoll, *J. Comput. Phys.*, 188 (2), 573-592 (2003)

Luis Chacón, chaconl@ornl.gov

Model

- Hall MHD considers the extended Ohm's law for electric field:

$$\vec{E} = -\vec{v} \times \vec{B} + \eta \vec{j} + \frac{d_i}{\rho}(\vec{j} \times \vec{B} - \nabla p_e) - \nu_e \nabla^2 \vec{j}$$

  where d_i is the ion inertial length scale.

- For incompressible ions in 2D:

$$\begin{aligned}
\nabla^2 \Phi &= \omega \\
(\partial_t + \vec{v} \cdot \nabla - \eta \nabla^2 + \eta_2 \nabla^4)\Psi + E_0 &= d_i \vec{B} \cdot \nabla B_z \\
(\partial_t + \vec{v} \cdot \nabla - \eta \nabla^2 + \eta_2 \nabla^4) B_z + \dot{S}_{B_z} &= \vec{B} \cdot \nabla v_z - d_i \vec{B} \cdot \nabla(\nabla^2 \Psi) \\
(\partial_t + \vec{v} \cdot \nabla - \nu \nabla^2) v_z + \dot{S}_{v_z} &= \vec{B} \cdot \nabla B_z \\
(\partial_t + \vec{v} \cdot \nabla - \nu \nabla^2)\omega + \dot{S}_{\omega} &= \vec{B} \cdot \nabla(\nabla^2 \Psi)
\end{aligned}$$

- Supports whistler wave: dispersive ($\omega \propto d_i k^2$)!
- Hyperresistivity $\eta_2 \nabla^4$ is required numerically to damp shortest scales on the grid.

Luis Chacón, chaconl@ornl.gov

Hall MHD physics-based preconditioner

- Based on physics, we assume the following time-step ordering:

$$\Delta t_{CFL}(\sim \frac{1}{\Delta x^2}) \ll \Delta t < \Delta t_{Alfven}(\sim \frac{1}{\Delta x})$$

- SEMI-IMPLICIT approximation of the Jacobian matrix:

$$J_k = \begin{bmatrix} D_\Phi & 0 & 0 & 0 & I \\ L_{\Phi,\Psi} & D_\Psi & U_{B_z,\Psi} & 0 & 0 \\ L_{\Phi,B_z} & L_{\Psi,B_z} & D_{B_z} & U_{v_z,B_z} & 0 \\ L_{\Phi,v_z} & L_{\Psi,v_z} & L_{B_z,v_z} & D_{v_z} & 0 \\ L_{\Phi,\omega} & L_{\Psi,\omega} & 0 & 0 & D_\omega \end{bmatrix} \approx \begin{bmatrix} D_\Phi & 0 & 0 & 0 & 0 \\ L_{\Phi,\Psi} & D_\Psi & U_{B_z,\Psi} & 0 & 0 \\ L_{\Phi,B_z} & L_{\Psi,B_z} & D_{B_z} & 0 & 0 \\ L_{\Phi,v_z} & L_{\Psi,v_z} & L_{B_z,v_z} & D_{v_z} & 0 \\ L_{\Phi,\omega} & L_{\Psi,\omega} & 0 & 0 & D_\omega \end{bmatrix}$$

- PARABOLIZATION via Schur decomposition of red 2x2 block (whistler wave):

$$\begin{bmatrix} D_\Psi & U_{B_z,\Psi} \\ L_{\Psi,B_z} & D_{B_z} \end{bmatrix} = \begin{bmatrix} I & U_{B_z,\Psi} D_{B_z}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} P_{SC}^{\Psi} & 0 \\ 0 & D_{B_z} \end{bmatrix} \begin{bmatrix} I & 0 \\ D_{B_z}^{-1} L_{\Psi,B_z} & I \end{bmatrix},$$

$$P_{SC}^{\Psi} = D_\Psi - U_{\Psi,B_z} D_{B_z}^{-1} L_{\Psi,B_z} \approx D_\Psi + \Delta t \theta^2 d_i^2 (\vec{B}_{p0} \cdot \nabla)^2 \nabla^2.$$

Preconditioner implementation details

- Semi-implicit system:

$$\left[\frac{1}{\Delta t} + \theta \vec{v}_{e0}\cdot\nabla - \theta\eta\nabla^2 + \theta\eta_2\nabla^4 + \Delta t\theta^2 d_i^2(\vec{B}_{p0}\cdot\nabla)^2\nabla^2\right]\delta\Psi = rhs_\Psi .$$

- Coupled MG solver:

$$\begin{aligned}\left[\frac{1}{\Delta t} + \theta \vec{v}_{e0}\cdot\nabla - \theta\eta\nabla^2\right]\delta\Psi + \left[\frac{\eta_2}{\Delta t\theta d_i^2}\nabla^2 + (\vec{B}_{p0}\cdot\nabla)^2\right]\xi &= rhs_\Psi ,\\ \Delta t\theta^2 d_i^2\nabla^2\delta\Psi - \xi &= 0.\end{aligned}$$

  - Matrix-light implementation
  - Block Jacobi smoothing

Luis Chacón, chaconl@ornl.gov

Efficiency performance: grid scaling

$(\Delta t = \Delta t_{Alfven})$

$d_i = 0.2$

Grid	Δt	Newton/Δt	GM/Δt	CPU (s)	CPU_{exp}/CPU	$\Delta t / \Delta t_{CFL}$
64x64	0.02	3.0	0.8	14	3.3	74
128x128	0.01	2.6	0.6	46	8.5	147
256x256	0.005	2.0	0	123	28.0	294

$d_i = 0.4$

Grid	Δt	Newton/Δt	GM/Δt	CPU (s)	CPU_{exp}/CPU	$\Delta t / \Delta t_{CFL}$
64x64	0.02	4.0	4	27	3.1	154
128x128	0.01	3.4	2.2	80	17.2	294
256x256	0.005	3.0	1.2	248	26.0	588

Luis Chacón, chaconl@ornl.gov

Efficiency performance: Δt scaling
$(d_i = 0.2)$

128×128

$\Delta t/\Delta t_A$	Newton/Δt	GM/Δt	CPU (s)	CPU_{exp}/CPU	$\Delta t/\Delta t_{CFL}$
1	2.6	0.6	46	8.5	147
2	3.6	1.8	78	9.4	294
4	4.8	5.8	147	9.3	588

256×256

$\Delta t/\Delta t_A$	Newton/Δt	GM/Δt	CPU (s)	CPU_{exp}/CPU	$\Delta t/\Delta t_{CFL}$
1	2	0	123	28.0	294
2	2.8	0.8	214	30.0	588
4	4.2	3.8	460	26.5	1176

Luis Chacón, chaconl@ornl.gov

A simple example for BOUT++: isothermal sound wave

- Outer "nonlinear" system (1D or multi-D):

$$
\begin{aligned}
\rho^{n+1} &= \rho^n - \Delta t \nabla \cdot \vec{p}^{n+1} \\
\vec{p}^{n+1} &= \vec{p}^n - \Delta t c_s^2 \nabla \rho^{n+1}.
\end{aligned}
$$

  - This PDE system supports a **single time scale**: $\omega = \pm c_s k$ (sound wave).
    - Physics is obliterated for $\Delta t > 1/k$, with $k = 2\pi n/L$.
  - **System is linear**, and hence a single Newton iteration in JFNK is enough.
    - GMRES is needed, since system is not symmetric.

- Preconditioner: $P\delta\vec{x} = \vec{z}$; $\delta\vec{x} = (\delta\rho, \delta\vec{p})$

$$
\begin{aligned}
(I - \Delta t^2 c_s^2 \nabla^2)\delta\rho &= z_\rho - \Delta t \nabla \cdot \vec{z}_p \\
\delta\vec{p} &= -\Delta t c_s^2 \nabla \delta\rho + \vec{z}_p
\end{aligned}
$$

  - Only **one scalar parabolic operator** (magenta) needs be inverted in preconditioner.
  - Use a **scalable solver in PC** (direct solver in 1D, MG in multi-D) for optimal results.
  - **Play with GMRES tolerance** of outer iteration to test impact of truncation error.