On preconditioning variable Poisson equation with extreme contrasts in the coefficients

Àdel Alsalti-Baldellou[1,2], F. Xavier Trias[1], Andrey Gorobets[3] and Assensi Oliva[1]

[1]Heat and Mass Transfer Technological Center, Technical University of Catalonia
C/ Colom 11, 08222 Terrassa (Barcelona), Spain; adel@cttc.upc.edu

[2]Termo Fluids SL, C/ Magí Colet 8, 08204 Sabadell (Barcelona), Spain

[3] Keldysh Institute of Applied Mathematics, 4A, Miusskaya Sq., Moscow 125047, Russia

11-15 January 2021, ECCOMAS 2020

Index

General concepts

Introducing Poisson equation

Arising in multiple situations such as Computational Fluid Dynamics (CFD), heat transfer (HT) simulations or computational electromagnetics (CEM).

General variable coefficients Poisson equation

Let $\rho(\mathbf{r}, t), \phi(\mathbf{r}, t), \psi(\mathbf{r}, t) \in \mathbb{R}$ be scalar fields. Then,

$$\nabla \cdot \left(\frac{1}{\rho} \nabla \phi \right) = \psi$$

Introducing Poisson equation

Arising in multiple situations such as Computational Fluid Dynamics (CFD), heat transfer (HT) simulations or computational electromagnetics (CEM).

General variable coefficients Poisson equation

Let $\rho(\mathbf{r},t), \phi(\mathbf{r},t), \psi(\mathbf{r},t) \in \mathbb{R}$ be scalar fields. Then,

$$\nabla \cdot \left(\frac{1}{\rho} \nabla \phi \right) = \psi$$

Discretized Poisson equation with variable coefficients

Let $\mathsf{M}, \mathsf{G}, \phi_h$ and ψ_h be the discretized divergence, gradient, ϕ and ψ, respectively; and $\mathsf{R} = \text{diag}(\rho_h)$. Then,

$$\mathsf{M}\mathsf{R}^{-1}\mathsf{G}\phi_h = \psi_h$$

Poisson equation in CFD

Governing equations ($\mu \equiv$ ct.)

Navier-Stokes: $$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = \nu \Delta \mathbf{v} - \frac{1}{\rho}\nabla p$$

Incompressibility: $$\nabla \cdot \mathbf{v} = 0$$

Poisson equation in CFD

Governing equations ($\mu \equiv$ ct.)

Navier-Stokes: $$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = \nu \Delta \mathbf{v} - \frac{1}{\rho} \nabla p$$

Incompressibility: $$\nabla \cdot \mathbf{v} = 0$$

Fractional Step Method (FSM)

1. Evaluate the auxiliar vector field $\mathbf{r}(\mathbf{v}^n) := -(\mathbf{v} \cdot \nabla)\mathbf{v} + \nu \Delta \mathbf{v}$
2. Evaluate the predictor velocity $\mathbf{v}^p := \mathbf{v}^n + \Delta t \left(\frac{3}{2}\mathbf{r}(\mathbf{v}^n) - \frac{1}{2}\mathbf{r}(\mathbf{v}^{n-1}) \right)$
3. Obtain the pressure field by solving a **Poisson equation**:

$$\nabla \cdot \left(\frac{1}{\rho} \nabla p^{n+1} \right) = \frac{1}{\Delta t} \nabla \cdot \mathbf{v}^p$$

4. Obtain the new divergence-free velocity $\mathbf{v}^{n+1} = \mathbf{v}^p - \nabla p^{n+1}$

Discretization of Poisson equation

$$\Omega \frac{d\mathbf{v}_h}{dt} = -\mathsf{C}(\mathbf{v}_h)\mathbf{v}_h + \mathsf{N}\mathsf{D}\mathbf{v}_h - \mathsf{R}^{-1}\Omega\mathsf{G}p_h, \text{ with } \begin{cases} \text{Convective operator: } \mathsf{C}(\mathbf{v}_h) \\ \text{Diffusive operator: } \mathsf{D} \\ \text{Mesh volumes: } \Omega = \text{diag}(V_h) \\ \mathsf{R} = \text{diag}(\rho_h),\ \mathsf{N} = \text{diag}(\nu_h) \end{cases}$$

Discretization of Poisson equation

$$\Omega \frac{d\mathbf{v}_h}{dt} = -\mathsf{C}(\mathbf{v}_h)\mathbf{v}_h + \mathsf{N}\mathsf{D}\mathbf{v}_h - \mathsf{R}^{-1}\Omega\mathsf{G}p_h, \text{ with } \begin{cases} \text{Convective operator: } \mathsf{C}(\mathbf{v}_h) \\ \text{Diffusive operator: } \mathsf{D} \\ \text{Mesh volumes: } \Omega = \mathrm{diag}(V_h) \\ \mathsf{R} = \mathrm{diag}(\rho_h), \ \mathsf{N} = \mathrm{diag}(\nu_h) \end{cases}$$

Symmetry-preserving staggered discretization of Navier-Stokes equations

In absence of diffusion ($\mathsf{D} = 0$), global kinetic energy $E_k = \left\langle \frac{1}{2}\mathsf{R}\mathbf{v}_h, \mathbf{v}_h \right\rangle_\Omega$ is conserved if:

$$\frac{dE_k}{dt} = 0$$

Discretization of Poisson equation

$$\Omega \frac{d\mathbf{v}_h}{dt} = -\mathsf{C}(\mathbf{v}_h)\mathbf{v}_h + \mathsf{N}\mathsf{D}\mathbf{v}_h - \mathsf{R}^{-1}\Omega\mathsf{G}p_h, \text{ with } \begin{cases} \text{Convective operator: } \mathsf{C}(\mathbf{v}_h) \\ \text{Diffusive operator: } \mathsf{D} \\ \text{Mesh volumes: } \Omega = \text{diag}(V_h) \\ \mathsf{R} = \text{diag}(\rho_h), \ \mathsf{N} = \text{diag}(\nu_h) \end{cases}$$

Symmetry-preserving staggered discretization of Navier-Stokes equations

In absence of diffusion ($\mathsf{D} = 0$), global kinetic energy $E_k = \left\langle \frac{1}{2}\mathsf{R}\mathbf{v}_h, \mathbf{v}_h \right\rangle_\Omega$ is conserved if:

$$\frac{dE_{k_{\mathsf{C}(\mathbf{v}_h)}}}{dt} + \frac{dE_{k_{\nabla p}}}{dt} = 0$$

Discretization of Poisson equation

$$\Omega \frac{d\mathbf{v}_h}{dt} = -\mathsf{C}(\mathbf{v}_h)\mathbf{v}_h + \mathsf{N}\mathsf{D}\mathbf{v}_h - \mathsf{R}^{-1}\Omega\mathsf{G}p_h, \text{ with } \begin{cases} \text{Convective operator: } \mathsf{C}(\mathbf{v}_h) \\ \text{Diffusive operator: } \mathsf{D} \\ \text{Mesh volumes: } \Omega = \mathrm{diag}(V_h) \\ \mathsf{R} = \mathrm{diag}(\rho_h),\ \mathsf{N} = \mathrm{diag}(\nu_h) \end{cases}$$

Symmetry-preserving staggered discretization of Navier-Stokes equations

In absence of diffusion ($\mathsf{D} = 0$), global kinetic energy $E_k = \left\langle \frac{1}{2}\mathsf{R}\mathbf{v}_h, \mathbf{v}_h \right\rangle_\Omega$ is conserved if:

$$\frac{dE_{k_{\mathsf{C}(\mathbf{v}_h)}}}{dt} + \frac{dE_{k_{\nabla p}}}{dt} = 0$$

$$\frac{dE_{k_{\nabla p}}}{dt} \stackrel{(*)}{=} 0$$

$(*)$ Symmetry-preserving discrete gradient[1] satisfies: $\mathsf{G} = -\Omega^{-1}\mathsf{M}^t$.

[1] Roel W.C.P. Verstappen and Arthur E.P. Veldman. "Symmetry-preserving discretization of turbulent flow". In: *Journal of Computational Physics* 187.1 (2003), pp. 343–368. ISSN: 00219991. DOI: 10.1016/S0021-9991(03)00126-8

Constant vs Variable coefficients Poisson equation

Combining FSM with a symmetry-preserving discretization leads to:

- $\rho \equiv$ ct. $\Rightarrow$ Constant Poisson equation:

$$\mathsf{L}p = \rho \mathsf{M} v^p, \text{ where } \mathsf{L} = \mathsf{M}\mathsf{G}$$

- $\rho \not\equiv$ ct. $\Rightarrow$ Variable coefficients Poisson equation:

$$\tilde{\mathsf{L}}p = \mathsf{M} v^p, \text{ where } \tilde{\mathsf{L}} := \mathsf{M}\mathsf{R}^{-1}\mathsf{G}$$

Constant vs Variable coefficients Poisson equation

Combining FSM with a symmetry-preserving discretization leads to:

- $\rho \equiv$ ct. $\Rightarrow$ Constant Poisson equation:

$$\mathsf{L}p = \rho \mathsf{M} v^p, \text{ where } \mathsf{L} = \mathsf{M}\mathsf{G} = -\mathsf{M}\Omega^{-1}\mathsf{M}^t$$

- $\rho \not\equiv$ ct. $\Rightarrow$ Variable coefficients Poisson equation:

$$\tilde{\mathsf{L}}p = \mathsf{M} v^p, \text{ where } \tilde{\mathsf{L}} := \mathsf{M}\mathsf{R}^{-1}\mathsf{G} = -\mathsf{M}\mathsf{R}^{-1}\Omega^{-1}\mathsf{M}^t$$

Constant vs Variable coefficients Poisson equation

Combining FSM with a symmetry-preserving discretization leads to:

- $\rho \equiv \text{ct.} \Rightarrow$ Constant Poisson equation:

$$\mathsf{L}p = \rho \mathsf{M} v^p, \text{ where } \mathsf{L} = \mathsf{M}\mathsf{G} = -\mathsf{M}\Omega^{-1}\mathsf{M}^t$$

- $\rho \not\equiv \text{ct.} \Rightarrow$ Variable coefficients Poisson equation:

$$\tilde{\mathsf{L}}p = \mathsf{M} v^p, \text{ where } \tilde{\mathsf{L}} \coloneqq \mathsf{M}\mathsf{R}^{-1}\mathsf{G} = -\mathsf{M}\mathsf{R}^{-1}\Omega^{-1}\mathsf{M}^t$$

Indeed, defining $\tilde{\Omega} \coloneqq \Omega\mathsf{R}$:

$$\tilde{\mathsf{L}} = -\mathsf{M}\tilde{\Omega}^{-1}\mathsf{M}^t$$

Poisson solvers in modern HPC systems

Direct solvers

Numerical methods that directly compute the exact solution (up to machine precision), such as LU or Cholesky factorization methods.

Iterative solvers

Numerical methods that iteratively approximate the exact solution. Further divided into:

- Stationary: Relaxation methods such as Jacobi or Gauss-Seidel methods.
- Non-stationary: such as Krylov subspace methods, e.g. CG, GMRES, BICGSTAB...

Poisson solvers in modern HPC systems

Direct solvers

Numerical methods that directly compute the exact solution (up to machine precision).

- Pros: Case-independent performance and machine accuracy.
- Cons: High memory requirements and very high complexity.

Iterative solvers

Numerical methods that iteratively approximate the exact solution.

- Pros: Highly parallelizable and, in many cases, much faster (especially considering well-conditioned large sparse systems).
- Cons: Less robust, convergence highly affected by the system.

Poisson solvers in modern HPC systems

Iterative solvers

- Pros: Highly parallelizable and, in many cases, much faster (especially considering well-conditioned large sparse systems).
- Cons: Less robust, convergence highly affected by the system.

Conjugate Gradient method

- Direct method converging to the solution after n steps (in exact arithmetic), being n the number of unknowns.
- Very low memory requirements.
- Lower computational costs per iteration compared to other Krylov subspace methods.
- Intrinsically only applicable to symmetric positive-definite (SPD) matrices.
- Convergence theorem:

$$\|e_k\|_A \leq 2\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^k \|e_0\|_A, \text{ where } \kappa(A)=\frac{\lambda_{max}(A)}{\lambda_{min}(A)}$$

Preconditioning techniques

Left, right and split preconditioning

Given the linear system $Ax = b$ and the preconditioner $M = M_1 M_2$, we can consider the following preconditioning techniques:

Left preconditioning:	$M^{-1}Ax = M^{-1}b$
Right preconditioning:	$AM^{-1}y = b,$ where $Mx = y$
Split preconditioning:	$M_1^{-1}AM_2^{-1}y = M_1^{-1}b,$ where $M_2 x = y$

Preconditioning techniques

Left, right and split preconditioning

Given the linear system $Ax = b$ and the preconditioner $M = M_1 M_2$, we can consider the following preconditioning techniques:

Left preconditioning:	$M^{-1}Ax = M^{-1}b$
Right preconditioning:	$AM^{-1}y = b$, where $Mx = y$
Split preconditioning:	$M_1^{-1}AM_2^{-1}y = M_1^{-1}b$, where $M_2 x = y$

Thus, applying a preconditioner:

- reduces to operations of the type $y = M^{-1}x$.
- is intended to improve the convergence of iterative solvers by modifying the spectrum of the system: $\kappa(M^{-1}A) < \kappa(A)$. Indeed,

$$M^{-1} \simeq A^{-1} \Rightarrow \kappa(M^{-1}A) \simeq \kappa(\mathbb{I}) = 1.$$

- needs to seek a balance between building/application costs and reduction in the number of iterations.
- if the solver being used requires the system to satisfy a certain condition, then the preconditioned system needs to satisfy it, too.

Variable Poisson equation with extreme contrasts in the coefficients

Recalling variable coefficients Poisson equation:

$$\tilde{\mathsf{L}} p = \mathsf{M} v^p,$$

where:

$$\tilde{\mathsf{L}} \coloneqq \mathsf{M}\mathsf{R}^{-1}\mathsf{G}$$

Variable Poisson equation with extreme contrasts in the coefficients

Recalling variable coefficients Poisson equation:

$$\tilde{\mathsf{L}}p = \mathsf{M}v^p,$$

where:

$$\tilde{\mathsf{L}} \coloneqq \mathsf{M}\mathsf{R}^{-1}\mathsf{G} \overset{\mathsf{G}=-\Omega^{-1}\mathsf{M}^t}{=} -\mathsf{M}\mathsf{R}^{-1}\Omega^{-1}\mathsf{M}^t$$

Variable Poisson equation with extreme contrasts in the coefficients

Recalling variable coefficients Poisson equation:

$$\tilde{\mathsf{L}}p = \mathsf{M}v^p,$$

where:

$$\tilde{\mathsf{L}} := \mathsf{M}\mathsf{R}^{-1}\mathsf{G} \overset{\mathsf{G}=-\Omega^{-1}\mathsf{M}^t}{=} -\mathsf{M}\mathsf{R}^{-1}\Omega^{-1}\mathsf{M}^t \overset{\tilde{\Omega}:=\Omega\mathsf{R}}{=} -\mathsf{M}\tilde{\Omega}^{-1}\mathsf{M}^t.$$

Variable Poisson equation with extreme contrasts in the coefficients

Recalling variable coefficients Poisson equation:

$$\tilde{\mathsf{L}} p = \mathsf{M} v^p,$$

where:

$$\tilde{\mathsf{L}} \coloneqq \mathsf{M}\mathsf{R}^{-1}\mathsf{G} \overset{\mathsf{G}=-\Omega^{-1}\mathsf{M}^t}{=} -\mathsf{M}\mathsf{R}^{-1}\Omega^{-1}\mathsf{M}^t \overset{\tilde{\Omega}:=\Omega\mathsf{R}}{=} -\mathsf{M}\tilde{\Omega}^{-1}\mathsf{M}^t.$$

Hence:

High contrasts in Ω or $\mathsf{R} \Rightarrow$ High contrasts in $\tilde{\Omega} \Rightarrow$ High contrasts in $\tilde{\mathsf{L}}$

Multiphase flow testcase for ratio $\in \{1, 10^2, 10^4, 10^6\}$

Idealized parameters

- Dynamic viscosity: $\mu = 10^{-4}$ Ns/m^2
- Surface tension: $\sigma = \rho_1/1000$ N/m
- Density: $\begin{cases} \rho_0 = 1.0, & \text{internal fluid} \\ \rho_1 = \text{ratio}^{-1}, & \text{external fluid} \end{cases}$ $\left(\text{kg/m}^3\right)$
- Initial ellipse axis: $(a, b) = (1.0\text{m}, 0.5\text{m})$
- Homogeneous mesh $\Rightarrow \Omega = (\Delta x \Delta y \Delta z)\,\mathbb{I}$ and $\tilde{\Omega} = (\Delta x \Delta y \Delta z)\,\mathrm{R}$

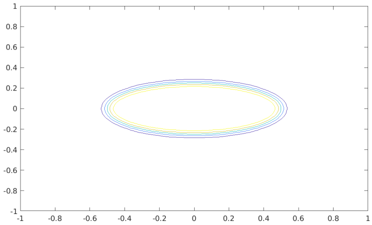


Figure: Initial "bubble" configuration.

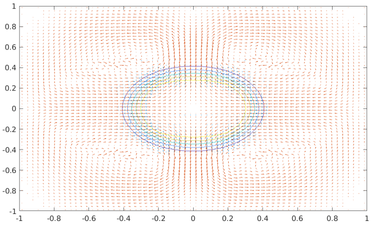


Figure: Evolved "bubble".

Spectrum of $\tilde{L} = -M\tilde{\Omega}^{-1}M^t$ for various ratios

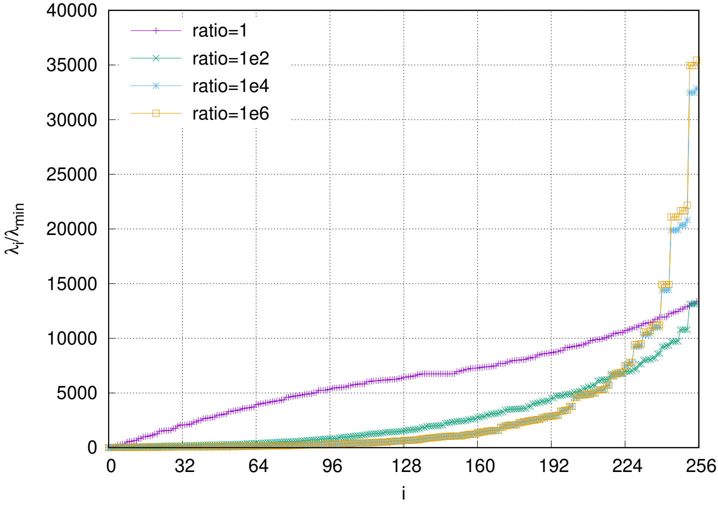


Figure: Normalized spectrum of $\tilde{L}$ for various density ratios on a 16×16 mesh.

Variable Poisson equation with extreme contrasts in the coefficients

$$\text{Variable high contrasts in } \Omega \text{ or } \mathrm{R} \Rightarrow \tilde{\mathrm{L}} \text{ is: } \begin{cases} \text{very ill-conditioned} \\ \text{variable} \\ \text{possibly not built explicitly} \end{cases}$$

Variable Poisson equation with extreme contrasts in the coefficients

$$\text{Variable high contrasts in } \Omega \text{ or } R \Rightarrow \tilde{L} \text{ is: } \begin{cases} \text{very ill-conditioned} \\ \text{variable} \\ \text{possibly not built explicitly} \end{cases}$$

$\Rightarrow$ Preconditioning becomes crucial to use iterative methods.

Variable Poisson equation with extreme contrasts in the coefficients

$$\text{Variable high contrasts in } \Omega \text{ or } R \Rightarrow \tilde{L} \text{ is: } \begin{cases} \text{very ill-conditioned} \\ \text{variable} \\ \text{possibly not built explicitly} \end{cases}$$

$\Rightarrow$ Preconditioning becomes crucial to use iterative methods.

Indeed:

$$\tilde{L} \text{ is: } \begin{cases} \text{very ill-conditioned} \\ \text{variable} \\ \text{possibly not built explicitly} \end{cases}$$

Variable Poisson equation with extreme contrasts in the coefficients

Variable high contrasts in Ω or R $\Rightarrow$ $\tilde{\mathsf{L}}$ is: $\begin{cases} \text{very ill-conditioned} \\ \text{variable} \\ \text{possibly not built explicitly} \end{cases}$

$\Rightarrow$ Preconditioning becomes crucial to use iterative methods.

Indeed:

$$\tilde{\mathsf{L}} \text{ is: } \begin{cases} \text{very ill-conditioned} \\ \text{variable} \\ \text{possibly not built explicitly} \end{cases} \quad \Rightarrow \quad M^{-1} \simeq \tilde{\mathsf{L}}^{-1} \text{ is required}$$

Variable Poisson equation with extreme contrasts in the coefficients

$$\text{Variable high contrasts in } \Omega \text{ or } \mathsf{R} \Rightarrow \tilde{\mathsf{L}} \text{ is: } \begin{cases} \text{very ill-conditioned} \\ \text{variable} \\ \text{possibly not built explicitly} \end{cases}$$

$\Rightarrow$ Preconditioning becomes crucial to use iterative methods.

Indeed:

$$\tilde{\mathsf{L}} \text{ is: } \begin{cases} \text{very ill-conditioned} & \Rightarrow \quad M^{-1} \simeq \tilde{\mathsf{L}}^{-1} \text{ is required} \\ \text{variable} & \Rightarrow \quad M \text{ is variable} \\ \text{possibly not built explicitly} & \end{cases}$$

Variable Poisson equation with extreme contrasts in the coefficients

$$\text{Variable high contrasts in } \Omega \text{ or R} \Rightarrow \tilde{\mathsf{L}} \text{ is: } \begin{cases} \text{very ill-conditioned} \\ \text{variable} \\ \text{possibly not built explicitly} \end{cases}$$

$\Rightarrow$ Preconditioning becomes crucial to use iterative methods.

Indeed:

$$\tilde{\mathsf{L}} \text{ is: } \begin{cases} \text{very ill-conditioned} & \Rightarrow & M^{-1} \simeq \tilde{\mathsf{L}}^{-1} \text{ is required} \\ \text{variable} & \Rightarrow & M \text{ is variable} \\ \text{possibly not built explicitly} & \Rightarrow & M \text{ shouldn't require full } \tilde{\mathsf{L}} \end{cases}$$

Variable Poisson equation with extreme contrasts in the coefficients

$$\text{Variable high contrasts in } \Omega \text{ or } \mathsf{R} \Rightarrow \tilde{\mathsf{L}} \text{ is: } \begin{cases} \text{very ill-conditioned} \\ \text{variable} \\ \text{possibly not built explicitly} \end{cases}$$

$\Rightarrow$ Preconditioning becomes crucial to use iterative methods.

Indeed:

$$\tilde{\mathsf{L}} \text{ is: } \begin{cases} \text{very ill-conditioned} & \Rightarrow & M^{-1} \simeq \tilde{\mathsf{L}}^{-1} \text{ is required} \\ \text{variable} & \Rightarrow & M \text{ is variable} \\ \text{possibly not built explicitly} & \Rightarrow & M \text{ shouldn't require full } \tilde{\mathsf{L}} \end{cases}$$

Arising not only in multiphase flows but also in many other situations such as: oil reservoir simulations, electromagnetics modeling or under AMR with high mesh aspect ratios, among others.

The preconditioner itself

Introducing Jacobi preconditioner

Jacobi preconditioner

Given the linear system $\tilde{\mathsf{L}}x = b$, Jacobi preconditioner is defined as:

$$M_{\mathsf{Jac}} = \mathsf{diag}(\tilde{\mathsf{L}}).$$

Introducing Jacobi preconditioner

Jacobi preconditioner

Given the linear system $\tilde{\mathsf{L}}x = b$, Jacobi preconditioner is defined as:

$$M_{\mathsf{Jac}} = \mathsf{diag}(\tilde{\mathsf{L}}).$$

- Pros:
  - If $\tilde{\mathsf{L}}$ is available, cheap to build.
  - Easily invertible and highly parallelizable.
  - Can be used with CG, given that by definition M_{Diag} is SPD.
  - Extremely easy to implement.
- Cons:
  - Requires full matrix $\tilde{\mathsf{L}}$.
  - In many cases, doesn't really improve convergence.

Introducing Jacobi preconditioner

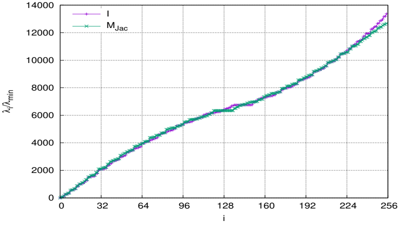
Jacobi preconditioner

Given the linear system $\tilde{\mathsf{L}}x = b$, Jacobi preconditioner is defined as:

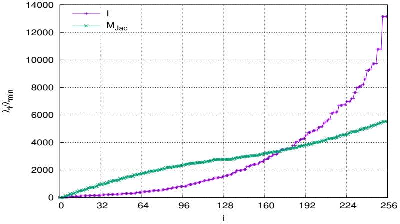
$$M_{\mathsf{Jac}} = \mathsf{diag}(\tilde{\mathsf{L}}).$$

- Pros:
  - If $\tilde{\mathsf{L}}$ is available, cheap to build.
  - Easily invertible and highly parallelizable.
  - Can be used with CG, given that by definition M_{Diag} is SPD.
  - Extremely easy to implement.
  - **Well-suited for high-ratio Poisson equation.**
- Cons:
  - Requires full matrix $\tilde{\mathsf{L}}$.
  - In many cases, doesn't really improve convergence.

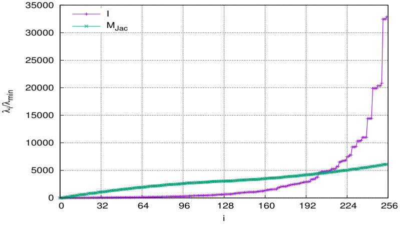
Spectrum of $M^{-1}\tilde{\mathsf{L}}$ for various preconditioners and ratios



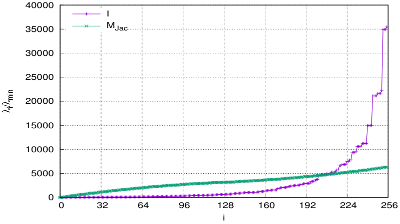
(a) ratio $= 1$



(b) ratio $= 10^2$



(c) ratio $= 10^4$



(d) ratio $= 10^6$

Figure: Normalized spectrum of $M^{-1}\tilde{\mathsf{L}}$ for $M \in \{\mathbb{I}, M_{\mathsf{Jac}}\}$.

Our proposal as a feasible alternative to the Jacobi preconditioner

Our proposal: an adaptive diagonal preconditioner

Given the linear system $\tilde{\mathsf{L}}x = b$, our adaptive diagonal preconditioner is defined as:

$$M_{\mathsf{Diag}} = \tilde{\Omega}^{-1}.$$

Our proposal as a feasible alternative to the Jacobi preconditioner

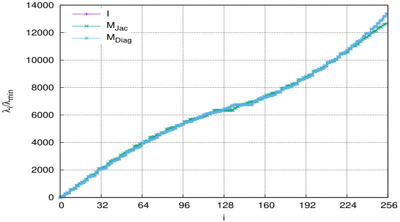
Our proposal: an adaptive diagonal preconditioner

Given the linear system $\tilde{\mathsf{L}}x = b$, our adaptive diagonal preconditioner is defined as:

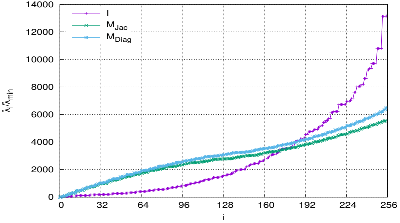
$$M_{\mathsf{Diag}} = \tilde{\Omega}^{-1}.$$

- Pros:
  - Does not require $\tilde{\mathsf{L}}$ (only R and Ω).
  - "Free" to build, as is based on available fields R and Ω.
  - Easily invertible and highly parallelizable.
  - Can be used with CG, given that by definition M_{Diag} is SPD.
  - Extremely easy to implement.
- Cons:
  - Compared to M_{Jac}, it requires one extra diagonal matrix product (if Ω and R are both considered).
  - For lower contrasts in the coefficients, it doesn't improve much the convergence.

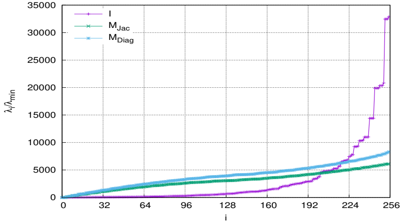
Spectrum of $M^{-1}\tilde{\mathsf{L}}$ for various preconditioners and ratios



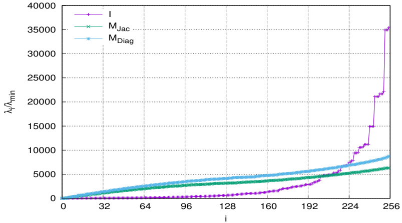
(a) ratio $= 1$



(b) ratio $= 10^2$



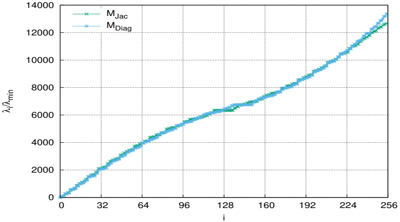
(c) ratio $= 10^4$



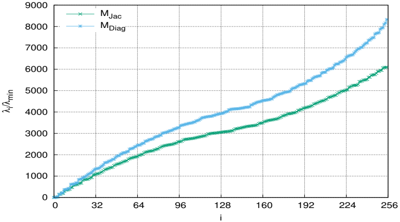
(d) ratio $= 10^6$

Figure: Normalized spectrum of $M^{-1}\tilde{\mathsf{L}}$ for $M \in \left\{\mathbb{I}, M_{\mathsf{Jac}}, M_{\mathsf{Diag}}\right\}$.

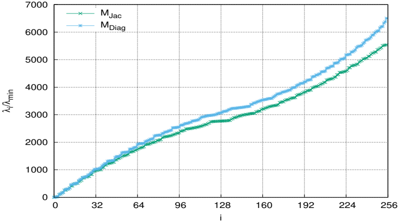
Spectrum of $M^{-1}\tilde{\mathsf{L}}$ for various preconditioners and ratios



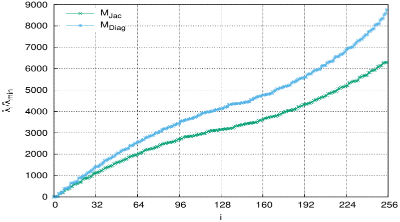
(a) ratio $= 1$



(c) ratio $= 10^4$



(b) ratio $= 10^2$



(d) ratio $= 10^6$

Figure: Normalized spectrum of $M^{-1}\tilde{\mathsf{L}}$ for $M \in \left\{M_{\mathsf{Jac}}, M_{\mathsf{Diag}}\right\}$.

Our proposal in combination with constant preconditioners

Combination of constant and adaptive diagonal preconditioners

Given the linear system $\tilde{\mathsf{L}}x = b$, and a constant preconditioner $M_{\mathsf{L}} = L_{\mathsf{L}} L_{\mathsf{L}}^t$ based on $\mathsf{L} = \mathsf{MG}$, our adaptive diagonal preconditioner can be applied to $\tilde{\mathsf{L}}$ as:

$$\tilde{M}_{\mathsf{Diag}} = L_{\mathsf{L}} M_{\mathsf{Diag}} L_{\mathsf{L}}^t.$$

Our proposal in combination with constant preconditioners

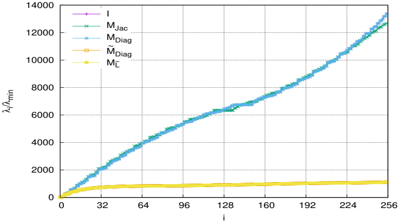
Combination of constant and adaptive diagonal preconditioners

Given the linear system $\tilde{\mathsf{L}}x = b$, and a constant preconditioner $M_{\mathsf{L}} = L_{\mathsf{L}} L_{\mathsf{L}}^t$ based on $\mathsf{L} = \mathsf{MG}$, our adaptive diagonal preconditioner can be applied to $\tilde{\mathsf{L}}$ as:

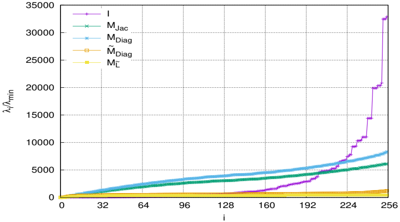
$$\tilde{M}_{\mathsf{Diag}} = L_{\mathsf{L}} M_{\mathsf{Diag}} L_{\mathsf{L}}^t.$$

- Pros:
  - Does not require $\tilde{\mathsf{L}}$ (only L, R and Ω).
  - Compatible with more complex preconditioners, as they only need to be calculated once.
  - Achieves further improvements in convergence compared to M_{Jac} and M_{Diag} thanks to the constant preconditioner M_{L}.
- Cons:
  - Compared to M_{L}, it requires two extra diagonal matrix products (if Ω and R are both considered).
  - It will always work worse than $M_{\tilde{\mathsf{L}}}$, a variable (and unaffordable) version of M_{L} based on $\tilde{\mathsf{L}}$.

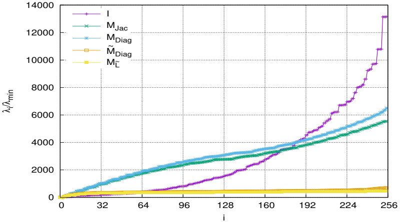
Spectrum of $M^{-1}\tilde{\mathsf{L}}$ for various preconditioners and ratios



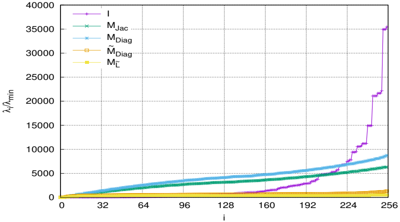
(a) ratio $= 1$



(c) ratio $= 10^4$



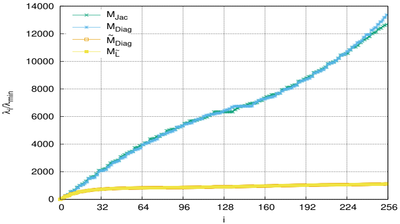
(b) ratio $= 10^2$



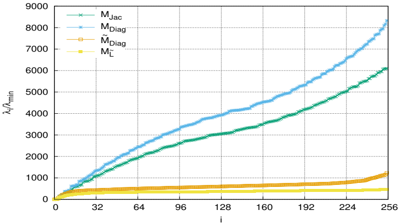
(d) ratio $= 10^6$

Figure: Normalized spectrum of $M^{-1}\tilde{\mathsf{L}}$ for $M \in \left\{\mathbb{I}, M_{\mathsf{Jac}}, M_{\mathsf{Diag}}, \tilde{M}_{\mathsf{Diag}}, M_{\tilde{\mathsf{L}}}\right\}$.

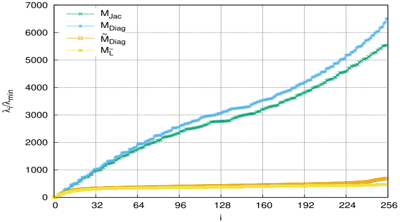
Spectrum of $M^{-1}\tilde{\mathrm{L}}$ for various preconditioners and ratios



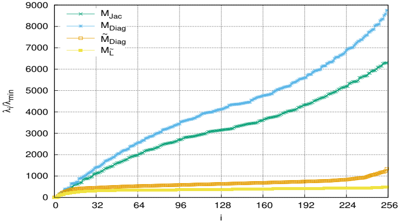
(a) ratio $= 1$



(c) ratio $= 10^4$



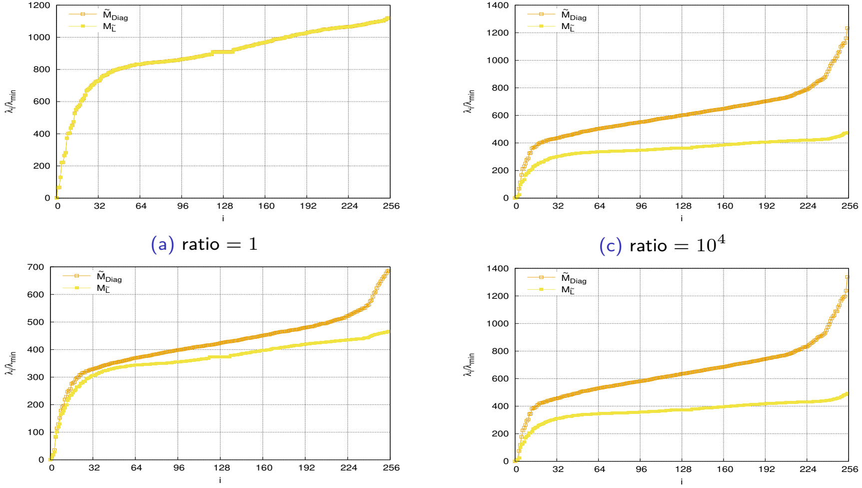
(b) ratio $= 10^2$



(d) ratio $= 10^6$

Figure: Normalized spectrum of $M^{-1}\tilde{\mathrm{L}}$ for $M \in \left\{M_{\mathrm{Jac}}, M_{\mathrm{Diag}}, \tilde{M}_{\mathrm{Diag}}, M_{\tilde{\mathrm{L}}}\right\}$.

Spectrum of $M^{-1}\tilde{\mathsf{L}}$ for various preconditioners and ratios



(a) ratio $= 1$

(c) ratio $= 10^4$

(b) ratio $= 10^2$

(d) ratio $= 10^6$

Figure: Normalized spectrum of $M^{-1}\tilde{\mathsf{L}}$ for $M \in \left\{\tilde{M}_{\text{Diag}}, M_{\tilde{\mathsf{L}}}\right\}$.

Numerical results

Numerical results

	tol $= 1.0\mathrm{e}{-6}$				
ratio	$\mathbb{I}$	M_{Diag}	M_{Jac}	$\tilde{M}_{Diag}$	$M_{\tilde{\mathsf{L}}}$
1	93	93	91	28	28
10^2	310	83	80	33	23
10^4	2828	86	86	32	22
10^6	30286	154	152	74	64

	tol $= 1.0\mathrm{e}{-8}$				
ratio	$\mathbb{I}$	M_{Diag}	M_{Jac}	$\tilde{M}_{Diag}$	$M_{\tilde{\mathsf{L}}}$
1	285	285	282	100	100
10^2	625	313	310	119	93
10^4	4024	323	318	142	98
10^6	37711	358	351	157	108

Table: Number of iterations required by PCG to solve the variable coefficients Poisson equation arising from the testcase for various preconditioners, ratios and convergence criteria. All tests are performed on a 64×64 mesh and convergence is achieved when the relative residual is smaller than the tolerance: $|b - \tilde{\mathsf{L}} x_k| / |b - \tilde{\mathsf{L}} x_0| <$ tol.

Concluding remarks

Conclusions

- M_{Diag} has been proposed as a **computationally cheaper alternative to the Jacobi preconditioner**, not requiring $\tilde{\mathsf{L}}$ to be built, being extremely easy to implement and leading to comparable reductions in the number of iterations.

- $\tilde{M}_{\text{Diag}}$ has been proposed as a **computationally affordable variable version of more complex fixed preconditioners** (based on L rather than $\tilde{\mathsf{L}}$), not requiring $\tilde{\mathsf{L}}$ to be built and leading to comparable reductions in the number of iterations (with respect to its analogue based on $\tilde{\mathsf{L}}$).

Conclusions

- M_{Diag} has been proposed as a **computationally cheaper alternative to the Jacobi preconditioner**, not requiring $\tilde{\mathrm{L}}$ to be built, being extremely easy to implement and leading to comparable reductions in the number of iterations.

- $\tilde{M}_{\text{Diag}}$ has been proposed as a **computationally affordable variable version of more complex fixed preconditioners** (based on L rather than $\tilde{\mathrm{L}}$), not requiring $\tilde{\mathrm{L}}$ to be built and leading to comparable reductions in the number of iterations (with respect to its analogue based on $\tilde{\mathrm{L}}$).

- Especially **for higher ratios of the coefficients, very important reductions in the number of iterations have been shown** for all the preconditioners considered.

- Numerical experiments confirm that **the preconditioners we propose achieve similar rates of convergence while being better suit for variable (in time) problems**.

Future lines of work

- Implementation of M_{Diag} and $\tilde{M}_{\text{Diag}}$ to real simulation codes to **quantify the reduction in the execution time of the simulations** based on variable Poisson equation with high (and not necessarily extreme) contrasts in the coefficients.

Future lines of work

- Implementation of M_{Diag} and $\tilde{M}_{\text{Diag}}$ to real simulation codes to **quantify the reduction in the execution time of the simulations** based on variable Poisson equation with high (and not necessarily extreme) contrasts in the coefficients.
- Study the impact of face-to-cell interpolators in M_{Diag} and $\tilde{M}_{\text{Diag}}$.
- Study other possible combinations of M_{Diag} with more complex fixed preconditioners (based on L rather than $\tilde{\mathsf{L}}$).

Future lines of work

- Implementation of M_{Diag} and $\tilde{M}_{\text{Diag}}$ to real simulation codes to **quantify the reduction in the execution time of the simulations** based on variable Poisson equation with high (and not necessarily extreme) contrasts in the coefficients.
- Study the impact of face-to-cell interpolators in M_{Diag} and $\tilde{M}_{\text{Diag}}$.
- Study other possible combinations of M_{Diag} with more complex fixed preconditioners (based on L rather than $\tilde{\mathsf{L}}$).
- Study ways to **combine M_{Diag} with deflation techniques applied to the variable matrix** $\tilde{\mathsf{L}}$. Thus, finding efficient and highly parallelizable ways to compute updated deflation vectors, similarly to what was proposed by van der Linden et al.[2]

[2] J.H. van der Linden et al. "The parallel subdomain-levelset deflation method in reservoir simulation". In: *Journal of Computational Physics* 304 (Jan. 2016), pp. 340–358. ISSN: 00219991. DOI: 10.1016/j.jcp.2015.10.016

Thanks for your attention!