On preconditioning variable Poisson equation with extreme contrasts in the coefficients

Àdel Alsalti-Baldellou\textsuperscript{1,2}, F. Xavier Trias\textsuperscript{1}, Andrey Gorobets\textsuperscript{3} and Assensi Oliva\textsuperscript{1}

\textsuperscript{1}Heat and Mass Transfer Technological Center, Technical University of Catalonia
\hspace{1em} C/ Colom 11, 08222 Terrassa (Barcelona), Spain; adel@cttc.upc.edu

\textsuperscript{2}Termo Fluids SL, C/ Magí Colet 8, 08204 Sabadell (Barcelona), Spain

\textsuperscript{3}Keldysh Institute of Applied Mathematics, 4A, Miusskaya Sq., Moscow 125047, Russia

11-15 January 2021, ECCOMAS 2020
1 General concepts
   - Poisson equation in CFD
   - Poisson solvers and HPC
   - The need for preconditioning
   - High-ratio Poisson equation
2 The preconditioner itself
   - Arising from Jacobi preconditioner
   - In combination with constant preconditioners
3 Numerical results
4 Concluding remarks
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 $\rho(r, t), \phi(r, t), \psi(r, t) \in \mathbb{R}$ 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 $\rho(r, t), \phi(r, t), \psi(r, t) \in \mathbb{R}$ be scalar fields. Then,

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

**Discretized Poisson equation with variable coefficients**

Let $M, G, \phi_h$ and $\psi_h$ be the discretized divergence, gradient, $\phi$ and $\psi$, respectively; and $R = \text{diag}(\rho_h)$. Then,

$$MR^{-1}G\phi_h = \psi_h$$
Poisson equation in CFD

**Governing equations \( (\mu \equiv \text{ct.}) \)**

- 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 ($\mu \equiv \text{ct.}$)

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 auxiliary vector field \( r(\mathbf{v}^n) := - (\mathbf{v} \cdot \nabla)\mathbf{v} + \nu \Delta \mathbf{v} \)
2. Evaluate the predictor velocity \( \mathbf{v}^p := \mathbf{v}^n + \Delta t \left( \frac{3}{2} r(\mathbf{v}^n) - \frac{1}{2} r(\mathbf{v}^{n-1}) \right) \)
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

\[ \Omega \frac{d\mathbf{v}_h}{dt} = -\mathbf{C}(\mathbf{v}_h)\mathbf{v}_h + \mathbf{N}\mathbf{D}\mathbf{v}_h - \mathbf{R}^{-1}\mathbf{G}\mathbf{p}_h, \text{ with} \]

\[
\left\{
\begin{array}{l}
\text{Convective operator: } \mathbf{C}(\mathbf{v}_h) \\
\text{Diffusive operator: } \mathbf{D} \\
\text{Mesh volumes: } \Omega = \text{diag}(\mathbf{V}_h) \\
\text{R = diag}(\rho_h), \text{ N = diag}(\nu_h) \\
\end{array}
\right.
\]
Discretization of Poisson equation

\[ \Omega \frac{d\mathbf{v}_h}{dt} = -C(\mathbf{v}_h)\mathbf{v}_h + ND\mathbf{v}_h - R^{-1} \Omega G p_h, \quad \text{with} \]

\[ \begin{aligned}
\text{Convective operator: } & C(\mathbf{v}_h) \\
\text{Diffusive operator: } & D \\
\text{Mesh volumes: } & \Omega = \text{diag}(V_h) \\
\text{ } & R = \text{diag}(\rho_h), \; \text{N} = \text{diag}(\nu_h)
\end{aligned} \]

Symmetry-preserving staggered discretization of Navier-Stokes equations

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

\[ \frac{dE_k}{dt} = 0 \]
Discretization of Poisson equation

\( \Omega \frac{d\mathbf{v}_h}{dt} = -C(\mathbf{v}_h)\mathbf{v}_h + N D \mathbf{v}_h - R^{-1} \Omega G p_h \), with

\[
\begin{align*}
\text{Convective operator: } & C(\mathbf{v}_h) \\
\text{Diffusive operator: } & D \\
\text{Mesh volumes: } & \Omega = \text{diag}(V_h) \\
\text{R = diag}(\rho_h), & N = \text{diag}(\nu_h)
\end{align*}
\]

Symmetry-preserving staggered discretization of Navier-Stokes equations

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

\[
\frac{dE_{kC(\mathbf{v}_h)}}{dt} + \frac{dE_{k\nabla p}}{dt} = 0
\]
Discretization of Poisson equation

\[ \Omega \frac{d\mathbf{v}_h}{dt} = -C(\mathbf{v}_h)\mathbf{v}_h + \text{ND}\mathbf{v}_h - R^{-1}\Omega \mathbf{G}\mathbf{p}_h, \quad \text{with} \]

\[ \begin{align*}
\text{Convective operator: } C(\mathbf{v}_h) \\
\text{Diffusive operator: } D \\
\text{Mesh volumes: } \Omega = \text{diag}(V_h) \\
R = \text{diag}(\rho_h), \quad N = \text{diag}(\nu_h)
\end{align*} \]

Symmetry-preserving staggered discretization of Navier-Stokes equations

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

\[
\frac{dE_{kC(\mathbf{v}_h)}}{dt} + \frac{dE_{k\nabla p}}{dt} = 0
\]

\[
\frac{dE_{k\nabla p}}{dt} \quad \text{(*)} \quad 0
\]

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

Constant vs Variable coefficients Poisson equation

Combining FSM with a symmetry-preserving discretization leads to:

- \( \rho \equiv \text{ct.} \Rightarrow \) Constant Poisson equation:
  \[
  L_p = \rho M v^P, \quad \text{where } L = MG
  \]

- \( \rho \not\equiv \text{ct.} \Rightarrow \) Variable coefficients Poisson equation:
  \[
  \tilde{L}_p = M v^P, \quad \text{where } \tilde{L} := MR^{-1}G
  \]
Constant vs Variable coefficients Poisson equation

Combining FSM with a symmetry-preserving discretization leads to:

- $\rho \equiv \text{ct.} \Rightarrow$ Constant Poisson equation:
  \[ Lp = \rho Mv^p, \quad \text{where} \quad L = MG = -M\Omega^{-1}M^t \]

- $\rho \not\equiv \text{ct.} \Rightarrow$ Variable coefficients Poisson equation:
  \[ \tilde{L}p = Mv^p, \quad \text{where} \quad \tilde{L} := MR^{-1}G = -MR^{-1}\Omega^{-1}M^t \]
Constant vs Variable coefficients Poisson equation

Combining FSM with a symmetry-preserving discretization leads to:

- $\rho \equiv \text{ct.} \Rightarrow$ Constant Poisson equation:
  \[ Lp = \rho M v^p, \text{ where } L = MG = -M\Omega^{-1}M^t \]

- $\rho \not\equiv \text{ct.} \Rightarrow$ Variable coefficients Poisson equation:
  \[ \tilde{L}p = M v^p, \text{ where } \tilde{L} := MR^{-1}G = -MR^{-1}\Omega^{-1}M^t \]

Indeed, defining $\tilde{\Omega} := \Omega R$:

\[ \tilde{L} = -M\tilde{\Omega}^{-1}M^t \]
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.
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 $M_2x = y$
Preconditioning techniques

Left, right and split preconditioning

Given the linear system $Ax = b$ and the preconditioner $M = M_1M_2$, 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 $M_2x = y$

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,

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

\[ \tilde{L}_p = M v^p, \]

where:

\[ \tilde{L} := MR^{-1} G \]
Variable Poisson equation with extreme contrasts in the coefficients

Recalling variable coefficients Poisson equation:

\[ \tilde{L}_p = M v^p, \]

where:

\[ \tilde{L} := MR^{-1} G = -\Omega^{-1} M^t = -MR^{-1} \Omega^{-1} M^t \]
Recalling variable coefficients Poisson equation:

\[ \tilde{L}_p = Mv^p, \]

where:

\[
\tilde{L} \equiv MR^{-1} G = \Omega^{-1} M^t \quad -MR^{-1} \Omega^{-1} M^t \quad \tilde{\Omega} \equiv \Omega R \equiv -M\tilde{\Omega}^{-1} M^t.
\]
Variable Poisson equation with extreme contrasts in the coefficients

Recalling variable coefficients Poisson equation:

\[ \tilde{L} p = M v^p, \]

where:

\[ \tilde{L} := M R^{-1} G = -\Omega^{-1} M^t \quad -M R^{-1} \Omega^{-1} M^t \quad \tilde{\Omega} := \Omega R = -M \tilde{\Omega}^{-1} M^t. \]

Hence:

High contrasts in \( \Omega \) or \( R \) \( \Rightarrow \) High contrasts in \( \tilde{\Omega} \) \( \Rightarrow \) High contrasts in \( \tilde{L} \)
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**: 
  - $\rho_0 = 1.0$, internal fluid (kg/m$^3$)
  - $\rho_1 = \text{ratio}^{-1}$, external fluid (kg/m$^3$)
- **Initial ellipse axis**: $(a, b) = (1.0 \text{m}, 0.5 \text{m})$
- **Homogeneous mesh** ⇒ $\Omega = (\Delta x \Delta y \Delta z) \mathbb{I}$ and $\tilde{\Omega} = (\Delta x \Delta y \Delta z) \mathbb{R}$

**Figure**: Initial “bubble” configuration.

**Figure**: Evolved “bubble”.
Spectrum of $\tilde{L} = -\tilde{M} \tilde{\Omega}^{-1} M^t$ for various ratios

Figure: Normalized spectrum of $\tilde{L}$ for various density ratios on a $16 \times 16$ mesh.
Variable Poisson equation with extreme contrasts in the coefficients

Variable high contrasts in $\Omega$ or $R \Rightarrow \tilde{L}$ 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 $\Omega$ or $R$ $\Rightarrow$ $\tilde{L}$ is:

\[
\begin{cases}
\text{very ill-conditioned} \\
\text{variable} \\
\text{possibly not built explicitly}
\end{cases}
\]

$\Rightarrow$ Preconditioning becomes crucial to use iterative methods.
Variable Poisson equation with extreme contrasts in the coefficients

Variable high contrasts in $\Omega$ or $R \Rightarrow \tilde{L}$ 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:

\[
\tilde{L} \text{ is: } \begin{cases}
\text{very ill-conditioned} \\
\text{variable} \\
\text{possibly not built explicitly}
\end{cases}
\]
Variable high contrasts in $\Omega$ or $R \Rightarrow \tilde{L}$ 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:
\[
\begin{cases} 
\text{very ill-conditioned} & \Rightarrow M^{-1} \simeq \tilde{L}^{-1} \text{ is required} \\
\text{variable} \\
\text{possibly not built explicitly}
\end{cases}
\]
Variable Poisson equation with extreme contrasts in the coefficients

Variable high contrasts in $\Omega$ or $R \Rightarrow \tilde{L}$ is: \begin{align*}
\text{very ill-conditioned} & \quad \Rightarrow M^{-1} \simeq \tilde{L}^{-1} \text{ is required} \\
\text{variable} & \quad \Rightarrow M \text{ is variable} \\
\text{possibly not built explicitly} & \quad \Rightarrow M \text{ is variable}
\end{align*}
Variable Poisson equation with extreme contrasts in the coefficients

Variable high contrasts in $\Omega$ or $R \Rightarrow \tilde{L}$ is:

- very ill-conditioned
- variable
- possibly not built explicitly

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

Indeed:

$\tilde{L}$ is:

- very ill-conditioned $\Rightarrow M^{-1} \approx \tilde{L}^{-1}$ is required
- variable $\Rightarrow M$ is variable
- possibly not built explicitly $\Rightarrow M$ shouldn’t require full $\tilde{L}$
Variable Poisson equation with extreme contrasts in the coefficients

Variable high contrasts in $\Omega$ or $R \Rightarrow \tilde{L}$ 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:

\[
\tilde{L} \text{ is: } \begin{cases}
\text{very ill-conditioned} & \Rightarrow M^{-1} \approx \tilde{L}^{-1} \text{ is required} \\
\text{variable} & \Rightarrow M \text{ is variable} \\
\text{possibly not built explicitly} & \Rightarrow M \text{ shouldn’t require full } \tilde{L}
\end{cases}
\]

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

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

$$M_{\text{Jac}} = \text{diag}(\tilde{L}).$$
Introducing Jacobi preconditioner

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

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

**Pros:**
- If $\tilde{L}$ is available, cheap to build.
- Easily invertible and highly parallelizable.
- Can be used with CG, given that by definition $M_{\text{Diag}}$ is SPD.
- Extremely easy to implement.

**Cons:**
- Requires full matrix $\tilde{L}$.
- In many cases, doesn’t really improve convergence.
Introducing Jacobi preconditioner

Jacobi preconditioner

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

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

- **Pros:**
  - If $\tilde{L}$ is available, cheap to build.
  - Easily invertible and highly parallelizable.
  - Can be used with CG, given that by definition $M_{\text{Diag}}$ is SPD.
  - Extremely easy to implement.
  - **Well-suited for high-ratio Poisson equation.**

- **Cons:**
  - Requires full matrix $\tilde{L}$.
  - In many cases, doesn’t really improve convergence.
Spectrum of $M^{-1}\tilde{L}$ for various preconditioners and ratios

Figure: Normalized spectrum of $M^{-1}\tilde{L}$ for $M \in \{I, M_{Jac}\}$.
Our proposal as a feasible alternative to the Jacobi preconditioner

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

$$M_{\text{Diag}} = \tilde{\Omega}^{-1}.$$
Our proposal as a feasible alternative to the Jacobi preconditioner

**Our proposal: an adaptive diagonal preconditioner**

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

\[
M_{\text{Diag}} = \tilde{\Omega}^{-1}.
\]

**Pros:**
- Does not require \( \tilde{L} \) (only \( R \) and \( \Omega \)).
- “Free” to build, as is based on available fields \( R \) and \( \Omega \).
- Easily invertible and highly parallelizable.
- Can be used with CG, given that by definition \( M_{\text{Diag}} \) is SPD.
- Extremely easy to implement.

**Cons:**
- Compared to \( M_{\text{Jac}} \), it requires one extra diagonal matrix product (if \( \Omega \) and \( R \) are both considered).
- For lower contrasts in the coefficients, it doesn’t improve much the convergence.
Spectrum of $M^{-1}\tilde{L}$ for various preconditioners and ratios

Figure: Normalized spectrum of $M^{-1}\tilde{L}$ for $M \in \{I, M_{\text{Jac}}, M_{\text{Diag}}\}$.
Spectrum of $M^{-1}\tilde{L}$ for various preconditioners and ratios

Figure: Normalized spectrum of $M^{-1}\tilde{L}$ for $M \in \{M_{\text{Jac}}, M_{\text{Diag}}\}$. 
Combination of constant and adaptive diagonal preconditioners

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

$$\tilde{M}_{\text{Diag}} = L_L M_{\text{Diag}} L_L^t.$$
Our proposal in combination with constant preconditioners

Combination of constant and adaptive diagonal preconditioners

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

$$\tilde{M}_{\text{Diag}} = L_L M_{\text{Diag}} L_L^t.$$ 

- **Pros:**
  - Does not require $\tilde{L}$ (only $L$, $R$ and $\Omega$).
  - Compatible with more complex preconditioners, as they only need to be calculated once.
  - Achieves further improvements in convergence compared to $M_{\text{Jac}}$ and $M_{\text{Diag}}$ thanks to the constant preconditioner $M_L$.

- **Cons:**
  - Compared to $M_L$, it requires two extra diagonal matrix products (if $\Omega$ and $R$ are both considered).
  - It will always work worse than $M_L$, a variable (and unaffordable) version of $M_L$ based on $\tilde{L}$. 
Spectrum of $M^{-1}\tilde{L}$ for various preconditioners and ratios

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

(a) ratio = 1

(b) ratio = $10^2$

(c) ratio = $10^4$

(d) ratio = $10^6$
Spectrum of $M^{-1}\tilde{L}$ for various preconditioners and ratios

Figure: Normalized spectrum of $M^{-1}\tilde{L}$ for $M \in \{M_{\text{Jac}}, M_{\text{Diag}}, \tilde{M}_{\text{Diag}}, M_{\tilde{L}}\}$.
Spectrum of $M^{-1}\tilde{L}$ for various preconditioners and ratios

(a) ratio = 1

(b) ratio = $10^2$

(c) ratio = $10^4$

(d) ratio = $10^6$

Figure: Normalized spectrum of $M^{-1}\tilde{L}$ for $M \in \{ \tilde{M}_{\text{Diag}}, M_{\tilde{L}} \}$. 
### Numerical results

<table>
<thead>
<tr>
<th>ratio</th>
<th>$\mathbb{I}$</th>
<th>$M_{Diag}$</th>
<th>$M_{Jac}$</th>
<th>$\tilde{M}_{Diag}$</th>
<th>$M_{\tilde{\mathbb{L}}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>93</td>
<td>93</td>
<td>91</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>$10^2$</td>
<td>310</td>
<td>83</td>
<td>80</td>
<td>33</td>
<td>23</td>
</tr>
<tr>
<td>$10^4$</td>
<td>2828</td>
<td>86</td>
<td>86</td>
<td>32</td>
<td>22</td>
</tr>
<tr>
<td>$10^6$</td>
<td>30286</td>
<td>154</td>
<td>152</td>
<td>74</td>
<td>64</td>
</tr>
</tbody>
</table>

**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 $64 \times 64$ mesh and convergence is achieved when the relative residual is smaller than the tolerance: $|b - \tilde{L}x_k|/|b - \tilde{L}x_0| < \text{tol}$. 

<table>
<thead>
<tr>
<th>ratio</th>
<th>$\mathbb{I}$</th>
<th>$M_{Diag}$</th>
<th>$M_{Jac}$</th>
<th>$\tilde{M}_{Diag}$</th>
<th>$M_{\tilde{\mathbb{L}}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>285</td>
<td>285</td>
<td>282</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>$10^2$</td>
<td>625</td>
<td>313</td>
<td>310</td>
<td>119</td>
<td>93</td>
</tr>
<tr>
<td>$10^4$</td>
<td>4024</td>
<td>323</td>
<td>318</td>
<td>142</td>
<td>98</td>
</tr>
<tr>
<td>$10^6$</td>
<td>37711</td>
<td>358</td>
<td>351</td>
<td>157</td>
<td>108</td>
</tr>
</tbody>
</table>

**tol = 1.0e−6**

**tol = 1.0e−8**
Concluding remarks
Conclusions

- $M_{\text{Diag}}$ has been proposed as a **computationally cheaper alternative to the Jacobi preconditioner**, not requiring $\tilde{L}$ 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 $\tilde{L}$), not requiring $\tilde{L}$ to be built and leading to comparable reductions in the number of iterations (with respect to its analogue based on $\tilde{L}$).
Conclusions

- $M_{\text{Diag}}$ has been proposed as a **computationally cheaper alternative to the Jacobi preconditioner**, not requiring $\tilde{L}$ 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 $\tilde{L}$), not requiring $\tilde{L}$ to be built and leading to comparable reductions in the number of iterations (with respect to its analogue based on $\tilde{L}$).

- 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 $M_{\text{Diag}}$ 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 $M_{\text{Diag}}$ 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 $M_{\text{Diag}}$ and $\tilde{M}_{\text{Diag}}$.

- Study other possible combinations of $M_{\text{Diag}}$ with more complex fixed preconditioners (based on $L$ rather than $\tilde{L}$).
Future lines of work

- Implementation of $M_{\text{Diag}}$ 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 $M_{\text{Diag}}$ and $\tilde{M}_{\text{Diag}}$.

- Study other possible combinations of $M_{\text{Diag}}$ with more complex fixed preconditioners (based on $L$ rather than $\tilde{L}$).

- Study ways to combine $M_{\text{Diag}}$ with deflation techniques applied to the variable matrix $\tilde{L}$. Thus, finding efficient and highly parallelizable ways to compute updated deflation vectors, similarly to what was proposed by van der Linden et al.\(^2\)

---

Thanks for your attention!