A new approach to flexible preconditioning for the iterative solution of the time-harmonic elastic wave equation at multiple frequencies

M. Baumann[*,†] and M. B. van Gijzen[†]

[*]Email: M.M.Baumann@tudelft.nl
[†]Delft Institute of Applied Mathematics
Delft University of Technology
Delft, the Netherlands

Progress meeting, October 25, 2013

Table of contents

The time-harmonic elastic wave equation

For many frequencies ω_k, $k = 1, ..., N$, we consider

$$-\mu \Delta \mathbf{u} - (\mu + \lambda)\nabla(\nabla \cdot \mathbf{u}) - \omega_k^2 \rho_s \mathbf{u} = \mathbf{r}, \quad \text{in } \Omega \subset \mathbb{R}^D,$$
$$i\gamma\omega_k \rho_s B\mathbf{u} + \left[\lambda(\nabla \cdot \mathbf{u}) + \mu\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right)\right]\mathbf{n} = 0, \quad \text{on } \partial\Omega,$$

where

Assumptions

- $\mathbf{u} \in \mathbb{R}^D$ is the displacement vector in two $(D = 2)$ or three $(D = 3)$ dimensions,
- $\mu, \lambda \in \mathbb{R}$ are the so-called Lamé parameters,
- $\rho_s \in \mathbb{R}$ is the (constant) material density,
- $B \in \mathbb{R}^{D \times D}$ is defined componentwise, $B_{i,j} \equiv c_p n_i n_j + c_s t_i t_j$, for $D = 2$.

The time-harmonic elastic wave equation

For many frequencies ω_k, $k = 1, ..., N$, we consider

$$-\mu\Delta\mathbf{u} - (\mu + \lambda)\nabla(\nabla \cdot \mathbf{u}) - \omega_k^2 \rho_s \mathbf{u} = \mathbf{r}, \quad \text{in } \Omega \subset \mathbb{R}^D,$$
$$i\gamma\omega_k\rho_s B\mathbf{u} + \left[\lambda(\nabla \cdot \mathbf{u}) + \mu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)\right]\mathbf{n} = 0, \quad \text{on } \partial\Omega,$$

where

Assumptions

- $\mathbf{u} \in \mathbb{R}^D$ is the displacement vector in two ($D = 2$) or three ($D = 3$) dimensions,
- $\mu, \lambda \in \mathbb{R}$ are the so-called Lamé parameters,
- $\rho_s \in \mathbb{R}$ is the (constant) material density,
- $B \in \mathbb{R}^{D\times D}$ is defined componentwise, $B_{i,j} \equiv c_p n_i n_j + c_s t_i t_j$, for $D = 2$.

The time-harmonic elastic wave equation

For many frequencies ω_k, $k = 1, ..., N$, we consider

$$-\mu\Delta\mathbf{u} - (\mu+\lambda)\nabla(\nabla\cdot\mathbf{u}) - \omega_k^2\rho_s\mathbf{u} = \mathbf{r}, \quad \text{in } \Omega \subset \mathbb{R}^D,$$
$$i\gamma\omega_k\rho_s B\mathbf{u} + \left[\lambda(\nabla\cdot\mathbf{u}) + \mu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)\right]\mathbf{n} = 0, \quad \text{on } \partial\Omega,$$

where

Assumptions

- $\mathbf{u} \in \mathbb{R}^D$ is the displacement vector in two ($D = 2$) or three ($D = 3$) dimensions,
- $\mu, \lambda \in \mathbb{R}$ are the so-called Lamé parameters,
- $\rho_s \in \mathbb{R}$ is the (constant) material density,
- $B \in \mathbb{R}^{D\times D}$ is defined componentwise, $B_{i,j} \equiv c_p n_i n_j + c_s t_i t_j$, for $D = 2$.

The time-harmonic elastic wave equation

For many frequencies ω_k, $k = 1, ..., N$, we consider

$$-\mu\Delta\mathbf{u} - (\mu+\lambda)\nabla(\nabla\cdot\mathbf{u}) - \omega_k^2\rho_s\mathbf{u} = \mathbf{r}, \quad \text{in } \Omega \subset \mathbb{R}^D,$$
$$i\gamma\omega_k\rho_s B\mathbf{u} + \left[\lambda(\nabla\cdot\mathbf{u}) + \mu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)\right]\mathbf{n} = 0, \quad \text{on } \partial\Omega,$$

where

Assumptions

- $\mathbf{u} \in \mathbb{R}^D$ is the displacement vector in two ($D = 2$) or three ($D = 3$) dimensions,
- $\mu, \lambda \in \mathbb{R}$ are the so-called Lamé parameters,
- $\rho_s \in \mathbb{R}$ is the (constant) material density,
- $B \in \mathbb{R}^{D\times D}$ is defined componentwise, $B_{i,j} \equiv c_p n_i n_j + c_s t_i t_j$, for $D = 2$.

The time-harmonic elastic wave equation

For many frequencies ω_k, $k = 1, ..., N$, we consider

$$-\mu\Delta\mathbf{u} - (\mu + \lambda)\nabla(\nabla \cdot \mathbf{u}) - \omega_k^2 \rho_s \mathbf{u} = \mathbf{r}, \quad \text{in } \Omega \subset \mathbb{R}^D,$$

$$i\gamma\omega_k\rho_s B\mathbf{u} + \left[\lambda(\nabla \cdot \mathbf{u}) + \mu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)\right]\mathbf{n} = 0, \quad \text{on } \partial\Omega,$$

where

Assumptions

- $\mathbf{u} \in \mathbb{R}^D$ is the displacement vector in two $(D = 2)$ or three $(D = 3)$ dimensions,
- $\mu, \lambda \in \mathbb{R}$ are the so-called Lamé parameters,
- $\rho_s \in \mathbb{R}$ is the (constant) material density,
- $B \in \mathbb{R}^{D\times D}$ is defined componentwise, $B_{i,j} \equiv c_p n_i n_j + c_s t_i t_j$, for $D = 2$.

TUDelft

Discretization

using the finite element method (FEM)

After FEM-discretization, we obtain the linear systems

$$(K + i\omega_k C - \omega_k^2 M)\underline{\mathbf{u}} = \underline{\mathbf{r}},$$

with K, C, M being symmetric and sparse.

Re-formulation yields

$$\left[\begin{pmatrix} iM^{-1}C & M^{-1}K \\ I & 0 \end{pmatrix} - \omega_k \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}\right] \begin{pmatrix} \omega_k \underline{\mathbf{u}} \\ \underline{\mathbf{u}} \end{pmatrix} = \begin{pmatrix} M^{-1}\underline{\mathbf{r}} \\ 0 \end{pmatrix},$$

which is of the form

$$(A - \omega_k I)\mathbf{x}_k = \mathbf{b}, \quad \omega_k = 1, \ldots, N.$$

TUDelft

Krylov methods for shifted linear systems

Main observation

For shifted linear systems

$$(A - \omega I)\mathbf{x} = \mathbf{b}, \quad \omega \in \mathbb{R},$$

the Krylov subspaces are invariant, i.e.:

$$\mathcal{K}_m(A, b) \equiv \text{span}\ \{b, Ab, ..., A^{m-1}b\} = \mathcal{K}_m(A - \omega I, b).$$

Main advantage:

- Perform Arnoldi algorithm only once, and use:

$$\begin{aligned} AV_m &= V_{m+1}\underline{H}_m \\ \Rightarrow \ (A - \omega I)V_m &= V_{m+1}(\underline{H}_m - \omega \underline{I}_m) \end{aligned}$$

TU Delft

Krylov methods for shifted linear systems

Main observation

For shifted linear systems

$$(A - \omega I)\mathbf{x} = \mathbf{b}, \quad \omega \in \mathbb{R},$$

the Krylov subspaces are invariant, i.e.:

$$\mathcal{K}_m(A, b) \equiv \text{span } \{b, Ab, ..., A^{m-1}b\} = \mathcal{K}_m(A - \omega I, b).$$

Main advantage:

- Perform Arnoldi algorithm only once, and use:

$$\begin{aligned} AV_m &= V_{m+1}\underline{H}_m \\ \Rightarrow \ (A - \omega I)V_m &= V_{m+1}(\underline{H}_m - \omega \underline{I}_m) \end{aligned}$$

TU Delft

Krylov methods for shifted linear systems

Preconditioning multishift problems

Main disadvantage:

- Preconditioners are in general not easy to apply. For

$$(A - \omega I)M_\omega^{-1} y_\omega = b, \quad x_\omega = M_\omega^{-1} y_\omega$$

it does not hold:

$$\mathcal{K}_m(AM_\omega^{-1}, b) \neq \mathcal{K}_m(AM_\omega^{-1} - \omega M_\omega^{-1}, b)$$

However, there is a way...

References

- Y. A. Erlangga, *A robust and efficient iterative method for the numerical solution of the Helmholtz equation.* PhD thesis, TU Delft, 2005.

TU Delft

Krylov methods for shifted linear systems

Preconditioning multishift problems

Main disadvantage:

- Preconditioners are in general not easy to apply. For

$$(A - \omega I)M_\omega^{-1} y_\omega = b, \quad x_\omega = M_\omega^{-1} y_\omega$$

it does not hold:

$$\mathcal{K}_m(AM_\omega^{-1}, b) \neq \mathcal{K}_m(AM_\omega^{-1} - \omega M_\omega^{-1}, b)$$

However, there is a way...

References

- Y. A. Erlangga, *A robust and efficient iterative method for the numerical solution of the Helmholtz equation.* PhD thesis, TU Delft, 2005.

TUDelft

The shift-and-invert preconditioner

Can we find M_ω^{-1} such that

$$(A - \omega I)M_\omega^{-1} = AM^{-1} - \eta(\omega)I \quad ?$$

For $M \equiv (A - \sigma I)$, we get:

$$\begin{aligned}(A - \omega I)M_\omega^{-1} &= A(A - \sigma I)^{-1} - \eta(\omega)I \\ &= (1 - \eta(\omega))(A + \underbrace{\frac{\sigma\eta(\omega)}{1 - \eta(\omega)}}_{\equiv -\omega} I)(A - \sigma I)^{-1}\end{aligned}$$

and, therefore,

$$\eta(\omega) = \frac{\omega}{\omega - \sigma}$$

$$M_\omega = \frac{1}{1 - \eta(\omega)}(A - \sigma I) = \frac{1}{1 - \eta(\omega)}M$$

The shift-and-invert preconditioner

Can we find M_ω^{-1} such that

$$(A - \omega I)M_\omega^{-1} = AM^{-1} - \eta(\omega)I \quad ?$$

For $M \equiv (A - \sigma I)$, we get:

$$\begin{aligned}(A - \omega I)M_\omega^{-1} &= A(A - \sigma I)^{-1} - \eta(\omega)I \\ &= (1 - \eta(\omega))(A + \underbrace{\frac{\sigma\eta(\omega)}{1 - \eta(\omega)}}_{\equiv -\omega} I)(A - \sigma I)^{-1}\end{aligned}$$

and, therefore,

$$\eta(\omega) = \frac{\omega}{\omega - \sigma}$$

$$M_\omega = \frac{1}{1 - \eta(\omega)}(A - \sigma I) = \frac{1}{1 - \eta(\omega)} M$$

The shift-and-invert preconditioner

Can we find M_ω^{-1} such that

$$(A - \omega I)M_\omega^{-1} = AM^{-1} - \eta(\omega)I \quad ?$$

For $M \equiv (A - \sigma I)$, we get:

$$\begin{aligned}(A - \omega I)M_\omega^{-1} &= A(A - \sigma I)^{-1} - \eta(\omega)I \\ &= (1 - \eta(\omega))(A + \underbrace{\frac{\sigma\eta(\omega)}{1 - \eta(\omega)}}_{\equiv -\omega} I)(A - \sigma I)^{-1}\end{aligned}$$

and, therefore,

$$\begin{aligned}\eta(\omega) &= \frac{\omega}{\omega - \sigma} \\ M_\omega &= \frac{1}{1 - \eta(\omega)}(A - \sigma I) = \frac{1}{1 - \eta(\omega)} M\end{aligned}$$

FOM preconditioning for flexible Krylov methods

A new idea

What does *flexible* stand for?

- allows a different preconditioner in every iteration
- suggestion: use many shift-and-invert preconditioner
- Can we do better?

References

- A. K. Saibaba, T. Bakhos, P.K. Kitanidis, *A Flexible Krylov Solver for Shifted Systems with Application to Oscillatory Hydraulic Tomography.* Stanford University, 2012.

TU Delft

FOM preconditioning for flexible Krylov methods

A new idea

What does *flexible* stand for?

- allows a different preconditioner in every iteration
- suggestion: use many shift-and-invert preconditioner
- Can we do better?

References

- A. K. Saibaba, T. Bakhos, P.K. Kitanidis, *A Flexible Krylov Solver for Shifted Systems with Application to Oscillatory Hydraulic Tomography.* Stanford University, 2012.

FOM preconditioning for flexible Krylov methods

A new idea

What does *flexible* stand for?

- allows a different preconditioner in every iteration
- suggestion: use many shift-and-invert preconditioner
- Can we do better?

References

- A. K. Saibaba, T. Bakhos, P.K. Kitanidis, *A Flexible Krylov Solver for Shifted Systems with Application to Oscillatory Hydraulic Tomography.* Stanford University, 2012.

FOM preconditioning for flexible Krylov methods

Algorithm 1 Flexible GMRES for $A\mathbf{x} = \mathbf{b}$

$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0,\ \beta = \|\mathbf{r}_0\|_2,\ \mathbf{v}_1 = \mathbf{r}_0/\beta$
for $j = 1$ to m **do**
  Compute $\mathbf{z}_j = M_j^{-1}\mathbf{v}_j$
  Compute $\mathbf{w} = A\mathbf{z}_j$
  for $i = 1$ to j **do**
    $h_{i,j} = \langle \mathbf{w}, \mathbf{v}_i \rangle$
    $\mathbf{w} = \mathbf{w} - h_{i,j}\mathbf{v}_i$
  end for
  Set $h_{j+1,j} = \|\mathbf{w}\|_2$ and $\mathbf{v}_{j+1} = \mathbf{w}/h_{j+1,j}$
end for

$\mathbf{x}_m = \mathbf{x}_0 + Z_m\mathbf{y}_m$, where

$\mathbf{y}_m = \underset{\mathbf{y}}{\operatorname{argmin}} \|\beta\mathbf{e}_1 - \underline{H}_m\mathbf{y}\|_2$

References

- Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm.* SIAM J. Sci. Comput., 14, 461-469, 1993.

FOM preconditioning for flexible Krylov methods

Algorithm 1 Flexible GMRES for $A\mathbf{x} = \mathbf{b}$

$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0, \ \beta = \|\mathbf{r}_0\|_2, \ \mathbf{v}_1 = \mathbf{r}_0/\beta$
for $j = 1$ to m **do**
    Compute $\mathbf{z}_j = M_j^{-1}\mathbf{v}_j$
    Compute $\mathbf{w} = A\mathbf{z}_j$
    for $i = 1$ to j **do**
        $h_{i,j} = \langle \mathbf{w}, \mathbf{v}_i \rangle$
        $\mathbf{w} = \mathbf{w} - h_{i,j}\mathbf{v}_i$
    end for
    Set $h_{j+1,j} = \|\mathbf{w}\|_2$ and $\mathbf{v}_{j+1} = \mathbf{w}/h_{j+1,j}$
end for

$\mathbf{x}_m = \mathbf{x}_0 + Z_m\mathbf{y}_m$, where

$$\mathbf{y}_m = \underset{\mathbf{y}}{\operatorname{argmin}} \|\beta\mathbf{e}_1 - \underline{H}_m\mathbf{y}\|_2$$

References

- Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm.* SIAM J. Sci. Comput., 14, 461-469, 1993.

FOM preconditioning for flexible Krylov methods

Algorithm 1 Flexible GMRES for $A\mathbf{x} = \mathbf{b}$

$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0,\ \beta = \|\mathbf{r}_0\|_2,\ \mathbf{v}_1 = \mathbf{r}_0/\beta$
for $j = 1$ to m **do**
  Compute $\mathbf{z}_j = M_j^{-1}\mathbf{v}_j$
  Compute $\mathbf{w} = A\mathbf{z}_j$
  for $i = 1$ to j **do**
    $h_{i,j} = \langle \mathbf{w}, \mathbf{v}_i \rangle$
    $\mathbf{w} = \mathbf{w} - h_{i,j}\mathbf{v}_i$
  end for
  Set $h_{j+1,j} = \|\mathbf{w}\|_2$ and $\mathbf{v}_{j+1} = \mathbf{w}/h_{j+1,j}$
end for

$$\mathbf{x}_m = \mathbf{x}_0 + Z_m\mathbf{y}_m, \text{ where}$$
$$\mathbf{y}_m = \underset{\mathbf{y}}{\operatorname{argmin}} \|\beta\mathbf{e}_1 - \underline{H}_m\mathbf{y}\|_2$$

References

- Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm.* SIAM J. Sci. Comput., 14, 461-469, 1993.

FOM preconditioning for flexible Krylov methods

A new idea

Our new approach:

Now, we need:

$$(A-\omega I)M_{\omega,j}{}^{-1} = \alpha_j(\omega)AM_j{}^{-1} - \beta_j(\omega)I$$

1. Solve the preconditioning step with FOM:

$$\mathbf{z}_j = M_j^{-1}\mathbf{v}_j = \texttt{fom(A,vj)}$$
$$\mathbf{z}_j^{(\omega)} = M_{\omega,j}^{-1}\mathbf{v}_j = \texttt{fom(A-wI,vj)}$$

2. In FOM, residuals of shifted systems are <u>collinear</u>, i.e.:

$$\exists \gamma_j^{(\omega)} \in \mathbb{R} : \quad \gamma_j^{(\omega)}\mathbf{r}_j = \mathbf{r}_j^{(\omega)}$$

TUDelft

FOM preconditioning for flexible Krylov methods

A new idea

3. Determine α_j and β_j:

$$
\begin{aligned}
(A-\omega I)\mathbf{z}_j^{(\omega)} &= \alpha_j A\mathbf{z}_j - \beta_j \mathbf{v}_j \\
\Leftrightarrow \quad \mathbf{v}_j - (A-\omega I)\mathbf{z}_j^{(\omega)} &= \alpha_j \mathbf{v}_j - \alpha_j A\mathbf{z}_j + (1-\alpha_j+\beta_j)\mathbf{v}_j \\
\Leftrightarrow \quad \mathbf{r}_j &= \underbrace{\alpha_j}_{\equiv \gamma_j^{(\omega)}} \mathbf{r}_j^{(\omega)} + \underbrace{(1-\alpha_j+\beta_j)}_{\equiv 0} \mathbf{v}_j
\end{aligned}
$$

4. Solve the shifted linear systems with:

$$
\gamma_j^{(\omega)} A M_j^{-1} - (\gamma_j^{(\omega)} - 1) I
$$

Numerical results

Recap

$$(A - \omega_k I)\mathbf{x}_k = \mathbf{b}, \quad k = 1, ..., N$$

We want to compare:

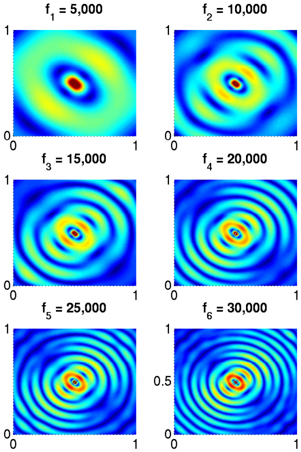
1. plain simultaneous solution: `msGMRES`, `msIDR(s)`
2. using the shift-and-invert preconditioner: `pmsGMRES`, `pmsIDR(s)`
3. using `msFOM` within a flexible method: `fmsGMRES`, `fmsIDR(s)`

Numerical results

The setting

Test case from literature:

- $\Omega = [0,1] \times [0,1]$
- $h = 0.01$ implying
  $n = 10.201$ grid points
- system size:
  $4n = 40.804$
- $N = 6$ frequencies
- point source at center



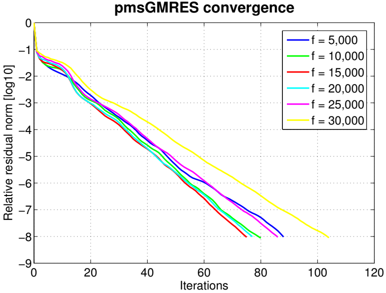
References

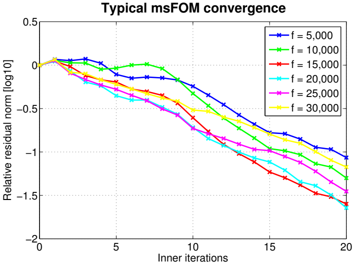
- T. Airaksinen, A. Pennanen, J. Toivanen, *A damping preconditioner for time-harmonic wave equations in flid and elastic material.* Journal of Computational Physics, 2009.

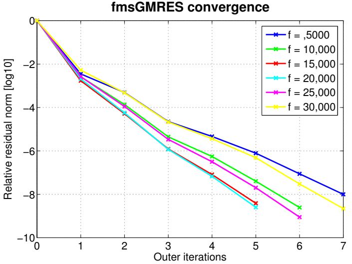
TUDelft

Numerical results

Convergence behavior







Numerical results

Time measurements

For a fair time comparison:

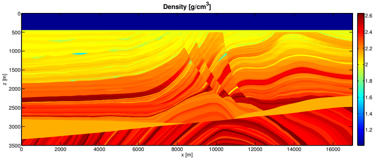
- optimize the (complex) *seed shift* σ,
- optimize number of inner `msFOM` iterations,
- guarantee same order of relative residuals at convergence.

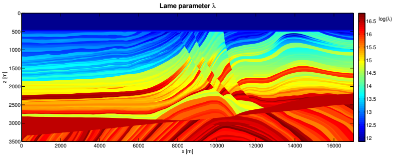
	msK	pmsK	fmsK
K = GMRES	$>63k$	16.21	7.69
K = IDR(1)	-	18.42	8.65
K = IDR(4)	-	17.16	8.83
K = IDR(8)	617.5	24.10	7.88

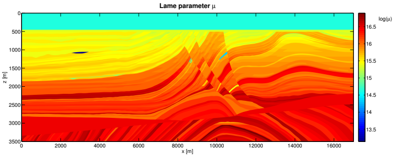
Discussion and Outlook

My plan for the next months...

1. Consider the `marmousi2` model (right):
   - parameters $\rho_s(\mathbf{x}), \mu(\mathbf{x}), \lambda(\mathbf{x})$ depend on space
   - $\sim 40,000,000$ grid points
2. Improve flexible pre-conditioning
   - avoid LU decomposition
   - publication







Discussion and Outlook

Large-scale outlook...

- High performance computing
  - When is a good point to switch from `MATLAB` to e.g. `C++` or `Python`?
  - Make use of parallel architecture, i.e. use `MPI` and/or `Cuda`.
- Consider many right-hand sides:

$$(A - \omega_k I)\mathbf{x}_k = \mathbf{b}_\ell, \qquad k = 1, .., N, \;\; \ell = 1, ..., M$$

- Solve inverse problem:

$$\min_{\rho_s, \lambda, \mu} \|\mathbf{u}_{sim}^{(\omega_k)} - \mathbf{u}_{data}^{(\omega_k)}\|_2 \quad s.t. \; \mathbf{f}(\mathbf{u}_{sim}, \omega_k, \rho_s, \lambda, \mu) = 0$$

TU Delft

Discussion and Outlook

Large-scale outlook...

- High performance computing
  - When is a good point to switch from `MATLAB` to e.g. `C++` or `Python`?
  - Make use of parallel architecture, i.e. use `MPI` and/or `Cuda`.
- Consider many right-hand sides:

$$(A - \omega_k I)\mathbf{x}_k = \mathbf{b}_\ell, \qquad k = 1, .., N, \;\; \ell = 1, ..., M$$

- Solve inverse problem:

$$\min_{\rho_s, \lambda, \mu} \|\mathbf{u}_{sim}^{(\omega_k)} - \mathbf{u}_{data}^{(\omega_k)}\|_2 \quad s.t. \; \mathbf{f}(\mathbf{u}_{sim}, \omega_k, \rho_s, \lambda, \mu) = 0$$

TU Delft

Discussion and Outlook

Large-scale outlook...

- High performance computing
  - When is a good point to switch from `MATLAB` to e.g. `C++` or `Python`?
  - Make use of parallel architecture, i.e. use `MPI` and/or `Cuda`.
- Consider many right-hand sides:

$$(A - \omega_k I)\mathbf{x}_k = \mathbf{b}_\ell, \qquad k = 1, .., N, \ \ell = 1, ..., M$$

- Solve inverse problem:

$$\min_{\rho_s, \lambda, \mu} \|\mathbf{u}_{sim}^{(\omega_k)} - \mathbf{u}_{data}^{(\omega_k)}\|_2 \quad s.t. \ \mathbf{f}(\mathbf{u}_{sim}, \omega_k, \rho_s, \lambda, \mu) = 0$$