Computational advances in quasi-optimal domain decomposition methods for time-harmonic electromagnetic wave problems

Nicolas Marsic   Alexandre Vion   Christophe Geuzaine

Department of Electrical Engineering and Computer Science
Montefiore Institute
University of Liège
Belgium

E-mail: {nicolas.marsic,a.vion,cgeuzaine}@ulg.ac.be

DD XXIII

Why Domain Decomposition Methods ?

- We want to solve high-frequency time-harmonic wave problems.
- We also want to use the finite element method.

Why Domain Decomposition Methods ?

- We want to solve high-frequency time-harmonic wave problems.
- We also want to use the finite element method.

Unfortunately, the resulting matrices are hard to handle.

- Large. ⟶ Direct solvers do not scale.
- Non-Hermitian.
- Highly indefinite.

⟶ Iterative solvers do not converge well.

Why Domain Decomposition Methods ?

- We want to solve high-frequency time-harmonic wave problems.
- We also want to use the finite element method.

Unfortunately, the resulting matrices are hard to handle.

- Large. ⟶ Direct solvers do not scale.
- Non-Hermitian.
- Highly indefinite.

(Non-Hermitian, Highly indefinite) ⟶ Iterative solvers do not converge well.

Solution: Domain Decomposition Method!

Outline

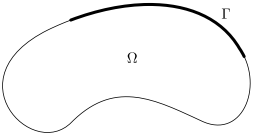
Outline

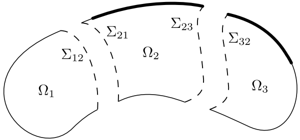
Optimized Schwarz algorithm

- We start by splitting the computational domain into sub-domains Ω_i.
- And we apply the following iterative scheme:

$$\left\{\begin{array}{rcll} \mathbf{curl}\,\mathbf{curl}\,\mathbf{e}_i^k - k^2\mathbf{e}_i^k & = & \mathbf{0} & \text{in } \Omega_i, \\ \gamma_T(\mathbf{e}_i^k) & = & -\gamma_T(\mathbf{e}^{inc}) & \text{on } \Gamma_i, \\ \gamma_t(\mathbf{curl}\,\mathbf{e}_i^k) + \mathcal{B}\Big[\gamma_T(\mathbf{e}_i^k)\Big] & = & \mathbf{0} & \text{on } \Gamma_i^\infty, \\ \gamma_t(\mathbf{curl}\,\mathbf{e}_i^k) + \mathcal{S}\Big[\gamma_T(\mathbf{e}_i^k)\Big] & = & \mathbf{g}_{ij}^{k-1} & \text{on } \Sigma_{ij}, \end{array}\right.$$

$$\text{with:} \qquad \mathbf{g}_{ij}^k = -\mathbf{g}_{ji}^{k-1} + 2\mathcal{S}\Big[\gamma_T(\mathbf{e}_j^k)\Big] \qquad \text{on } \Sigma_{ij}.$$





Optimized Schwarz algorithm

- We start by splitting the computational domain into sub-domains Ω_i.
- And we apply the following iterative scheme:

$$\begin{cases} \mathbf{curl}\,\mathbf{curl}\,\mathbf{e}_i^k - k^2\mathbf{e}_i^k &= \mathbf{0} & \text{in } \Omega_i, \\ \gamma_{\mathsf{T}}(\mathbf{e}_i^k) &= -\gamma_{\mathsf{T}}(\mathbf{e}^{\mathrm{inc}}) & \text{on } \Gamma_i, \\ \gamma_{\mathrm{t}}(\mathbf{curl}\,\mathbf{e}_i^k) + \mathcal{B}\Big[\gamma_{\mathsf{T}}(\mathbf{e}_i^k)\Big] &= \mathbf{0} & \text{on } \Gamma_i^{\infty}, \\ \gamma_{\mathrm{t}}(\mathbf{curl}\,\mathbf{e}_i^k) + \mathcal{S}\Big[\gamma_{\mathsf{T}}(\mathbf{e}_i^k)\Big] &= \mathbf{g}_{ij}^{k-1} & \text{on } \Sigma_{ij}, \end{cases}$$

$$\text{with:} \qquad \mathbf{g}_{ij}^k = -\mathbf{g}_{ji}^{k-1} + 2\mathcal{S}\Big[\gamma_{\mathsf{T}}(\mathbf{e}_j^k)\Big] \qquad \text{on } \Sigma_{ij}.$$

- It can be recast into the following linear system:

$$\mathbf{g}^k = \mathcal{A}\Big[\mathbf{g}^{k-1}\Big] + \mathbf{b} \quad \longrightarrow \quad (\mathcal{I} - \mathcal{A})\mathbf{g} = \mathbf{b}.$$

Outline

Transmission operators — Zeroth and second orders

- Zeroth-order: [Després, 1992]

$$\mathcal{S}\Big[\, \gamma_{\mathsf{T}}\,(\mathbf{e}) \Big] = \jmath k\, \gamma_{\mathsf{T}}\,(\mathbf{e}).$$

Transmission operators — Zeroth and second orders

- Zeroth-order: [Després, 1992]

$$\mathcal{S}\Big[\,\gamma_{\mathsf{T}}\,(\mathbf{e})\Big] = \jmath k\,\gamma_{\mathsf{T}}\,(\mathbf{e}).$$

- Optimized second-order: [Dolean, Gander and Gerardo-Giorda, 2009]

[Rawat and Lee, 2010]

[Dolean, Gander, Lanteri, Lee and Peng, 2015]

$$\mathcal{S}\Big[\,\gamma_{\mathsf{T}}\,(\mathbf{e})\Big] = \jmath k\Big[\mathcal{I} + \frac{\beta}{k^2}\,\mathbf{grad}_{\Sigma}\,\mathrm{div}_{\Sigma}\,\Big]^{-1}\Big[\mathcal{I} - \frac{\alpha}{k^2}\,\mathbf{curl}_{\Sigma}\,\mathrm{curl}_{\Sigma}\,\Big]\Big[\,\gamma_{\mathsf{T}}\,(\mathbf{e})\Big].$$

  - Where α and β are chosen such that optimal convergence rate is obtained.

Transmission operators — On-surface radiation condition

- On-surface radiation condition: [El Bouajaji, Antoine and Geuzaine, 2014]

$$\mathcal{S}\left[\gamma_{\mathsf{T}}(\mathbf{e})\right] = \jmath k \left[\mathcal{I} + \mathbf{grad}_{\Sigma} \frac{1}{k_{\varepsilon}^{2}} \mathrm{div}_{\Sigma} - \mathbf{curl}_{\Sigma} \frac{1}{k_{\varepsilon}^{2}} \mathrm{curl}_{\Sigma}\right]^{-1/2} \left[\mathcal{I} - \mathbf{curl}_{\Sigma} \frac{1}{k_{\varepsilon}^{2}} \mathrm{curl}_{\Sigma}\right]\left[\gamma_{\mathsf{T}}(\mathbf{e})\right].$$

  - With $k_{\varepsilon} = k + \jmath\varepsilon$, and ε chosen to reach the optimal convergence rate.
  - The square-root operator is non-local.

Transmission operators — On-surface radiation condition

- On-surface radiation condition: [El Bouajaji, Antoine and Geuzaine, 2014]

$$\mathcal{S}\left[\gamma_T(\mathbf{e})\right] = \jmath k \left[\mathcal{I} + \mathbf{grad}_\Sigma \frac{1}{k_\varepsilon^2}\,\mathrm{div}_\Sigma - \mathbf{curl}_\Sigma \frac{1}{k_\varepsilon^2}\,\mathrm{curl}_\Sigma\right]^{-1/2} \left[\mathcal{I} - \mathbf{curl}_\Sigma \frac{1}{k_\varepsilon^2}\,\mathrm{curl}_\Sigma\right]\left[\gamma_T(\mathbf{e})\right].$$

- With $k_\varepsilon = k + \jmath\varepsilon$, and ε chosen to reach the optimal convergence rate.
- The square-root operator is non-local.
- It is thus localized using Padé decomposition of order N_p, with a rotation of the branch-cut α:

$$(\mathcal{I} + \mathbf{\Delta})^{1/2} \approx R_0(\alpha) - \sum_{l=1}^{N_p} \frac{A_l(\alpha)}{B_l(\alpha)}\left[\mathcal{I} + B_l(\alpha)\mathbf{\Delta}\right]^{-1}.$$

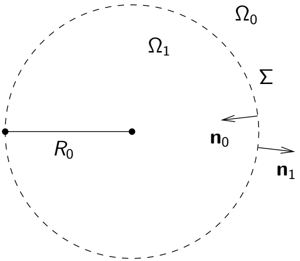
⟶ Where $R_0(\alpha)$, $A_l(\alpha)$ and $B_l(\alpha)$ are the Padé coefficients.

⟶ And $\mathbf{\Delta} = \mathbf{grad}_\Sigma \frac{1}{k_\varepsilon^2}\,\mathrm{div}_\Sigma - \mathbf{curl}_\Sigma \frac{1}{k_\varepsilon^2}\,\mathrm{curl}_\Sigma$.

Transmission operators — Benchmark

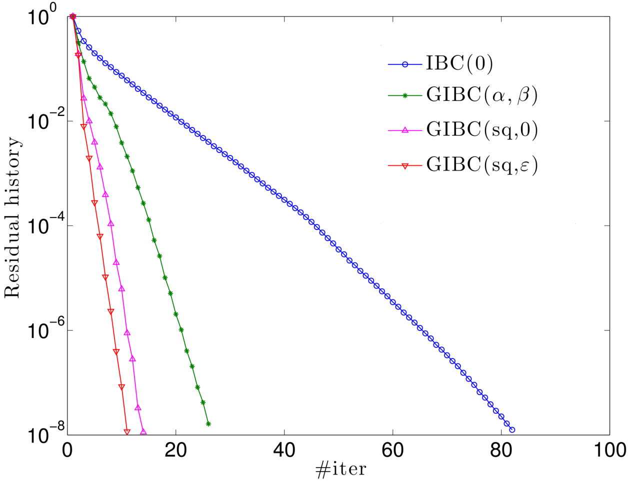
- Let us now study the convergence rate of the previous operators.

- Scattering by a sphere taken as test case.



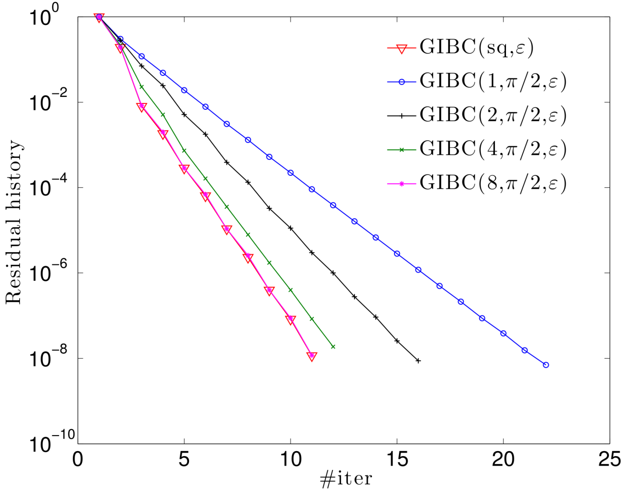
Transmission operators — Benchmark: convergence

- Convergence history for different transmission operators.
  ⟶ No FEM used — 2 sub-domains — No Padé used.



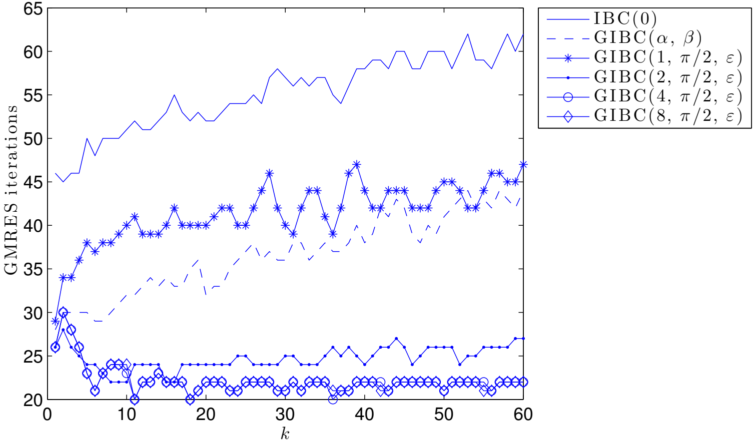
Transmission operators — Benchmark: Padé orders

- Convergence history for different Padé orders.
  ⟶ No FEM used — 2 sub-domains.



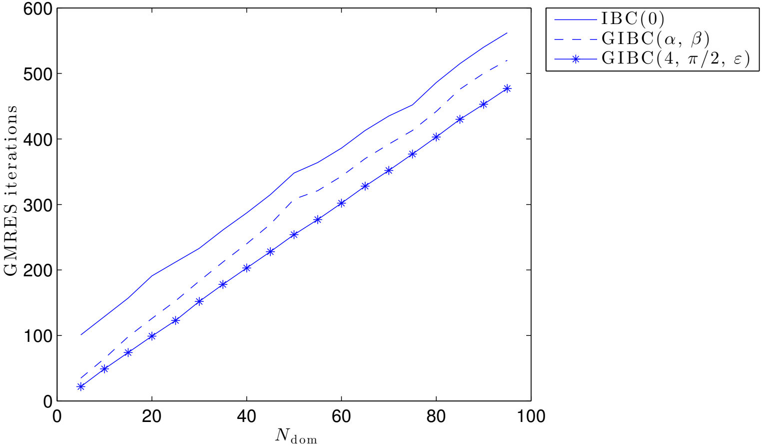
Transmission operators — Benchmark: k variations

- Iteration count for different wavenumbers.
  - ⟶ FEM used — 5 sub-domains.
  - ⟶ 20 mesh elements per wavelength.



Transmission operators — Benchmark: sub-domains

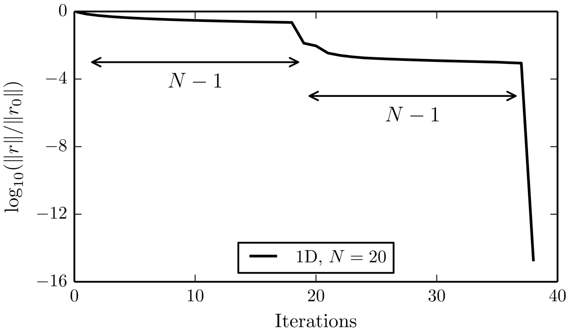
- Iteration count for different number of sub-domains.
  - ⟶ FEM used — 5 sub-domains.
  - ⟶ 20 mesh elements per wavelength.



Outline

Preconditioner

- Even with the best possible transmission operator, the Schwarz algorithm requires $\mathcal{O}(N)$ iterations.

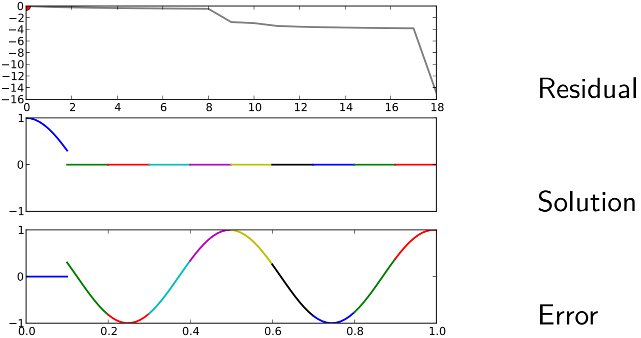


Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a `1D` waveguide with 10 sub-domains.

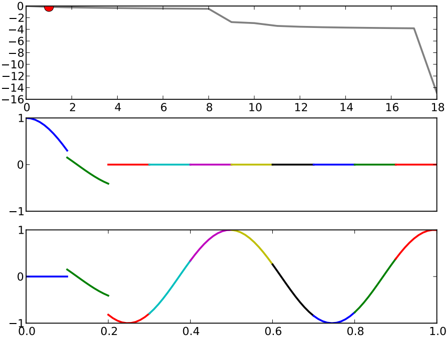
Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a 1D waveguide with 10 sub-domains.



Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a 1D waveguide with 10 sub-domains.



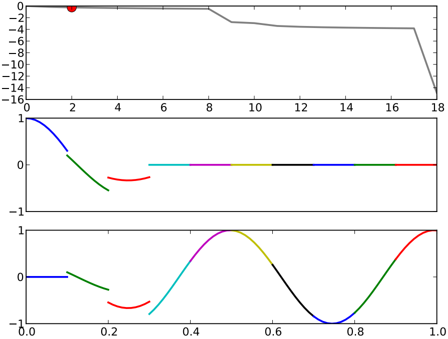
Residual

Solution

Error

Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a 1D waveguide with 10 sub-domains.



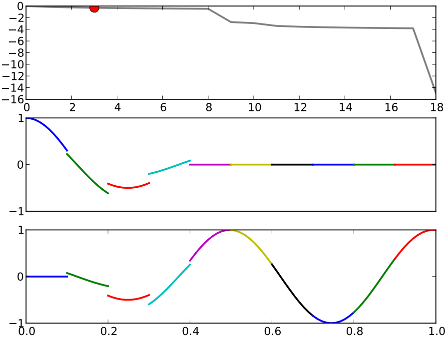
Residual

Solution

Error

Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a 1D waveguide with 10 sub-domains.



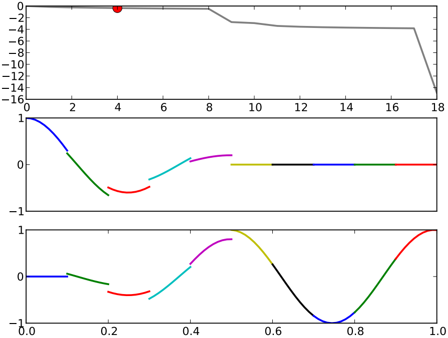
Residual

Solution

Error

Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a 1D waveguide with 10 sub-domains.



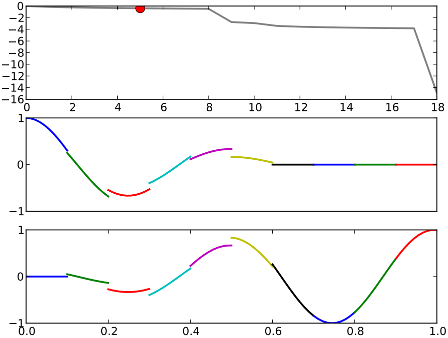
Residual

Solution

Error

Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a 1D waveguide with 10 sub-domains.



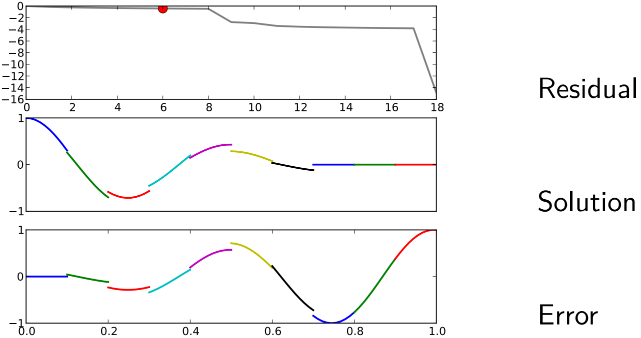
Residual

Solution

Error

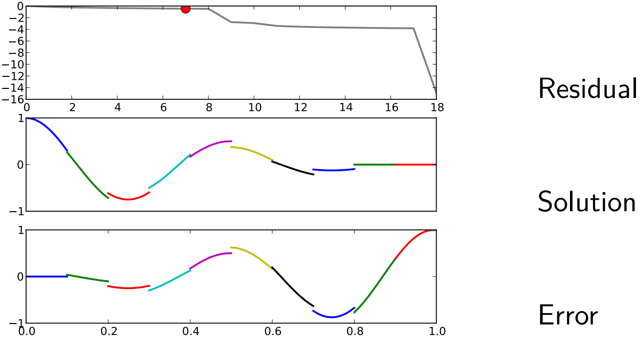
Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a 1D waveguide with 10 sub-domains.



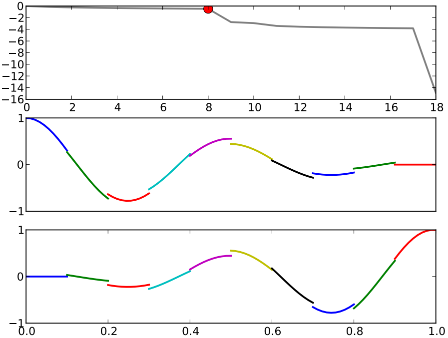
Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a `1D` waveguide with 10 sub-domains.



Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a 1D waveguide with 10 sub-domains.



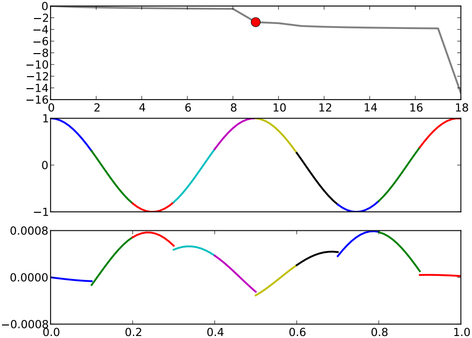
Residual

Solution

Error

Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a `1D` waveguide with 10 sub-domains.



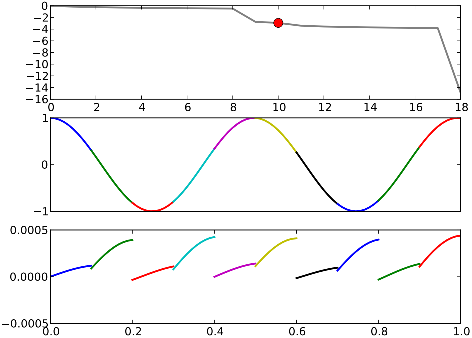
Residual

Solution

Error

Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a `1D` waveguide with 10 sub-domains.



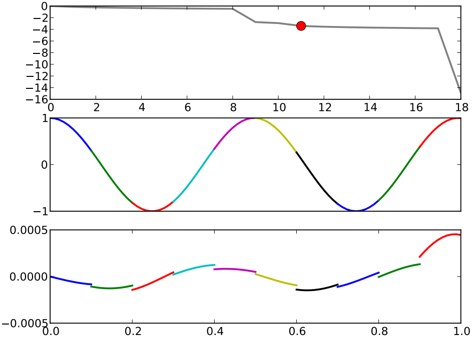
Residual

Solution

Error

Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a 1D waveguide with 10 sub-domains.



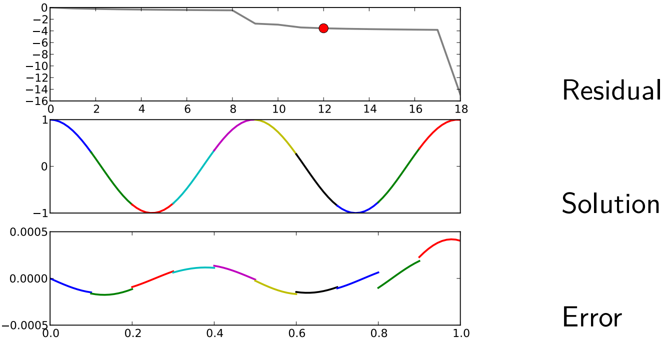
Residual

Solution

Error

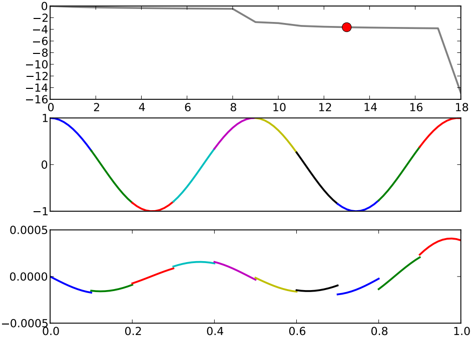
Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a `1D` waveguide with 10 sub-domains.



Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a 1D waveguide with 10 sub-domains.



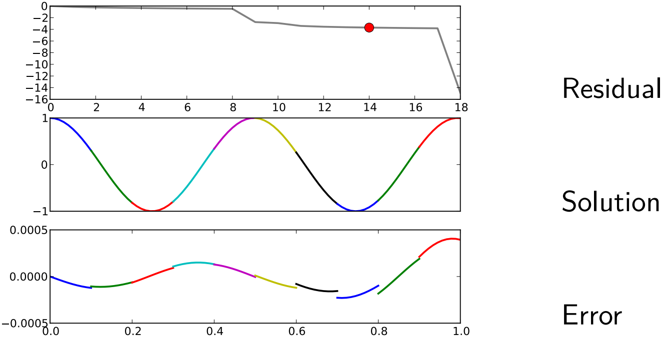
Residual

Solution

Error

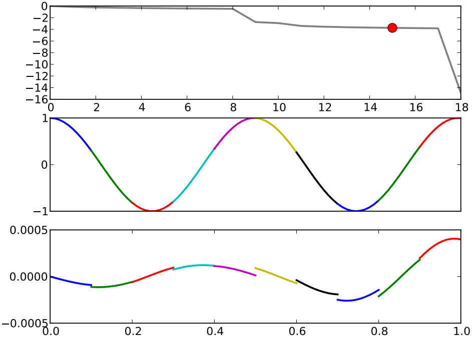
Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a 1D waveguide with 10 sub-domains.



Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a 1D waveguide with 10 sub-domains.



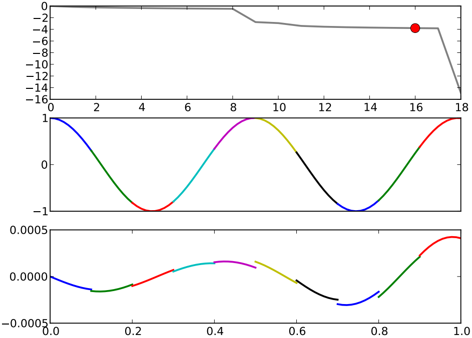
Residual

Solution

Error

Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a 1D waveguide with 10 sub-domains.



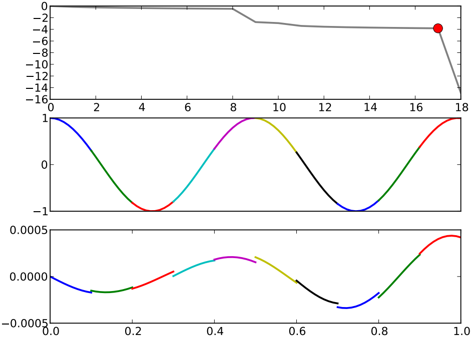
Residual

Solution

Error

Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a 1D waveguide with 10 sub-domains.



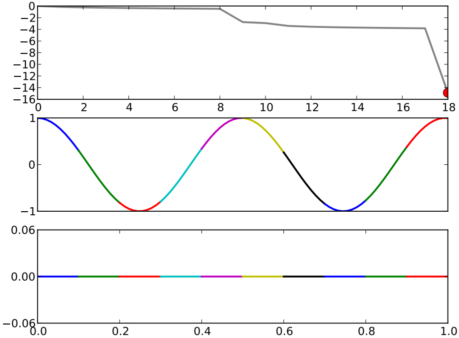
Residual

Solution

Error

Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a `1D` waveguide with 10 sub-domains.



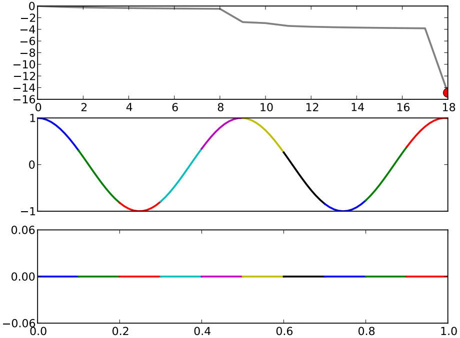
Residual

Solution

Error

Preconditioner — Intuitive explanation

- The information is propagated only between neighbour sub-domains.
  - Let us consider a 1D waveguide with 10 sub-domains.



Residual

Solution

Error

- We need to propagate information globally.

Preconditioner — A simple case

- For simplicity, the iteration operator is renamed.

$$(\mathcal{I}-\mathcal{A})\mathbf{g}=\mathbf{b} \iff \mathcal{F}\mathbf{g}=\mathbf{b}$$

- For a simple waveguide with no internal reflections, $\mathcal{F}$ is known:

$$\mathcal{F}_A = \left[\begin{array}{ccccccc}
\mathcal{I} & & \mathcal{B}_2^b & & & & \\
 & \mathcal{I} & & & & & \\
 & & \mathcal{I} & & \ddots & & \\
 & \mathcal{B}_2^f & & \mathcal{I} & & & \\
 & & \ddots & & \ddots & \mathcal{B}_{N-1}^b & \\
 & & & & & \mathcal{I} & \\
 & & & & \mathcal{B}_{N-1}^f & & \mathcal{I}
\end{array}\right]$$

Preconditioner — A simple case

- For simplicity, the iteration operator is renamed.

$$(\mathcal{I} - \mathcal{A})\mathbf{g} = \mathbf{b} \Longleftrightarrow \mathcal{F}\mathbf{g} = \mathbf{b}$$

- For a simple waveguide with no internal reflections, $\mathcal{F}$ is known:

$$\mathcal{F}_A^{-1} = \left[\begin{array}{cc|cc|cc|cc}
\mathcal{I} & & -\mathcal{B}_2^b & & \mathcal{B}_2^b\mathcal{B}_3^b & & -\mathcal{B}_2^b\mathcal{B}_3^b\mathcal{B}_4^b & \\
 & \mathcal{I} & & & & & & \\
\hline
 & & \mathcal{I} & & -\mathcal{B}_3^b & & \mathcal{B}_3^b\mathcal{B}_4^b & \\
 & -\mathcal{B}_2^f & & \mathcal{I} & & & & \\
\hline
 & & & & \mathcal{I} & & -\mathcal{B}_4^b & \\
 & \mathcal{B}_3^f\mathcal{B}_2^f & & -\mathcal{B}_3^f & & \mathcal{I} & & \\
\hline
 & & & & & & \mathcal{I} & \\
 & -\mathcal{B}_4^f\mathcal{B}_3^f\mathcal{B}_2^f & & \mathcal{B}_4^f\mathcal{B}_3^f & & -\mathcal{B}_4^f & & \mathcal{I}
\end{array}\right]$$

- Moreover $\mathcal{F}_A^{-1}$ is easy to compute explicitly.

Preconditioner — A simple case

- For simplicity, the iteration operator is renamed.

$$(\mathcal{I} - \mathcal{A})\mathbf{g} = \mathbf{b} \Longleftrightarrow \mathcal{F}\mathbf{g} = \mathbf{b}$$

- For a simple waveguide with no internal reflections, $\mathcal{F}$ is known:

$$\mathcal{F}_A^{-1} = \left[\begin{array}{cc|cc|cc|cc}
\mathcal{I} & & -\mathcal{B}_2^b & & \mathcal{B}_2^b\mathcal{B}_3^b & & -\mathcal{B}_2^b\mathcal{B}_3^b\mathcal{B}_4^b & \\
 & \mathcal{I} & & & & & & \\
\hline
 & & \mathcal{I} & & -\mathcal{B}_3^b & & \mathcal{B}_3^b\mathcal{B}_4^b & \\
 & -\mathcal{B}_2^f & & \mathcal{I} & & & & \\
\hline
 & & & & \mathcal{I} & & -\mathcal{B}_4^b & \\
 & \mathcal{B}_3^f\mathcal{B}_2^f & & -\mathcal{B}_3^f & & \mathcal{I} & & \\
\hline
 & & & & & & \mathcal{I} & \\
 & -\mathcal{B}_4^f\mathcal{B}_3^f\mathcal{B}_2^f & & \mathcal{B}_4^f\mathcal{B}_3^f & & -\mathcal{B}_4^f & & \mathcal{I}
\end{array}\right]$$

- Moreover $\mathcal{F}_A^{-1}$ is easy to compute explicitly.

$\mathcal{F}_A^{-1}$ can be a good preconditioner!

Preconditioner — Double sweep

- Constructing explicitly $\mathcal{F}_A^{-1}$ is not needed.
- For iterative solvers only the application of $\mathcal{F}_A^{-1}$ is needed.
- The following recursion can be used:

$$\mathbf{h} = \mathcal{F}_A^{-1}\mathbf{r} = \begin{cases} \begin{cases} h_{2,1} = r_{2,1}, \\ h_{\mathbf{i+1},\mathbf{i}} = r_{i+1,i} - \mathcal{B}_i^f h_{\mathbf{i},\mathbf{i-1}}, \end{cases} & \text{Forward sweep} \\ \\ \begin{cases} h_{N-1,N} = r_{N-1,N}, \\ h_{\mathbf{i-1},\mathbf{i}} = r_{i-1,i} - \mathcal{B}_i^b h_{\mathbf{i},\mathbf{i+1}}. \end{cases} & \text{Backward sweep} \end{cases}$$

⟶ Double-sweep.

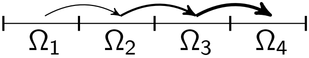
Preconditioner — Double sweep

- Constructing explicitly $\mathcal{F}_A^{-1}$ is not needed.
- For iterative solvers only the application of $\mathcal{F}_A^{-1}$ is needed.
- The following recursion can be used:

$$\mathbf{h} = \mathcal{F}_A^{-1}\mathbf{r} = \begin{cases} \begin{cases} h_{2,1} = r_{2,1}, \\ h_{\mathbf{i+1},\mathbf{i}} = r_{i+1,i} - \mathcal{B}_i^f h_{\mathbf{i},\mathbf{i-1}}, \end{cases} & \text{Forward sweep} \\ \\ \begin{cases} h_{N-1,N} = r_{N-1,N}, \\ h_{\mathbf{i-1},\mathbf{i}} = r_{i-1,i} - \mathcal{B}_i^b h_{\mathbf{i},\mathbf{i+1}}. \end{cases} & \text{Backward sweep} \end{cases}$$

$\longrightarrow$ Double-sweep.

- Information is accumulated from sub-domain to sub-domain.



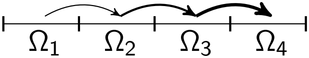
Preconditioner — Double sweep

- Constructing explicitly $\mathcal{F}_A^{-1}$ is not needed.
- For iterative solvers only the application of $\mathcal{F}_A^{-1}$ is needed.
- The following recursion can be used:

$$\mathbf{h} = \mathcal{F}_A^{-1}\mathbf{r} = \begin{cases} \begin{cases} h_{2,1} = r_{2,1}, \\ h_{\mathbf{i+1},\mathbf{i}} = r_{i+1,i} - \mathcal{B}_i^f h_{\mathbf{i},\mathbf{i-1}}, \end{cases} & \text{Forward sweep} \\ \\ \begin{cases} h_{N-1,N} = r_{N-1,N}, \\ h_{\mathbf{i-1},\mathbf{i}} = r_{i-1,i} - \mathcal{B}_i^b h_{\mathbf{i},\mathbf{i+1}}. \end{cases} & \text{Backward sweep} \end{cases}$$

$\longrightarrow$ Double-sweep.

- Information is accumulated from sub-domain to sub-domain.



Unfortunately this is a sequential scheme.

Preconditioner — Benchmark

- 2D waveguide benchmark — TE mode.

	$\omega = 20\pi$					$\omega = 40\pi$				
N	5	10	25	50	100	5	10	25	50	100
0^{th} order + Precond.	3	3	4	4	4	3	3	4	4	4
0^{th} order	8	18	48	98	198					
2^{nd} order + Precond.	3	3	4	4	4	3	3	3	3	4
2^{nd} order	8	18	46	98	201					
OSRC + Precond.	3	3	3	4	4	3	3	4	4	8
OSRC	8	18	48	119	239					

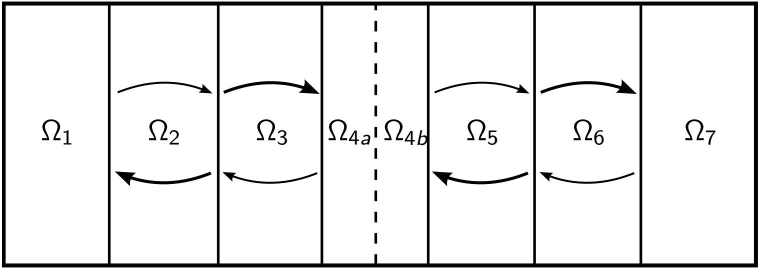
Preconditioner is robust with respect to ω and N variations.

Preconditioner — Partial sweeps

- How to circumvent the sequential nature of the preconditioner?

Preconditioner — Partial sweeps

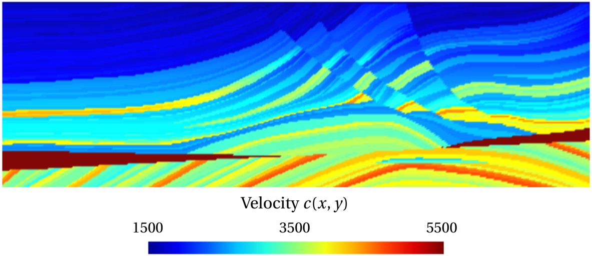
- How to circumvent the sequential nature of the preconditioner?

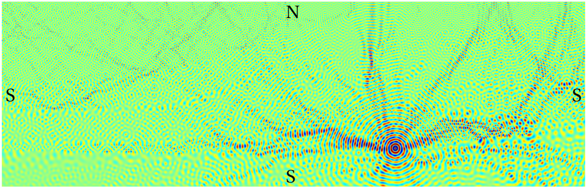


Preconditioner applied on independent blocks.
⟶ Equivalent to a Block-Jacobi applied to $\mathcal{F}_A^{-1}$.

Preconditioner — Benchmark: setup

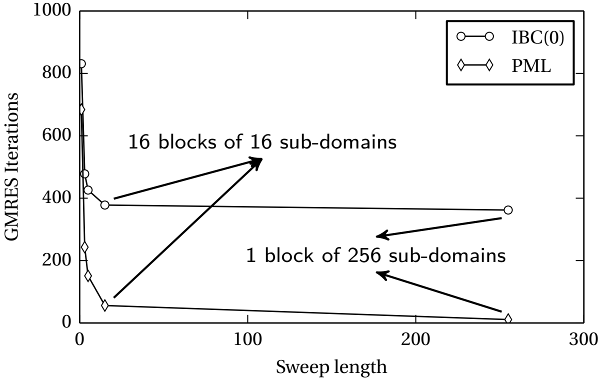
- Marmousi benchmark.
- Pressure field on a $[9 \times 3]$ $[\mathrm{km}]$ domain.
- Domain decomposed into 256 sub-domains.





Preconditioner — Benchmark: iteration count

- Iteration count.



Iteration count stable for sufficiently large sweeps.

Outline

High-order FEM discretizations

Time-harmonic wave problems are known to require very fine meshes.

High-order FEM discretizations

Time-harmonic wave problems are known to require very fine meshes.

The situation can be improved using high-order FEM discretizations.

High-order FEM discretizations

Time-harmonic wave problems are known to require very fine meshes.

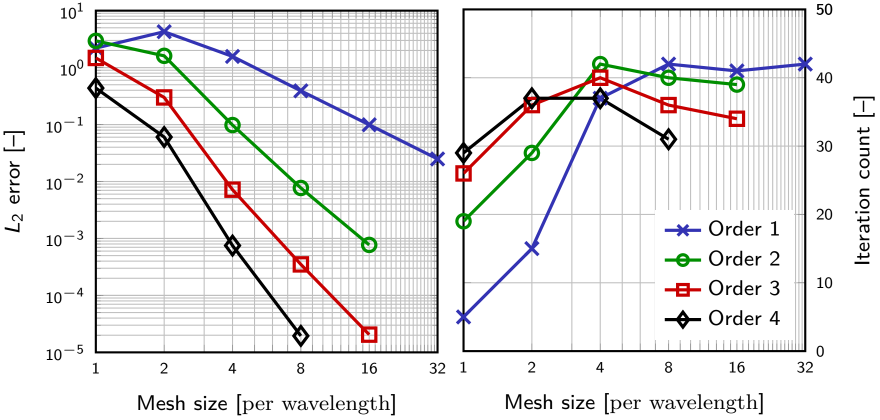
The situation can be improved using high-order FEM discretizations.

1. What happens to the DDM iteration count?
2. What happens when we mix orders for $\mathbf{e}_i$ and $\mathbf{g}$?

⟶ Tests on a waveguide with different orders and mesh sizes.
⟶ On-surface radiation condition is used.

High-order FEM discretizations — Iteration count

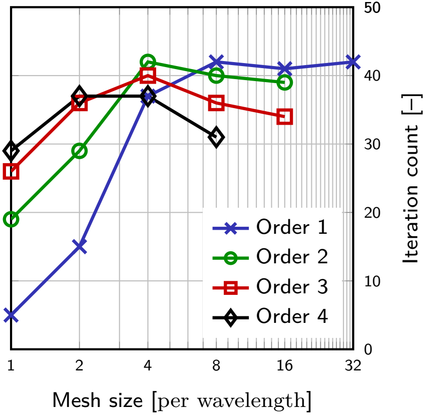
- What happens to the iteration count?



High-order FEM discretizations — Iteration count

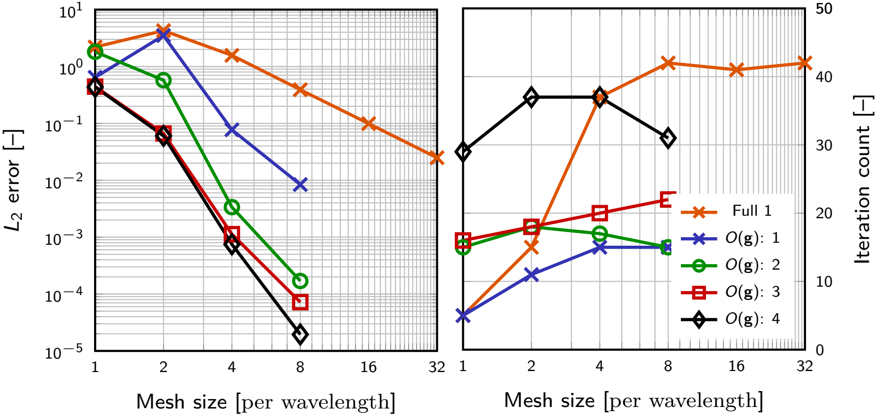
- What happens to the iteration count?

Order	Mesh size	Accuracy	Count
1	16	9.9×10^{-2}	**41**
2	4	9.8×10^{-2}	**42**
2	16	7.7×10^{-4}	**39**
4	4	7.5×10^{-4}	**37**
3	16	2.1×10^{-5}	**34**
4	8	1.9×10^{-5}	**31**



High-order FEM discretizations — Mixing orders

- What happens when we mix orders for $\mathbf{e}_i$ and $\mathbf{g}$?
  ⟶ Unknown $\mathbf{e}_i$ discretized with an order 4 basis.



- The DDM is no longer equivalent to the non-DDM problem.

Outline

Conclusion I

- On-surface radiation condition combined with Padé localization
  ⟶ Offers high linear solver convergence speed.
  ⟶ Robust with respect to a k increase.

- Double sweep
  ⟶ Limits the iteration count increase with sub-domains.

- High-order FEM
  ⟶ Iteration count decreases when accuracy is increased.
  ⟶ Accuracy decreases when mixing orders for $\mathbf{e}_i$ and $\mathbf{g}$...
  ⟶ ...but, iteration count decreases also.

Conclusion II

- Open-source implementation available for testing!

- Go to `http://onelab.info`.
- Download the pre-compiled `GetDDM` code.
  - Available for MS Windows, MacOS X, and Linux.
  - Makes use of `Gmsh` and `GetDP` codes.
- Open `Gmsh` executable.
- Open `./models/ddm_waves/main.pro`.
- Click `Run`.

- High-order version will come soon.

Conclusion II

- Open-source implementation available for testing!

- Go to `http://onelab.info`.
- Download the pre-compiled `GetDDM` code.
  - Available for MS Windows, MacOS X, and Linux.
  - Makes use of `Gmsh` and `GetDP` codes.
- Open `Gmsh` executable.
- Open `./models/ddm_waves/main.pro`.
- Click `Run`.

- High-order version will come soon.

Thank you for your attention!