NEMESIS 2024 - Montpellier, France

June 19-21, 2024

Space-time RAS for wave problems on polytopal meshes

Ilario Mazzieri

POLITECNICO di MILANO (Italy) Department of Mathematics
MOX Laboratory for Modeling and Scientific Computing

Joint work with:
G. Ciaramella (PoliMi), M. Gander (UniGe).

POLITECNICO MILANO 1863
DIPARTIMENTO DI MATEMATICA
DEPARTMENT OF EXCELLENCE 2023-2027

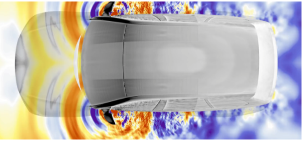
NEMESIS
New generation methods for numerical simulations

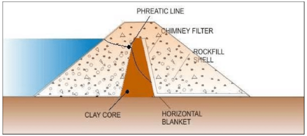
Introduction

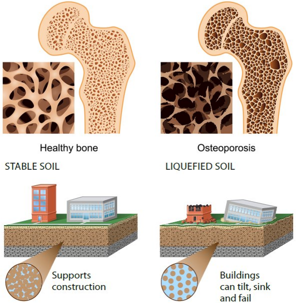
Motivations: challenging demands on computational methods

Features of the multi-physical model

- Thin structures and highly heterogeneous media
- Scattered fields at high-frequency/small-wavelength
- Proper representation of the hydraulic conctact at the interfaces
- Possibly nonlinear coupled problem

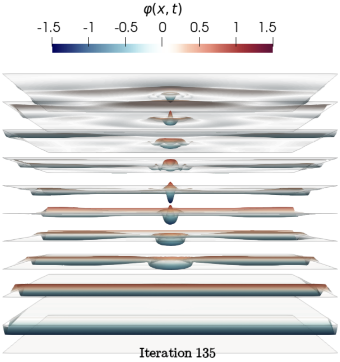






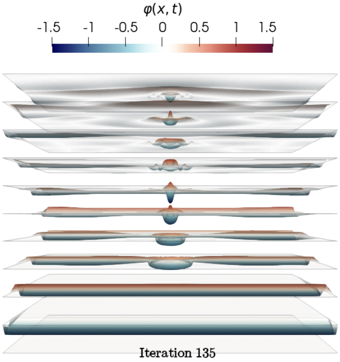
Why using space-time methods? (instead of space discretization & time stepping)

- High-order **accuracy** in both *space and time* can be easily obtained
- Spectral **convergence** can be otained by polynomial refinement
- **Implicit** methods or **explicit** methods with **local** CFL condition
- The numerical solution is available at **all times** in (0,T)



Why using space-time methods? (instead of space discretization & time stepping)

- High-order **accuracy** in both *space and time* can be easily obtained
- Spectral **convergence** can be otained by polynomial refinement
- **Implicit** methods or **explicit** methods with **local** CFL condition
- The numerical solution is available at **all times** in (0,T)



Major drawback: high complexity

- time dependent problems in d space dimensions $\rightarrow$ $(d+1)$-dimensional problems

Introduction

State of the art: minimal bibliography

Cartesian mesh

- [Hughes & Hulbert, 1988], [Monk & Richter 2005], [Dumbser et al. 2007], [Steinbach & Zank, 2017], [Ernesti & Wieners, 2019], [Bansal et al. 2020]

Unstructured mesh

- [Idesman 2007], [Abedi et al, 2006], [Dörfler et al. 2016], [Kretzschmar et al. 2016], [Banjai et al. 2017], [Barucq et al. 2018], [Gopalakrishnan et al. 2017], [Moiola & Perugia, 2018], [Perugia et al. 2020]

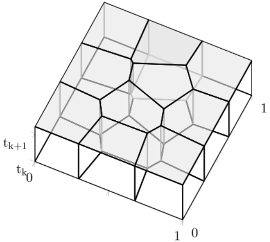
Cartesian mesh

- [Hughes & Hulbert, 1988], [Monk & Richter 2005], [Dumbser et al. 2007], [Steinbach & Zank, 2017], [Ernesti & Wieners, 2019], [Bansal et al. 2020]

Unstructured mesh

- [Idesman 2007], [Abedi et al, 2006], [Dörfler et al. 2016], [Kretzschmar et al. 2016], [Banjai et al. 2017], [Barucq et al. 2018], [Gopalakrishnan et al. 2017], [Moiola & Perugia, 2018], [Perugia et al. 2020]

Our contribution: Acceleration of the solution algorithm for dG methods applied to wave problems on prismatic meshes (**tensor-product** meshes obtained starting from a general **polygonal grid** for the space discretization).

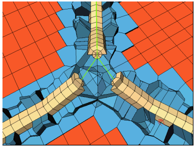


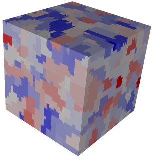
Space discretization by the PolydG method

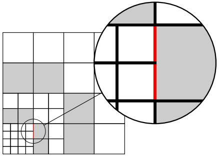
[Riviere, (2008); Di Pietro, Ern, (2012); Cangiani, Dong, Georgoulis, Houston, (2017)]

Advantages of the PolyDG discretization

- Support of **general polytopal** meshes, local **mesh refinement & coarsening**
- **High-order** accuracy
- **Robustness** with respect to heterogeneous media
- **Scalable** and **parallel implementation** algorithm





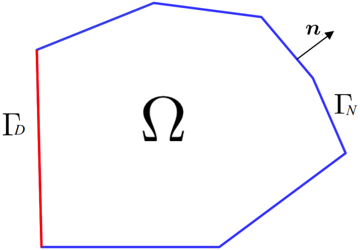


Under suitable hypothesis on the mesh, we can exploit trace-inverse inequality and optimal approximation results in $\mathbb{P}^{\ell}(\mathcal{T}_h)$.

Mathematical model: governing equations

Wave equation: find $(\varphi, \psi) : \Omega \times (0, T] \to \mathbb{R}$ such that

$$
\begin{cases}
\partial_t \varphi - \psi = 0, & \text{in } \Omega \times (0, T], \\
\partial_t \psi - c^2 \Delta \varphi = f, & \text{in } \Omega \times (0, T], \\
\varphi = 0, & \text{in } \Gamma_D \times (0, T], \\
\nabla \varphi \cdot \boldsymbol{n} = g_a, & \text{in } \Gamma_N \times (0, T], \\
(\varphi, \psi)(t = 0) = (\varphi_0, \psi_0) & \text{in } \Omega.
\end{cases}
$$



- Ω is an open bounded polygonal domain with Lipschitz boundary.
- c is the wave speed velocity
- g_a, φ_0, and ψ_0 are regular data.

Space-discretization: PolydG discretization

Functional space: $V_h = \{\varphi_h \in L^2(\Omega) : \varphi_{h|\kappa} \in [\mathcal{P}_p(\kappa)],\ p \geqslant 1\ \forall \kappa \in \mathcal{T}_h\}.$

Space-discretization: PolydG discretization

Functional space: $V_h = \{\varphi_h \in L^2(\Omega) : \varphi_{h|\kappa} \in [\mathcal{P}_p(\kappa)],\ p \geqslant 1\ \forall \kappa \in \mathcal{T}_h\}.$

Variational formulation: For $t \in (0, T]$, find $(\varphi_h, \psi_h)(t) \in \boldsymbol{V}_h = V_h \times V_h$, s.t.

$$\begin{cases} \mathcal{M}(\dot{\varphi}_h, w) - \mathcal{M}(\psi_h, w) = 0 & \forall w \in V_h \\ \mathcal{M}(\dot{\psi}_h, z) + \mathcal{A}_h(\varphi_h, z) = \mathcal{F}(z) & \forall z \in V_h. \end{cases}$$

Space-discretization: PolydG discretization

Functional space: $V_h = \{\varphi_h \in L^2(\Omega) : \varphi_{h|\kappa} \in [\mathcal{P}_p(\kappa)],\ p \geqslant 1\ \forall \kappa \in \mathcal{T}_h\}$.

Variational formulation: For $t \in (0, T]$, find $(\varphi_h, \psi_h)(t) \in \boldsymbol{V}_h = V_h \times V_h$, s.t.

$$\begin{cases} \mathcal{M}(\dot{\varphi}_h, w) - \mathcal{M}(\psi_h, w) = 0 & \forall w \in V_h \\ \mathcal{M}(\dot{\psi}_h, z) + \mathcal{A}_h(\varphi_h, z) = \mathcal{F}(z) & \forall z \in V_h. \end{cases}$$

$$\mathcal{M}(u, v) = (u, v)_\Omega \quad \forall u, v \in V_h,$$

$$\mathcal{A}_h(u, v) = (c^2 \nabla_h u, \nabla_h v)_\Omega - \langle \{c^2 \nabla_h u\}, [\![v]\!] \rangle_{\mathcal{F}_h} - \langle [\![u]\!], \{c^2 \nabla_h v\} \rangle_{\mathcal{F}_h} + \langle \chi [\![u]\!], [\![v]\!] \rangle_{\mathcal{F}_h} \quad \forall u, v \in V_h.$$

- Standard notation for jump $[\![\cdot]\!]$ and average $\{\cdot\}$ operators
- $\chi \in L^\infty(\mathcal{F}_h)$ is a suitable *stabilization function*

Space-discretization: PolydG discretization

Functional space: $V_h = \{\varphi_h \in L^2(\Omega) : \varphi_{h|\kappa} \in [\mathcal{P}_p(\kappa)],\ p \geqslant 1\ \forall \kappa \in \mathcal{T}_h\}$.

Variational formulation: For $t \in (0,T]$, find $(\varphi_h, \psi_h)(t) \in \boldsymbol{V}_h = V_h \times V_h$, s.t.

$$\begin{cases} \mathcal{M}(\dot{\varphi}_h, w) - \mathcal{M}(\psi_h, w) = 0 & \forall w \in V_h \\ \mathcal{M}(\dot{\psi}_h, z) + \mathcal{A}_h(\varphi_h, z) = \mathcal{F}(z) & \forall z \in V_h. \end{cases}$$

$$\mathcal{M}(u,v) = (u,v)_\Omega \quad \forall u, v \in V_h,$$

$$\mathcal{A}_h(u,v) = (c^2 \nabla_h u, \nabla_h v)_\Omega - \langle \{\!\{c^2 \nabla_h u\}\!\}, [\![v]\!] \rangle_{\mathcal{F}_h} - \langle [\![u]\!], \{\!\{c^2 \nabla_h v\}\!\} \rangle_{\mathcal{F}_h} + \langle \chi [\![u]\!], [\![v]\!] \rangle_{\mathcal{F}_h} \quad \forall u, v \in V_h.$$

- Standard notation for jump $[\![\cdot]\!]$ and average $\{\!\{\cdot\}\!\}$ operators
- $\chi \in L^\infty(\mathcal{F}_h)$ is a suitable *stabilization function*

Remarks:

- Stability estimates are independent of the discretization parameters
- Optimal order of convergence (energy mesh dependent norm) w.r.t. the mesh sizes h and suboptimal w.r.t. the polynomial dregree p.

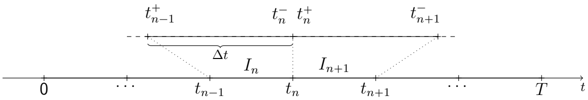
Differential system: for any $t \in (0,T]$, find $(\boldsymbol{\varphi}_h, \boldsymbol{\psi}_h) \in \mathbb{R}^{N_h} \times \mathbb{R}^{N_h}$ s.t.

$$\begin{bmatrix} I & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\varphi}}_h \\ \dot{\boldsymbol{\psi}}_h \end{bmatrix}(t) + \begin{bmatrix} 0 & -I \\ A & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\varphi}_h \\ \boldsymbol{\psi}_h \end{bmatrix}(t) = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{F}_h \end{bmatrix}(t), \quad t \in (0,T].$$

Differential system: for any $t \in (0,T]$, find $(\boldsymbol{\varphi}_h, \boldsymbol{\psi}_h) \in \mathbb{R}^{N_h} \times \mathbb{R}^{N_h}$ s.t.

$$\begin{bmatrix} I & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\varphi}}_h \\ \dot{\boldsymbol{\psi}}_h \end{bmatrix}(t) + \begin{bmatrix} 0 & -I \\ A & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\varphi}_h \\ \boldsymbol{\psi}_h \end{bmatrix}(t) = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{F}_h \end{bmatrix}(t), \quad t \in (0,T].$$

Mesh partition:



Time interval $I = (0,T]$ partitioned into N_T time-slabs $I_n = (t_{n-1}, t_n]$ having length $\Delta t = t_n - t_{n-1}$, for $n = 1, \ldots, N$ with $t_0 = 0$ and $t_N = T$.

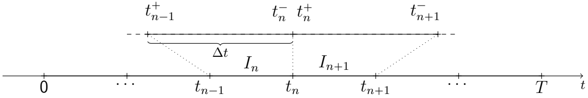
Time integration

dG method for first order differential system

Differential system: for any $t \in (0,T]$, find $(\boldsymbol{\varphi}_h, \boldsymbol{\psi}_h) \in \mathbb{R}^{N_h} \times \mathbb{R}^{N_h}$ s.t.

$$\begin{bmatrix} I & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\varphi}}_h \\ \dot{\boldsymbol{\psi}}_h \end{bmatrix}(t) + \begin{bmatrix} 0 & -I \\ A & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\varphi}_h \\ \boldsymbol{\psi}_h \end{bmatrix}(t) = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{F}_h \end{bmatrix}(t), \quad t \in (0,T].$$

Mesh partition:



Time interval $I = (0,T]$ partitioned into N_T time-slabs $I_n = (t_{n-1}, t_n]$ having length $\Delta t = t_n - t_{n-1}$, for $n = 1, \ldots, N$ with $t_0 = 0$ and $t_N = T$.

Polynomial spaces: $V_n^r = \{\boldsymbol{\varphi} : I_n \to \mathbb{R}^{N_h}$ s.t. $\boldsymbol{\varphi} \in [\mathcal{P}^r(I_n)]^{N_h}\}$, where $N_h = dim(V_h)$ and

$$\mathcal{V}_{dG} = \{\boldsymbol{\varphi} \in \boldsymbol{L}^2(0,T] \text{ s.t. } \boldsymbol{\varphi}|_{I_n} \in V_n^r\}.$$

Time integration

dG method for first order differential system

dG formulation: find $(\varphi_{dG}, \psi_{dG}) \in \mathcal{V}_{dG} \times \mathcal{V}_{dG}$ such that

$$
\begin{cases}
\displaystyle\sum_{n=1}^{N_T} (\dot{\boldsymbol{\varphi}}_{dG}, \boldsymbol{w})_{I_n} - (\boldsymbol{\psi}_{dG}, \boldsymbol{w})_{I_n} + \sum_{n=1}^{N_T-1} [\boldsymbol{\varphi}_{dG}]_n \cdot \boldsymbol{w}(t_n^+) + \boldsymbol{\varphi}_{dG}(0^+) \cdot \boldsymbol{w}(0^+), & = \boldsymbol{\varphi}_0 \cdot \boldsymbol{w}(0^+) \\
\displaystyle\sum_{n=1}^{N_T} (M\dot{\boldsymbol{\psi}}_{dG}, \boldsymbol{z})_{I_n} + (A\boldsymbol{\varphi}_{dG}, \boldsymbol{z})_{I_n} + \sum_{n=1}^{N_T-1} M[\boldsymbol{\psi}_{dG}]_n \cdot \boldsymbol{z}(t_n^+) + M\boldsymbol{\psi}_{dG}(0^+) \cdot \boldsymbol{z}(0^+) & = \displaystyle\sum_{n=1}^{N_T} (\boldsymbol{F}_h, \boldsymbol{z})_{I_n} \\
& \quad + M\boldsymbol{\psi}_0 \cdot \boldsymbol{z}(0^+),
\end{cases}
$$

for any $(\boldsymbol{w}, \boldsymbol{z}) \in \mathcal{V}_{dG} \times \mathcal{V}_{dG}$ and where $[\boldsymbol{w}]_n = \boldsymbol{w}(t_n^+) - \boldsymbol{w}(t_n^-)$ for any $\boldsymbol{w} \in \mathcal{V}_{dG}$.

Time integration

dG method for first order differential system

dG formulation: find $(\boldsymbol{\varphi}_{dG}, \boldsymbol{\psi}_{dG}) \in \mathcal{V}_{dG} \times \mathcal{V}_{dG}$ such that

$$\begin{cases} \displaystyle\sum_{n=1}^{N_T} (\dot{\boldsymbol{\varphi}}_{dG}, \boldsymbol{w})_{I_n} - (\boldsymbol{\psi}_{dG}, \boldsymbol{w})_{I_n} + \sum_{n=1}^{N_T-1} [\boldsymbol{\varphi}_{dG}]_n \cdot \boldsymbol{w}(t_n^+) + \boldsymbol{\varphi}_{dG}(0^+) \cdot \boldsymbol{w}(0^+), & = \boldsymbol{\varphi}_0 \cdot \boldsymbol{w}(0^+) \\ \displaystyle\sum_{n=1}^{N_T} (M\dot{\boldsymbol{\psi}}_{dG}, \boldsymbol{z})_{I_n} + (A\boldsymbol{\varphi}_{dG}, \boldsymbol{z})_{I_n} + \sum_{n=1}^{N_T-1} M[\boldsymbol{\psi}_{dG}]_n \cdot \boldsymbol{z}(t_n^+) + M\boldsymbol{\psi}_{dG}(0^+) \cdot \boldsymbol{z}(0^+) & = \displaystyle\sum_{n=1}^{N_T} (\boldsymbol{F}_h, \boldsymbol{z})_{I_n} \\ & \quad + M\boldsymbol{\psi}_0 \cdot \boldsymbol{z}(0^+), \end{cases}$$

for any $(\boldsymbol{w}, \boldsymbol{z}) \in \mathcal{V}_{dG} \times \mathcal{V}_{dG}$ and where $[\boldsymbol{w}]_n = \boldsymbol{w}(t_n^+) - \boldsymbol{w}(t_n^-)$ for any $\boldsymbol{w} \in \mathcal{V}_{dG}$.

Remarks:

- Stability estimate (mesh dependent seminorm) is independent of the discretization parameters
- Optimal order of convergence (mesh dependent seminorm) w.r.t. the time step Δt ($\mathcal{O}(\Delta t^{r+\frac{1}{2}})$) suboptimal w.r.t. the polynomial dregree r.

Space-time linear system

Block bi-diagonal matrix of the form

$$\begin{bmatrix} G_1 & 0 & \dots & \dots & 0 \\ G_0 & G_1 & \ddots & & \vdots \\ 0 & G_0 & G_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & G_0 & G_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \\ \vdots \\ \vdots \\ \boldsymbol{\alpha}_{N_T} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \vdots \\ \boldsymbol{b}_{N_T} \end{bmatrix}, \qquad \text{with} \qquad G_1 = \begin{bmatrix} I & * \\ 0 & \widehat{A} \end{bmatrix}$$

where

- I is the identity matrix in $\mathbb{R}^{N_h(r+1)}$,
- $\widehat{A} = M \otimes (L_1 + L_3) + A \otimes L_7$, with L_i time matrices,
- $\boldsymbol{\alpha}_i = [\boldsymbol{\alpha}_i^{\varphi}, \boldsymbol{\alpha}_i^{\psi}]^T$ are the unknown expansions coefficients in I_i and $\boldsymbol{b}_i$ are the right hand sides.

Solution strategy

Advance in time by solving:

$$G_1 \boldsymbol{\alpha}_1 = \boldsymbol{b}_1,$$

$$G_1 \boldsymbol{\alpha}_i = \boldsymbol{b}_i - G_0 \boldsymbol{\alpha}_{i-1}, \quad i = 2, \ldots, N_T$$

Solution strategy

Advance in time by solving:

$$G_1 \boldsymbol{\alpha}_1 = \boldsymbol{b}_1,$$
$$G_1 \boldsymbol{\alpha}_i = \boldsymbol{b}_i - G_0 \boldsymbol{\alpha}_{i-1}, \quad i = 2, \ldots, N_T$$

Remarks:

✓ Equivalence between time-dG and fully implicit Runge-Kutta methods

Solution strategy

Advance in time by solving:

$$G_1 \boldsymbol{\alpha}_1 = \boldsymbol{b}_1,$$

$$G_1 \boldsymbol{\alpha}_i = \boldsymbol{b}_i - G_0 \boldsymbol{\alpha}_{i-1}, \quad i = 2, ..., N_T$$

Remarks:

- ✓ Equivalence between time-dG and fully implicit Runge-Kutta methods
- ✓ Good properties of A-stability, L-stability and B-stability $\rightarrow$ good for stiff problems

Solution strategy

Advance in time by solving:

$$G_1\boldsymbol{\alpha}_1 = \boldsymbol{b}_1,$$
$$G_1\boldsymbol{\alpha}_i = \boldsymbol{b}_i - G_0\boldsymbol{\alpha}_{i-1}, \quad i = 2, ..., N_T$$

Remarks:

- ✓ Equivalence between time-dG and fully implicit Runge-Kutta methods
- ✓ Good properties of A-stability, L-stability and B-stability → good for stiff problems
- ✓ In practice one needs to invert only the matrix $\widehat{A}$

Solution strategy

Advance in time by solving:

$$G_1 \boldsymbol{\alpha}_1 = \boldsymbol{b}_1,$$
$$G_1 \boldsymbol{\alpha}_i = \boldsymbol{b}_i - G_0 \boldsymbol{\alpha}_{i-1}, \quad i = 2, ..., N_T$$

Remarks:

- ✓ Equivalence between time-dG and fully implicit Runge-Kutta methods
- ✓ Good properties of A-stability, L-stability and B-stability → good for stiff problems
- ✓ In practice one needs to invert only the matrix $\widehat{A}$
- ✗ System matrix $\widehat{A}$ is not symmetric

Solution strategy

Advance in time by solving:

$$G_1 \boldsymbol{\alpha}_1 = \boldsymbol{b}_1,$$
$$G_1 \boldsymbol{\alpha}_i = \boldsymbol{b}_i - G_0 \boldsymbol{\alpha}_{i-1}, \quad i = 2, \ldots, N_T$$

Remarks:

- ✓ Equivalence between time-dG and fully implicit Runge-Kutta methods
- ✓ Good properties of A-stability, L-stability and B-stability → good for stiff problems
- ✓ In practice one needs to invert only the matrix $\widehat{A}$
- ✗ System matrix $\widehat{A}$ is not symmetric
- ✗ Dimension of $\widehat{A}$ → iterative solver (matrix-free) to be used for large problems

Solution strategy

Advance in time by solving:

$$G_1 \boldsymbol{\alpha}_1 = \boldsymbol{b}_1,$$
$$G_1 \boldsymbol{\alpha}_i = \boldsymbol{b}_i - G_0 \boldsymbol{\alpha}_{i-1}, \quad i = 2, ..., N_T$$

Remarks:

- ✓ Equivalence between time-dG and fully implicit Runge-Kutta methods
- ✓ Good properties of A-stability, L-stability and B-stability → good for stiff problems
- ✓ In practice one needs to invert only the matrix $\widehat{A}$
- ✗ System matrix $\widehat{A}$ is not symmetric
- ✗ Dimension of $\widehat{A}$ → iterative solver (matrix-free) to be used for large problems

Example: $\Omega = (0,1)^2 \times (0,2]$ partioned into 400×40 polygonal prisms. Polynomial degree $p = r = 2$ and $dim(\widehat{A}) = 576.000$.

	Time to solution [s]
Direct	8
GMRES	109

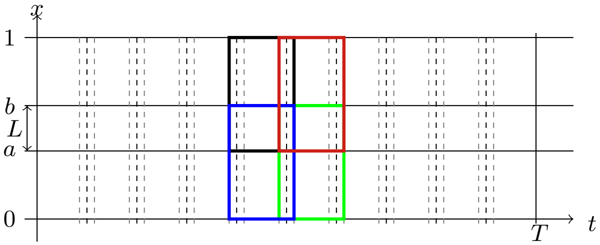
Fast solution techiques for space-time discretizations

Space-time decomposition and XT-RAS method

- Let us consider the fully discretize problem of the form: $A\boldsymbol{u} = \boldsymbol{f}, \quad A \in \mathbb{R}^{N\times N}, \boldsymbol{u}, \boldsymbol{f} \in \mathbb{R}^N, N >> 1.$

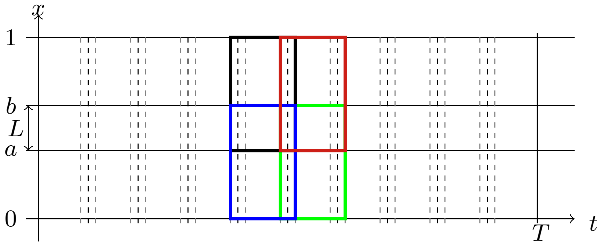
Space-time decomposition and XT-RAS method

- Let us consider the fully discretize problem of the form: $A\boldsymbol{u} = \boldsymbol{f}, \quad A \in \mathbb{R}^{N \times N}, \boldsymbol{u}, \boldsymbol{f} \in \mathbb{R}^{N}$, $N >> 1$.
- Set $N_X = 2$-subdomain space decomposition $\Omega_1 = (0, b)$ and $\Omega_2 = (a, 1)$ with $L = b - a > 0$ and split $[0, T]$ in N_T overlapping subdomains:



Space-time decomposition and XT-RAS method

- Let us consider the fully discretize problem of the form: $A\boldsymbol{u} = \boldsymbol{f}, \quad A \in \mathbb{R}^{N \times N}, \boldsymbol{u}, \boldsymbol{f} \in \mathbb{R}^N, N >> 1.$
- Set $N_X = 2$-subdomain space decomposition $\Omega_1 = (0, b)$ and $\Omega_2 = (a, 1)$ with $L = b - a > 0$ and split $[0, T]$ in N_T overlapping subdomains:



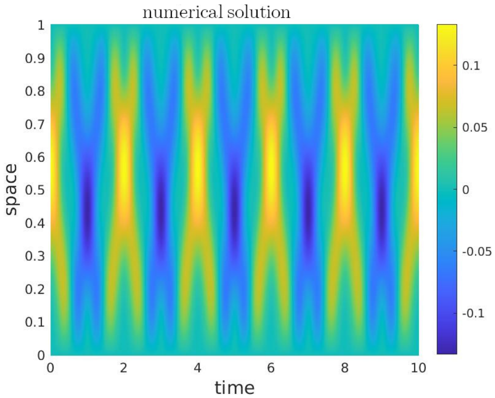
We consider the classical RAS form (now XT)

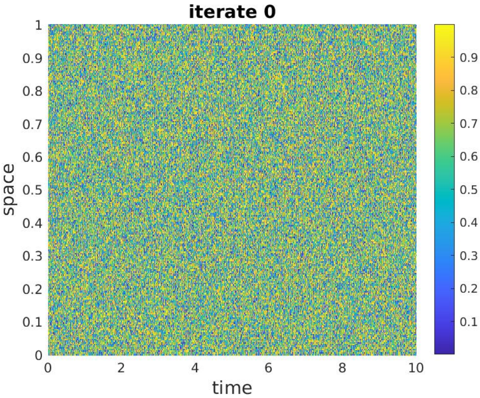
$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \sum_{k=1}^{N_X N_T} \widetilde{R}_k^\top A_k^{-1} R_k(\boldsymbol{f} - A\boldsymbol{u}^n),$$

where

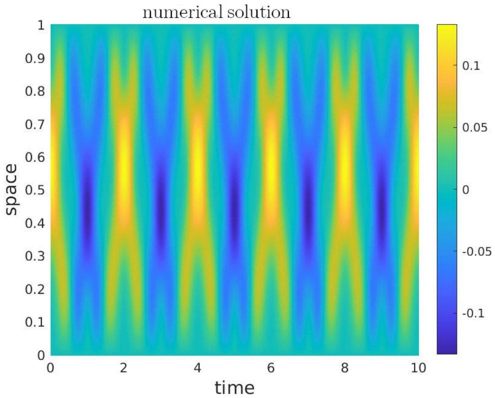
- R_k and $\widetilde{R}_k$ are the XT restriction matrices (including a partition of unity),
- A_k are XT matrices corresponding to the local subproblems.

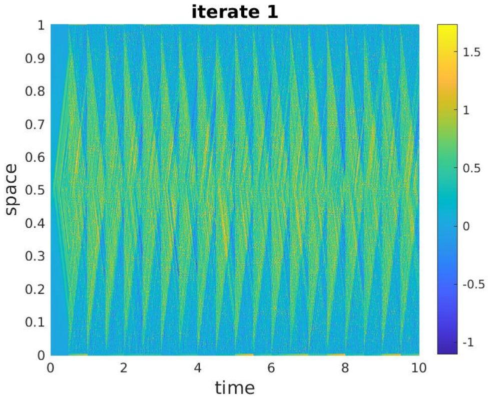
XT-RAS: Numerical experiments ($T = 10,\ N_X = 2,\ N_T = 20$)

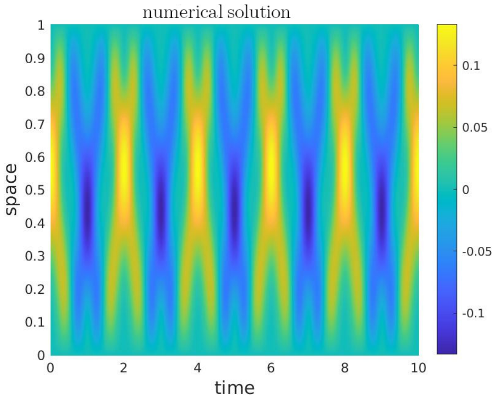




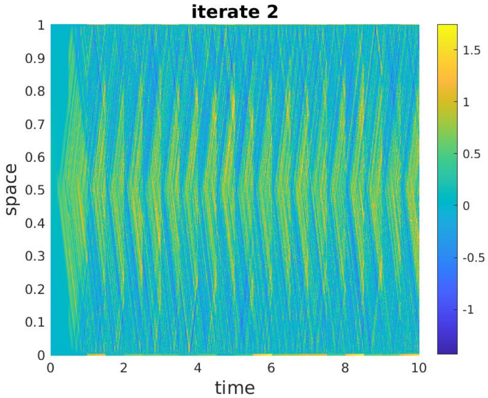
XT-RAS: Numerical experiments ($T = 10$, $N_X = 2$, $N_T = 20$)





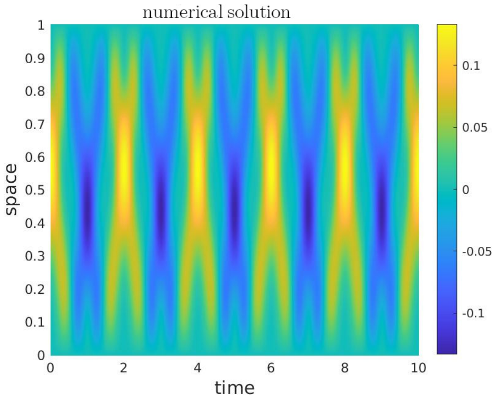


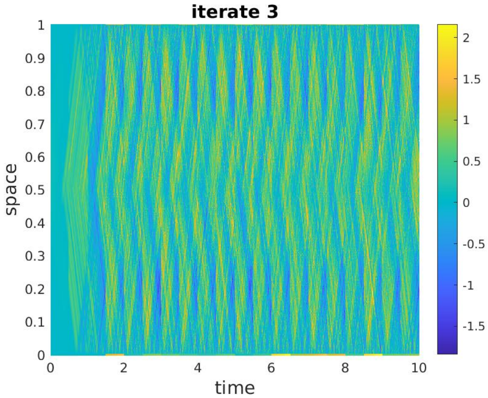
numerical solution
space
time
0
2
4
6
8
10
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.1
0.05
0
-0.05
-0.1



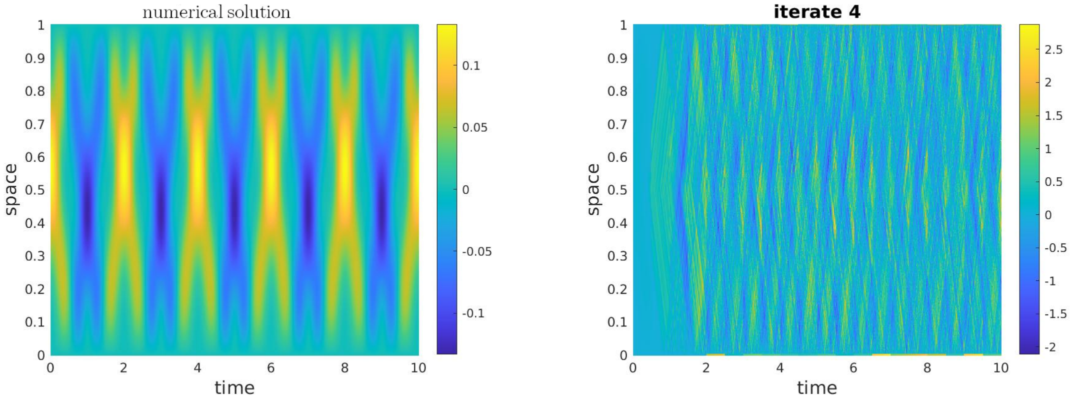
iterate 2
space
time
0
2
4
6
8
10
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.5
1
0.5
0
-0.5
-1

XT-RAS: Numerical experiments ($T = 10,\ N_X = 2,\ N_T = 20$)

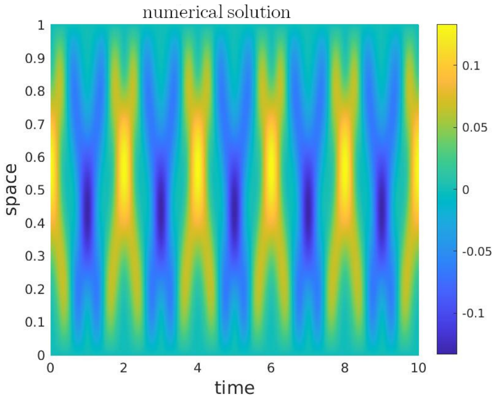


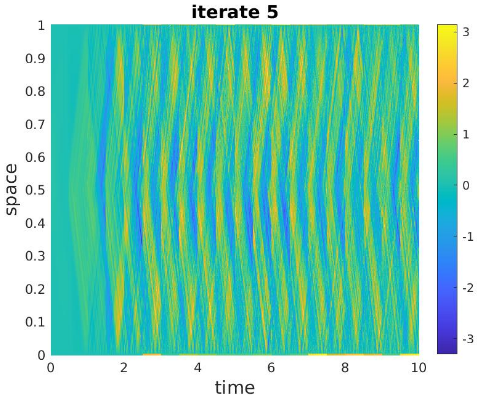


XT-RAS: Numerical experiments ($T = 10$, $N_X = 2$, $N_T = 20$)

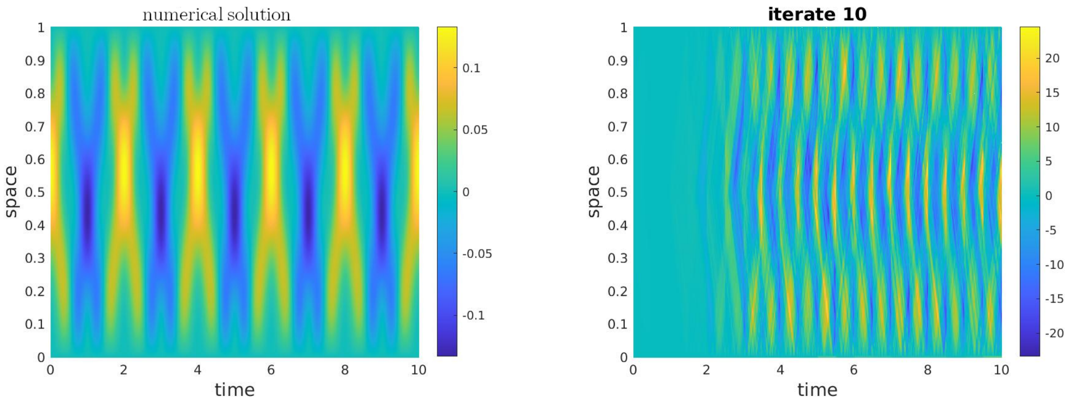


XT-RAS: Numerical experiments ($T = 10,\ N_X = 2,\ N_T = 20$)

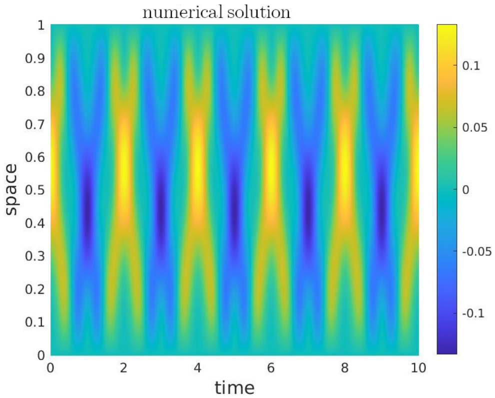


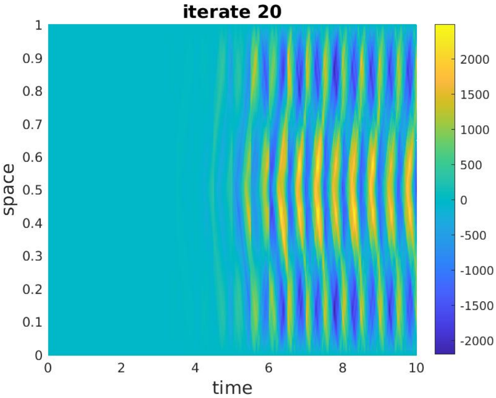


XT-RAS: Numerical experiments ($T = 10,\ N_X = 2,\ N_T = 20$)

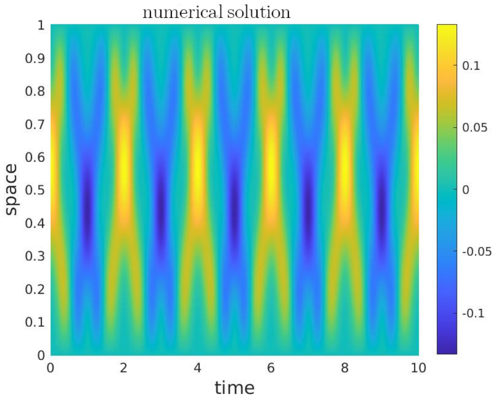


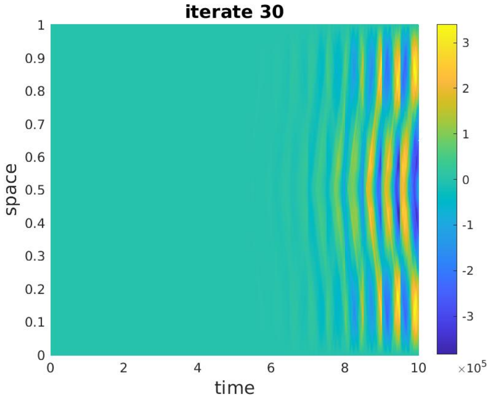
XT-RAS: Numerical experiments ($T = 10,\ N_X = 2,\ N_T = 20$)



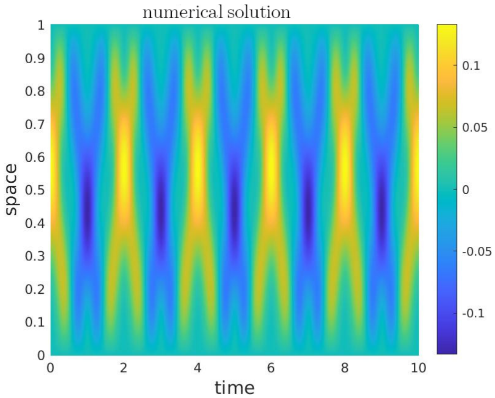


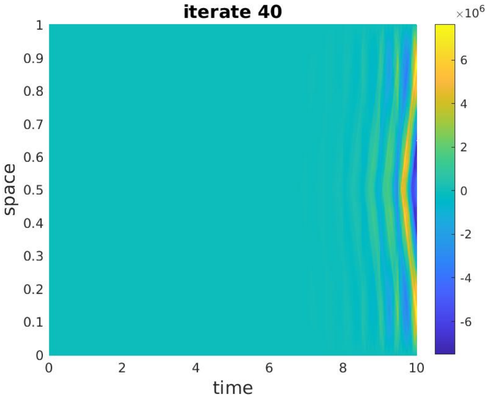
XT-RAS: Numerical experiments ($T = 10$, $N_X = 2$, $N_T = 20$)



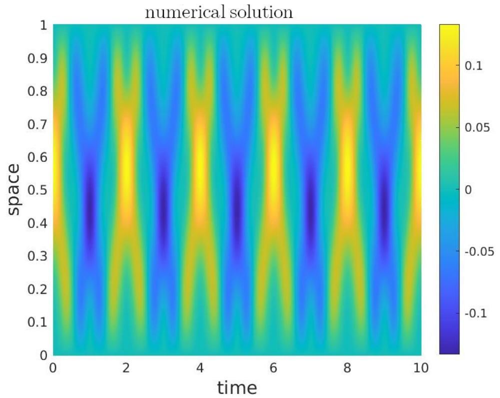


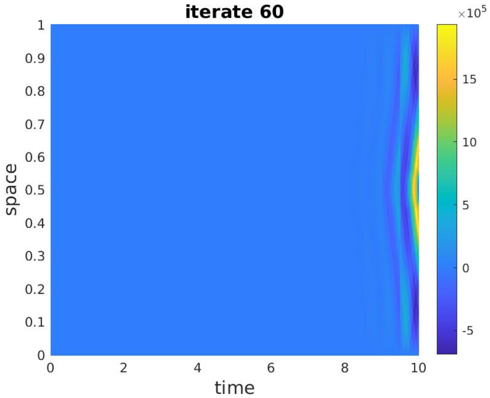
XT-RAS: Numerical experiments ($T = 10$, $N_X = 2$, $N_T = 20$)



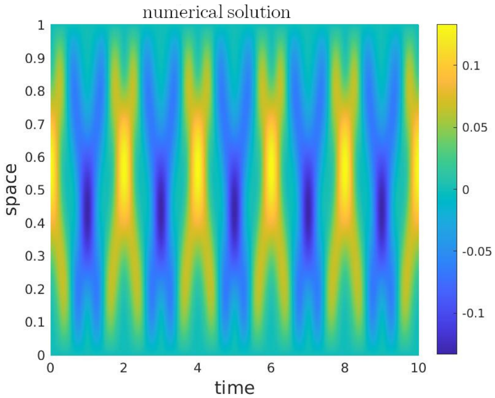


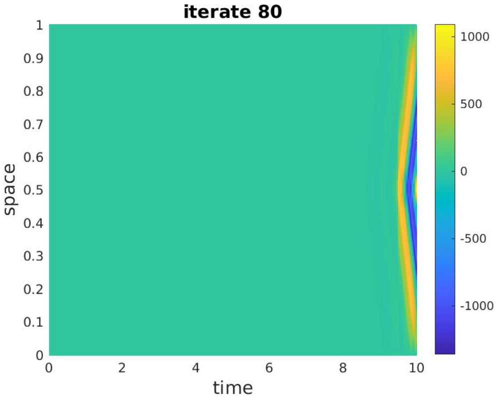
XT-RAS: Numerical experiments ($T = 10,\ N_X = 2,\ N_T = 20$)



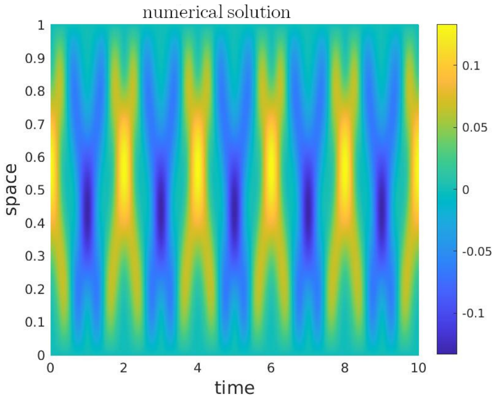


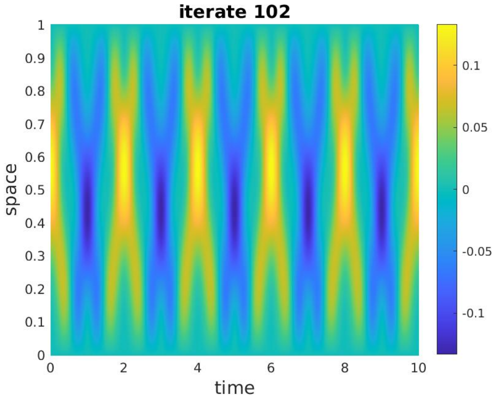
XT-RAS: Numerical experiments ($T = 10$, $N_X = 2$, $N_T = 20$)





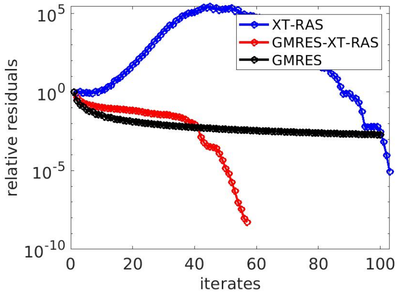
XT-RAS: Numerical experiments ($T = 10,\ N_X = 2,\ N_T = 20$)



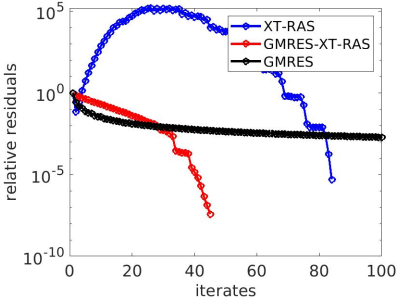


XT-RAS: Numerical experiments ($T = 10$, $N_X = 2$, $N_T = 20$)

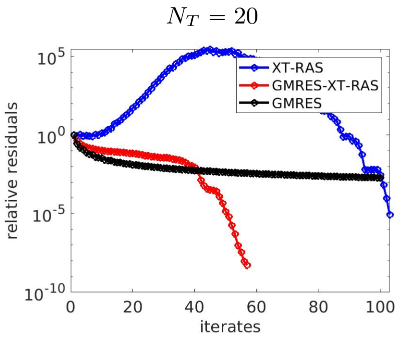
$N_T = 20$

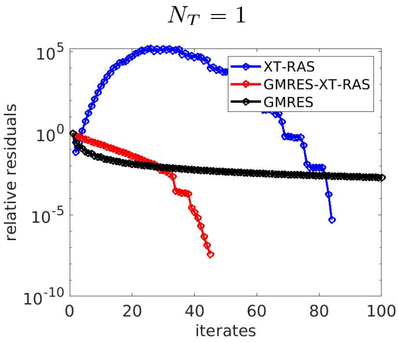


$N_T = 1$



XT-RAS: Numerical experiments ($T = 10$, $N_X = 2$, $N_T = 20$)

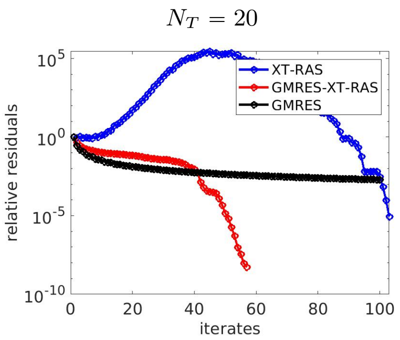


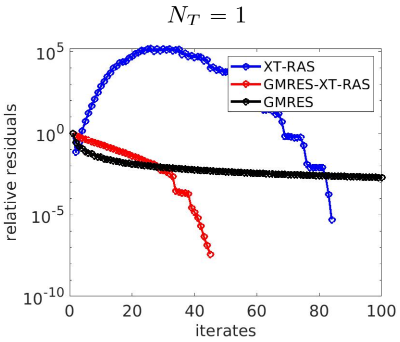


Remarks:

- There is no much gain in using a decomposition in time ($N_T = 1$ Waveform Relaxation (WR)).
- The "good information" need to propagate through the time subdomains to reach T.
- The error propagates and grows through the subdomains. $\rightarrow$ Useless subdomain solves.

XT-RAS: Numerical experiments ($T = 10$, $N_X = 2$, $N_T = 20$)



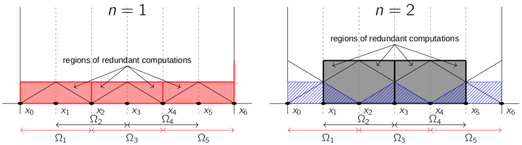


Remarks:

- There is no much gain in using a decomposition in time ($N_T = 1$ Waveform Relaxation (WR)).
- The "good information" need to propagate through the time subdomains to reach T.
- The error propagates and grows through the subdomains. $\rightarrow$ Useless subdomain solves.
- Error analysis: error contraction for $T < \frac{2\min(\tilde{L}_1, \tilde{L}_2)+2L}{c}$ $\rightarrow$ Better to use large overlap.

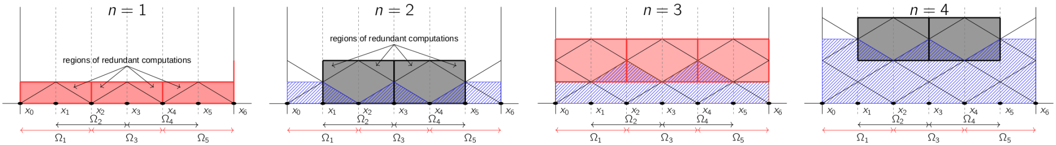
XT-WR/XT-RAS and Unmapped Tent-Pitching Scheme

- Consider a space-decomposition as in the figure (with generous overlap),
- a time decomposition with $T_{\text{sub}} = \frac{|\Omega_j|}{c}$ ($T_{\text{sub}} = \frac{|\Omega_j|}{2c}$ at first iteration ... rectangular tents),
- and a red-black WR iteration (RBSWR).



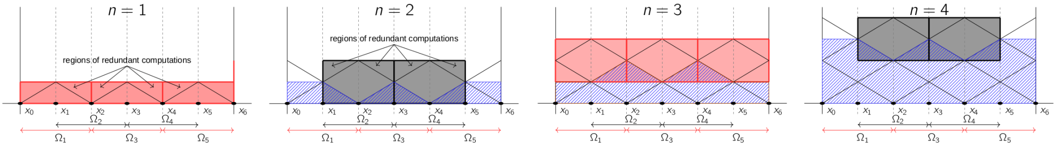
XT-WR/XT-RAS and Unmapped Tent-Pitching Scheme

- Consider a space-decomposition as in the figure (with generous overlap),
- a time decomposition with $T_{\text{sub}} = \frac{|\Omega_j|}{c}$ ($T_{\text{sub}} = \frac{|\Omega_j|}{2c}$ at first iteration ... rectangular tents),
- and a red-black WR iteration (RBSWR).



XT-WR/XT-RAS and Unmapped Tent-Pitching Scheme

- Consider a space-decomposition as in the figure (with generous overlap),
- a time decomposition with $T_{\text{sub}} = \frac{|\Omega_j|}{c}$ ($T_{\text{sub}} = \frac{|\Omega_j|}{2c}$ at first iteration ... rectangular tents),
- and a red-black WR iteration (RBSWR).



- Equivalent to a mapped tent pitching scheme [Gopalakrishnan, 2017].
  We call it Unmapped tent pitching scheme.
- Numerically one needs an explicit scheme (or accept a residual/iterate).
- It can be implemented as $\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \sum_{k \in \mathcal{K}_n^{\text{RBSWR}}} \widetilde{R}_k^\top A_k^{-1} R_k(\boldsymbol{f} - A\boldsymbol{u}^n)$.

XT-RAS and Pipeline strategies

One could solve all subdomains in space instead of alternating between the red and black ones:

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^{n} + \sum_{k \in \mathcal{K}_n^{\mathrm{space}}} \widetilde{R}_k^{\top} A_k^{-1} R_k (\boldsymbol{f} - A\boldsymbol{u}^{n}).$$

Using $T_{\mathrm{sub}} = \frac{|\Omega_j|}{c}$ and an explicit time-stepping scheme, one can advance every two iterations. More redundant computations are performed.

XT-RAS and Pipeline strategies

One could solve all subdomains in space instead of alternating between the red and black ones:

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \sum_{k \in \mathcal{K}_n^{\mathrm{space}}} \widetilde{R}_k^\top A_k^{-1} R_k(\boldsymbol{f} - A\boldsymbol{u}^n).$$

Using $T_{\mathrm{sub}} = \frac{|\Omega_j|}{c}$ and an explicit time-stepping scheme, one can advance every two iterations. More redundant computations are performed. However, since one already iterates,

- one may use implicit time-stepping schemes and smaller overlap,
- there is no need of red-black sequences

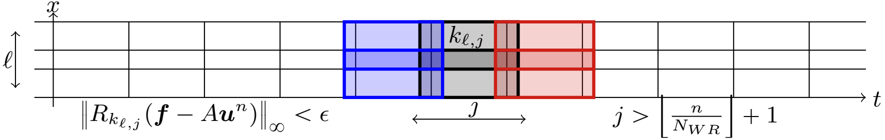
XT-RAS and Pipeline strategies

One could solve all subdomains in space instead of alternating between the red and black ones:

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \sum_{k \in \mathcal{K}_n^{\text{space}}} \widetilde{R}_k^\top A_k^{-1} R_k (\boldsymbol{f} - A\boldsymbol{u}^n).$$

Using $T_{\text{sub}} = \frac{|\Omega_j|}{c}$ and an explicit time-stepping scheme, one can advance every two iterations. More redundant computations are performed. However, since one already iterates,

- one may use implicit time-stepping schemes and smaller overlap,
- there is no need of red-black sequences
- one may adapt/control the advancing subdomain solution front and tail:



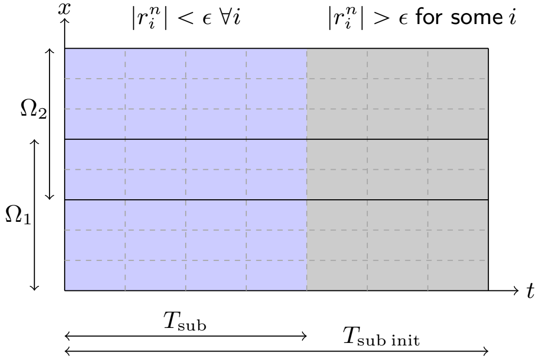
$$\mathcal{K}_n := \left\{ k_{\ell,j} \in \mathcal{I} \,:\, \ell \in \mathcal{K}_X,\, j \in \mathbb{N}_+ \text{ with } j \leqslant \min\left(N_T, \left\lfloor \frac{n}{N_{WR}} \right\rfloor + 1\right) \text{ and } \left\|R_{k_{\ell,j}}(\boldsymbol{f} - A\boldsymbol{u}^n)\right\|_\infty > \epsilon \right\},$$

where $\mathcal{I} := \{1, \dots, N_T N_X\}$, $\mathcal{K}_X := \{1, \dots, N_X\}$, $N_{WR} \in \mathbb{N}_+$, and $\epsilon > 0$.

Adaptive choice of the subdomains time length

By monitoring the local residuals $\|R_k(\boldsymbol{f} - A\boldsymbol{u}^n)\|_\infty$, one can estimate T_{sub} at the first iteration:

- Guess an initial $T_{\text{sub init}}$.
- Perform N_{WR} iterations on the first strip in time of subdomains of length $T_{\text{sub init}}$.
- Compute T_{sub} as the largest time such that the residual is smaller than ϵ.

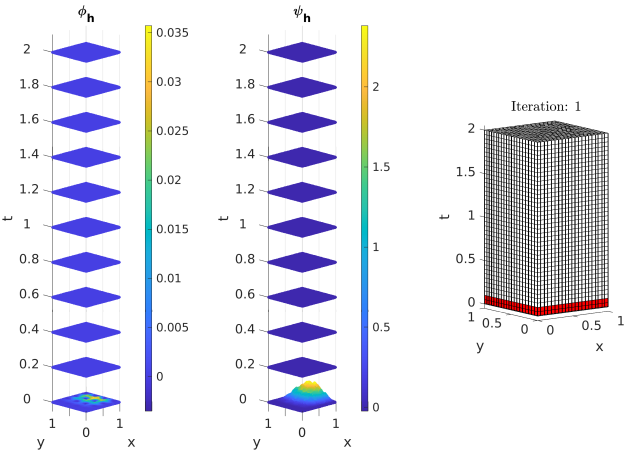


For $N_{WR} = 2$ and an explicit integrator, this leads to the tent heights in the (un)mapped strategy.

Numerical experiments

https://lymph.bitbucket.io/

Pipeline Algorithm: $T = 2$, $N_X = 16$, $N_T = 20$, $N_{WR} = 4$



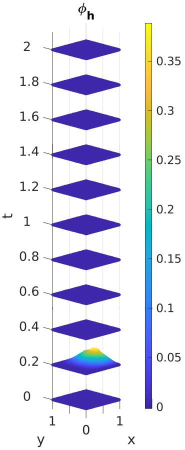
Active subdomains 16 / 320

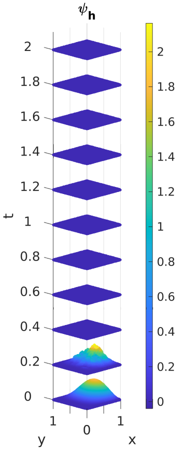
- **Analytical solution**

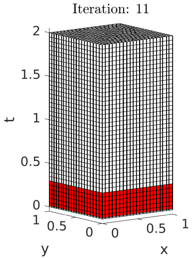
$$\begin{cases} \varphi(x,y,t) = x^2 \sin(\pi x)\sin(\pi y)\sin(\pi t\sqrt{2}), \\ \psi(x,y,t) = \partial_t \varphi(x,y,t), \end{cases}$$

- Dirichlet boundary conditions

Pipeline Algorithm: $T = 2$, $N_X = 16$, $N_T = 20$, $N_{WR} = 4$







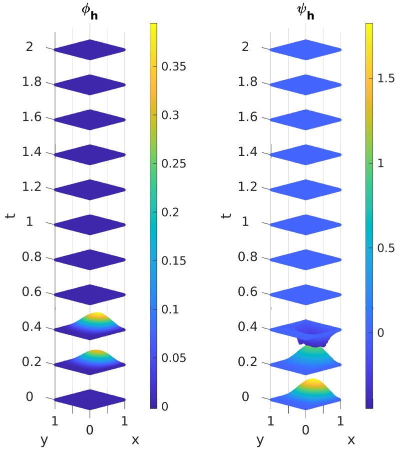
Active subdomains 48 / 320

- **Analytical solution**

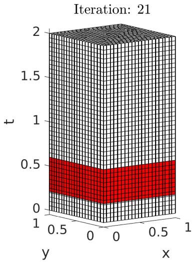
$$\begin{cases} \varphi(x,y,t) = x^2 \sin(\pi x)\sin(\pi y)\sin(\pi t\sqrt{2}), \\ \psi(x,y,t) = \partial_t \varphi(x,y,t), \end{cases}$$

- Dirichlet boundary conditions

Pipeline Algorithm: $T = 2$, $N_X = 16$, $N_T = 20$, $N_{WR} = 4$



Active subdomains 76 / 320

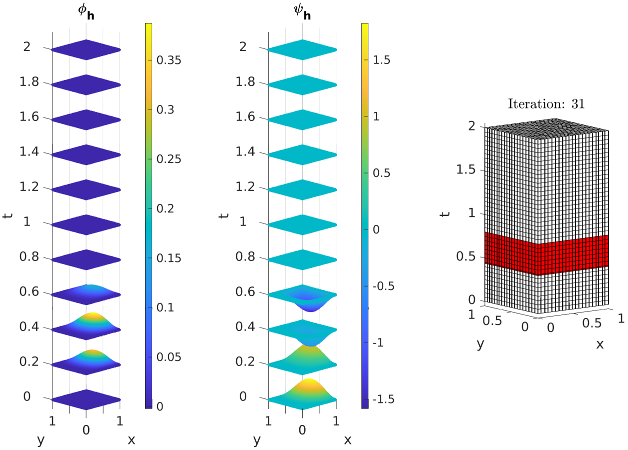


- **Analytical solution**

$$\begin{cases} \varphi(x,y,t) = x^2 \sin(\pi x)\sin(\pi y)\sin(\pi t\sqrt{2}), \\ \psi(x,y,t) = \partial_t \varphi(x,y,t), \end{cases}$$

- Dirichlet boundary conditions

Pipeline Algorithm: $T = 2$, $N_X = 16$, $N_T = 20$, $N_{WR} = 4$



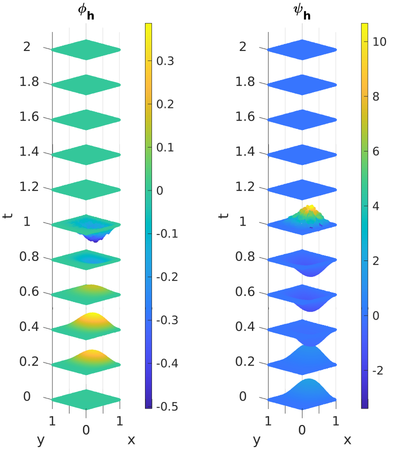
Active subdomains 75 / 320

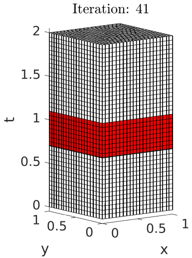
- **Analytical solution**

$$\begin{cases} \varphi(x,y,t) = x^2 \sin(\pi x)\sin(\pi y)\sin(\pi t\sqrt{2}), \\ \psi(x,y,t) = \partial_t \varphi(x,y,t), \end{cases}$$

- Dirichlet boundary conditions

Pipeline Algorithm: $T = 2$, $N_X = 16$, $N_T = 20$, $N_{WR} = 4$





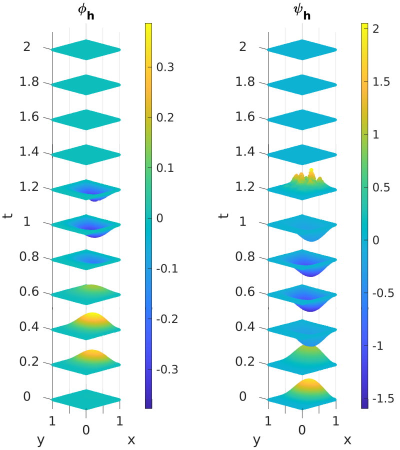
Active subdomains 68 / 320

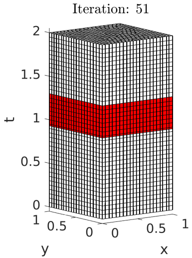
- **Analytical solution**

$$\begin{cases} \varphi(x,y,t) = x^2 \sin(\pi x)\sin(\pi y)\sin(\pi t\sqrt{2}), \\ \psi(x,y,t) = \partial_t \varphi(x,y,t), \end{cases}$$

- Dirichlet boundary conditions

Pipeline Algorithm: $T = 2$, $N_X = 16$, $N_T = 20$, $N_{WR} = 4$





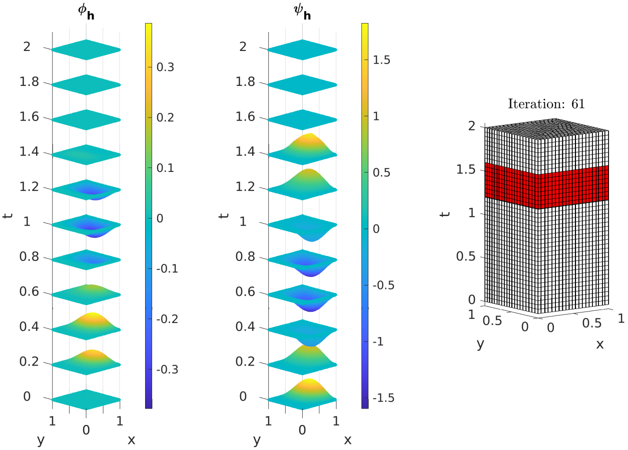
- **Analytical solution**

$$\begin{cases} \varphi(x,y,t) = x^2 \sin(\pi x)\sin(\pi y)\sin(\pi t\sqrt{2}), \\ \psi(x,y,t) = \partial_t \varphi(x,y,t), \end{cases}$$

- Dirichlet boundary conditions

Active subdomains 64 / 320

Pipeline Algorithm: $T = 2$, $N_X = 16$, $N_T = 20$, $N_{WR} = 4$



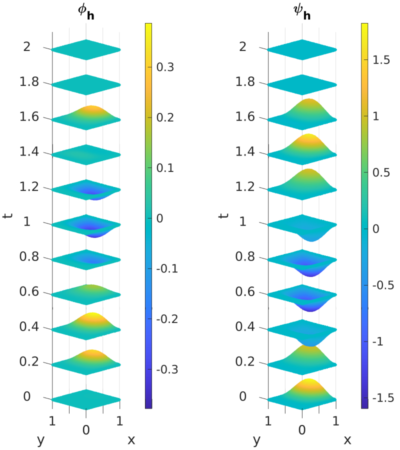
Active subdomains 79 / 320

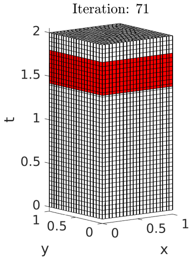
- **Analytical solution**

$$\begin{cases} \varphi(x,y,t) = x^2 \sin(\pi x)\sin(\pi y)\sin(\pi t\sqrt{2}), \\ \psi(x,y,t) = \partial_t \varphi(x,y,t), \end{cases}$$

- Dirichlet boundary conditions

Pipeline Algorithm: $T = 2$, $N_X = 16$, $N_T = 20$, $N_{WR} = 4$





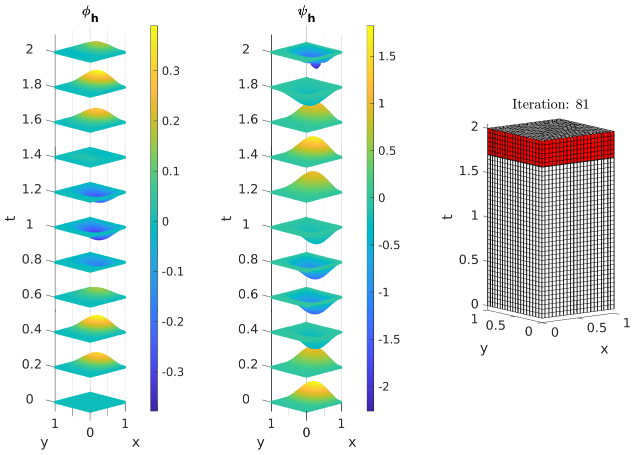
Active subdomains 74 / 320

- **Analytical solution**

$$\begin{cases} \varphi(x,y,t) = x^2 \sin(\pi x)\sin(\pi y)\sin(\pi t\sqrt{2}), \\ \psi(x,y,t) = \partial_t \varphi(x,y,t), \end{cases}$$

- Dirichlet boundary conditions

Pipeline Algorithm: $T = 2,\ N_X = 16,\ N_T = 20,\ N_{WR} = 4$



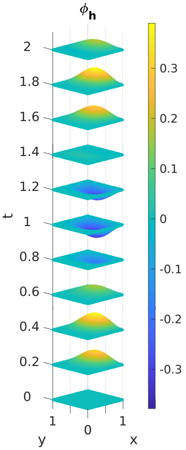
Active subdomains 60 / 320

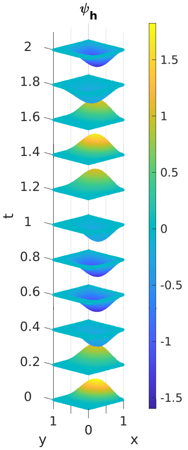
- **Analytical solution**

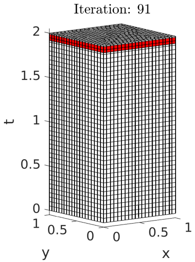
$$\begin{cases} \varphi(x,y,t) = x^2 \sin(\pi x)\sin(\pi y)\sin(\pi t\sqrt{2}), \\ \psi(x,y,t) = \partial_t \varphi(x,y,t), \end{cases}$$

- Dirichlet boundary conditions

Pipeline Algorithm: $T = 2$, $N_X = 16$, $N_T = 20$, $N_{WR} = 4$







Active subdomains 27 / 320

- **Analytical solution**

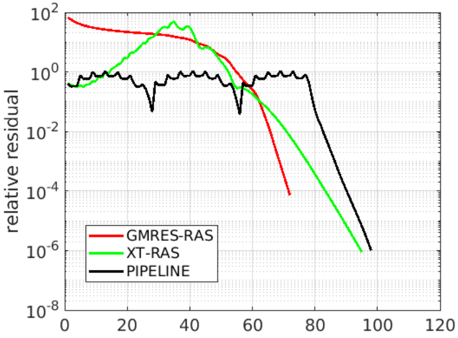
$$\begin{cases} \varphi(x,y,t) = x^2 \sin(\pi x)\sin(\pi y)\sin(\pi t\sqrt{2}), \\ \psi(x,y,t) = \partial_t \varphi(x,y,t), \end{cases}$$

- Dirichlet boundary conditions

Comparison: $T = 2$, $N_X = 16$, $N_T = 20$, $N_{WR} = 4$

Relative residual ϵ =1.e-6

	dofs	iterations	time [s]
XT-RAS	192.000	85	209
	576.000	100	1753
	1.280.000	95	8901
GMRES-RAS	192.000	75	205
	576.000	77	1392
	1.280.000	72	6956
Pipeline	192.000	92	39
	576.000	97	369
	1.280.000	98	1937

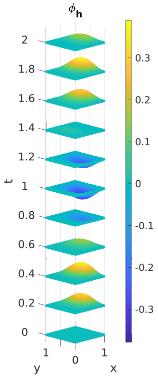


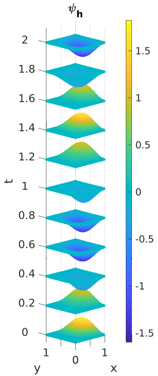
Remarks: No parallelization used!

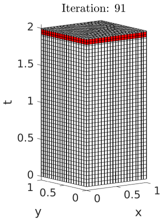
Comparison with GMRES in the single slab

Relative residual $\epsilon = $1.e-6

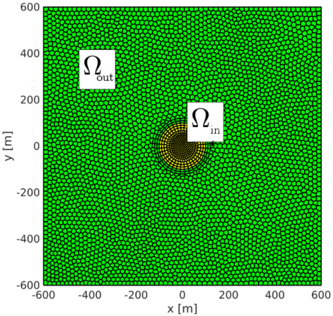
	dofs	time [s]
GMRES (slab)	192.000	**10**
	576.000	**136**
	1.280.000	**1536**
	2.560.000	18795
Pipeline	192.000	39
	576.000	369
	1.280.000	1937
	2.560.000	**4185**

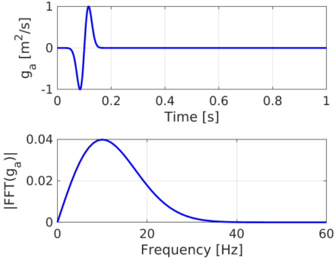






Acoustic wave impacting a cylinder [Artoni, Ciaramella, Mazzieri, 2024]





Discretization paramters

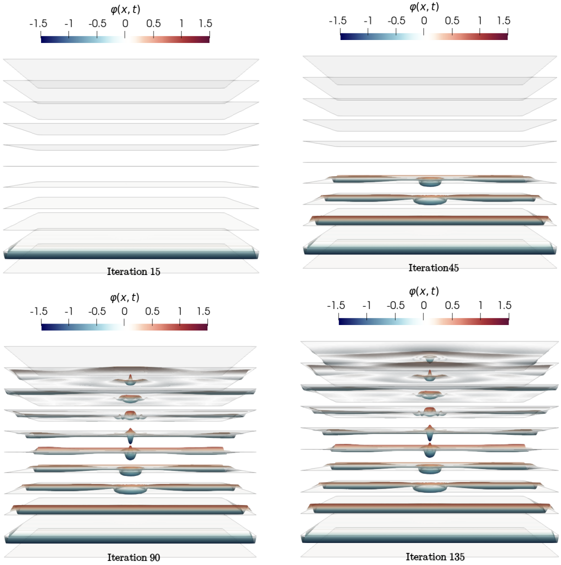
- $h_a \approx 36.5\ m$ in Ω_{out}
- $h_p \approx 11.8\ m$ in Ω_{in}
- $p = 3$,
- $\Delta t = 0.01\ s$, $r = 1$

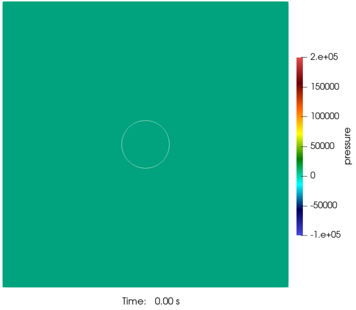
Space-time partition

- $N_X = 16$, $N_T = 25$

- Dirichlet boundary condition $\varphi(\boldsymbol{x}, t) = g_a(\boldsymbol{x}, t)$ on the bottom edge
- $\nabla\varphi(\boldsymbol{x}, t) \cdot \boldsymbol{n} = 0$ on the lateral and top edges
- $c = 150\ m/s$ in Ω_{out} and $c = 1500\ m/s$ in Ω_{in}

Acoustic wave impacting a cylinder [Artoni, Ciaramella, Mazzieri, 2024]





Conclusions & perspectives

Conclusions

- For Space-time dG discretization of wave problems on polytopal meshes we proposed:
    - a parallel XT-RAS framework;
    - relations and comparison with the tent-pitching algorithms;
    - choice/computation of subdomain time length.

Conclusions & perspectives

Conclusions

- For Space-time dG discretization of wave problems on polytopal meshes we proposed:
  - a parallel XT-RAS framework;
  - relations and comparison with the tent-pitching algorithms;
  - choice/computation of subdomain time length.

Ongoing work

- Unmapped Tent Pitching in higher dimension.
- Improve the strategy for non-constant coefficients.
- Relation with other pipeline approaches (Kwok, Ong (2019), Malas, Keyes (2016))
- Adaptivity

Thank you for the attention

Aknowledgements

- This work received funding by the European Union: **ERC SyG**, **NEMESIS**, project number 101115663 **https://erc-nemesis.eu/**
- This work received funding by the PRIN grant ASTICE - CUP: D53D23005710006
- The present research has been partially supported by MUR, grant Dipartimento di Eccellenza 2023-2027