MOX-Report No. 52/2026

On the Unmapped Tent Pitching for the Heterogeneous Wave Equation

Bonazzoli M.; Ciaramella G.; Mazzieri I.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it https://mox.polimi.it

On the Unmapped Tent Pitching for the Heterogeneous Wave Equation

Marcella Bonazzoli[0000−0002−0284−5643],
Gabriele Ciaramella[0000−0002−5877−4426],
Ilario Mazzieri[0000−0003−4121−8092]

1 Introduction

The Unmapped Tent Pitching algorithm (UTP) is a novel space-time domain decomposition technique for wave-propagation problems, first introduced in [2] for the parallel solution of the homogeneous one-dimensional wave equation and later extended in [1] to three-dimensional space domains. It shares the idea of the Mapped Tent Pitching algorithm (MTP) [3], which computes the solution by iteratively constructing polytopal space-time subdomains, called *tents*. Tents are pitched following the characteristic lines of the wave equation (CFL condition), so that the local problems can be solved exactly within each tent. More precisely, in MTP the solution is obtained by: (i) transforming the physical tents into Cartesian space-time cylinders, (ii) computing the local solutions in these transformed subdomains and (iii) mapping the solution back into the original tents.

The idea of UTP is to avoid the non-linear (and singular) mapping process by computing the solution directly in physical cartesian cylinders containing the tents, but at the cost of redundant computations [2]. The height of these space-time cylinders is determined by the CFL condition, as for the MTP algorithm, while the length is the size of the space subdomain. However, it can be observed that when the space–time subdomains of the UTP and the mapped tents of the MTP coincide, the computational costs of the corresponding subproblems are of the same order,

Marcella Bonazzoli
Inria, Unité de Mathématiques Appliqueés, ENSTA, Institut Polytechnique de Paris, 91120 Palaiseau, France, e-mail: marcella.bonazzoli@inria.fr

Gabriele Ciaramella
MOX Lab, Dipartimento di Matematica, Politecnico di Milano, Italy, e-mail: gabriele.ciaramella@polimi.it

Ilario Mazzieri
MOX Lab, Dipartimento di Matematica, Politecnico di Milano, Italy, e-mail: ilario.mazzieri@polimi.it

provided that the same space–time grid and time integrator are used. In the UTP, the chosen time integrator must satisfy a CFL condition, whereas in the MTP this restriction applies not only within the mapped tents but also through an additional limitation on the tent height imposed by the singularity of the mapping. Furthermore, this singularity also leads to a reduction in the accuracy of the resulting numerical solution; see, e.g., [4].

In this paper, we study the UTP when applied to the heterogeneous problem

$$\begin{cases} \partial_{tt} u(x,t) = c(x)^2 \partial_{xx} u(x,t), & \text{for } (x,t) \in \Omega \times (0,T), \\ u(x,0) = f(x), \text{ and } \partial_t u(x,0) = g(x), & \text{for } x \in \Omega, \\ u(0,t) = u(L,t) = 0, & \text{for } t \in (0,T]. \end{cases} \tag{1}$$

Here, $\Omega = (0, L)$, $T > 0$. We assume for simplicity that the wave speed is piecewise constant with $c(x) = c_1 > 0$ for $x \in (0, L/2]$ and $c(x) = c_2 > 0$ for $x \in (L/2, L)$, and f, g sufficiently regular functions.

While in the homogeneous case the choice of the space-time UTP subdomains is rather natural, in this work we aim at answering the question:

What is the optimal space-time decomposition in the case of a heterogeneous wave speed?

Here, the optimality is understood in terms of computational time, and will be precisely defined in the following sections.

The article is structured as follows. In section 2, we recall the UTP for a homogeneous medium, and briefly discuss its properties. Section 3 is devoted to the study of UTP for the heterogeneous problem (1). Conclusions are drawn in section 4.

2 UTP in the homogeneous case

Before extending the UTP to the heterogeneous case, let us recall the algorithm in the homogeneous setting, where $c(x) = c = c_1 = c_2$ and the slope of the characteristic lines is $\pm 1/c$ (see also [2, Algorithm 2 and Fig. 3]). Consider a set $\{x_j\}_{j=0}^{N-1} \subset \overline{\Omega}$ of N equispaced points $x_j = jL/(N-1)$, $j = 0, \ldots, N-1$, with N odd, and a decomposition of Ω into $N-2$ overlapping subintervals

$$I_j = (x_{j-1}, x_{j+1}), \ j = 1, \ldots, N-2. \tag{2}$$

By setting $\mathcal{R} = \{1, 3, 5, \ldots, N-2\}$ and $\mathcal{B} = \{2, 4, 6, \ldots, N-3\}$, we distinguish between the non-overlapping *red* subintervals $I_j, j \in \mathcal{R}$, and the non-overlapping *black* subintervals $I_j, j \in \mathcal{B}$. In the UTP space-time rectangular subdomains are pitched alternately on the red and black subintervals. Following the characteristic lines, we set $H = \frac{L}{c(N-1)}$. In the first iteration ($k = 1$), the red rectangles $\mathcal{T}_j^k$, $j \in \mathcal{R}$, of height H are pitched. In the next iterations ($k > 1$), black and red rectangles $\mathcal{T}_j^k$ of height $2H$ are pitched alternately, with $j \in \mathcal{B}$ when k is even and $j \in \mathcal{R}$ when k is

Algorithm 1 Unmapped Tent Pitching in the homogeneous case

Require: The subintervals $I_j \subset \Omega$ defined in (2) and an initial guess function u^0 in $\overline{\Omega \times (0,T)}$ such that $u^0 = f$ and $\partial_t u^0 = g$ on $\Omega \times \{0\}$.

Set $H = \frac{L}{c(N-1)}$, $\mathcal{R} = \{1, 3, 5, \ldots, N-2\}$, $\mathcal{B} = \{2, 4, 6, \ldots, N-3\}$.
Set $k = 1$ and $v_j^0 = 0$ for all $j = 1, \ldots, N$.
while $\exists\, j \in \{1, \ldots, N-2\}$ such that $v_j^{k-1} \neq T$ **do**
    Set $u^k = u^{k-1}$.
    Set $J_k = \mathcal{R}$ if k is odd, and $J_k = \mathcal{B}$ if k is even.
    if $k = 1$ **then**
        For all $j \in J_k$, set $v_j^k = \min(T, H)$ and pitch the rectangle $\mathcal{T}_j^k = I_j \times (0, v_j^k)$.
    else
        For all $j \in J_k$, set $v_j^k = \min(T, v_j^{k-2} + 2H)$ and pitch $\mathcal{T}_j^k = I_j \times (v_j^{k-2}, v_j^k)$.
    end if
    For $j \in J_k$, solve problems (3) in parallel to get u_j^k in $\mathcal{T}_j^k$.
    Update $u^k = u_j^k$ in $\mathcal{T}_j^k$ for all $j \in J_k$.
    Update $k = k + 1$.
end while

odd. Thus, in each iteration k, local problems are solved in parallel in the space-time rectangles

$$\mathcal{T}_j^k := I_j \times (v_j^{k-2}, v_j^k),$$

with $v_j^k = v_j^{k-2} + 2H$, $j \in \mathcal{R}$ for k odd, and $j \in \mathcal{B}$ for k even. Notice that for the first red iteration ($k = 1$) the length in time is different: $\mathcal{T}_j^1 := I_j \times (0, H)$. The same can happen for the last iteration. Thus, given an approximation u^{k-1} in $\Omega \times (0,T)$, the new approximation u^k is computed by first solving (in parallel) the subproblems

$$\begin{cases} \partial_{tt} u_j^k = c^2 \partial_{xx} u_j^k & \text{in } \mathcal{T}_j^k = I_j \times (v_j^{k-2}, v_j^k), \\ u_j^k = u^{k-1} & \text{on } \{x_{j-1}, x_{j+1}\} \times (v_j^{k-2}, v_j^k), \\ u_j^k = u^{k-1} & \text{on } I_j \times \{v_j^{k-2}\}, \\ \partial_t u_j^k = \partial_t u^{k-1} & \text{on } I_j \times \{v_j^{k-2}\}, \end{cases} \tag{3}$$

for $j \in \mathcal{R}$ for k odd and $j \in \mathcal{B}$ for k even, and then setting $u^k = u_j^k$ in $\mathcal{T}_j^k$ for the solved j subdomains and $u^k = u_j^{k-1}$ elsewhere. Clearly, we complement problem (3) with Dirichlet boundary conditions, i.e. $u_1^k = 0$ on $\{0\} \times (v_j^{k-2}, v_j^k)$, $u_{N-2}^k = 0$ on $\{L\} \times (v_j^{k-2}, v_j^k)$ for any k odd. Moreover, at the first two iterations ($k = 0$ and $k = 1$), one considers the initial conditions $u_j^1(x, 0) = f(x)$ and $\partial_t u_j^1(x, 0) = g(x)$. The overall UTP procedure is detailed in Algorithm 1.

An example of UTP iterations is shown in Fig. 1 (left column). The red and black subdomain solutions alternate. Notice that the red and black rectangles have height $2H = \frac{2L}{c(N-1)}$. As explained in [2], Algorithm 1 computes the exact solution below the tents (the hatched regions in Fig. 1), and wrong approximations in the areas above the tents, which correspond to the regions where redundant computations are performed. However, note that the area of the rectangles (space-time subdomains) in

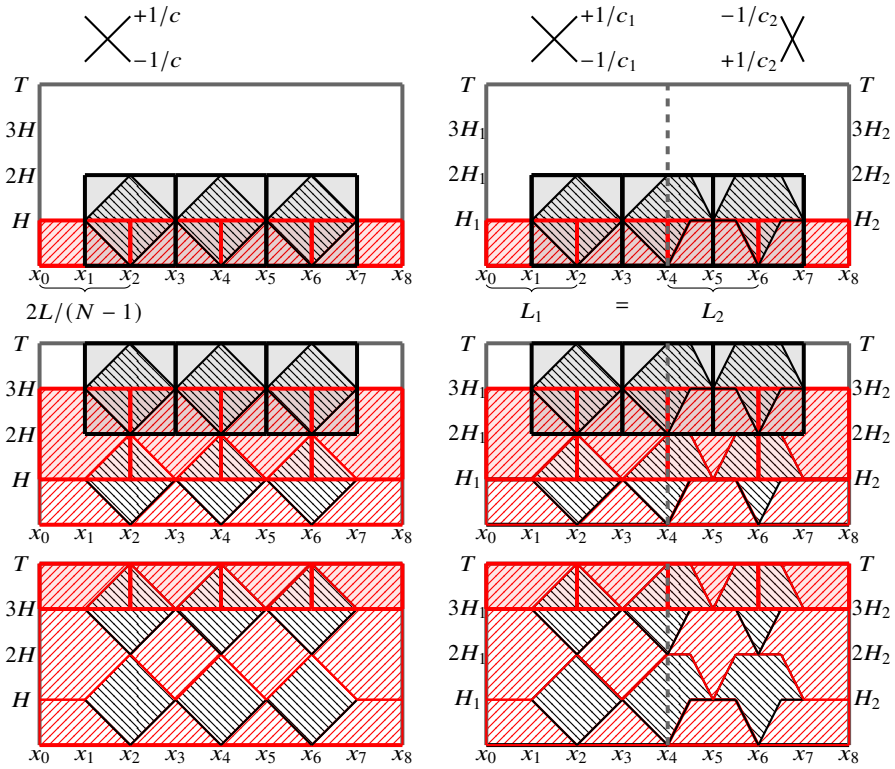


Fig. 1 Iterations of UTP with $N = 9$ in the homogeneous case (left), and in the heterogeneous case with the choice $H_2 = H_1$, $m_2 = m_1 = 2$ (right). Here $c_1 = 2c_2$. Red and black rectangles are the space-time subdomains constructed by UTP at odd and even iterations. We display iterations $k = 1$, $k = 2$ at the top, iterations $k = 3$, $k = 4$ in the middle, iteration $k = 5$ at the bottom of the figure. The hatched regions are the portions of the domain where UTP computes the exact solution.

the UTP scheme is the same as that of the MTP once the non-linear map is applied to the tent. For this reason, the local problem resolution has the same computation cost for both algorithms. The algorithm terminates when the exact solution is computed in the entire space-time domain $\Omega \times (0, T)$.

3 UTP in the heterogeneous case

We consider now the heterogeneous case (1) with $c(x) = c_1 > 0$ for $x \in (0, L/2]$ and $c(x) = c_2 > 0$ for $x \in (L/2, L)$. Without loss of generality, we assume that $c_1 > c_2$, so the absolute value of the slope of the characteristic lines is smaller in the left region of the domain, i.e., $|1/c_1| < |1/c_2|$. For simplicity, we assume $T = L/(2c_1)$. In contrast to the homogeneous case, the space-time subdomains in the two regions

can have different lengths and heights. We denote by L_j and H_j, for $j = 1, 2$, the subdomain lengths and heights in the two regions. Without loss of generality, we fix L_1 and H_1 and study the UTP behavior for varying L_2 and H_2. In particular, we divide the left region $(0, L/2)$ into m_1 non-overlapping subintervals ($m_1 \in \mathbb{N}\backslash\{0\}$) of equal length $L_1 = L/(2m_1)$ and, following the characteristic lines, we set the height of the space-time rectangles (at the first iteration) to $H_1 = L_1/(2c_1)$. In this way, the problem we aim to address in this work is the following:

Find the optimal values for the length $L_2 = L/(2m_2)$, with $m_2 \in \mathbb{N}\backslash\{0\}$, and the height H_2 of the space-time rectangles for the right region $(L/2, L)$ such that the computational cost to solve the problem in the entire $\Omega \times (0, T)$ is minimal.

Now, we assume that the computational cost of each subdomain problem is proportional to its space-time volume. This is the case, e.g., when an explicit time-stepping scheme is used. Furthermore, noting that at each iteration $m_1 + m_2$ red subdomain problems or $m_1 + m_2 - 1$ black subdomain problems are solved, we assume that $m_1 + m_2$ parallel processes are available. Moreover, since at each iteration the subdomain problems are solved in parallel, we assume that the cost of iteration is just biggest volume among those subdomains.

To find the optimal subdomains configuration, having fixed L_1 and H_1, it is necessary to identify all possible choices for m_2 (hence L_2) and H_2. These are summarized in Fig. 2. Here, we show the space-time subdomains at the first iteration: the subdomains in the region with $c(x) = c_1$ in the first column of Fig. 2, and those in the region with $c(x) = c_2$ in the second, third and fourth columns of Fig. 2. Moreover, the different rows correspond to different sizes of the subdomain length L_2 given by m_2, while the different columns (second to third) correspond to different subdomain heights H_2. For each case, the hatched region is the portion in which the exact solution can be computed following the characteristic lines if only exact initial data are available. The integer m_2^* is the one corresponding to $L_2^* = L/(2m_2^*)$ such that $L_2^*/(2c_2) = L_1/(2c_1)$, that is, the peaks of the 'classical' triangular tents are reached at the same height in the two regions. We have $m_2^*/m_1 = c_1/c_2$, so $m_2^* > m_1$ for $c_1 > c_2$. Moreover, the different choices are classified into three configurations: the peak of the triangular tent (a) is not reached within the rectangular subdomain, (b) is reached exactly at the top of the rectangular subdomain, and (c) is reached strictly below the top of the subdomain. Now, the goal is to compute the computational cost in all these cases, in order to identify the optimal configuration.

$H_2 = H_1$ and $m_2 = m_1$. In this case, the space-time rectangles have the same size in the left and right regions, with the height determined by c_1, cf. Fig. 2, second row and second column. As an example, with $c_1 = 2c_2$ and $m_2 = m_1 = 2$, we show in Fig. 1 (right column) the 5 iterations needed for the UTP to compute the exact solution in the entire space-time domain $\Omega \times (0, T)$, with $T = L/(2c_1)$. More precisely, we display iterations $k = 1, k = 2$ at the top, iterations $k = 3, k = 4$ in the middle, iteration $k = 5$ at the bottom of the figure. With this first choice, UTP essentially proceeds as in the homogeneous case. The only difference is that in the heterogeneous case the portion where the exact solution is computed is not the same in the two regions of the domain, see the hatched portions in Fig. 1. Notice that

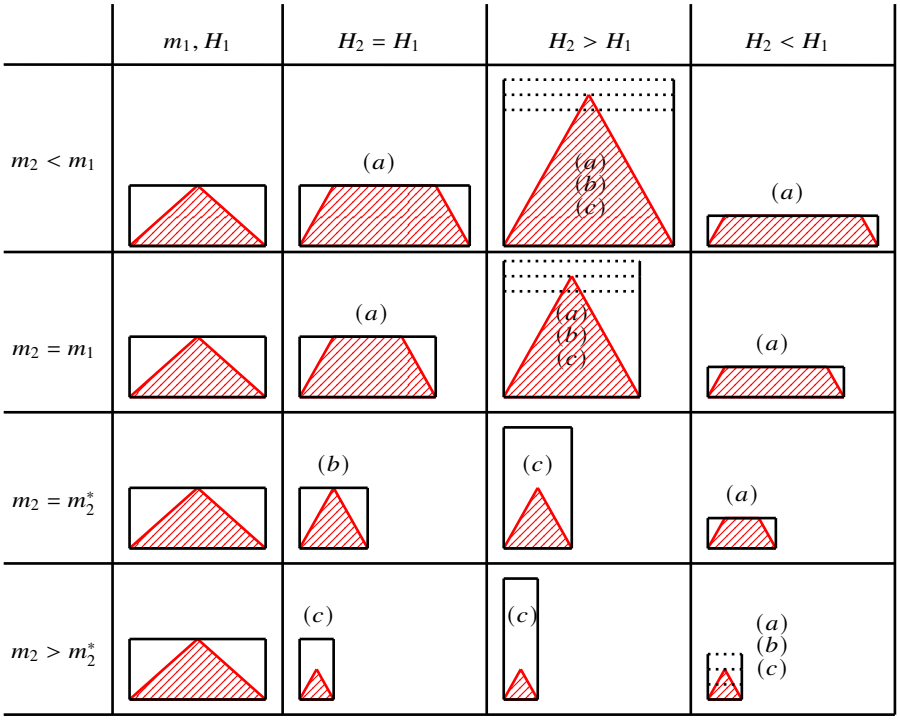


Fig. 2 The possible choices for the length $L_2 = L/(2m_2)$, with $m_2 \in \mathbb{N}\backslash\{0\}$, and the height H_2 of the space-time rectangles in the right region $(L/2, L)$, with respect to the reference rectangle of length $L_1 = L/(2m_1)$ and height $H_1 = L_1/(2c_1)$ in the left region $(0, L/2]$. We define m_2^* such that $m_2^*/m_1 = c_1/c_2$, so $m_2^* > m_1$ for the present case $c_1 > c_2$. The hatched region is the portion where the exact solution can be computed following the characteristic lines.

Algorithm 1 can be applied straightforwardly (with $N = 2(2m_1) + 1$) by using $c = c_1$ to pitch the rectangular subdomains and the true $c = c(x)$ for the local subdomain solves.

Let us now calculate the computational cost. At each iteration k, we use $m_1 + m_2 = 2m_1$ processes to solve in parallel the local problems on the rectangles: for k odd, we have $2m_1$ red rectangles of the same size (the processes are perfectly balanced); for k even, we have $2m_1 - 1$ black rectangles of the same size (one process is not used). The computational cost at $k = 1$ is proportional to the area $A = L_1 H_1 = L_1^2/(2c_1) = L^2/(8m_1^2 c_1)$. At the next iterations $k > 1$ the cost is $2A$, except the last one for which it is A again. To easily compute the total cost, we observe that the total height $T = L/(2c_1)$ can be covered with $T/H_1 = L/L_1 = 2m_1$ non overlapping red rectangles of area A (by considering the large ones as the union of two small ones), and with the same number $2m_1$ of non overlapping black rectangles of area A. Therefore, the total computational cost is

$$2(2m_1A) = \frac{L^2}{2m_1c_1}.$$

This is also shown in Fig. 6 (b), where we report the pipeline of the computational work (time to solution) employed by the $m_1 + m_2$ parallel processes. It is easy to see that the computational workload is balanced and well-distributed among all the processes, with only one inactive process during the black iteration (the lighter-colored region at the bottom line).

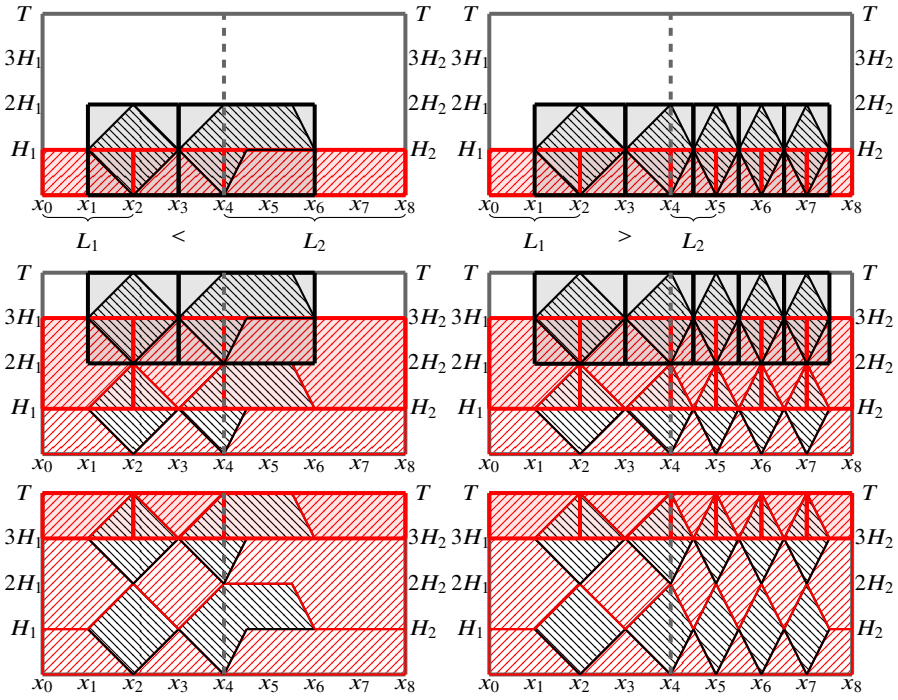


Fig. 3 Iterations of UTP with the choice $H_2 = H_1$, $1 = m_2 < m_1 = 2$ (left), and the choice $H_2 = H_1$, $4 = m_2 = m_2^* (> m_1 = 2)$ (right).

$H_2 = H_1$ and $m_2 < m_1$. We now consider the choice $H_2 = H_1, m_2 < m_1$, cf. Fig. 2 first row and second column, in which the rectangles are larger in the right region, see an example with $m_1 = 2, m_2 = 1$ in Fig. 3 (left). By using again $m_1 + m_2$ processes, the local problems in the right part of the domain are more expensive to solve. Hence, $A = L_2H_1 = L_2L_1/(2c_1) = L^2/(8m_1m_2c_1)$, and the total computational cost is

$$2(2m_1A) = \frac{L^2}{2m_2c_1} > \frac{L^2}{2m_1c_1}.$$

Thus, this strategy is worse than the previous one ($H_2 = H_1$ and $m_2 = m_1$), as confirmed by comparing the corresponding pipelines in Fig. 6 (a).

$H_2 = H_1$ and $m_2 = m_2^* > m_1$. An example for this choice, with $m_1 = 2, m_2 = m_2^* = 4$, is displayed in Fig. 3 (right), cf. also Fig. 2 third row and second column. The rectangular subdomains are larger in the left region and have area $A = L_1 H_1$. Thus, the total computational cost is the same as the first strategy ($H_2 = H_1$ and $m_2 = m_1$), but at the price of using more processes since $m_1 + m_2 > 2m_1$. Moreover, no computational time improvement is achieved by increasing the number of processes, because m_2 processes are partially inactive during the red or black iterations, cf. the pipeline in Fig. 7 (a).

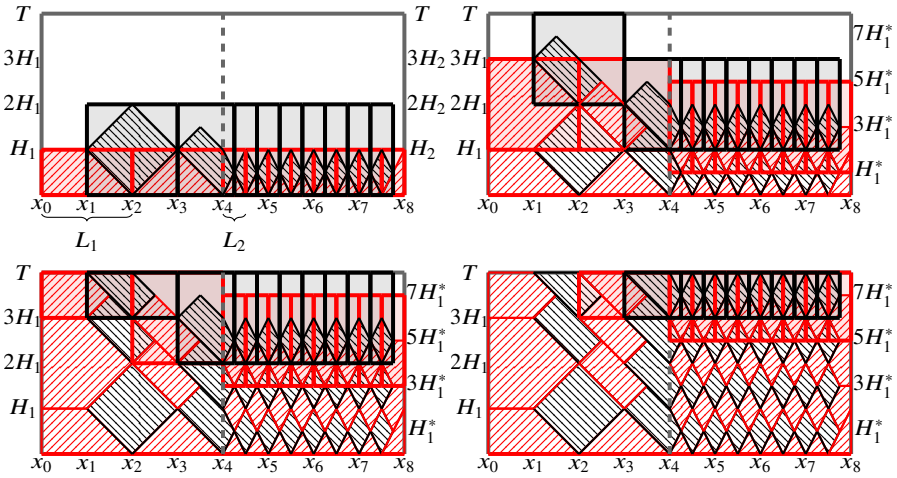


Fig. 4 First eight iterations of UTP with the choice $H_2 = H_1$, $8 = m_2 > m_2^* = 4\ (> m_1 = 2)$. We display iterations $k = 1$, $k = 2$ at the top left, iterations $k = 3$, $k = 4$ at the top right, iterations $k = 5$, $k = 6$ at the bottom left, and iterations $k = 7$, $k = 8$ at the bottom right of the figure.

$H_2 = H_1$ and $m_2 > m_2^* > m_1$. This case corresponds to Fig. 2 fourth row and second column, and it requires more iterations to compute the exact solution. For instance, in Fig. 4, we show the first eight iterations with $m_1 = 2, m_2 = 8$, but nine iterations are actually needed, to be compared to five iterations in all the previous examples. Indeed, in the right region, since the portion where the exact solution is computed does not reach the top of the rectangular subdomains, the red rectangles at different iterations need to overlap (and similarly for the black ones). This leads to an exact solution advancing more slowly in the right region with multiples of an actual height $H_1^* < H_1$, which slows down the overall computation, as shown in the corresponding pipeline in Fig. 7 (b). Thus, the strategy $H_2 = H_1$ and $m_2 > m_2^* > m_1$ is outperformed by the previously discussed ones.

$H_2 > H_1$. This case corresponds to the third column in Fig. 2, which could be regarded as a natural choice, since the characteristic lines in the right region are

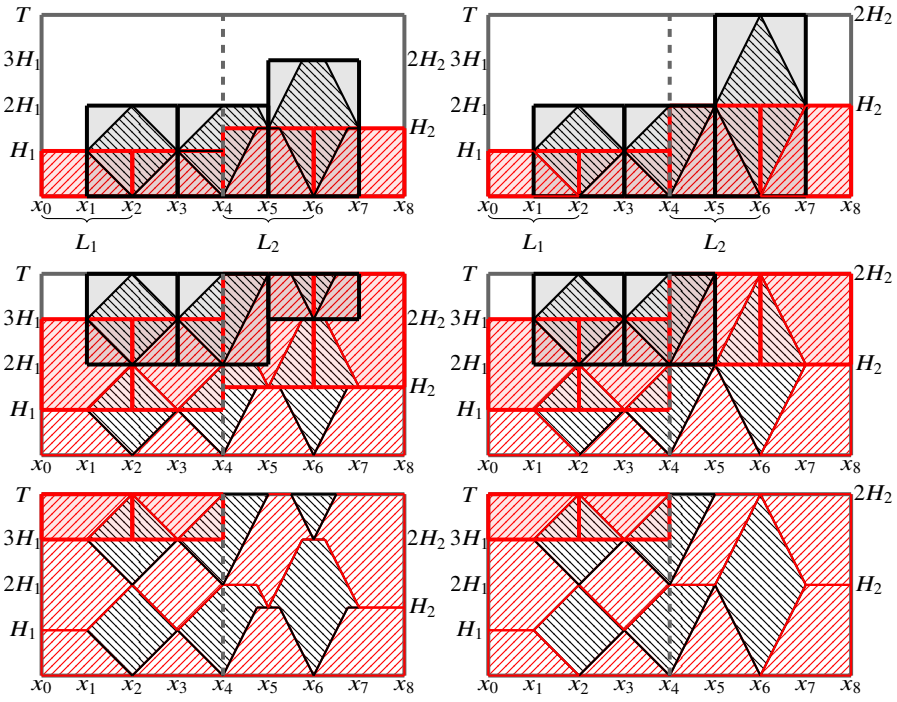


Fig. 5 Iterations of UTP with the choice $H_2 > H_1$, $m_2 = m_1 = 2$, case (a) (left), and the choice $H_2 > H_1$, $m_2 = m_1 = 2$, case (b) (right).

steeper. See two examples with $m_2 = m_1 = 2$ in Fig. 5. With $H_2 > H_1$, the m_2 processes assigned to subdomains in the right region of the domain reach the final time T sooner than the m_1 ones in the left area. Nevertheless, this does not reduce the computational cost but rather causes communication delay within the processes, similarly to Fig. 7 (b).

$H_2 < H_1$. This choice, corresponding to the fourth column in Fig. 2, produces two effects. First, it conflicts with the characteristic slopes in the two regions. Moreover, it creates a symmetric issue to the one described above, that is, the final time T is reached earlier in the left region than in the right one, resulting in a communication issue between processes.

In conclusion:

The optimal value for the length is $L_2 = L_1 = L/(2m_1)$, i.e., $m_2 = m_1$, and for the height is $H_2 = H_1 = L_1/(2c_1)$.

As discussed above, the resulting UTP algorithm for the heterogeneous wave equation (1) is essentially Algorithm 1 using $c = c_1$ (the maximum propagation speed) to build the rectangular subdomains, considering $N = 2(2m_1) + 1$, and solving, instead of (3), heterogeneous local problems:

$$\begin{cases} \partial_{tt} u_j^k(x,t) = c(x)^2 \partial_{xx} u_j^k(x,t) & \text{for } (x,t) \text{ in } \mathcal{T}_j^k = I_j \times (v_j^{k-2}, v_j^k), \\ u_j^k = u^{k-1} & \text{on } \{x_{j-1}, x_{j+1}\} \times (v_j^{k-2}, v_j^k), \\ u_j^k = u^{k-1} & \text{on } I_j \times \{v_j^{k-2}\}, \\ \partial_t u_j^k = \partial_t u^{k-1} & \text{on } I_j \times \{v_j^{k-2}\}, \end{cases} \tag{4}$$

with the Dirichlet boundary conditions at $x = 0$ for u_1^k, $x = L$ for u_{N-2}^k, for every k odd, and the initial conditions $u_j^1(x,0) = f(x)$, $\partial_t u_j^1(x,0) = g(x)$ for $x \in I_j$ at the first iteration.

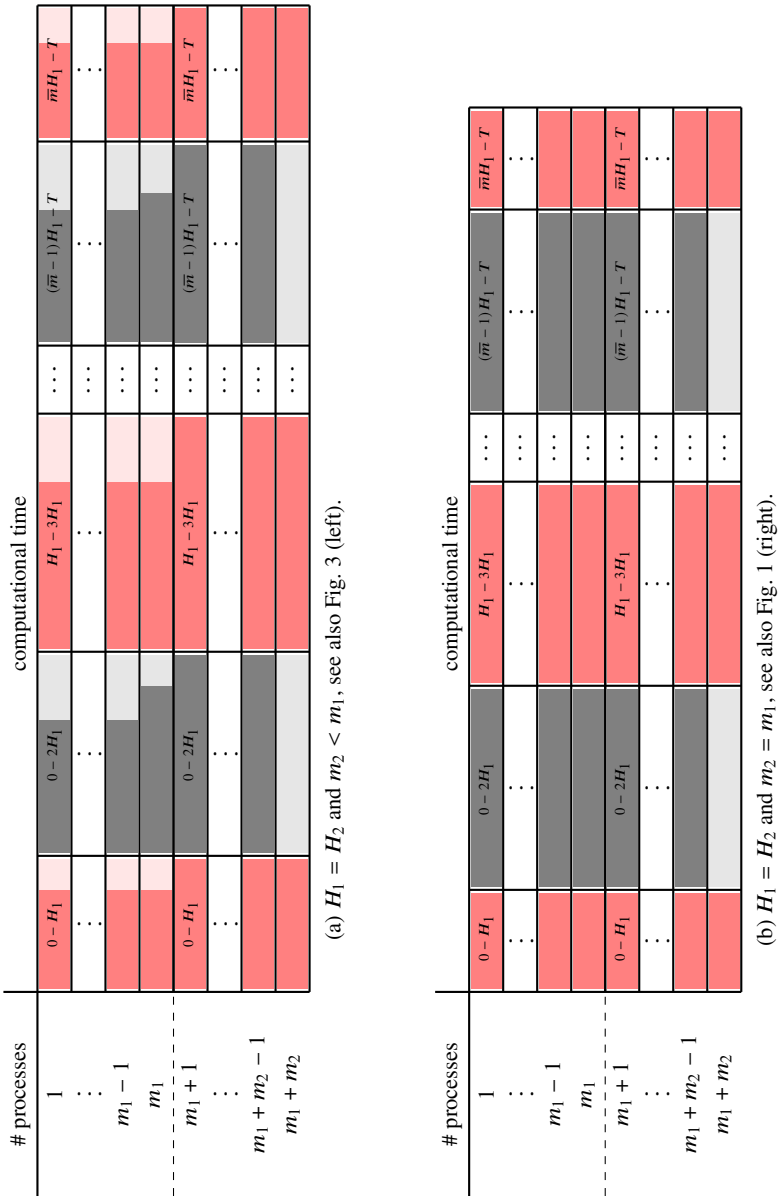
4 Conclusion

In this work we studied the UTP for the wave equation (in one-dimensional space domains) with a piecewise constant propagation speed. In particular, the study focuses on analyzing the computational cost of different strategies for decomposing the space-time domain in order to identify the optimal one. We prove that the most efficient approach is the one employing space–time subdomains with identical spatial and temporal dimensions in the different material regions, determined by the maximum propagation speed. Future work will address the extension of these results to wave equations in two- and three-dimensional space domains.

Acknowledgements The work of G. Ciaramella and I. Mazzieri has been partially supported by the PRIN2022 grant ASTICE - CUP: D53D23005710006. G. Ciaramella and I. Mazzieri are members of INdAM-GNCS group. The present research is part of the activities of "Dipartimento di Eccellenza 2023-2027".

References

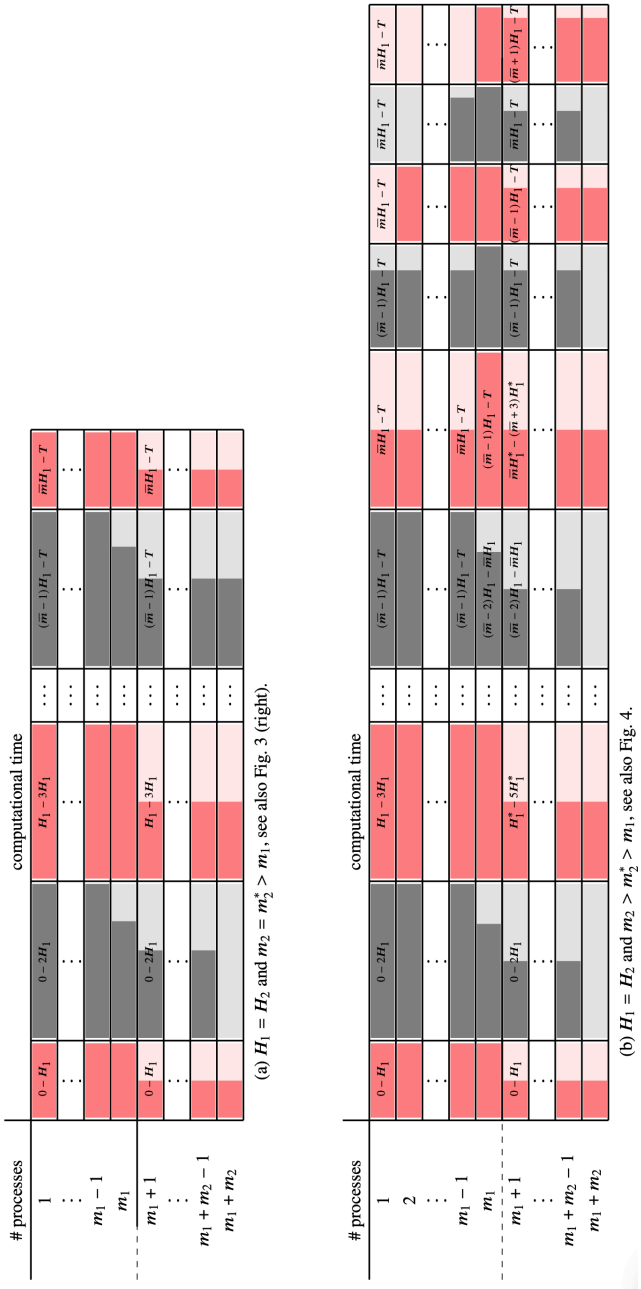
1. Artoni, A., Ciaramella, G., Gander, M.J., Mazzieri, I.: On the unmapped tent-pitching in 3D. In: Domain Decomposition Methods in Science and Engineering XXVIII, Lecture Notes in Computational Science and Engineering. Springer-Verlag (2025). In print
2. Ciaramella, G., Gander, M.J., Mazzieri, I.: Unmapped tent pitching schemes by waveform relaxation. In: Domain Decomposition Methods in Science and Engineering XXVII, Lecture Notes in Computational Science and Engineering, pp. 455–462. Springer-Verlag (2024)
3. Gopalakrishnan, J., Schöberl, J., Wintersteiger, C.: Mapped tent pitching schemes for hyperbolic systems. SIAM J. Sci. Comput. **39**(6), B1043–B1063 (2017)
4. Gopalakrishnan, J., Schöberl, J., Wintersteiger, C.: Structure aware Runge–Kutta time stepping for spacetime tents. SN Partial Differ. Equ. Appl. **1**, 19 (2020)



(a) $H_1 = H_2$ and $m_2 < m_1$, see also Fig. 3 (left).

(b) $H_1 = H_2$ and $m_2 = m_1$, see also Fig. 1 (right).

Fig. 6 Computational cost (time to solution) for different red-black domain decomposition strategies with $H_2 = H_1$, cf. Fig. 2 second column. The lighter-colored areas represent moments of inactivity of the corresponding process and $\overline{m} = 2m_1 - 1$.



(a) $H_1 = H_2$ and $m_2 = m_2^* > m_1$, see also Fig. 3 (right).

(b) $H_1 = H_2$ and $m_2 > m_2^* > m_1$, see also Fig. 4.

Fig. 7 Computational cost (time to solution) for different red-black domain decomposition strategies with $H_2 = H_1$, cf. Fig. 2 second column. The lighter-colored areas represent moments of inactivity of the corresponding process and $\overline{m} = 2m_1 - 1$.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

51/2026 Bellezza P.; Ciaramella G.; Macchini C.; Mazzieri I.; Verani M.
ParaFlow: Parareal Acceleration of Gradient-Flow Minimization

50/2026 Donnarumma, A.; Guagliardi, O.; Di Stazio F.; Mazza E.; Tanelli M.; Paganoni A.M.
Modelling Well-Being and Psychological Risk in Doctoral Education: An Integrated Latent Trait Approach

49/2026 Bortolotti, T.; Troilo, R.; Casu, F.; Vantini, S.; Menafoglio, A.
Regularized covariance estimation from partially observed interferometric data

48/2026 Antonietti, P.F.; Corti, M.; Orlando, G.
Optimized high-order IMEX-RK schemes for degenerate diffusion-reaction problems with application to travelling waves phenomena

47/2026 Torri, V.; Barbieri, E.; Cantarutti, A.; Giaquinto, C.; Ieva, F.
Automatic identification of diagnosis from hospital discharge letters via weakly supervised Natural Language Processing

46/2026 Cancrini, A.; Ciaramella, G.; Antonietti, P.F.
A Scalable Deflated Conjugate Gradient Solver for the Time-Dependent Pseudo-Stress Stokes Problem

44/2026 Bonetti, S; Botti, M.; Antonietti, P.F.
Splitting strategies for the fully-coupled nonlinear thermo-hydro-mechanical problem

45/2026 Antonietti, P.F.; Botti, M.; Parolini, N.; Pederzoli, V.; Verani, M.
Polytopal Discontinuous Galerkin Discretizations of Coupled Non-Newtonian Stokes-Darcy Systems

43/2026 Micheletti, S.
A validated MATLAB framework for sparse vectorized finite element assembly

42/2026 Fumagalli, I.; Campioni, M.; Sirtori, A.; Pagani, S.; Levi, R.; Politi, L. S.; Capo, G.; Antonietti, P. F.
Patient-specific computational mechanics of functional lumbar spine units