Multi-Level Explicit Local Time-Stepping Methods for Second-Order Wave Equations

Julien Diaz, Marcus J. Grote

Institute of Mathematics
University of Basel
Rheinsprung 21
CH - 4051 Basel
Switzerland

Preprint No. 2014-10
August, 2014

www.math.unibas.ch

Multi-Level Explicit Local Time-Stepping Methods for Second-Order Wave Equations

Julien Diaz[1], Marcus J. Grote[2]

Abstract

Local mesh refinement severly impedes the efficiency of explicit time-stepping methods for numerical wave propagation. Local time-stepping (LTS) methods overcome the bottleneck due to a few small elements by allowing smaller time-steps precisely where those elements are located. Yet when the region of local mesh refinement itself contains a sub-region of even smaller elements, any local time-step again will be overly restricted. To remedy the repeated bottleneck caused by hierarchical mesh refinement, multi-level local time-stepping methods are proposed, which permit the use of the appropriate time-step at every level of mesh refinement. Based on the LTS methods from [1], these multi-level LTS methods are explicit, yield arbitrarily high accuracy and conserve the energy. Numerical experiments illustrate the theoretical properties and the usefulness of these methods.

Keywords: wave propagation, finite element methods, adaptivity, explicit time integration, leap-frog method, local time-stepping, multirate methods

1. Introduction

Second-order wave equations are ubiquitous across a wide range of applications from acoustics, electromagnetics, and elasticity. Their spatial discretization by standard finite difference or finite element methods tyically leads to a large system of second-order ordinary differential equations. When explicit time integration is subsequently used, the time-step will be governed by the smallest elements in the mesh for numerical stability. Near corners, material interfaces or

Email addresses: julien.diaz@inria.fr (Julien Diaz), marcus.grote@unibas.ch (Marcus J. Grote)

[1]INRIA Research Center Bordeaux-Sud Ouest, Team-project Magique 3D and Laboratoire de Mathématiques Appliquées, UMR CNRS 5142. Université de Pau et des Pays de l'Adour - Bât. IPRA, BP 1155, 64013 PAU Cedex, France

[2]Institute of Mathematics, University of Basel, Rheinsprung 21, 4051 Basel, Switzerland

other small-scale geometric features, adaptive mesh refinement is certainly key for the accurate simulation of wave phenomena [2]. Local mesh refinement, however, severely impedes the efficiency of explicit time-marching methods because of the overly small time-step dictated by but a few tiny elements. When mesh refinement is restricted to a small portion of the computational domain, the use of implicit methods or a small time-step everywhere, is rather high a price to pay.

Local time-stepping (LTS) methods overcome the bottleneck due to local refinement by dividing the mesh into two distinct regions: the "coarse" region, which contains the larger elements and is integrated in time using an explicit method, and the "fine" region, which contains the smaller elements and is integrated in time using either smaller time-steps or an implicit scheme.

Locally implicit methods build on the long tradition of hybrid implicit-explicit (IMEX) algorithms for operator splitting in computational fluid dynamics – see [3, 4] and the references therein. Here, a linear system needs to be solved inside the refined region at every time-step, which becomes not only increasingly expensive with decreasing mesh size, but also increasingly ill-conditioned as the grid-induced stiffness increases [5]. Moreover, even when each individual method has order two, the implicit-explicit component splitting can reduce by one the overall space-time convergence rate of the resulting scheme [6, 7]. Recently, Descombes, Lanteri and Moya [7] remedied that unexpected loss in accuracy and hence recovered second-order convergence, by using the LF/CN-IMEX approach of Verwer [8] instead, yet at the price of a significantly larger albeit sparse linear system.

In contrast, locally explicit time-stepping methods remain fully explicit by taking smaller time-steps in the "fine" region, that is precisely where the smaller elements are located. Also known as multirate or multiple time-stepping methods in the ODE literature [9], various LTS methods have been proposed for numerical wave propagation based on classical Adams-Bashforth multistep methods [10]; they can also be interpreted as particular approximations of exponential-Adams multi-step methods [11]. Recently, Runge-Kutta based explicit LTS of arbitrarily high-order were proposed for wave propagation in [12].

In the absence of forcing and dissipation, the classical wave equation conserves the total energy. When a symmetric spatial FD or FE discretization is combined with a centered time-marching scheme, such as the standard leap-frog (LF) (also known as Newmark or Störmer-Verlet) method, the resulting fully discrete formulation will also conserve (a discrete version of)

the energy. Highly efficient in practice, centered time discretizations also display remarkably high accuracy over long times and remain even nowadays probably the most popular methods for the time integration of wave equations. In [13] Collino, Fouquet and Joly proposed an LTS method for the wave equation in first-order form, which conserves a discrete energy yet requires every time-step the solution of a linear system on the interface between the coarse and the fine mesh. It was analyzed in [14, 15] and later extended to elastodynamics [16] and Maxwell's equations [17]. By combining a symplectic integrator with a DG discretization of Maxwell's equations in first-order form, Piperno [18] proposed a second-order explicit local time-stepping scheme, which also conserves a discrete energy. Starting from the standard LF method, the authors proposed energy conserving fully explicit LTS integrators of arbitrarily high accuracy for the wave equation [1]; that approach was extended to Maxwell's equations in [19]. An *hp*-version, where not only the time-step but also the order of approximation is adapted within different regions of the mesh, was proposed in [20] and later applied to a realistic geological model [21].

When a region of local refinement itself contains sub-regions of further refinement, those "very fine" elements yet again will dictate the time-step, albeit local, to the entire "fine" region. Then, it becomes more efficient to let the time-marching strategy mimic the multilevel hierarchy of the mesh organized into tiers of "coarse", "fine", "very fine", etc. elements by introducing a corresponding hierarchy into the time-stepping method. Hence, the resulting multi-level local time-stepping (MLTS) method will advance in time by using within each tier of equally sized elements the corresponding optimal time-step.

The outline of our paper is as follows. Starting from a semi-discrete Galerkin finite element formulation of the wave equation, we derive in Section 2 local time-stepping (LTS) methods of arbitrarily high order based on the leap-frog (LF) method; we also recall some of their key properties from [1]. Although first presented in [1], the present derivation is different and crucial for the derivation of the multi-level local time-stepping (MLTS) methods in Section 3. In Section 4, we prove that the second-order MLTS method conserves a discrete energy regardless of the number of intermediate levels. Finally, in Section 5, we present numerical experiments in one and two space dimensions which illustrate the stability and convergence properties of these MLTS schemes.

2. Local time-stepping

We consider the acoustic wave equation

$$\frac{1}{\mu}u_{tt} - \nabla\cdot\left(\frac{1}{\rho}\nabla u\right) = 0 \qquad \text{in } (0,T)\times\Omega, \tag{1}$$

$$u = 0 \qquad \text{on } (0,T)\times\partial\Omega, \tag{2}$$

$$u|_{t=0} = u_0, \quad u_t|_{t=0} = v_0 \qquad \text{in } \Omega, \tag{3}$$

a standard model for second-order hyperbolic problems. Here Ω is a bounded domain in $\mathbb{R}^d$, whereas $u_0 \in H_0^1(\Omega)$ and $v_0 \in L^2(\Omega)$ are prescribed initial conditions. For simplicity, we impose homogeneous Dirichlet conditions at the boundary, $\partial\Omega$, and assume that Ω is source-free. The density, ρ, and the bulk modulus, μ, are piecewise smooth, strictly positive and bounded, and hence so is the wave speed, $c = \sqrt{\mu/\rho}$. Because sources are absent, the (continuous) *energy*,

$$E[u](t) = \frac{1}{2}\left\{\left\|\frac{1}{\sqrt{\mu}}u_t(t,.)\right\|^2 + \left\|\frac{1}{\sqrt{\rho}}\nabla u(t,.)\right\|^2\right\}, \qquad t \geq 0, \tag{4}$$

is conserved for all time.

Various finite element methods (FEM) are available for the spatial discretization of (1)–(3). For instance, the standard H^1-conforming FEM with mass-lumping starts from the weak formulation: Find $u : [0,T] \to H_0^1(\Omega)$ such that

$$\begin{aligned}\left(\frac{1}{\mu}u_{tt}, v\right) + \left(\frac{1}{\rho}\nabla u, c\nabla v\right) &= 0 \qquad \forall\, v \in H_0^1(\Omega),\ t \in (0,T),\\ u|_{t=0} &= u_0 \qquad \text{in } \Omega,\\ u_t|_{t=0} &= v_0 \qquad \text{in } \Omega,\end{aligned} \tag{5}$$

where $(\cdot,\cdot)$ denotes the standard inner product on $L^2(\Omega)$.

Next, we consider a family of shape-regular meshes $\{\mathcal{T}_h\}_h$ that each partition Ω into disjoint elements K, i.e. $\overline{\Omega} = \cup_{K\in\mathcal{T}_h}\overline{K}$; for simplicity, we assume that Ω is polygonal. The diameter of element K, a triangle or a quadrilateral in two space dimensions, and a tetrahedron or hexahedron in three dimensions, is denoted by h_K; hence, the mesh size, h, is given by $h = \max_{K\in\mathcal{T}_h} h_K$. Let $V_h \subset H_0^1(\Omega)$ denote the finite dimensional subspace

$$V_h = \{v \in H_0^1(\Omega) \,:\, v|_K \in \mathcal{S}^\ell(K),\ \forall K \in \mathcal{T}_h\}, \qquad \ell \geq 1,$$

where $\mathcal{S}^\ell(K)$ corresponds to the space $\mathcal{P}^\ell(K)$ of polynomials of total degree at most ℓ, if K is a triangle or tetrahedron, or to the space $\mathcal{Q}^\ell(K)$ of polynomials of maximal degree ℓ in each variable, if K is a quadrilateral or hexahedron.

The semi-discrete Galerkin approximation, $u^h(t) \in V_h$, is then defined for $0 < t < T$ by the restriction of (5) to V_h. Let $y(t) \in \mathbb{R}^N$ denote the coefficients of $u^h(t)$ with respect to the standard Lagrangian basis $\{\phi_i\}_{i=1,\dots,N}$ of V_h. Then, $y(t)$ satisfies

$$\begin{aligned} &M\,\frac{d^2y}{dt^2}(t) + K\,y(t) = 0\,, \qquad t \in (0,T), \\ &M\,y(0) = u_0^h, \quad M\,\frac{dy}{dt}(0) = v_0^h, \end{aligned} \tag{6}$$

where u_0^h, v_0^h are suitable approximations to the initial conditions. Moreover, the mass matrix, M, and the stiffness matrix, K, are given by

$$M_{ij} = \left(\frac{1}{\mu}\phi_j, \phi_i\right), \qquad K_{ij} = \left(\frac{1}{\rho}\nabla\phi_j, \nabla\phi_i\right), \qquad 1 \le i, j \le N.$$

The matrix M is sparse, symmetric and positive definite, whereas the matrix K is sparse, symmetric but, in general, only positive semi-definite.

Even though explicit numerical time integration may be applied directly to (6), every time-step then requires the solution of a linear system involving M. To avoid that computational work, various mass-lumping techniques have been developed [22, 23, 24], which replace M by a diagonal approximation while retaining the rate of convergence [25]. Alternatively, the spectral element method [26] and the symmetric interior penalty discontinuous Galerkin (DG) method [27] both waive the need for mass-lumping altogether: The former inherently leads to a diagonal mass matrix, whereas the latter leads to a block-diagonal mass matrix with block size equal to the number of degrees of freedom per element. Thus, both alternative FE discretizations also lead to (6) with an essentially diagonal mass matrix M,

so that M^{-1} can be explicitly computed with little effort independently of the mesh size.

Next, we multiply (6) by $M^{-\frac{1}{2}}$ which yields

$$\frac{d^2z}{dt^2} + M^{-\frac{1}{2}}KM^{-\frac{1}{2}}z = 0, \tag{7}$$

with $z = M^{\frac{1}{2}}y$. If we let A denote the matrix $M^{-\frac{1}{2}}KM^{-\frac{1}{2}}$, which is also sparse and symmetric positive semidefinite, we can rewrite (7) as

$$\frac{d^2z}{dt^2} + Az = 0\,. \tag{8}$$

This equivalent formulation of (6) is useful for the analysis; in practice, however, we apply our LTS schemes directly to

$$\frac{d^2y}{dt^2} + By = 0\,, \tag{9}$$

with $B = M^{-1}K$.

2.1. Global time-stepping

To discretize (8) in time, we opt for the popular leap-frog (LF) scheme,

$$\frac{z_{n+1} - 2z_n + z_{n-1}}{\Delta t^2} = -Az_n, \tag{10}$$

where z_n represents an approximation of $z(n\Delta t)$. Not only explicit and thus easy to implement, it also conserves the discrete energy

$$E^{n+\frac{1}{2}} = \frac{1}{2}\left\langle\left(I - \frac{\Delta t^2}{4}A\right)\frac{z_{n+1} - z_n}{\Delta t}, \frac{z_{n+1} - z_n}{\Delta t}\right\rangle + \frac{1}{2}\left\langle A\frac{z_{n+1} + z_n}{2}, \frac{z_{n+1} + z_n}{2}\right\rangle, \tag{11}$$

which approximates $E[u]((n + 1/2)\Delta t)$. For Δt sufficiently small, that is

$$\Delta t \leq \beta h,$$

where β depends on the type of FE discretization used, but not on h, $E^{n+\frac{1}{2}}$ is positive and the LF method is stable.

High-order LF type schemes of order $2s$, $s \geq 1$ can be derived via the modified equation (ME) technique [28, 29]:

$$\frac{z_{n+1} - 2z_n + z_{n-1}}{\Delta t^2} = 2\sum_{i=1}^{s} \frac{\Delta t^{2(i-1)}}{(2i)!}(-A)^i z_n. \tag{12}$$

They also conserve a discrete approximation of the energy,

$$\begin{aligned} E^{n+\frac{1}{2}} &= \frac{1}{2}\left\langle\left(I - \frac{\Delta t^2}{4}\sum_{i=1}^{s}\frac{\Delta t^{2(i-1)}}{(2i)!}(-A)^i\right)\frac{z_{n+1} - z_n}{\Delta t}, \frac{z_{n+1} - z_n}{\Delta t}\right\rangle \\ &+ \frac{1}{2}\left\langle\sum_{i=1}^{s}\frac{\Delta t^{2(i-1)}}{(2i)!}(-A)^i\frac{z_{n+1} + z_n}{2}, \frac{z_{n+1} + z_n}{2}\right\rangle. \end{aligned} \tag{13}$$

Again, $E^{n+\frac{1}{2}}$ is positive under a CFL (Courant-Friedrichs-Lewy) condition, that is

$$\Delta t \leq \alpha_s \beta h,$$

where α_s depends on the time accuracy [30]. For instance, $\alpha_1 = 1$, $\alpha_2 = \sqrt{3} \approx 1.73$ and $\alpha_3 = 1.38$; thus, higher order methods actually allow larger time-steps

2.2. Local time-stepping method

We now assume that the underlying finite element mesh consists of both "coarse" and "fine" elements and let h_{coarse} and h_{fine}, respectively, denote the characteristic length of the smallest element in each sub-region of the mesh. Hence, the time-step of any global explicit time-marching scheme would be dictated by h_{fine} through the stability condition $\Delta t \leq \alpha_s \beta\, h_{\text{fine}}$. To circumvent that severe stability restriction, we shall instead use a larger time-step, $\Delta t = \alpha_s \beta\, h_{\text{coarse}}$, inside the "coarse" part and a smaller time-step, $\Delta\tau = \Delta t/p_1 \leq \alpha_s \beta\, h_{\text{fine}}$, inside the "fine" part of mesh, where p_1 denotes the integer mesh size ratio defined by

$$h_{\text{coarse}}/(p_1 - 1) < h_{\text{fine}} \leq h_{\text{coarse}}/p_1.$$

However, in doing so we must ensure that the resulting LTS method retains the accuracy of the original global time-marching scheme.

To derive an LTS method, we first consider for any fixed time t the auxiliary function

$$z^t(\tau) = \frac{z(t-\tau) + z(t+\tau)}{2}, \qquad 0 \leq \tau \leq \Delta t. \tag{14}$$

Clearly, it satisfies

$$z^t(0) = z(t), \qquad \frac{dz^t}{d\tau}(0) = 0, \qquad \frac{d^2 z^t}{d\tau^2}(\tau) = -\frac{1}{2} A\left(z(t-\tau) + z(t+\tau)\right) = -A z^t(\tau).$$

Therefore, $z^t(\tau)$ is equivalently defined as the solution of

$$\begin{cases} \dfrac{d^2 z^t}{d\tau^2}(\tau) = -A z^t(\tau), & 0 < \tau < \Delta t, \\ z^t(0) = z(t), \qquad \dfrac{dz^t}{d\tau}(0) = 0. \end{cases} \tag{15}$$

Since (14) with $t = t_n = n\Delta t$ and $\tau = \Delta t$ implies that

$$z((n+1)\Delta t) = -z((n-1)\Delta t) + 2z^{n\Delta t}(\Delta t), \tag{16}$$

we can advance $z(t)$ until time t_{n+1} once $z^{n\Delta t}$ is known, which we shall compute by solving (15) with $t = t_n$ numerically.

Next, we partition the unknowns in $z^{n\Delta t}(\tau)$ into a "coarse" and a "fine" subset,

$$z^{n\Delta t}(\tau) = (I - P) z^{n\Delta t}(\tau) + P z^{n\Delta t}(\tau) = z^{n\Delta t,[\text{coarse}]}(\tau) + z^{n\Delta t,[\text{fine}]}(\tau)\,,$$

where the partitioning matrix, P, is diagonal: its diagonal entries, equal to zero or one, identify the unknowns associated with the locally refined region, that is where smaller time-steps are needed. Then, we rewrite (15) with $t = t_n$ as

$$\begin{cases} \dfrac{d^2 z^{n\Delta t}}{d\tau^2}(\tau) = -Az^{n\Delta t,[\text{coarse}]}(\tau) - Az^{n\Delta t,[\text{fine}]}(\tau), \qquad 0 < \tau < \Delta t, \\ z^{n\Delta t}(0) = z(n\Delta t), \qquad \dfrac{dz^{n\Delta t}}{d\tau}(0) = 0. \end{cases} \tag{17}$$

To circumvent the severe CFL restriction on Δt caused by the smaller elements, we shall now treat $z^{n\Delta t,[\text{fine}]}(\tau)$ differently from $z^{n\Delta t,[\text{coarse}]}(\tau)$, depending on the required accuracy.

2.2.1. Second-order local time-stepping

To derive a second-order LTS scheme, we approximate in (17) the function $z^{n\Delta t,[\text{coarse}]}(\tau)$ by its value at $\tau = 0$ and denote by $z^n(\tau)$ the solution of the modified problem

$$\begin{cases} \dfrac{d^2 z^n}{d\tau^2}(\tau) = -A(I-P)z(n\Delta t) - APz^n(\tau), \qquad 0 < \tau < \Delta t, \\ z^n(0) = z(n\Delta t), \qquad \dfrac{dz^n}{d\tau}(0) = 0. \end{cases} \tag{18}$$

Since $z^n(\tau) \simeq z^{n\Delta t}(\tau)$ for $0 \leq \tau \leq \Delta t$, we shall approximate $z^{n\Delta t}(\Delta t)$ by solving (18) with the standard LF method yet with a smaller time-step $\Delta\tau = \Delta t/p$. Because of (16), we thus obtain z_{n+1} as

$$z_{n+1} = -z_{n-1} + 2\,\text{LTS}_2(z_n, -A(I-P)z_n, \Delta t, P, p), \tag{19}$$

where the function LTS_2, defined by Algo. 1, corresponds to the standard LF method applied to (18) with time-step $\Delta\tau = \Delta t/p$. For simplicity, we henceforth assume that A is globally defined.

Function $y_{new} = \text{LTS}_2(y_{inter}, w, \Delta t, P, p)$

$y_{new} := y_{inter} + \frac{1}{2}\left(\frac{\Delta t}{p}\right)^2 (w - APy_{inter})$

for $m = 1$ **to** $p - 1$ **do**

  $y_{old} := y_{inter}$, $y_{inter} := y_{new}$

  $y_{new} := 2y_{inter} - y_{old} + \left(\frac{\Delta t}{p}\right)^2 (w - APy_{inter})$

end

Algorithm 1: Second-order local time-stepping

To compute z_{n+1} in (19), the LTS_2 method requires in Algo. 1 a single multiplication by $A(I-P)$ (to compute w) and p multiplications by AP. Because P vanishes outside the fine region, those p multiplications only affect the unknowns in the refined region, or immediately next to it. The successive updates of the coarse unknowns involving $(\Delta t/p)^2 w$ during sub-steps reduce to a single standard LF step of size Δt and, in fact, can be replaced by it. In that sense, Algo. 1 together with (19) yields a local time-stepping method. In [1], we have proved the following result.

Proposition 2.1. *The* LTS_2 *method* (19) *is equivalent to*

$$z_{n+1} = -z_{n-1} + 2z_n - \Delta t^2 A_p z_n,$$

where the matrix A_p *is defined by*

$$A_p = A - \frac{2}{p^2} \sum_{j=1}^{p-1} \left(\frac{\Delta t}{p}\right)^{2j} \alpha_j^p (AP)^j A \tag{20}$$

with α_j^p *constant. Moreover, it is second-order accurate and conserves the discrete energy,*

$$E^{n+\frac{1}{2}} = \frac{1}{2}\left[\left\langle \left(I - \frac{\Delta t^2}{4} A_p\right) \frac{z_{n+1}-z_n}{\Delta t}, \frac{z_{n+1}-z_n}{\Delta t} \right\rangle + \left\langle A_p \frac{z_{n+1}+z_n}{2}, \frac{z_{n+1}+z_n}{2} \right\rangle\right], \tag{21}$$

for Δt *sufficiently small.*

The coefficients α_j^p are explicitly given in [1] but are never needed in practice, as (20) is only used to prove the symmetry of A_p. In fact, A_p itself is never needed, except to numerically determine the stability range of the scheme. Instead of (20), however, it is often more convenient to directly use algorithm LTS_2 to compute A_p. Indeed since

$$(2I - \Delta t^2 A_p) z_n = z_{n+1} + z_{n-1} = 2\,\mathrm{LTS}_2(z_n, -A(I-P)z_n, \Delta t, P, p)),$$

we can compute A_p merely by replacing in (19) z_n by successive columns of I, which yields

$$A_p = \frac{2}{\Delta t^2}(I - \mathrm{LTS}_2(I, -A(I-P), \Delta t, P, p)).$$

2.2.2. High-order local time stepping

To obtain an LTS scheme of arbitrarily high order $2s$, $s \geq 1$, we now approximate in (17) the function $z^{n\Delta t,[\mathrm{coarse}]}(\tau)$ through Taylor expansion as

$$z^{n\Delta t,[\mathrm{coarse}]}(\tau) = \sum_{i=0}^{2s-2} \frac{\tau^i}{i!} \frac{d^i z^{n\Delta t,[\mathrm{coarse}]}}{d\tau^i}(0) + O(\tau^{2s-1})). \tag{22}$$

By successive differentiation of (15) with $t = n\Delta t$, we infer that

$$\frac{d^{2i} z^{n\Delta t,[\text{coarse}]}}{d\tau^{2i}}(0) = (I-P)\frac{d^{2i} z^{n\Delta t}}{d\tau^{2i}}(0) = (I-P)(-A)^i z(n\Delta t),$$

and

$$\frac{d^{2i+1} z^{n\Delta t,[\text{coarse}]}}{d\tau^{2i+1}}(0) = (I-P)\frac{d^{2i+1} z^{n\Delta t}}{d\tau^{2i+1}}(0) = (I-P)(-A)^i \frac{dz^{n\Delta t}}{d\tau}(0) = 0;$$

therefore, all terms with odd indices vanish in (22). Again we denote by $z^n(\tau)$ the solution of the corresponding modified problem:

$$\begin{cases} \dfrac{d^2 z^n}{d\tau^2}(\tau) = -\displaystyle\sum_{i=0}^{s-1} \frac{\tau^{2i}}{(2i)!} A(I-P)(-A)^i z(n\Delta t) - APz^n(\tau), \\ z^n(0) = z(n\Delta t), \qquad \dfrac{dz^n}{d\tau}(0) = 0, \end{cases} \tag{23}$$

Since $z^n(\tau) \simeq z^{n\Delta t}(\tau)$ for $0 \le \tau \le \Delta t$, we shall approximate $z^{n\Delta t}(\Delta t)$ by solving (23) with the standard ME method of order $2s$ but with a smaller time-step $\Delta\tau = \Delta t/p$.

For $s = 2$, we thus obtain the fourth-order LTS method

$$z_{n+1} = -z_{n-1} + 2\text{LTS}_4(z_n, w_1, w_2, \tilde{w}, \Delta t, P, p), \tag{24}$$

where w_1, w_2, and $\tilde{w}$ are defined as

$$w_1 = -A(I-P)z_n, \qquad w_2 = -APz_n, \qquad \tilde{w} = A(I-P)Az_n = -A(I-P)(w_1 + w_2),$$

and the function LTS_4, defined by Algo. 2, corresponds to the standard ME method applied to (23) with time-step $\Delta\tau = \Delta t/p$. To compute z_{n+1} in (24), the LTS_4 method requires in Algo. 2 two multiplications by $A(I-P)$ (to compute w_1 and $\tilde{w}$) and $2p$ multiplications by AP. Because P vanishes outside the fine region, those $2p$ multiplications only affect the unknowns in the refined region, or immediately next to it. In that sense, Algo. 2 together with (24) yields a local time-stepping method. In [1], we have proved the following result.

Proposition 2.2. *The* LTS_4 *method* (24) *is fourth-order accurate. For* Δt *sufficiently small, it conserves the discrete energy*

$$E^{n+\frac{1}{2}} = \frac{1}{2}\left[\left\langle \left(A - \frac{\Delta t^2}{4} AA_p\right)\frac{z_{n+1}-z_n}{\Delta t}, \frac{z_{n+1}-z_n}{\Delta t}\right\rangle \right. \tag{25}$$

$$\left. + \left\langle AA_p \frac{z_{n+1}+z_n}{2}, \frac{z_{n+1}+z_n}{2}\right\rangle\right], \tag{26}$$

Function $y_{new} = \mathrm{LTS}_4(y_{inter}, w_1, w_2, \tilde{w}, \Delta t, P, p)$

$\tau := 0$

$$y_{new} := y_{inter} + \frac{1}{2}\left(\frac{\Delta t}{p}\right)^2 (w_1 + w_2) + \frac{1}{24}\left(\frac{\Delta t}{p}\right)^4 (\tilde{w} - AP(w_1 + w_2))$$

for $m = 1$ **to** $p - 1$ **do**

$$y_{old} := y_{inter},\ y_{inter} := y_{new},\ \tau := \tau + \Delta t/p$$

$$v := w_1 + \frac{\tau^2}{2}\tilde{w} - APy_{new}$$

$$y_{new} := 2y_{inter} - y_{old} + \left(\frac{\Delta t}{p}\right)^2 v + \frac{1}{12}\left(\frac{\Delta t}{p}\right)^4 (\tilde{w} - APv)$$

end

Algorithm 2: Fourth-order local time-stepping

where the matrix A_p is defined by

$$A_p = A - \frac{\Delta t^2}{12}A^2 - \frac{2}{p^2}\sum_{j=1}^{2(p-1)} \left(\frac{\Delta t}{p}\right)^{2(j+1)} \beta_{j,p}(AP)^j A^2, \tag{27}$$

with $\beta_{j,p}$ constant. Moreover, the matrix AA_p is symmetric.

Again, the constants $\beta_{j,p}$ are explicitly given in (Prop. 4.3, [1]), but never needed in practice. If need be, the matrix A_p can easily be computed by applying the LTS_4 algorithm, with z_n replaced by the identity matrix, as

$$A_p = \frac{2}{\Delta t^2}(I - \mathrm{LTS}_4(I, -A(I-P), -AP, -A(I-P)A, -APA, \Delta t, P, p)).$$

3. Multilevel local time-stepping

If the refined part of the mesh itself contains a small subregion of even further local space refinement, it becomes more efficient to introduce yet another level of local time-stepping associated with it. Thus, we let P_1 denote the diagonal partitioning matrix whose diagonal entries, equal to zero or one, identify the unknowns associated with the first level of local mesh refinement. Similarly, we let P_2 denote the diagonal partioning matrix associated with the second level of local mesh refinement. Since that subregion of "very fine" elements lies inside the former subregion of "fine" elements, we have $P_1P_2 = P_2$. Hence the solution $z(t)$ now separates into

three distinct non-overlapping parts as

$$z = (I - P_1)z + (P_1 - P_2)z + P_2 z,$$

associated with the "coarse", the "fine", and the "very fine" elements, respectively, while excluding unknowns that also pertain to any subsequent finer level.

Next, we denote by $p_1, p_2 \geq 2$ the relative mesh size ratio associated with the first and second level of local refinement, respectively. To advance the solution from t_n to $t_n + \Delta t$, we shall again solve (15) for $t = t_n$ and $0 \leq \tau \leq \Delta t$ with time-step $\Delta\tau = \Delta t/p_1$ in the "fine" part of the mesh. Inside the embedded "very fine" subregion, however, we shall use a new, even smaller time-step $\Delta\theta = \Delta\tau/p_2$. Clearly, in doing so we must preserve both the accuracy and energy conservation properties of the original time-stepping method. In analogy to (14), we thus define for a fixed value of τ the auxiliary function

$$z^{n,\tau}(\theta) = \frac{z^n(\tau - \theta) + z^n(\tau + \theta)}{2}, \qquad 0 \leq \theta \leq \Delta\tau. \tag{28}$$

Hence, (28) with $\tau = m\Delta\tau$ and $\theta = \Delta\tau$ yields $z^n((m+1)\Delta\tau)$, once $z^{n,m\Delta\tau}(\Delta\tau)$ is known. To calculate $z^{n,m\Delta\tau}(\theta)$ at $\theta = \Delta\tau$, we first derive a differential equation satisfied by $z^{n,m\Delta\tau}$ and then approximate its solution numerically.

3.1. Second-order multilevel local time-stepping method

Since z^n is the solution of (18), we deduce from (28) that

$$\begin{aligned}
\frac{d^2 z^{n,\tau}}{d\theta^2}(\theta) &= \frac{1}{2}\left(\frac{d^2 z^n}{d\tau^2}(\tau - \theta) + \frac{d^2 z^n}{d\tau^2}(\tau + \theta)\right) \\
&= -A(I - P_1)\, z(n\Delta t) - \frac{1}{2} A P_1 \left(z^n(\tau - \theta) + z^n(\tau + \theta)\right) \\
&= -A(I - P_1)\, z(n\Delta t) - A P_1 z^{n,\tau}(\theta), \qquad -\Delta\tau < \theta < \Delta\tau.
\end{aligned}$$

By setting $\tau = m\Delta\tau$, we thus find that $z^{n,m\Delta\tau}(\theta)$ is equivalently defined as the solution of

$$\begin{cases}
\dfrac{d^2 z^{n,m\Delta\tau}}{d\theta^2}(\theta) = -A(I - P_1)\, z(n\Delta t) - A P_1 z^{n,m\Delta\tau}(\theta), \\
z^{n,m\Delta\tau}(0) = z^n(m\Delta\tau), \qquad \dfrac{dz^{n,m\Delta\tau}}{d\theta}(0) = 0.
\end{cases} \tag{29}$$

Note that the first term on the right of (29) does not depend on θ.

Next, we introduce the partitioning

$$z^{n,m\Delta\tau}(\theta) = (I - P_2)\, z^{n,m\Delta\tau}(\theta) + P_2 z^{n,m\Delta\tau}(\theta) =: z^{n\Delta t, \frac{m}{p_1}\Delta t,[\text{coarse}]}(\theta) + z^{n\Delta t, \frac{m}{p_1}\Delta t,[\text{fine}]}(\theta),$$

and rewrite (29) as

$$\begin{cases} \dfrac{d^2 z^{n,m\Delta\tau}}{d\theta^2}(\theta) = -A(I - P_1)z(n\Delta t) - A(P_1 - P_2)z^{n,m\Delta\tau}(\theta) - AP_2 z^{n,m\Delta\tau}(\theta), \\ z^{n,m\Delta\tau}(0) = z^n\,(m\Delta\tau)\,, \qquad \dfrac{dz^{n,m\Delta\tau}}{d\theta}(0) = 0, \end{cases} \tag{30}$$

In (30), we now use that $(z^{n,m\Delta\tau})'(0) = 0$ to approximate the second term on the right by Taylor expansion as

$$A(P_1 - P_2)z^{n,m\Delta\tau}(\theta) = A(P_1 - P_2)z^{n,m\Delta\tau}(0) + O(\theta^2),$$

and denote by $z^{n,\frac{m}{p_1}}(\theta)$ the solution of the resulting differential equation:

$$\begin{cases} \dfrac{d^2 z^{n,\frac{m}{p_1}}}{d\theta^2}(\theta) \;=\; -A(I - P_1)z(n\Delta t) - A(P_1 - P_2)z^n(m\Delta\tau) - AP_2 z^{n,\frac{m}{p_1}}(\theta), \qquad 0 < \theta < \Delta\tau \\ z^{n,\frac{m}{p_1}}(0) = z^n(m\Delta\tau), \qquad \dfrac{dz^{n,\frac{m}{p_1}}}{d\theta}(0) = 0. \end{cases} \tag{31}$$

Finally, we use the LF method with time-step $\Delta\theta$ to solve (31) until $\theta = \Delta\tau$ and update the solution z^n for $m \geq 1$ as

$$z^n\,((m+1)\Delta\tau) = -z^n\,((m-1)\Delta\tau) + 2z^{n,\frac{m}{p_1}}(\Delta\tau),$$

and for $m = 0$ as $z^n(\Delta\tau) = z^{n,0}(\Delta\tau)$, since $z^n(\tau)$ is an even function. Successive application for $m = 0, \ldots, p_1 - 1$ yields Algorithm 3, which computes z_{n+1}, approximation of $z((n+1)\Delta t)$, given z_n and z_{n-1}, by

$$z_{n+1} = -z_{n-1} + 2\,\text{TLTS}_2(z_n, -A(I - P_1)z_n, \Delta t, P_1, P_2, p_1, p_2).$$

It requires a single multiplication by $A(I - P_1)$ at the coarsest level, p_1 multiplications by $A(P_1 - P_2)$ at the "fine" level, and $p_1 p_2$ multiplications by AP_2 at the "very fine" level.

We are now in position to define a multilevel LTS algorithm for any number of refinement levels. Assume that the first locally refined subgrid, $\mathcal{T}_1$, itself contains a hierachy of increasingly finer grids $\{\mathcal{T}_\ell\}$, such that $\mathcal{T}_\ell \subset \mathcal{T}_{\ell-1}$ for $\ell = 2, \ldots, N_{level}$. To each level $\ell \geq 1$, we associate a diagonal projection matrix, P_ℓ, so that left multiplication with P_ℓ selects precisely those unknowns that belong to $\mathcal{T}_\ell$. At the top level, $\ell = 0$, we set $\mathcal{T}_0 = \mathcal{T}_h$ and $P_0 = I$. Next for $\ell = 0, .., N_{level} - 1$,

Function $y_{new} = \mathrm{TLTS}_2(y_{inter}, w, \Delta t, P_1, P_2, p_1, p_2)$

$y_{new} := \mathrm{LTS}_2(y_{inter}, w - A(P_1 - P_2)y_{inter}, \Delta t/p_1, P_2, p_2)$

for $m = 1$ **to** $p_1 - 1$ **do**

  $y_{old} := y_{inter}, y_{inter} := y_{new}$

  $y_{new} := -y_{old} + 2\mathrm{LTS}_2(y_{inter}, w - A(P_1 - P_2)y_{inter}, \Delta t/p_1, P_2, p_2)$

end

Algorithm 3: Two-level second-order local time-stepping

we denote by h_ℓ the size of the smallest element of subgrid $\mathcal{T}_\ell \setminus \mathcal{T}_{l+1}$, and by $h_{N_{level}}$ the size of the smallest element of $\mathcal{T}_{N_{level}}$, and hence across the entire mesh $\mathcal{T}_h$. Moreover for $\ell = 1, \ldots, N_{level}$, we denote by p_ℓ the mesh size ratio between two subsequent grids, that is the integer such that $h_{\ell-1}/(p_\ell - 1) < h_\ell \leq h_{\ell-1}/p_\ell$, and set $p_0 = 1$.

Recursive application of the above derivation then yields Algorithm 4, which computes z_{n+1} as

$$z_{n+1} = -z_{n-1} + 2\,\mathrm{MLTS}_2(z_n, -A(I - P_1)z_n, \Delta t, 1). \tag{32}$$

Here for simplicity, we assume that the number of levels, N_{level}, the mesh size ratios, p_ℓ, and the projection matrices, P_ℓ, $\ell = 0, \ldots, N_{level}$, are global variables.

Function $y_{new} = \mathrm{MLTS}_2(y_{inter}, w, \Delta t, l)$

if $l < N_{level}$ **then**

  $y_{new} := \mathrm{MLTS}_2(y_{inter}, w - A(P_l - P_{l+1})y_{inter}, \Delta t/p_l, l + 1)$

  for $m = 1$ **to** $p_l - 1$ **do**

    $y_{old} := y_{inter}, y_{inter} := y_{new}$

    $y_{new} := -y_{old} + 2\mathrm{MLTS}_2(y_{inter}, w - A(P_l - P_{l+1})y_{inter}, \Delta t/p_l, l + 1)$

  end

else

  $y_{new} := \mathrm{LTS}_2(y_{inter}, w, \Delta t, P_{N_{Level}}, p_{N_{Level}})$

end

return y_{new}

Algorithm 4: Multi-level second-order local time-stepping.

3.2. Higher order multilevel local time-stepping method

Since z^n now is the solution of (23), we deduce from (28) that for $-\Delta\tau < \theta < \Delta\tau$

$$\frac{d^2 z^{n,\tau}}{d\theta^2}(\theta) = -\frac{1}{2}\sum_{i=0}^{s-1} \frac{(\tau-\theta)^{2i} + (\tau+\theta)^{2i}}{(2i)!} A(I-P_1)(-A)^i z(n\Delta t) - AP_1 \frac{1}{2}\left(z^n(\tau-\theta) + z^n(\tau+\theta)\right).$$

By setting $\tau = m\Delta\tau$, we thus find that $z^{n,m\Delta\tau}(\theta)$ is equivalently defined as the solution of

$$\begin{aligned}
&\frac{d^2 z^{n,m\Delta\tau}}{d\theta^2}(\theta) = -\frac{1}{2}\sum_{i=0}^{s-1} \frac{(m\Delta\tau-\theta)^{2i} + (m\Delta\tau+\theta)^{2i}}{(2i)!} A(I-P_1)(-A)^i z(n\Delta t) - AP_1 z^{n,m\Delta\tau}(\theta),\\
&z^{n,m\Delta\tau}(0) = z^n(m\Delta\tau), \qquad \frac{dz^{n,m\Delta\tau}}{d\theta}(0) = 0.
\end{aligned} \tag{33}$$

Both terms on the right of (33) now depend on θ.

Again, we introduce the partitioning

$$z^{n,m\Delta\tau}(\theta) = (I-P_2)\, z^{n,m\Delta\tau}(\theta) + P_2 z^{n,m\Delta\tau}(\theta) =: z^{n\Delta t, \frac{m}{p_1}\Delta t,[\text{coarse}]}(\theta) + z^{n\Delta t, \frac{m}{p_1}\Delta t,[\text{fine}]}(\theta),$$

and rewrite (33) as

$$\begin{aligned}
\frac{d^2 z^{n,m\Delta\tau}}{d\theta^2}(\theta) = &-\frac{1}{2}\sum_{i=0}^{s-1} \frac{(m\Delta\tau-\theta)^{2i} + (m\Delta\tau+\theta)^{2i}}{(2i)!} A(I-P_1)(-A)^i z(n\Delta t)\\
&-A(P_1-P_2) z^{n,m\Delta\tau}(\theta) - AP_2 z^{n,m\Delta\tau}(\theta),\\
z^{n,m\Delta\tau}(0) = z^n(m\Delta\tau), &\qquad \frac{dz^{n,m\Delta\tau}}{d\theta}(0) = 0.
\end{aligned} \tag{34}$$

In (34), we now approximate the second term on the right by Taylor expansion of order $2s$,

$$A(P_1-P_2) z^{n,m\Delta\tau}(\theta) = A(P_1-P_2)\sum_{i=0}^{2s-1} \frac{\theta^i}{i!} \frac{d^i z^{n,m\Delta\tau}}{d\theta^i}(0) + O(\theta^{2s}),$$

and denote by $z^{n,\frac{m}{p_1}}(\theta)$ the solution of the resulting differential equation. Since $z^{n,m\Delta\tau}(\theta)$ is even in θ, all odd derivatives vanish at $\theta = 0$ and $z^{n,\frac{m}{p_1}}(\theta)$ thus satisfies

$$\begin{aligned}
\frac{d^2 z^{n,\frac{m}{p_1}}}{d\theta^2}(\theta) = &-\frac{1}{2}\sum_{i=0}^{s-1} \frac{(m\Delta\tau-\theta)^{2i} + (m\Delta\tau+\theta)^{2i}}{i!} A(I-P_1)(-A)^i z(n\Delta t)\\
&-A(P_1-P_2)\sum_{i=0}^{s-1} \frac{\theta^{2i}}{(2i)!} \frac{d^{2i} z^{n,\frac{m}{p_1}}}{d\theta^{2i}}(0) - AP_2 z^{n,\frac{m}{p_1}}(\theta),\\
z^{n,\frac{m}{p_1}}(0) = z^n(m\Delta\tau), &\qquad \frac{dz^{n,\frac{m}{p_1}}}{d\theta}(0) = 0.
\end{aligned} \tag{35}$$

Finally, we use the ME approach of order $2s$ to solve (35) with time-step $\Delta\theta$ until $\theta = \Delta\tau$. Then, we update the solution for $m \geq 1$ as

$$z^n\left((m+1)\Delta\tau\right) = -z^n\left((m-1)\Delta\tau\right) + 2z^{n,\frac{m}{p_1}}(\Delta\tau),$$

and for $m = 0$ as $z^n(\Delta\tau) = z^{n,0}(\Delta\tau)$, since $z^n(\tau)$ is an even function.

3.3. Fourth-order multilevel local time-stepping algorithm

In practice, the case $s = 2$ is probably most relevant. Then, (35) reduces to

$$\begin{aligned}\frac{d^2 z^{n,\frac{m}{p_1}}}{d\theta^2}(\theta) &= -A(I-P_1)z(n\Delta t) + \frac{(m\Delta\tau)^2+\theta^2}{2}A(I-P_1)Az(n\Delta t)\\ &\quad -A(P_1-P_2)z^n(m\Delta\tau) - \frac{\theta^2}{2}A(P_1-P_2)\frac{d^2 z^{n,\frac{m}{p_1}}}{d\theta^2}(0) - AP_2 z^{n,\frac{m}{p_1}}(\theta). \qquad (36)\\ z^{n,\frac{m}{p_1}}(0) &= z^n(m\Delta\tau), \qquad \frac{dz^{n,\frac{m}{p_1}}}{d\theta}(0) = 0.\end{aligned}$$

To simplify notation, we now define the auxiliary variables w_1^n, $w_2^{n,m}$, and $\tilde{w}_1^n$ by

$$w_1^n = -A(I-P_1)z(n\Delta t), \qquad w_2^{n,m} = -A(P_1-P_2)z^n(m\Delta\tau), \qquad \tilde{w}_1^n = A(I-P_1)Az(n\Delta t),$$

and thus rewrite (36) as

$$\frac{d^2 z^{n,\frac{m}{p_1}}}{d\theta^2}(\theta) = w_1^n + w_2^{n,m} + \frac{(m\Delta\tau)^2+\theta^2}{2}\tilde{w}_1^n - \frac{\theta^2}{2}A(P_1-P_2)\frac{d^2 z^{n,\frac{m}{p_1}}}{d\theta^2}(0) - AP_2 z^{n,\frac{m}{p_1}}(\theta). \qquad (37)$$

Next, we compute $\dfrac{d^2 z^{n,\frac{m}{p_1}}}{d\theta^2}(0)$ by setting $\theta = 0$ in (37) and using (36) in the resulting expression, which yields:

$$\frac{d^2 z^{n,\frac{m}{p_1}}}{d\theta^2}(0) = w_1^n + w_2^{n,m} - AP_2 z^n(m\Delta\tau) + \frac{(m\Delta\tau)^2}{2}\tilde{w}_1^n.$$

We now define $w_3^{n,m} = -AP_2 z^n(m\Delta\tau)$ and

$$\tilde{w}_2^{n,m} = -A(P_1-P_2)\frac{d^2 z^{n,\frac{m}{p_1}}}{d\theta^2}(0) = -A(P_1-P_2)\left(w_1^n + w_2^{n,m} + w_3^{n,m} + \frac{(m\Delta\tau)^2}{2}\tilde{w}_1^n\right),$$

and thus rewrite (36) as

$$\begin{aligned}\frac{d^2 z^{n,\frac{m}{p_1}}}{d\theta^2}(\theta) &= w_1^n + w_2^{n,m} + \frac{(m\Delta\tau)^2}{2}\tilde{w}_1^n + \frac{\theta^2}{2}\left(\tilde{w}_1^n + \tilde{w}_2^{n,m}\right) - AP_2 z^{n,\frac{m}{p_1}}(\theta), \qquad (38)\\ z^{n,\frac{m}{p_1}}(0) &= z^n(m\Delta\tau), \qquad \frac{dz^{n,\frac{m}{p_1}}}{d\theta}(0) = 0.\end{aligned}$$

Finally, we discretize (38) by using the fourth-order ME approach with time-step $\Delta\theta = \Delta\tau/p_2$, which is based on the approximation:

$$\frac{z^{n,\frac{m}{p_1}}(\theta+\Delta\theta) - 2z^{n,\frac{m}{p_1}}(\theta) + z^{n,\frac{m}{p_1}}(\theta-\Delta\theta)}{\Delta\theta^2} \approx \frac{d^2 z^{n,\frac{m}{p_1}}}{d\theta^2}(\theta) + \frac{\Delta\theta^2}{12}\frac{d^4 z^{n,\frac{m}{p_1}}}{d\theta^4}(\theta).$$

The second-order derivative follows immediately from (38) whereas the fourth-order derivative is obtained by first differentiating it twice:

$$\frac{d^4 z^{n,\frac{m}{p_1}}}{d\theta^4}(\theta) = \tilde{w}_1^n + \tilde{w}_2^{n,m} - AP_2\frac{d^2 z^{n,\frac{m}{p_1}}}{d\theta^2}(\theta).$$

After p_2 steps of the ME method, we update the solution z^n for $m \geq 1$ as

$$z^n((m+1)\Delta\tau) = -z^n((m-1)\Delta\tau) + 2z^{n,\frac{m}{p_1}}(\Delta\tau),$$

and for $m = 0$ as $z^n(\Delta\tau) = z^{n,0}(\Delta\tau)$, since $z^n(\tau)$ is an even function. Successive application for $m = 0, \ldots, p_1 - 1$ yields a two-level fourth-order LTS method, which computes z_{n+1}, approximation of $z((n+1)\Delta t)$, given z_n and z_{n-1}, by

$$z_{n+1} = -z_{n-1} + 2\mathrm{TLTS}_4(z_n, w_1, w_2, w_3, \tilde{w}, \Delta t, P_1, P_2, p_1, p_2),$$

where w_1, w_2, w_3 are defined as

$$w_1 = -A(I - P_1)z_n, \qquad w_2 = -A(P_1 - P_2)z_n, \qquad w_3 = -AP_2 z_n,$$

and $\tilde{w}$ is defined as

$$\tilde{w} = A(I - P_1)Az_n = -A(I - P_1)(w_1 + w_2 + w_3),$$

– see Algo. 5. It requires two multiplications by $A(I - P_1)$ at the coarsest level, $2p_1$ multiplications by $A(P_1 - P_2)$ at the "fine" level, and $2p_1p_2$ multiplications by AP_2 at the "very fine" level.

Again, we can extend the above derivation to any number of refinement levels $\{\mathcal{T}_\ell\}$, $\mathcal{T}_\ell \subset \mathcal{T}_{\ell-1}$, $\ell = 2, \ldots, N_{level}$, each associated with its diagonal projection matrix P_ℓ and local mesh size ratio p_ℓ. This yields the following fourth-order MLTS algorithm, which first requires the definition of the auxiliary variables

1. For $k = 1, \ldots, N_{level}$, $w_k = -A(P_{k-1} - P_k)z_n$
2. $w_{N_{Level}+1} = -AP_{N_{Level}}z_m$

Function $y_{new} = \mathrm{TLTS}_4(y_{inter}, w_1, w_2, w_3, \tilde{w}, \Delta t, P_1, P_2, p_1, p_2)$

$\tau := 0$

$v := w_1 + w_2 + w_3$

$y_{new} := \mathrm{LTS}_4(y_{inter}, w_1 + w_2, w_3, \tilde{w} - A(P_1 - P_2)v, \Delta t/p_1, P_2, p_2)$

for $m = 1$ **to** $p_1 - 1$ **do**

$y_{old} := y_{inter}$, $y_{inter} := y_{new}$, $\tau := \tau + \Delta t/p_1$

$w_2 := -A(P_1 - P_2)y_{new}$, $w_3 := -AP_2 y_{new}$

$v := w_1 + w_2 + w_3 + \tau^2/2\tilde{w}$

$y_{new} := -y_{old} + 2\mathrm{LTS}_4(y_{inter}, w_1 + w_2 + \tau^2/2\tilde{w}, w_3, \tilde{w} - A(P_1 - P_2)v, \Delta t/p_1, P_2, p_2)$

end

Algorithm 5: Two-level fourth-order local time-stepping

3. $\tilde{w} = -A(I - P_1)\sum_{k=1}^{N_{Level}+1} w_k$

and then calls the recursive MLTS_4 function listed in Algo. 6:

$$z_{n+1} = z_{n-1} - 2\mathrm{MLTS}_4(z_n, w_1, \tilde{w}, \Delta t, 1). \tag{39}$$

For simplicity, we assume that the number of levels, N_{level}, the mesh size ratios, p_ℓ, and the projection matrices, P_ℓ, $\ell = 0, \ldots, N_{level}$, together with w_ℓ, $\ell = 1..N_{Level} + 1$, are globally defined.

4. Energy conservation

In [1], we proved that both the second-order and the fourth-order two-level LTS methods conserve a discrete energy – see Prop. 2.1 and 2.2. To extend that analysis to an arbitrary number of levels $N = N_{level}$, we first rewrite the MLTS_2 algorithm (32) in LF fashion:

Proposition 4.1. *The* MLTS_2 *algorithm (32) is equivalent to*

$$\frac{z_{n+1} - 2z_n + z_{n-1}}{\Delta t^2} + A_{p_1 p_2 \ldots p_N} z_n = 0,$$

where $A_{p_1 p_2 \ldots p_N}$ *is a symmetric matrix.*

The proof follows directly from the next two lemmas, which are proved in the appendix.

Function $y_{new} = \mathrm{MLTS}_4(y_{inter}, w, \tilde{w}, \Delta t, l)$

if $l < N_{level}$ **then**

$\tau := 0$

$v := w + \sum_{k=l+1}^{N_{Level}+1} w_k$

$y_{new} := \mathrm{MLTS}_4(y_{inter}, w + w_{l+1}, \tilde{w} - A(P_l - P_{l+1})v), \Delta t/p_l)$

for $m = 1$ **to** $p_l - 1$ **do**

$y_{old} := y_{inter}$, $y_{inter} := y_{new}$, $\tau := \tau + \Delta t/p_l$

for $k = l + 1$ **to** N_{level} **do**

$w_k := -A(P_{k-1} - P_k)y_{inter}$

end

$w_{N_{Level}+1} := -AP_{N_{Level}}y_{inter}$

$v := w + \sum_{k=l+1}^{N_{Level}+1} w_k + \tau^2/2\tilde{w}$

$y_{new} := -y_{old} + 2\mathrm{MLTS}_4(y_{inter}, w + w_{l+1} + \tau^2/2\tilde{w}, \tilde{w} - A(P_l - P_{l+1})v, \Delta t/p_l, l + 1)$

end

else

$y_{new} := \mathrm{LTS}_4(y_{inter}, w, w_{N_{Level}+1}, \tilde{w}, \Delta t, AP_{N_{Level}}, p_{N_{Level}})$

end

return y_{new}.

Algorithm 6: Multi-level fourth-order local time-stepping

Lemma 4.1. *Let* $z = \mathrm{LTS}_2\left(\tilde{z}, w, \frac{\Delta t}{p_1 p_2 \cdots p_{N-1}}, P_N, p_N\right)$. *Then*

$$z = \tilde{z} + B_{p_N}(w - AP_N\tilde{z}),$$

with

$$B_{p_N} = \frac{\Delta t^2}{p_1^2 p_2^2 \cdots p_{N-1}^2}\left(\frac{I}{2} + \sum_{i=1}^{p_N-1} \beta_{i,N} \frac{\Delta t^{2i}}{p_1^{2i} p_2^{2i} \cdots p_{N-1}^{2i}} (AP_N)^i\right), \qquad \beta_{i,N} \in \mathbb{R}, \quad i = 1, \ldots, p_N - 1.$$

Note that the matrix $B_{p_N}A$ is symmetric, since

$$((AP_N)^i A)^\top = A(P_N A)^i = (AP_N)^i A$$

by the symmetry of A and P_N.

Lemma 4.2. *Let* $z = \mathrm{MLTS}_2(\tilde{z}, w, \frac{\Delta t}{p_1 p_2 \cdots p_{N-2}}, N-1)$. *Then,*

$$z = \tilde{z} + B_{p_{N-1}p_N}(w - AP_{N-1}\tilde{z}),$$

with

$$B_{p_{N-1}p_N} = p_{N-1}^2 B_{p_N} + \sum_{i=1}^{p_{N-1}-1} \beta_{i,N-1} B_{p_N}(AP_{N-1}B_{p_N})^i, \qquad \beta_{i,N-1} \in \mathbb{R}, \quad i = 1, \ldots, p_{N-1} - 1.$$

Again the matrix $B_{p_{N-1}p_N}A$ is symmetric, since

$$(B_{p_N}(AP_{N-1}B_{p_N})^i A)^\top = (B_{p_N}A(P_{N-1}B_{p_N}A)^i)^\top = (B_{p_N}AP_{N-1})^i B_{p_N}A = B_{p_N}(AP_{N-1}B_{p_N})^i A$$

by the symmetry of P_{N-1} and $B_{p_N}A$.

Proof. First, we show by induction that

$$z = \mathrm{MLTS}_2(\tilde{z}, w, \frac{\Delta t}{p_1 p_2 \cdots p_{i-1}}, i) = \tilde{z} + B_{p_i p_{i+1} \ldots p_N}(w - AP_i\tilde{z}), \qquad 1 \le i \le N-1, \qquad (40)$$

where the matrix $B_{p_i p_{i+1} \ldots p_N}$ is defined as

$$B_{p_i p_{i+1} \ldots p_N} = p_i^2 B_{p_{i+1} \ldots p_N} + \sum_{j=1}^{p_i - 1} \beta_{j,i} B_{p_{i+1} \ldots p_N}(AP_i B_{p_{i+1} \ldots p_N})^j, \qquad \beta_{j,i} \in \mathbb{R}. \qquad (41)$$

Moreover, the matrix $B_{p_i p_{i+1} \ldots p_N}A$ is symmetric.

For $i = N - 1$, (40) indeed holds by Lemma 4.2.

Next, we assume that (40) holds for any fixed i, $2 \le i \le N-1$. To show that (40) also holds for $i-1$, we now let

$$z = \mathrm{MLTS}_2(\tilde{z}, w, \frac{\Delta t}{p_1 p_2 ... p_{i-2}}, i-1). \tag{42}$$

Following Algo. 4 with $l = i-1 < N-1$ and $y_{inter} = \tilde{z}$, we thus obtain

$$\begin{aligned} y_{new} &= \mathrm{MLTS}_2(\tilde{z}, w - A(P_{i-1} - P_i)\,\tilde{z}, \Delta t/(p_1 p_2 ... p_{i-2} p_{i-1}), i) \\ &= \tilde{z} + B_{p_i p_{i+1} ... p_N}(w - A(P_{i-1} - P_i)\,\tilde{z} - AP_i\tilde{z}) \\ &= \tilde{z} + B_{p_i p_{i+1} ... p_N}(w - AP_{i-1}\tilde{z}), \end{aligned} \tag{43}$$

where we have used the induction hypothesis (40) for the second equality.

Similarly, entering the loop over $m = 1$ to $p_{i-1} - 1$, we deduce that

$$y_{new} = \tilde{z} + \left(4B_{p_i p_{i+1} ... p_N} - 2B_{p_i p_{i+1} ... p_N} A P_{i-1} B_{p_i p_{i+1} ... p_N}\right)(w - AP_{i-1}\tilde{z})$$

after the first iteration. In fact, after $m \ge 1$ iterations, we have

$$y_{new} = \tilde{z} + \left((m+1)^2 B_{p_i p_{i+1} ... p_N} + \sum_{j=1}^{m} \gamma_{j,m} B_{p_i p_{i+1} ... p_N} \left(AP_{i-1} B_{p_i p_{i+1} ... p_N}\right)^j\right)(w - AP_{i-1}\tilde{z}),$$

as one can easily show by induction over m; see also the proof of Lemma 4.2 in the Appendix. In particular, for $m = p_{i-1} - 1$ we have

$$\begin{aligned} y_{new} &= \tilde{z} + \left(p_{i-1}^2 B_{p_i p_{i+1} ... p_N} + \sum_{j=1}^{p_{i-1}-1} \gamma_{j,p_{i-1}-1} B_{p_i p_{i+1} ... p_N} \left(AP_{i-1} B_{p_i p_{i+1} ... p_N}\right)^j\right)(w - AP_{i-1}\tilde{z}) \\ &= \tilde{z} + B_{p_{i-1} p_i p_{i+1} ... p_N}(w - AP_{i-1}\tilde{z}), \end{aligned}$$

by definition (41) of $B_{p_{i-1} p_i p_{i+1} ... p_N}$ with $\beta_{j,i-1} = \gamma_{j,p_{i-1}-1}$.

Again the matrix $B_{p_{i-1} p_i p_{i+1} ... p_N} A$ is symmetric, since

$$\begin{aligned} &\left(B_{p_i p_{i+1} ... p_N} (AP_{i-1} B_{p_i p_{i+1} ... p_N})^j A\right)^\top = \left(B_{p_i p_{i+1} ... p_N} A (P_{i-1} B_{p_i p_{i+1} ... p_N} A)^j\right)^\top \\ &= (B_{p_i p_{i+1} ... p_N} A P_{N-1})^j B_{p_i p_{i+1} ... p_N} A = B_{p_i p_{i+1} ... p_N} (AP_{i-1} B_{p_i p_{i+1} ... p_N})^j A, \end{aligned}$$

by the symmetry of P_{N-1} and $B_{p_i p_{i+1} ... p_N} A$. This concludes the proof of (40).

Finally, we use (40) with $i = 1$ in (32) to infer that

$$\begin{aligned} z_{n+1} &= -z_{n-1} + 2\,\mathrm{MLTS}_2(z_n, -A(I - P_1)z_n, \Delta t, 1) \\ &= -z_{n-1} + 2\,\left(z_n + B_{p_1 ... p_N}[-A(I - P_1)z_n - AP_1 z_n]\right), \\ &= -z_{n-1} + 2\,z_n - 2B_{p_1 ... p_N} A z_n. \end{aligned}$$

This concludes the proof with $A_{p_1 \dots p_N} = (2/\Delta t^2)\, B_{p_1 \dots p_N} A$, which is symmetric. □

Corollary 4.1. *The* MLTS_2 *algorithm (32) is second-order accurate and conserves the discrete energy*

$$\begin{aligned} E^{n+\frac{1}{2}} &= \frac{1}{2}\left\langle \left(I - \frac{\Delta t^2}{4} A_{p_1 \dots p_N}\right) \frac{z_{n+1} - z_n}{\Delta t}, \frac{z_{n+1} - z_n}{\Delta t} \right\rangle \\ &+ \frac{1}{2}\left\langle A_{p_1 \dots p_N} \frac{z_{n+1} + z_n}{2}, \frac{z_{n+1} + z_n}{2} \right\rangle \end{aligned} \tag{44}$$

for Δt *sufficiently small.*

Proof. By Prop. 4.1, the MLTS_2 algorithm (32) can be rewritten in LF fashion (10) with A replaced by the symmetric matrix $A_{p_1 \dots p_N}$. Therefore, it also conserves the energy in (11) with A replaced by $A_{p_1 \dots p_N}$, which corresponds to (44). To ensure that the energy in (44) is positive for Δt sufficiently small, we still need to show that the matrices $I - \frac{\Delta t^2}{4} A_{p_1 \dots p_N}$ and $A_{p_1 \dots p_N}$ are positive definite for Δt sufficiently small.

From Lemma 4.1 we immediately obtain that

$$B_{p_N} = \frac{\Delta t^2}{p_1^2 \cdots p_{N-1}^2} \left(\frac{I}{2} + O(\Delta t^2)\right), \qquad \Delta t \to 0,$$

and similarly from Lemma 4.2 that

$$B_{p_{N-1} p_N} = \frac{\Delta t^2}{p_1^2 \cdots p_{N-2}^2} \left(\frac{I}{2} + O(\Delta t^2)\right), \qquad \Delta t \to 0.$$

From the recursive definition (41) of $B_{p_i \dots p_N}$, we thus easily infer by induction that for $i \geq 1$

$$B_{p_i \dots p_N} = \frac{\Delta t^2}{p_1^2 \cdots p_{i-1}^2} \left(\frac{I}{2} + O(\Delta t^2)\right), \qquad \Delta t \to 0.$$

In particular, for $i = 1$ we have

$$B_{p_1 \dots p_N} = \Delta t^2 \left(\frac{I}{2} + O(\Delta t^2)\right), \qquad \Delta t \to 0,$$

and hence

$$A_{p_1 \dots p_N} = \frac{2}{\Delta t^2} B_{p_1 \dots p_N} A = A + O(\Delta t^2), \qquad \Delta t \to 0.$$

Therefore, since A is positive definite, so are both $A_{p_1 \dots p_N}$ and $I - \frac{\Delta t^2}{4} A_{p_1 \dots p_N}$ for Δt sufficiently small. Moreover, the effective stiffness matrix $A_{p_1 \dots p_N}$ of the MLTS_2 algorithm corresponds to an $O(\Delta t^2)$ perturbation of the original stiffness matrix A. Hence, it is second-order accurate, which completes the proof. □

Proposition 4.1 implies that the MLTS_2 method is equivalent to the standard LF method with A replaced by the matrix $A_{p_1 p_2 \ldots p_{N_{level}}}$. Because $A_{p_1 p_2 \ldots p_{N_{level}}}$ is also symmetric, the MLTS_2 method conserves a discrete energy and is stable when $I - (\Delta t^2/4) A_{p_1 \ldots p_N}$ is positive definite. Hence, the MLTS_2 method is stable for any particular Δt, if all eigenvalues of $(\Delta t^2/4) A_{p_1 \ldots p_N}$ lie between zero and one; otherwise, it is unstable.

Remark 1. *Although the matrix $A_{p_1 \ldots p_N}$ is never used in practice, it is useful for determining the range of values Δt for which the* MLTS_2 *method is stable. To determine $A_{p_1 \ldots p_N}$, we simply apply once Algorithm 4 with $z_{n-1} = 0$ and z_n replaced by the $n \times n$ identity matrix. According to Proposition 4.1, the matrix $A_{p_1 \ldots p_N}$ is then immediately given by*

$$A_{p_1 \ldots p_N} = \frac{1}{\Delta t^2} [2I - z_{n+1}].$$

5. Numerical Results

We shall now present numerical experiments that confirm the expected order of convergence and demonstrate the versatility of the multilevel local time-stepping (MLTS) methods from Section 3. First, we consider a simple one-dimensional test problem to show that the different MLTS schemes are stable and indeed yield the expected overall rate of convergence when combined with a spatial finite element discretization of comparable accuracy. Then, we consider wave propagation in two space dimensions with hierarchical local mesh refinement to illustrate the usefulness of MLTS in the presence of complex geometry.

5.1. Stability and CFL condition

We consider the one-dimensional wave equation (1) with constant $\rho = \mu = 1$ on the interval $\Omega = [0\,;\,3]$ with periodic boundary conditions, which we divide into three equal parts. The two outer intervals, $\Omega_c = [0\,;\,1] \cup [2\,;\,3]$, are discretized with an equidistant mesh of size h_{coarse}: they correspond to the "coarse" region. Next, we divide the inner interval $[1\,;\,2]$ itself into three parts: the two intervals $\Omega_1 = [1\,;\,1.25] \cup [1.75\,;\,2]$ are discretized with an equidistant mesh of size $h_1 = h_{\text{coarse}}/p_1$ and hence correspond to the first level of refinement. Finally, the innermost interval, $\Omega_2 = [1.25\,;\,1.75]$, is discretized with an even finer equidistant mesh of size $h_2 = h_1/p_2 = h_{\text{coarse}}/(p_1 p_2)$; it corresponds to the second level of local refinement. Hence $\Omega = \Omega_c \cup \Omega_1 \cup \Omega_2$, as shown in Fig.1.

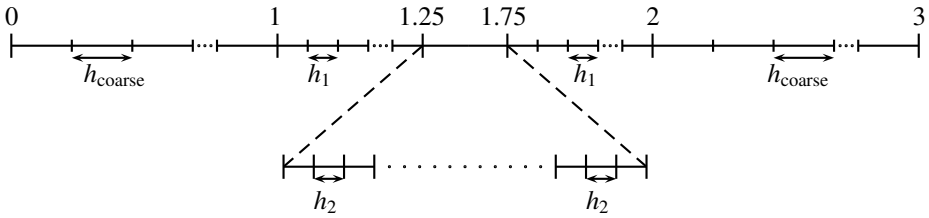

Figure 1: One-dimensional computational mesh: the "coarse", "fine" and "very fine" regions have mesh size h_{coarse}, h_1 and h_2, respectively.

During every time-step Δt, we take p_1 steps of size $\Delta\tau = \Delta t/p_1$ inside Ω_1 and $p_1 p_2$ steps of size $\Delta\theta = \Delta\tau/p_2$ inside Ω_2. In the absence of local refinement, i.e. $p_1 = p_2 = 1$, the mesh is equidistant throughout Ω. Then, the (local) time-stepping algorithm corresponds to the standard leap-frog (LF) method and we denote by Δt_{LF} the largest time-step allowed. Else if either $p_1 \geq 2$ or $p_2 \geq 2$, we denote by $\Delta t_{p_1,p_2}$ the maximal time-step of the considered MLTS method. If $\Delta t_{p_1,p_2} = \Delta t_{LF}$, the MLTS algorithm imposes no further restriction on Δt and we then call the CFL condition of the new scheme optimal.

We now consider the interior point (IP) DG discretization from [27] with $\mathcal{P}^1$–elements and (small) penalization, $\alpha = 2$. At the coarsest level we set $h_{\text{coarse}} = 0.2$, which yields the maximal time-step $\Delta t_{LF} = 0.55\, h_{\text{coarse}} = 0.11$ for the equidistant mesh. We then refine by a factor $p_1 = 2$ those elements that lie inside the interval $[1, 2]$, that is set $h_1 = 0.1$ and to one all corresponding entries in P_1. Next, we refine once again by a factor $p_2 = 2$ those elements that lie inside the interval $[1.25, 1.75]$, that is set $h_2 = 0.5$ and to one all corresponding entries in P_2. Hence for every time-step Δt, we shall take two steps of size $\Delta\tau = \Delta t/2$ inside Ω_1 and four steps of size $\Delta\theta = \Delta t/4$ inside Ω_2.

To determine the range of time-steps for which the MLTS_2 method is stable, we verify for any particular Δt whether all eigenvalues of $(\Delta t^2/4)A_{2,2}$ lie between zero and one – see Remark 1. As shown in the left frame of Fig. 2, the smallest eigenvalue dips below zero for $\Delta t \approx 0.65\Delta t_{LF}$; hence, the largest time-step allowed, though more than twice that dictated by $h_2 = h_{\text{coarse}}/4$, still falls short of the optimum at Δt_{LF}.

To increase the maximal permissible time-step, we now slightly enlarge the set of unknowns where each local time-step is used by adding those degrees of freedom that are associated with elements directly adjacent to the refined region. By setting the corresponding entries in P_1 and

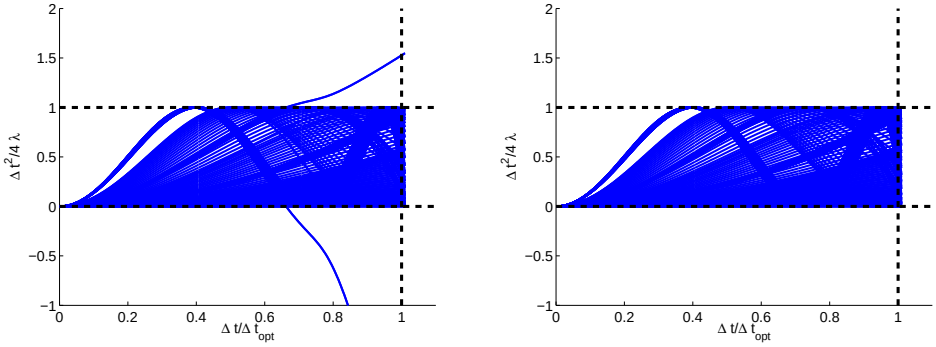

Figure 2: The eigenvalues of $(\Delta t^2/4)A_{2,2}$: without overlap (left); with an overlap by one element (right).

P_2 to one, we easily realize this overlap by one element in $z^{n\Delta t,[\text{fine}]}$ and $z^{n\Delta t,\frac{m}{p_1}\Delta t,[\text{fine}]}$. In the right frame of Fig. 2 we observe that all eigenvalues now lie essentially between zero and one. However, a thousand-fold magnification of that same figure, shown in the left frames of Figs. 3 and 4, reveals that some eigenvalues still transgress the strict stability limit at one. As shown in the right frames of Figs. 3 and 4, further extension of the overlap by one additional element removes all unstable values below $0.9\Delta t_{LF}$, while narrow bands of (barely) unstable values between $0.91 \leq \Delta t/\Delta t_{LF} \leq 0.98$ remain. Here we shall not attempt to elucidate that peculiar and somewhat sensitive behavior, due to weak resonances caused by the underlying regular, one-dimensional grid.

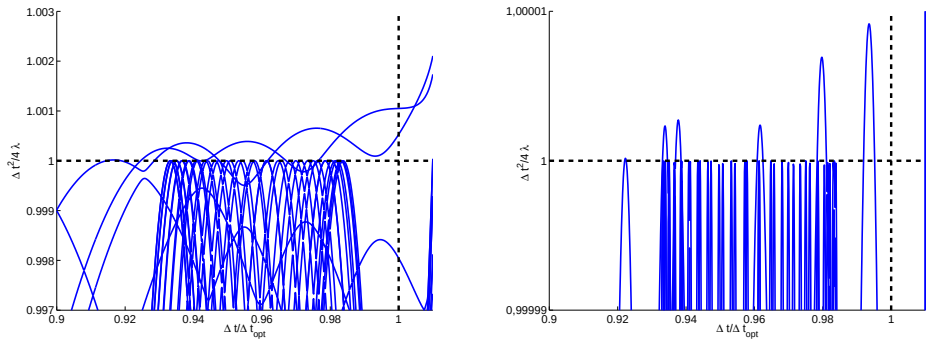

Figure 3: The eigenvalues of $(\Delta t^2/4)A_{2,2}$: overlap by one element (left); overlap by two elements (right). The vertical scale is strongly magnified on the right: $-0.00004 < \lambda_{\min} < .00004$.

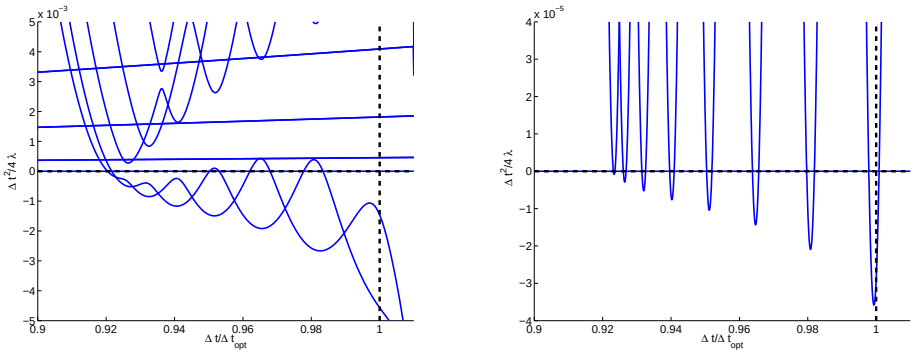

Figure 4: The eigenvalues of $(\Delta t^2/4)A_{2,2}$: overlap by one element (left); overlap by two elements (right). The vertical scale is strongly magnified on the right: $0.99999 < \lambda_{\max} < 1.00001$.

5.2. Convergence study

Again, we consider an IP-DG discretization of (1) with $\rho = \mu = 1$ on $\Omega = (0, 3)$ using $\mathcal{P}^1$ elements and (small) penalty parameter $\alpha = 2$. The initial conditions u_0, v_0 are set to yield the exact solution

$$u_0 = \cos\left(\frac{8\pi}{3}(t - x)\right).$$

As described in Section 5.1, we let $\Omega = \Omega_c \cup \Omega_1 \cup \Omega_2$ and denote the mesh size inside Ω_1 by $h_1 = h_{\text{coarse}}/p_1$ and inside Ω_2 by $h_2 = h_1/p_2$ – see Fig.1. For time integration, we use the three-level MLTS_2 method, which takes p_1 steps of size $\Delta\tau = \Delta t/p_1$ inside Ω_1 and $p_1 p_2$ steps of size $\Delta\theta = \Delta\tau/p_2$ inside Ω_2 during each global time-step Δt. In all cases, we use an overlap of two which enables us to set $\Delta t = \Delta t_{LF}$, the largest time-step allowed by the LF method on an equidistant mesh with $h = h_{\text{coarse}}$.

We now systematically reduce the global mesh size, $h_{\text{coarse}} = 0.1,\ 0.05,\ 0.025,\ 0.0125,\ 0.00625$ together with Δt, while monitoring the L^2 space-time error in the numerical solution, $\|u - u_{ex}\|_{L^2(0,T;L^2(\Omega))}$ until time $T = 60$. In Fig. 5, the numerical error is shown vs. the mesh size, $h = h_{\text{coarse}}$, for the different mesh size ratios $(p_1, p_2) = (1, 1),\ (2, 2)\ (2, 3),\ (3, 2),\ (3, 5)$; for $(p_1, p_2) = (1, 1)$, the mesh is equidistant throughout Ω and the time integration reduces to the standard LF method. Regardless of the number of local time-steps, or their mutual ratio, the MLTS_2 method yields overall second-order space-time convergence, as expected.

Next, we repeat the above numerical experiment but now opt for an IP-DG spatial discretization with $\mathcal{P}^3$ elements and penalty parameter $\alpha = 10$. To reach overall fourth-order space-time convergence with respect to the L^2 norm, we combine it with the MLTS_4 method for time inte-

gration. Again, we choose an overlap of two and let $\Delta t = \Delta t_{ME}$, the largest possible time-step allowed by the modified equation approach on an equidistant mesh with $h = h_{\text{coarse}}$. We now systematically reduce the global mesh size, $h_{\text{coarse}} = 0.1, 0.05, 0.025, 0.0125$, together with Δt, while monitoring the L^2 space-time error in the numerical solution. As shown in Fig. 5, the MLTS$_4$ method leads to fourth-order space-time convergence, regardless of p_1 or p_2.

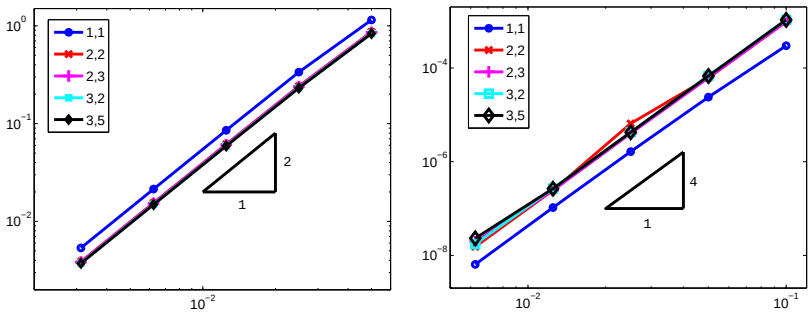

Figure 5: Error vs. $h = h_{\text{coarse}}$ with $(p_1, p_2) = (1,1), (2,2)\ (2,3), (3,2), (3,5)$: MLTS$_2$ (left) and MLTS$_4$(right).

5.3. Two-dimensional example

To illustrate the usefulness of the MLTS methods in the presence of complex geometry, we now consider (1) with constant $\rho = \mu = 1$ in a square domain $\Omega = (-1,1) \times (-1,1)$ with four rectangular slots. Located at $(\pm 0.5, \pm 0.5)$, the four slots are 0.1 in length each but become increasingly narrow, as their width successively decreases from 0.05 to 0.00625 – see Fig. 6. We impose homogeneous Neumann conditions on the boundary of Ω and set the initial conditions to a circular Gaussian of radius $r = 0.025$ centered at the origin:

$$u_0(x) = \begin{cases} \exp(\|x\|^2/r^2), & \|x\| \le \sqrt{2}\, r, \\ 0, & \text{else}, \end{cases}, \qquad v_0(x) = 0, \qquad x \in \Omega. \tag{45}$$

Inside Ω, we now generate a triangular mesh, shown in Fig. 7, by using the program Triangle [31]. To avoid a loss of mesh quality due to the increasingly high aspect ratio, the mesh surrounding each slot is organized into tiers of like-sized triangles, as shown in Figs. 7 and 8. Next to the widest slot in the upper left corner, the mesh size $h_1 = 3.9E-3$ in the locally refined (blue) region is about 3.2 times smaller than that of the coarse mesh, $h_{\text{coarse}} = 1.26E-2$; thus, we set $p_1 = 4$ inside the upper left locally refined region, which dictates the local time-step $\Delta\tau = \Delta t/p_1$. In the vicinity of the slot in the lower left corner, the locally refined mesh is

divided into two tiers with mesh size $h_1 = 4.37E-3$ in the outer (blue) and $h_2 = 1.45E-3$ in the inner (green) region. According to the respective mesh size ratios $h_1/h_{\text{coarse}} = 2.88$ and $h_2/h_{\text{coarse}} = 8.69$, we thus select the time-step ratios $p_1 = 3$ and $p_2 = 3$. Note that p_1, and therefore $\Delta\tau$, may have a different value in the upper and in the lower left corner, as the four regions are completely independent of each other within any of our MLTS algorithms. Similarly, the locally refined mesh surrounding the lower right slot leads to the time-step ratios $p_1 = 5$, $p_2 = 2$ and $p_3 = 6$, whereas that surrounding the narrowest slot in the upper right yields $p_1 = 4$, $p_2 = 2$, $p_3 = 5$, as summarized in Table 1.

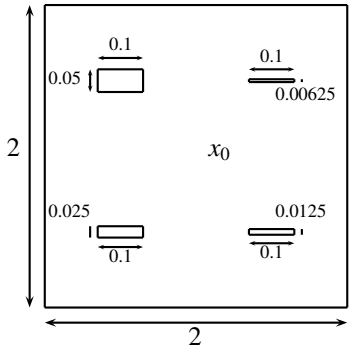

Figure 6: Two-dimensional example: computational domain Ω.

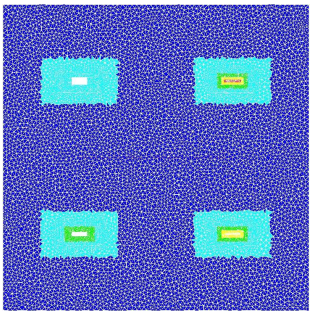

Figure 7: Finite element mesh

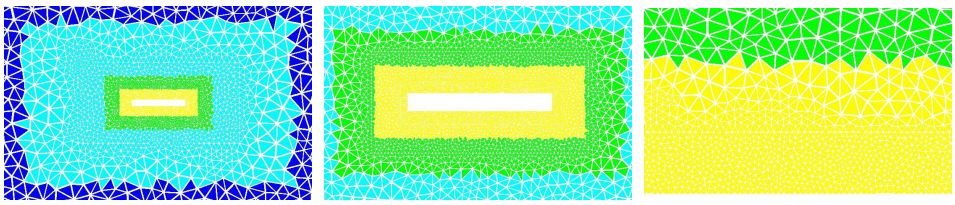

Figure 8: Locally refined mesh sourrounding the lower right slot at increasingly higher magnification. Each color designates a tier of like-sized elements.

For spatial discretization, we opt for the IP-DG method with $\mathcal{P}^3$ triangular elements and penalty parameter $\alpha = 11$. It leads to the CFL condition $\Delta t \leq 0.15\, h_{\text{min}}$, where h_{min} denotes the smallest element size of the mesh. To circumvent that stability restriction across the entire mesh hierarchy shown in Fig. 7 and Fig. 8, while maintaining overall fourth-order space-time accuracy, we combine it with the MLTS$_4$ method from Section 3.3 . Hence for every global time-step Δt, the MLTS$_4$ method will take $p_1 = 4$ steps inside the outer (blue) region, $p_1 p_2 = 8$

Corner	Region	$h_{\min}$	$h_{\min}/h_c$	p
top-left	blue	3.90E-3	3.23	4
bottom-left	blue	4.37E-3	2.88	3
	green	1.45E-3	8.69	9
bottom-right	blue	3.91E-3	3.22	5
	green	1.75E-3	7.2	10
	yellow	4.30E-4	29.3	30
top-right	blue	3.94E-3	3.19	4
	green	1.67E-3	7.54	8
	yellow	3.31E-4	38	40
	red	8.02E-5	157.1	160

Table 1: Properties of the four locally refined regions: smallest element size h_{min}, ratio to the coarse mesh size h_{min}/h_c, ratio to the global time step p.

steps inside the next (green) region, $p_1p_2p_3 = 40$ steps inside the subsequent (yellow) region, and $p_1p_2p_3p_4 = 160$ local time-steps inside the innermost "red" region. In Fig. 9, obtained with the program Paraview [32], we observe how the circular Gaussian wave expands until it impinges upon the four slots. Each slot then generates a circular wave, the smaller the hole, the weaker the reflection, while the main wave front reaches the outer square boundary, without any spurious reflection from mesh interfaces.

We have not proved that the MLTS$_4$ method with more than two levels conserves (a discrete approximation of) the energy. However, Proposition 2.2 and the theory from Section 4 suggest that the MLTS$_4$ method in fact is equivalent to

$$\frac{z_{n-1} - 2z_n + z_{n-1}}{\Delta t^2} + A_{p_1 \dots p_N} z_n = 0, \tag{46}$$

with $AA_{p_1 \dots p_N}$ symmetric, and hence that it does conserve the discrete energy

$$\begin{aligned} E^{n+\frac{1}{2}} \;=\; & \frac{1}{2}\left[\left\langle \left(A - \frac{\Delta t^2}{4} AA_{p_1 \dots p_N}\right) \frac{z_{n+1} - z_n}{\Delta t}, \frac{z_{n+1} - z_n}{\Delta t} \right\rangle \right. \\ & \left. + \left\langle AA_{p_1 \dots p_N} \frac{z_{n+1} + z_n}{2}, \frac{z_{n+1} + z_n}{2} \right\rangle \right]. \end{aligned} \tag{47}$$

As illustrated in Fig. 10, the MLTS$_4$ method indeed conserves to machine precision the discrete

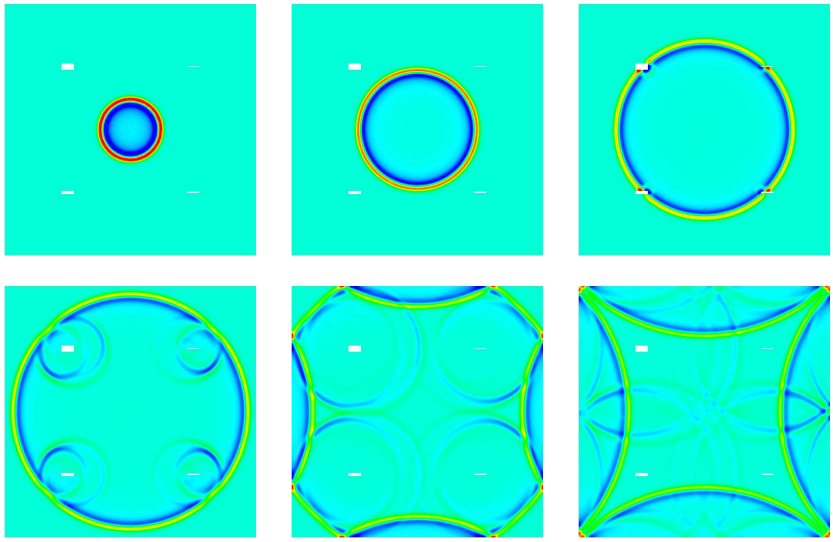

Figure 9: Two-dimensional example: the solution is shown at times $t = 0.23, 0.46, 0.69, 0.92, 1.15, 1.38$.

energy in (47). Note that it is not necessary to explicitly compute $A_{p_1 \dots p_N}$, since the product $A_{p_1 \dots p_N} z_n$ can be computed from (46) by using three subsequent time-steps.

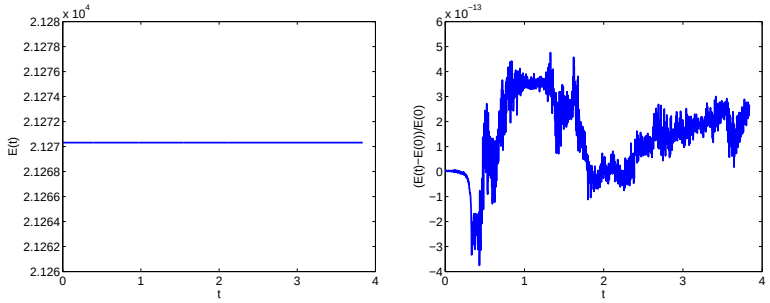

Figure 10: Two-dimensional example: The discrete energy $E^{n+\frac{1}{2}}$ (left), defined in (47), and its relative variation $[E^{n+\frac{1}{2}} - E^0]/E^0$ (right) are shown vs. time.

6. Concluding remarks

Starting from the local time-stepping (LTS) methods in [1], we have derived multi-level local time-stepping methods (MLTS) of arbitrarily high order for second-order wave equations. When the elements of the underlying mesh are naturally organized into tiers of "coarse", "fine", "very fine", etc. elements, our MLTS methods apply the same multi-level structure to the time-

stepping, without sacrificing accuracy or explicitness. Hence they permit inside every tier of like-sized elements the use of the appropriate time-step dictated by the local CFL condition. In particular, when the local mesh refinement occupies only a small portion of the entire computational mesh, such as in the vicinity of corners, point sources or material interfaces, our MLTS methods permit to overcome the stringent CFL stability restriction dictated by but a few elements at each level of refinement.

The second-order MLTS_2 method is given by (32), whereas the fourth-order MLTS_4 method is given by (39). Higher-order versions are derived in Section 3.2 . Although the algorithms are formulated recursively, they are nonetheless fully explicit and thus inherently parallel. In particular, any multiplication involving a projection matrix P_i is in fact performed elementwise; thus, it only affects the i-th sub-tier of like-sized elements and those elements right next to it.

We have proved that the MLTS_2 method conserves a discrete energy and hence is stable for Δt sufficiently small. Our numerical results also indicate that the resulting CFL condition is optimal in the sense that it corresponds to the minimal CFL condition at the i-th level multiplied by the mesh refinement ratio p_i. Moreover, our numerical experiments demonstrate that the $MLTS_4$ method also conserves the energy in (47) down to machine precision. Again, the MLTS_4 method achieves an optimal (global) CFL condition, even when the smallest elements are up to 160 times smaller than those at the coarsest level. Both yield the expected, optimal space-time convergence rates when combined with an appropriate (P^1 or P^3) finite element spatial discretization.

Our MLTS methods also apply to other second-order (vector) wave equations, such as in electromagnetics or elasticity, as long as the underlying semi-discrete formulation coincides with (6) with a (block-)diagonal mass matrix and a sparse symmetric stiffness matrix.

Appendix A. Proof of Lemma 4.1

Let z_1 denote the value of y_{new} prior to the loop in Algo. 1 and z_{m+1} the value of y_{new} after m iterations. We first show by induction over m that z_m satisfies

$$z_m = \tilde{z} + \frac{\Delta t^2}{p_1^2 p_2^2 \cdots p_{N-1}^2} \left(\frac{m^2}{2\, p_N^2} I + \sum_{i=1}^{m-1} \gamma_{m,i} \frac{\Delta t^{2i}}{p_1^{2i} p_2^{2i} \cdots p_{N-1}^{2i}} (AP_N)^i \right) (w - AP_N \tilde{z}), \qquad m \geq 1, \tag{A.1}$$

where the constants $\gamma_{m,i} \in \mathbb{R}$ are independent of Δt.

For $m = 1$, we clearly have

$$z_1 = \tilde{z} + \frac{1}{2}\frac{\Delta t^2}{p_1^2 p_2^2 \cdots p_{N-1}^2 p_N^2}\,(w - AP_N\tilde{z})\,,$$

which corresponds to (A.1) with an empty sum.

For $m = 2$, we infer following Algo. 1 that

$$z_2 = \tilde{z} + 2\frac{\Delta t^2}{p_1^2 p_2^2 \cdots p_{N-1}^2 p_N^2}(w - AP_N\tilde{z}) - \frac{1}{2}\frac{\Delta t^4}{p_1^2 p_2^4 \cdots p_{N-1}^4 p_N^4} AP_N(w - AP_N\tilde{z}),$$

which corresponds to (A.1) with $\gamma_{2,1} = -1/(2p_N^4)$.

Next, we proceed by induction and assume that (A.1) holds for $m-1$ and m. From Algo. 1, we have for $m \geq 2$ that

$$z_{m+1} = 2z_m - z_{m-1} + \frac{\Delta t^2}{p_1^2 p_2^2 \cdots p_{N-1}^2 p_N^2}\,(w - AP_N z_m)\,.$$

By using the induction hypothesis, we now replace z_{m-1} and z_m in the above by (A.1) to obtain

$$z_{m+1} = \tilde{z} + \frac{\Delta t^2}{p_1^2 p_2^2 \cdots p_{N-1}^2}\left(\frac{(m+1)^2}{2\,p_N^2} I + \sum_{i=1}^{m} \gamma_{m+1,i}\frac{\Delta t^{2i}}{p_1^{2i} p_2^{2i} \cdots p_{N-1}^{2i}}(AP_N)^i\right)(w - AP_N)\,\tilde{z},$$

where the constants $\gamma_{m+1,i}$, $i = 1, \ldots, m+1$ are determined recursively through $\gamma_{m,i}$ and $\gamma_{m-1,i}$. Hence (A.1) holds for arbitrary $m \geq 1$. Finally, we set $m = p_N$ in (A.1) and let $\beta_{i,N} = \gamma_{p_N,i}$, which concludes the proof.

Appendix B. Proof of Lemma 4.2

We consider Algo. 4 with $l = N - 1$ and let z_1 denote the value of y_{new} prior to the loop and z_{m+1} denote the value of y_{new} after m iterations. We first show by induction over m that z_m satisfies

$$z_m = \tilde{z} + \left(\sum_{i=0}^{m-1} \gamma_{m,i} B_{p_N}(AP_{N-1}B_{p_N})^i\right)(w - AP_{N-1}\tilde{z}), \qquad m \geq 1, \tag{B.1}$$

where the constants $\gamma_{m,i} \in \mathbb{R}$ are independent of Δt and $\gamma_{m,0} = m^2$.

For $m = 1$, we have

$$z_1 = \mathrm{MLTS}_2\left(\tilde{z}, w - A(P_{N-1} - P_N)\,\tilde{z}, \frac{\Delta t}{p_1 p_2 \cdots p_{N-1}}, N\right),$$

so that

$$z_1 = \mathrm{LTS}_2\left(\tilde{z}, w - A(P_{N-1} - P_N)\,\tilde{z}, \frac{\Delta t}{p_1 p_2 \cdots p_{N-1}}, P_N, p_N\right).$$

By using Lemma 4.1, we find that

$$z_1 = \tilde{z} + B_{p_N}(w - A(P_{N-1} - P_N)\tilde{z} - AP_N\tilde{z}) = \tilde{z} + B_{p_N}(w - AP_{N-1}\tilde{z}), \tag{B.2}$$

For $m = 2$, we infer following Algo. 4 that

$$z_2 = -\tilde{z} + 2\,\mathrm{MLTS}_2\left(z_1, w - A(P_{N-1} - P_N)\,z_1, \frac{\Delta t}{p_1 p_2 \cdots p_{N-1}}, N\right),$$

so that

$$z_2 = -\tilde{z} + 2\,\mathrm{LTS}_2\left(z_1, w - A(P_{N-1} - P_N)\,z_1, \frac{\Delta t}{p_1 p_2 \cdots p_{N-1}}, P_N, p_N\right).$$

Again by using Lemma 4.1, we have

$$z_2 = -\tilde{z} + 2\left(z_1 + B_{p_N}(w - A(P_{N-1} - P_N)z_1 - AP_N z_1)\right).$$

We now use (B.2) to replace z_1 and thus obtain

$$z_2 = \tilde{z} + 4B_{p_N}(w - AP_{N-1}\tilde{z}) - 2B_{p_N}AP_{N-1}B_{p_N}(w - AP_{N-1}\tilde{z}),$$

which corresponds to (B.1) with $m = 2$ and $\gamma_{2,0} = 4$.

Next, we proceed by induction and assume that (B.1) holds for $m - 1$ and m. From Algo. 4, we have for $m \geq 2$ that

$$z_{m+1} = -z_{m-1} + 2\,\mathrm{LTS}_2\left(z_m, w - A(P_{N-1} - P_N)z_m, \frac{\Delta t}{p_1 p_2 \cdots p_{N-1}}, P_N, p_N\right),$$

where z_{m-1}, z_m and z_{m+1} correspond to y_{old}, y_{inter} and y_{new}, respectively. We now use (B.2) to replace z_{m-1} and z_m, which after some calculations yields

$$z_{m+1} = \tilde{z} + \left(\sum_{i=0}^{m} \gamma_{m+1,i} B_{p_N}(AP_{N-1}B_{p_N})^i\right)(w - AP_{N-1}\tilde{z}), \tag{B.3}$$

where the constants $\gamma_{m+1,i}$ are determined recursively through $\gamma_{m,i}$ and $\gamma_{m-1,i}$. In particular, we have

$$\gamma_{m+1,0} = 2\gamma_{m,0} - \gamma_{m-1,0} + 2 = 2m^2 - (m-1)^2 + 2 = (m+1)^2.$$

Hence (B.2) holds for arbitrary $m \geq 1$. Finally, we set $m = p_{N-1}$ in (B.2) and let $\beta_{i,N-1} = \gamma_{p_{N-1},i}$, which concludes the proof.

References

[1] J. Diaz, M. J. Grote, Energy conserving explicit local time-stepping for second-order wave equations, SIAM Journal on Scientific Computing 31 (3) (2009) 1985–2014. doi:10.1137/070709414.

[2] F. Müller, C. Schwab, Finite Elements with mesh refinement for wave equations in polygons, Tech. rep., Seminar for Applied Mathematics, ETH Zurich (April 2013).
URL `http://www.sam.math.ethz.ch/sam_reports/index.php?id=2013-11`

[3] S. Osher, R. Sanders, Numerical approximations to nonlinear conservation laws with locally varying time and space grids, Mathematics of computation 41 (164) (1983) 321–336.

[4] U. M. Ascher, S. J. Ruuth, B. T. Wetton, Implicit-explicit methods for time-dependent partial differential equations, SIAM Journal on Numerical Analysis 32 (3) (1995) 797–823.

[5] A. Kanevsky, M. H. Carpenter, D. Gottlieb, J. S. Hesthaven, Application of implicit–explicit high order Runge–Kutta methods to discontinuous-Galerkin schemes, Journal of Computational Physics 225 (2) (2007) 1753–1781.

[6] V. Dolean, H. Fahs, L. Fezoui, S. Lanteri, Locally implicit discontinuous Galerkin method for time domain electromagnetics, Journal of Computational Physics 229 (2) (2010) 512–526.

[7] S. Descombes, S. Lanteri, L. Moya, Locally implicit time integration strategies in a discontinuous Galerkin method for Maxwell's equations, Journal of Scientific Computing 56 (1) (2013) 190–218.

[8] J. Verwer, Component splitting for semi-discrete Maxwell equations, BIT Numerical Mathematics 51 (2) (2011) 427–445.

[9] C. W. Gear, D. R. Wells, Multirate linear multistep methods, BIT Numerical Mathematics 24 (4) (1984) 484–502.

[10] M. J. Grote, T. Mitkova, High-order explicit local time-stepping methods for damped wave equations, Journal of Computational and Applied Mathematics 239 (2013) 270–289.

[11] M. Hochbruck, A. Ostermann, Exponential multistep methods of Adams-type, BIT Numerical Mathematics 51 (4) (2011) 889–908.

[12] M. J. Grote, M. Mehlin, T. Mitkova, Runge-Kutta based explicit local time-stepping, Tech. rep., Institute of Mathematics, University of Basel (February 2014).
URL `https://math.unibas.ch/fileadmin/mathe/redaktion/Preprints/14_alle/Preprint-1405.pdf`

[13] F. Collino, T. Fouquet, P. Joly, A conservative space-time mesh refinement method for the 1-D wave equation. I. Construction, Numer. Math. 95 (2) (2003) 197–221.

[14] F. Collino, T. Fouquet, P. Joly, A conservative space-time mesh refinement method for the 1-D wave equation. II. Analysis, Numer. Math. 95 (2) (2003) 223–251.

[15] P. Joly, J. Rodríguez, An error analysis of conservative space-time mesh refinement methods for the one-dimensional wave equation, SIAM J. Numer. Anal. 43 (2) (2005) 825–859.

[16] E. Bécache, P. Joly, J. Rodríguez, Space-time mesh refinement for elastodynamics. Numerical results, Comput. Methods Appl. Mech. Engrg. 194 (2-5) (2005) 355–366.

[17] F. Collino, T. Fouquet, P. Joly, Conservative space-time mesh refinement methods for the FDTD solution of Maxwell's equations, J. Comput. Phys. 211 (1) (2006) 9–35.

[18] S. Piperno, Symplectic local time-stepping in non-dissipative DGTD methods applied to wave propagation problems, Modél. Math. Anal. Numér. 40 (5) (2006) 815–841.

[19] M. J. Grote, T. Mitkova, Explicit local time-stepping methods for Maxwell's equations., J. Computational Applied Mathematics 234 (12) (2010) 3283–3302.

[20] C. Baldassari, H. Barucq, H. Calandra, J. Diaz, Numerical performances of a hybrid local-time stepping strategy applied to the reverse time migration, Geophysical Prospecting 59 (5) (2011) 907–919.

[21] S. Minisini, E. Zhebel, A. Kononov, W. A. Mulder, Local time stepping with the discontinuous Galerkin method for wave propagation in 3D heterogeneous media, Geophysics 78 (3) (2013) T67–T77.

[22] G. C. Cohen, Higher-order numerical methods for transient wave equations, Scientific Computation, Springer-Verlag, Berlin, 2002.

[23] G. C. Cohen, P. Joly, J. E. Roberts, N. Tordjman, Higher order triangular finite elements with mass lumping for the wave equation, SIAM J. Numer. Anal. 38 (2001) 2047–2078.

[24] W. A. Mulder, Higher-order mass-lumped finite elements for the wave equation, J. Comput. Acoust. 09 (2001) 671–680.

[25] G. A. Baker, V. A. Dougalis, The effect of quadrature errors on finite element approximations for second order hyperbolic equations, SIAM J. Numer. Anal. 13 (4) (1976) 577–598.

[26] C. Canuto, M. Y. Hussaini, A. Quarteroni, T. A. Zang, Spectral Methods: Fundamentals in Single Domains, Springer–Verlag, New York, 2006.

[27] M. J. Grote, A. Schneebeli, D. Schötzau, Discontinuous Galerkin finite element method for the wave equation, SIAM J. Numer. Anal. 44 (6) (2006) 2408–2431.

[28] M. A. Dablain, High order differencing for the scalar wave equation, SEG Technical Program Expanded Abstracts 3 (1) (1984) 854–854. doi:10.1190/1.1894340.

[29] G. R. Shubin, J. B. Bell, A modified equation approach to constructing fourth-order methods for acoustic wave propagation, SIAM J. Sci. Statist. Comput. 8 (2) (1987) 135–151.

[30] J.-C. Gilbert, P. Joly, Higher order time stepping for second order hyperbolic problems and optimal CFL conditions, in: Partial Differential Equations, Springer, 2008, pp. 67–93.

[31] J. R. Shewchuk, Triangle: Engineering a 2d quality mesh generator and delaunay triangulator, in: Applied computational geometry towards geometric engineering, Springer, 1996, pp. 203–222.

[32] A. Henderson, The ParaView Guide, A Parallel Visualization Application., Kitware Inc., 2007.
URL {http://www.paraview.org}

LATEST PREPRINTS

No.	Author: *Title*
2014-01	**Helmut Harbrecht, Michael Peters, Markus Siebenmorgen** *Efficient Approximation of Random Fields for Numerical Applications*
2014-02	**Ali Hyder, Luca Martinazzi** *Conformal Metrics on R^{2m} with Constant Q-Curvature, Prescribed Volume and Asymptotic Behavior*
2014-03	**Jürgen Dölz, Helmut Harbrecht, and Michael Peters** *H-Matrix Accelerated Second Moment Analysis for Potentials with Rough Correlation*
2014-04	**Tianling Jin, Ali Maalaoui, Luca Martinazzi, Jingang Xiong** *Existence and Asymptotics for Solutions of a Non-Local Q-Curvature Equation in Dimension Three*
2014-05	**Marcus J. Grote, Michaela Mehlin, Teodora Mitkova** *Runge-Kutta Based Explicit Local Time-Stepping Methods for Wave Propagation*
2014-06	**Hanspeter Kraft, Andriy Regeta** *Automorphisms of the Lie Algebra of Vector Fields on Affine n-Space*
2014-07	**Jérémy Blanc, Alberto Calabri** *On Degenerations of Plane Cremona Transformations*
2014-08	**Helmut Harbrecht, Michael Peters, Markus Siebenmorgen** *Numerical Solution of Elliptic Diffusion Problems on Random Domains*
2014-09	**H. Harbrecht, W.L. Wendland, and N. Zorii** *Rapid Solution of Minimal Riesz Energy Problems*
2014-10	**Julien Diaz, Marcus J. Grote** *Multi-Level Explicit Local Time-Stepping Methods for Second-Order Wave Equations*