Energy Conserving Explicit Local Time-Stepping for Second-Order Wave Equations

Julien Diaz, Marcus J. Grote

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Energy Conserving Explicit Local Time-Stepping for Second-Order Wave Equations

Julien Diaz — Marcus J. Grote

N° ????

December 2007

Thème NUM

ISSN 0249-6399 ISRN INRIA/RR--????--FR+ENG

INRIA
FUTURS

Energy Conserving Explicit Local Time-Stepping for Second-Order Wave Equations

Julien Diaz * , Marcus J. Grote †

Thème NUM — Systèmes numériques
Projet Magique-3D

Rapport de recherche n° ???? — December 2007 — 34 pages

Abstract: Locally refined meshes impose severe stability constraints on explicit time-stepping methods for the numerical simulation of time dependent wave phenomena. To overcome that stability restriction, local time-stepping methods are developed, which allow arbitrarily small time-steps precisely where small elements in the mesh are located. When combined with a symmetric finite element discretization in space with an essentially diagonal mass matrix, the resulting discrete numerical scheme is explicit, inherently parallel, and exactly conserves a discrete energy. Starting from the standard second-order "leap-frog" scheme, time-stepping methods of arbitrary order of accuracy are derived. Numerical experiments illustrate the efficiency and usefulness of these methods and validate the theory.

Key-words: Second-order hyperbolic problems, explicit methods, time reversible methods, energy conservation, discontinuous Galerkin methods, finite element methods, mass lumping, wave equation, acoustic waves, electromagnetic waves, elastic waves

* Magique-3D, LMA Pau, I.P.R.A - Université de Pau et des Pays de l'Adour avenue de l'Université BP 1155-64013 PAU CEDEX

† Department of Mathematics, University of Basel, Rheinsprung 21, 4051 Basel, Switzerland, email: Marcus.Grote@unibas.ch.

Unité de recherche INRIA Futurs
Parc Club Orsay Université, ZAC des Vignes,
4, rue Jacques Monod, 91893 ORSAY Cedex (France)
Téléphone : +33 1 72 92 59 00 — Télécopie : +33 1 60 19 66 08

Schéma à pas de temps local explicite et conservatif pour les équations d'ondes

Résumé : Les maillages raffinés localement imposent des fortes contraintes de stabilité aux méthodes explicites pour la résolution de problèmes hyperboliques. Pour surmonter cette difficulté, nous développons des méthodes de résolution permettant d'utiliser des petits pas de temps uniquement sur les éléments fins du maillage. Associée à une discrétisation en espace par éléments finis permettant d'obtenir une matrice de masse diagonale, cette méthode conduit à un schéma numérique explicite, parallélisable, et qui conserve une énergie discrète. À partir du schéma "saute-mouton" classique, nous construisons des méthodes à pas de temps local d'ordre arbitrairement élevé. Nous mettons en évidence l'utilité et l'efficacité de ces méthodes et nous validons nos résultats numériques au travers de résultats numériques.

Mots-clés : Problèmes hyperboliques du second ordre, méthodes explicites, méthodes réversibles en temps, conservation d'énergie, méthodes de Galerkine discontinu, méthodes d'éléments finis, condensation de masse, ondes acoustiques, ondes électromagnétiques, ondes élastiques.

1 Introduction

The efficient and accurate numerical solution of the wave equation is of fundamental importance for the simulation of time dependent acoustic, electromagnetic, or elastic wave phenomena. Finite difference methods are commonly used for the simulation of time dependent waves because of their simplicity and their efficiency on structured cartesian meshes [25, 26, 32]. However, in the presence of complex geometry or small geometric features that require locally refined meshes, their usefulness is somewhat limited. In contrast, finite element methods (FEMs) easily handle locally refined unstructured meshes; moreover, their extension to high order is straightforward, even in the presence of curved boundaries or material interfaces.

The finite element Galerkin discretization of second-order hyperbolic problems typically leads to a second-order system of ordinary differential equations. Even if explicit time-stepping is employed, the mass matrix arising from the spatial discretization by standard continuous finite elements must be inverted at each time-step: a major drawback in terms of efficiency. To overcome that problem, various "mass lumping" techniques have been proposed, which effectively replace the mass matrix by a diagonal approximation. While straightforward for piecewise linear elements [8, 30], mass lumping techniques require special quadrature rules at higher order to preserve the accuracy and guarantee numerical stability [12].

Alternatively, discontinuous Galerkin (DG) methods offer even greater flexibility for local mesh refinement by accomodating non-conforming grids and hanging nodes. Based on discontinuous finite element spaces, DG-FEMs weakly enforce continuity by adding suitable bilinear forms, so-called numerical fluxes, to standard variational formulations – see [9, 10, 11] for further details and recent reviews. Because individual elements decouple, the mass matrix arising from a spatial DG discretization is block-diagonal, with block size equal to the number of degrees of freedom per element; it can therefore be inverted at very low computational cost. In fact, for a judicious choice of (locally orthogonal) shape functions, the mass matrix is diagonal. Thus, when combined with explicit time integration, the resulting time marching scheme will be truly explicit.

Recently, Grote, Schneebeli and Schötzau [22] proposed the *symmetric* interior penalty (IP) DG method for the second-order wave equation; in particular, they derived optimal a priori error bounds in the energy norm and the L^2-norm for the semi-discrete formulation. A symmetric DG formulation of the wave equation in its second-order form offers the following advantage, which also pertains to the classical continuous Galerkin formulation. Since the stiffness matrix is positive semidefinite, the semi-discrete formulation inherently conserves (a discrete version of) the energy for all time. Moreover, when combined with a symmetric time marching scheme, such as the standard leap-frog (or Störmer-Verlet) method, the resulting fully discrete formulation will also conserve a discrete energy. Thus, both formulations will be free of any (unnecessary) damping. The dispersive properties of the symmetric IP-DG method were analyzed by Ainsworth, Monk and Muniz [1]. In [23, 24] the symmetric interior DG method was extended to Maxwell's Equations in second-order form.

Adaptivity and mesh refinement are certainly key for the efficient numerical solution of partial differential equations. However, locally refined meshes impose severe stability constraints on explicit time-stepping schemes, where the maximal time-step allowed by the CFL condition is dictated by the smallest elements in the mesh. When mesh refinement is restricted to a small region, the use of implicit methods, or a very small time-step in the entire computational domain, are very high a price to pay. To overcome that stability restriction, various local time-stepping schemes were proposed, which allow smaller time-steps precisely where

the smallest elements in the mesh are located. In [14] Collino, Fouquet and Joly proposed a local time-stepping method for the first order wave equation; it was analyzed in [15, 31] and extended to elastodynamics [6] and Maxwell's equations [16]. Their approach, which is based on the introdution of a Lagrange multiplier, conserves a discrete energy. However, it requires the solution of a linear system on the interface between the coarse and the fine mesh. By combining a symplectic integrator with a DG discretization of Maxwell's equations in first-order form, Piperno [37] proposed an explicit local time-stepping scheme, which also conserves a discrete energy. All these methods are second-order accurate in time. Alternatively, domain decomposition methods permit the use of different numerical methods or time steps in separate subdomains [20, 28].

The outline of our paper is as follows. In Section 2 we recall both the standard continuous and the symmetric IP-DG finite element discretizations of the scalar second-order wave equation, which typically serves as a model problem for general second-order hyperbolic problems. Starting from the well-known "leap-frog" scheme, we then derive a second-order a local time-stepping scheme in Section 3. With a symmetric finite element discretization in space with a (block-)diagonal mass matrix, the resulting fully discrete scheme is not only explicit and thus inherently parallel, but it also conserves (a discrete version of) the energy. We also show via numerical experiments how a small overlap between the fine and the coarse region achieves an optimal CFL condition. Next, in Section 4, we extend the second-order local time-stepping schemes first to fourth order, and then to arbitrarily high order of accuracy. In Section 5 we present numerical experiments in one and two space dimensions which validate the theory and illustrate the usefulness of these local time-stepping schemes.

2 Finite element discretizations of the wave equation

We consider the scalar wave equation

$$u_{tt} - \nabla \cdot \left(c^2 \, \nabla u \right) = f \qquad \text{in } (0,T) \times \Omega, \tag{1}$$

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

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

$$u_t|_{t=0} = v_0 \qquad \text{in } \Omega, \tag{4}$$

where Ω is a bounded domain in $\mathbb{R}^2$ or $\mathbb{R}^3$. Here $f \in L^2(0,T;L^2(\Omega))$ is a (known) source term, while $u_0 \in H_0^1(\Omega)$ and $v_0 \in L^2(\Omega)$ are prescribed initial conditions. We consider homogeneous Dirichlet boundary conditions, for simplicity, and assume that the speed of propagation, $c(x)$, is piecewise smooth and strictly positive. In the absence of forcing, the (continuous) *energy*,

$$E[u](t) = \frac{1}{2} \left\{ \|u_t(t,.)\|^2 + \|\nabla u(t,.)\|^2 \right\}, \qquad t \geq 0, \tag{5}$$

is conserved for all time.

We shall now discretize in space (1)–(4) by using either standard continuous (H^1-conforming) finite elements (with mass lumping) or a symmetric IP discontinuous Galerkin discretization from [22], while leaving time continuous. Thus, we consider shape-regular meshes $\mathcal{T}_h$ that partition the domain Ω into disjoint elements $\{K\}$, such that $\overline{\Omega} = \cup_{K \in \mathcal{T}_h} \overline{K}$. The elements are triangles or quadrilaterals in two space dimensions, and tetrahedra or hexaedra in three dimensions, respectively. The diameter of element K is denoted by h_K and the mesh size, h, is given by $h = \max_{K \in \mathcal{T}_h} h_K$.

2.1 Standard continuous Galerkin formulation

The standard continuous (H^1-conforming) Galerkin formulation of the wave equation (1)–(4) starts from its weak formulation [34]: find $u : [0,T] \times H_0^1(\Omega) \to \mathbb{R}$ such that

$$(u_{tt}, v) + (c\nabla u, c\nabla v) = (f, v) \qquad \forall v \in H_0^1(\Omega), \quad t \in (0,T), \tag{6}$$

$$u|_{t=0} = u_0, \tag{7}$$

$$u_t|_{t=0} = v_0. \tag{8}$$

Here $(.,.)$ denotes the standard L^2 inner product on Ω.

For a given partition $\mathcal{T}_h$ of Ω and an approximation order $\ell \geq 1$, we wish to approximate the solution $u(t,\cdot)$ of (6)–(8) in the finite element space

$$V^h := \{v \in H_0^1(\Omega) : v|_K \circ F_K \in \mathcal{S}^\ell(\hat{K}) \ \ \forall K \in \mathcal{T}_h\}, \tag{9}$$

where $\mathcal{S}^\ell(\hat{K})$ is the space $\mathcal{P}^\ell(\hat{K})$ (for triangles or tetrahedra) or $\mathcal{Q}^\ell(\hat{K})$ (for quadrilaterals or hexahedra) and $F_K : \hat{K} \to K$ is one-to-one and maps the boundary of the reference element, $\hat{K}$, to the boundary of K. Thus, we consider the following semi-discrete Galerkin approximation of (6)–(8): find $u^h : [0,T] \times V^h \to \mathbb{R}$ such that

$$(u^h_{tt}, v) + (c\nabla u^h, c\nabla v) = (f, v) \quad \forall v \in V^h, \quad t \in (0,T), \tag{10}$$

$$u^h|_{t=0} = \Pi_h u_0, \tag{11}$$

$$u^h_t|_{t=0} = \Pi_h v_0, \tag{12}$$

where Π_h denotes the L^2-projection onto V^h – see [4] for further details.

Let $(\phi_i)_{i=1..N}$ denote the standard nodal basis of V^h defined by $\phi_i(x_j) = \delta_{ij}$, where $(x_i)_{i=1..N}$ are the nodes of the mesh. Next, we denote by $y(t)$, $F(t)$, y_0, and $\dot{y}_0$ the N-vectors

$$y_i(t) = (u_h(t,.), \phi_i), \quad F_i(t) = (f(t,.), \phi_i), \quad y_{0,i} = (\Pi_h u_0, \phi_i), \quad \dot{y}_{0,i} = (\Pi_h v_0, \phi_i).$$

Then (10)–(12) is equivalent to the second-order system of ordinary differential equations

$$M\frac{d^2y}{dt^2} + Ky = F \tag{13}$$

$$y(0) = y_0, \quad \frac{dy}{dt}(0) = \dot{y}_0, \tag{14}$$

where the $N \times N$ mass and stiffness matrices, M and K, are defined by

$$M_{ij} = (\phi_i, \phi_j), \qquad K_{ij} = (c\nabla\phi_i, c\nabla\phi_j)),$$

respectively. The matrix M is symmetric positive definite whereas the matrix K is symmetric and, in general, positive semidefinite only.

Since the matrix M is not diagonal, it must be inverted at every time-step of any explicit time integration scheme. To overcome this difficulty, various so-called mass lumping techniques have been developed, which essentially replace M with a diagonal approximation by computing the integrals over each element K with judicious quadrature rules. For $\mathcal{P}^1$-elements in two space dimensions, for instance, one uses the approximation

$$\int_K f \simeq \frac{|K|}{3}\sum_{k=1}^{3} f(x_K^k),$$

where x_K^k are the vertices of triangle K. From the definition of the basis functions ϕ_i, it then immediately follows that

$$\int_K \phi_i \phi_j \simeq \frac{|K|}{3}\delta_{ij}.$$

Hence, M is now diagonal while the spatial discretization remains second-order accurate [8, 30].

For $\mathcal{P}^\ell$-elements up to order $\ell = 3$, mass lumping techniques are also available but more complicated [12], whereas the case of $\mathcal{Q}^\ell$ (or spectral)-elements for quadrilateral meshes and arbitrary ℓ, is well-understood [13, 29, 36, 38]. Here, the degrees of freedom associated with the basis functions (ϕ_i) coincide with the Gauss-Lobatto quadrature points on each elements. As the integrals are also computed elementwise through Gauss-Lobatto quadrature, the resulting mass matrix is diagonal while the spatial accuracy is not affected [4, 5].

2.2 Discontinuous Galerkin formulation

Here we briefly recall the symmetric interior penalty DG fomulation from [22]. For simplicity, we assume in this section that the elements are triangles or parallelograms in two space dimensions, and tetrahedra or parallelepipeds in three dimensions, respectively. Generally, we allow for irregular meshes with hanging nodes. An interior face of $\mathcal{T}_h$ is the (nonempty) interior of $\partial K^+ \cap \partial K^-$, where K^+ and K^- are two adjacent elements of $\mathcal{T}_h$. Similarly, a boundary face of $\mathcal{T}_h$ is the (nonempty) interior of $\partial K \cap \partial\Omega$, which consists of entire faces of ∂K. We denote by $\mathcal{F}_h^{\mathcal{I}}$ the set of all interior faces of $\mathcal{T}_h$, by $\mathcal{F}_h^{\mathcal{B}}$ the set of all boundary faces, and let $\mathcal{F}_h = \mathcal{F}_h^{\mathcal{I}} \cup \mathcal{F}_h^{\mathcal{B}}$. Here we generically refer to any element of $\mathcal{F}_h$ as a "face", both in two and in three dimensions.

For any piecewise smooth function v we now introduce the following trace operators. Let $F \in \mathcal{F}_h^{\mathcal{I}}$ be an interior face shared by two neighboring elements K^+ and K^- and let $x \in F$; we write $\mathbf{n}^\pm$ to denote the outward unit normal vectors on the boundaries $\partial K^\pm$. Denoting by $v^\pm$ the trace of v taken from within $K^\pm$, we define the jump and average of v at $x \in F$ by

$$[\![v]\!] := v^+\mathbf{n}^+ + v^-\mathbf{n}^-, \qquad \{\!\{v\}\!\} := (v^+ + v^-)/2,$$

respectively. On every boundary face $F \in \mathcal{F}_h^{\mathcal{B}}$, we set $[\![v]\!] := v\mathbf{n}$ and $\{\!\{v\}\!\} := v$. Here, $\mathbf{n}$ is the outward unit normal on $\partial\Omega$.

For a piecewise smooth vector-valued function $\mathbf{q}$, we analogously define the average across interior faces by $\{\!\{\mathbf{q}\}\!\} := (\mathbf{q}^+ + \mathbf{q}^-)/2$, and on boundary faces we set $\{\!\{\mathbf{q}\}\!\} := \mathbf{q}$. The jump of a vector-valued function will not be used. For a vector-valued function $\mathbf{q}$ with *continuous* normal components across a face f, the trace identity

$$v^+(\mathbf{n}^+ \cdot \mathbf{q}^+) + v^-(\mathbf{n}^- \cdot \mathbf{q}^-) = [\![v]\!] \cdot \{\!\{\mathbf{q}\}\!\} \quad \text{on } f,$$

immediately follows from the above definitions.

For a given partition $\mathcal{T}_h$ of Ω and an approximation order $\ell \geq 1$, we wish to approximate the solution $u(t,\cdot)$ of (1)–(4) in the finite element space

$$V^h := \{v \in L^2(\Omega) : \, v|_K \in \mathcal{S}^\ell(K) \;\; \forall K \in \mathcal{T}_h\}, \tag{15}$$

where $\mathcal{S}^\ell(K)$ is the space $\mathcal{P}^\ell(K)$ of polynomials of total degree at most ℓ on K, if K is a triangle or a tetrahedra, or the space $\mathcal{Q}^\ell(K)$ of polynomials of degree at most ℓ in each

variable on K, if K is a parallelogram or a parallelepiped. Thus, we consider the following (semi-discrete) discontinuous Galerkin approximation of (1)–(4): find $u^h : \overline{J} \times V^h \to \mathbb{R}$ such that

$$(u^h_{tt}, v) + a_h(u^h, v) = (f, v) \qquad \forall v \in V^h, \quad t \in J, \tag{16}$$

$$u^h|_{t=0} = \Pi_h u_0, \tag{17}$$

$$u^h_t|_{t=0} = \Pi_h v_0. \tag{18}$$

Here, Π_h denotes the L^2-projection onto V^h and the discrete bilinear form a_h on $V^h \times V^h$ is given by

$$\begin{aligned} a_h(u,v) := \quad & \sum_{K\in\mathcal{T}_h} \int_K c^2\, \nabla u \cdot \nabla v \, dx - \sum_{F\in\mathcal{F}_h} \int_F [\![u]\!] \cdot \{\!\{c^2\, \nabla v\}\!\}\, dA \\ & - \sum_{F\in\mathcal{F}_h} \int_F [\![v]\!] \cdot \{\!\{c^2\, \nabla u\}\!\}\, dA + \sum_{F\in\mathcal{F}_h} \int_F \mathtt{a}\, [\![u]\!] \cdot [\![v]\!]\, dA. \end{aligned} \tag{19}$$

The last three terms in (19) correspond to jump and flux terms at element boundaries; they vanish when $u, v \in H^1_0(\Omega) \cap H^{1+\sigma}(\Omega)$, for $\sigma > \frac{1}{2}$. Hence the above semi-discrete discontinuous Galerkin formulation (16) is consistent with the original continuous problem (6).

In (19) the function $\mathtt{a}$ penalizes the jumps of u and v over the faces of $\mathcal{T}_h$. It is referred to as interior penalty stabilization function and is defined as follows. We first introduce the function $\mathtt{h}$ by

$$\mathtt{h}|_F = \begin{cases} \min\{h_K, h_{K'}\}, & F \in \mathcal{F}_h^{\mathcal{I}},\ F = \partial K \cap \partial K', \\ h_K, & F \in \mathcal{F}_h^{\mathcal{B}},\ F = \partial K \cap \partial\Omega. \end{cases}$$

For $x \in F$, we further define $\mathtt{c}$ by

$$\mathtt{c}|_F(x) = \begin{cases} \max\{c|_K(x), c|_{K'}(x)\}, & F \in \mathcal{F}_h^{\mathcal{I}},\ F = \partial K \cap \partial K', \\ c|_K(x), & F \in \mathcal{F}_h^{\mathcal{B}},\ F = \partial K \cap \partial\Omega. \end{cases}$$

Then, on each $F \in \mathcal{F}_h$, we set

$$\mathtt{a}|_F := \alpha\, \mathtt{c}^2 \mathtt{h}^{-1}, \tag{20}$$

where α is a positive parameter independent of the local mesh sizes and the coefficient c. There exists a threshold value $\alpha_{\min} > 0$ which only depends on the shape-regularity of the mesh and the approximation order ℓ, such that for $\alpha \geq \alpha_{\min}$ the discontinuous Galerkin bilinear form a_h is coercive and, hence, the discretization stable [3]. Throughout the rest of the paper we shall assume that $\alpha \geq \alpha_{\min}$, so that the semi-discrete problem (16)–(18) has a unique solution which converges with optimal order [22].

The semi-discrete IP-DG formulation (16)–(18) is equivalent to the second-order system of ordinary differential equations

$$M \frac{d^2 y}{dt^2} + Ky = F \tag{21}$$

$$y(0) = y_0, \qquad \frac{dy}{dt}(0) = \dot{y}_0. \tag{22}$$

The $N \times N$ mass matrix M, with entries $M_{ij} = (\phi_i, \phi_j)$, again is symmetric positive definite. Yet because individual elements decouple, M is also block-diagonal, with block size equal

to the number of degrees of freedom per element. Thus, it can be inverted at very low computational cost. In fact, for a judicious choice of (locally orthogonal) shape functions, the mass matrix is diagonal and therefore the resulting time marching scheme truly explicit.

Remark 1. *Because the matrices M and K are symmetic, either for the symmetric IP-DG or the standard continuous Galerkin discretization, the two semi-discrete formulations (13)–(14) and (21)-(22) conserve the (discrete) energy*

$$E_h(t) := \frac{1}{2}\left\{\left\langle My'(t)y'(t)\right\rangle + \langle Ky(t), y(t)\rangle\right\}.$$

when $F = 0$. Here the angular brackets denote the standard Euclidean inner product on $\mathbb{R}^N$. If the underlying bilinear form a is strictly coercive, so are the two discrete bilinear forms a_h, either for the discontinuous Galerkin discretization with $\alpha \geq \alpha_{\min}$ or for the continuous Galerkin discretization with (or without) mass lumping. For $f = 0$, conservation of energy then implies that the solution remains bounded for all time. However, when the underlying bilinear form a is not (strictly) coercive, as in the presence of Neumann or periodic boundary conditions, for instance, the elliptic partial differential operator will have a (simple) zero eigenvalue (with constant eigenfunction). Then, the wave equation admits linear growth in time, although the (continuous or discrete) energy remains constant. Nonetheless, if the numerical method is consistent, stability (with respect to the energy) will imply convergence on any finite time interval. Moreover, since all time integration schemes considered here are at least second-order accurate, the numerical time integration of the linearly growing zeroth eigenmode will actually be exact.

3 Local time-stepping

We consider the semi-discrete wave equation,

$$M\frac{d^2y}{dt^2} + Ky = 0, \tag{23}$$

where M is an $N \times N$ symmetric positive definite (sparse) matrix and K is an $N \times N$ symmetric positive semidefinite (sparse) matrix. Moreover, we assume that $M^{\frac{1}{2}}$ can be explicitly computed and inverted at low cost, as in the case when M is (block-) diagonal, for instance. Next, we multiply (23) by $M^{-\frac{1}{2}}$ to obtain

$$\frac{d^2z}{dt^2} + M^{-\frac{1}{2}}KM^{-\frac{1}{2}}z = 0, \tag{24}$$

with

Let A denote the matrix $M^{-\frac{1}{2}}KM^{-\frac{1}{2}}$, which is also sparse and symmetric positive semidefinite. Then, we rewrite (24) as

$$\frac{d^2z}{dt^2} + Az = 0\ . \tag{25}$$

For any $f \in \mathcal{C}^1$ we have

$$f(t+\Delta t) - 2f(t) + f(t-\Delta t) = -\Delta t^2 \int_{-1}^{1} (1-|\theta|) f''(t+\theta\Delta t)\, d\theta. \tag{26}$$

Hence the (exact) solution $z(t)$ of (25) satisfies

$$z(t+\Delta t) - 2z(t) + z(t-\Delta t) = -\Delta t^2 \int_{-1}^{1} (1-|\theta|) A z(t+\theta\Delta t)\, d\theta. \tag{27}$$

3.1 Second-order local time-stepping method

The integral on the right side of (27) represents a weighted average of $Az(s)$ over the interval $[t-\Delta t, t+\Delta t]$, which needs to be approximated in any numerical algorithm. For instance, if we simply replace $Az(t+\theta\Delta t)$ by $Az(t)$ in (27) and evaluate the remaining θ-dependent integral, we obtain the well-known second-order leap-frog scheme with time-step Δt,

$$z_{n+1} - 2z_n + z_{n-1} = -\Delta t^2 A z_n, \qquad z_n \simeq z(t_n), \tag{28}$$

which, however, would require Δt to be comparable in size to the smallest elements in the mesh. Instead, we now split the vector $z(t)$ in two parts:

$$z(t) = (I-P)z(t) + Pz(t) = z^{[\text{coarse}]}(t) + z^{[\text{fine}]}(t)\ .$$

The projection 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. To circumvent the severe CFL restriction on Δt in (28), we shall treat $z^{[\text{fine}]}(t)$ differently from $z^{[\text{coarse}]}(t)$ in

$$z(t+\Delta t) - 2z(t) + z(t-\Delta t) = -\Delta t^2 \int_{-1}^{1} (1-|\theta|) \quad A \quad \Big[z^{[\text{coarse}]}(t+\theta\Delta t) + z^{[\text{fine}]}(t+\theta\Delta t) \Big]\, d\theta. \tag{29}$$

Following [27, 33], we approximate the integrand in (29) as

$$A\left[z^{[\text{coarse}]}(t+\theta\Delta t) + z^{[\text{fine}]}(t+\theta\Delta t) \right] \simeq A z^{[\text{coarse}]}(t) + AP\tilde{z}(\theta\Delta t),$$

where $\tilde{z}(\tau)$ solves the differential equation

$$\begin{cases} \dfrac{d^2\tilde{z}}{d\tau^2}(\tau) = -A(I-P)z(t) - AP\tilde{z}(\tau), \\ \tilde{z}(0) = z(t), \qquad \dfrac{d\tilde{z}}{d\tau}(0) = \nu, \end{cases} \tag{30}$$

and ν will be specified below. Thus,

$$z(t+\Delta t) - 2z(t) + z(t-\Delta t) \simeq -\Delta t^2 \int_{-1}^{1} (1-|\theta|) \left[A(I-P)z(t) + AP\tilde{z}(\theta\Delta t) \right] d\theta. \tag{31}$$

Note that A and P do not commute. Since $\tilde{z}$ solves (30), we deduce again from (26) that

$$\tilde{z}(\Delta t) - 2\tilde{z}(0) + \tilde{z}(-\Delta t) = -\Delta t^2 \int_{-1}^{1} (1-|\theta|) \left[A(I-P)z(t) + AP\tilde{z}(\theta\Delta t) \right] d\theta. \tag{32}$$

From the comparison of (31) and (32) we infer that

$$z(t+\Delta t) - 2z(t) + z(t-\Delta t) \simeq \tilde{z}(\Delta t) - 2\tilde{z}(0) + \tilde{z}(-\Delta t). \tag{33}$$

The quantity $\tilde{z}(\Delta t) - 2\tilde{z}(0) + \tilde{z}(-\Delta t)$ does not depend on the value of ν, which we choose equal to zero. Therefore, $\tilde{z}(\Delta t) = \tilde{z}(-\Delta t)$ by symmetry and (33) becomes

$$z(t+\Delta t) - 2z(t) + z(t-\Delta t) \simeq 2(\tilde{z}(\Delta t) - \tilde{z}(0)), \tag{34}$$

or equivalently

$$z(t+\Delta t) + z(t-\Delta t) \simeq 2\tilde{z}(\Delta t). \tag{35}$$

Hence, we shall approximate the right side of (35) by solving (30) on $[0, \Delta t]$, and then use (35) to compute $z(t+\Delta t)$. Since the first term on the right side of (30) *does not depend on* τ, the (high-frequency, oscillatory) evolution of $\tilde{z}(\tau)$ is solely determined by the second term, $AP\tilde{z}(\tau)$, which involves only the unknowns associated with the nonzero entries in P. If those nonzero entries occupy only a small fraction of all unknowns, the additional effort from solving (30) will be small since A is sparse. Clearly, in doing so we must also ensure that the overall numerical scheme remains second-order accurate in time, as we shall show below.

In summary, the local time-stepping algorithm for the solution of (23) computes $z_{n+1} \simeq z(t+\Delta t)$, given z_n and z_{n-1}, as follows:

Algorithm 2. *1. Set* $w = A(I-P)z_n$ *and* $\tilde{z}_0 = z_n$*;*

2. Compute $\tilde{z}_{1/p} = \tilde{z}_0 - \frac{1}{2}\left(\frac{\Delta t}{p}\right)^2 (w + AP\tilde{z}_0)$*;*

3. For $m = 1, \ldots, p-1$*, compute*

$$\tilde{z}_{(m+1)/p} = 2\tilde{z}_{m/p} - \tilde{z}_{(m-1)/p} - \left(\frac{\Delta t}{p}\right)^2 \left(w + AP\tilde{z}_{m/p}\right); \tag{36}$$

4. Compute $z_{n+1} = -z_{n-1} + 2\tilde{z}_1$.

Here Steps 1–3 correspond to the numerical solution of (30) with $\nu = 0$ until $\tau = \Delta t$ using the leap-frog scheme with the local time-step $\Delta\tau = \Delta t/p$. In fact, any other second-order method, either explicit or implicit, could be used there instead. For $P = 0$, that is without any local time-stepping, we have

$$\tilde{z}_1 = \tilde{z}(\Delta t) = \left(I - \frac{\Delta t^2}{2}A\right) z(t)$$

and hence we recover the standard leap-frog scheme (28). If the fraction of nonzero entries in P is small, the overall cost will be dominated by the computation of w, which requires a single multiplication by $A(I-P)$ per time-step Δt. All further matrix-vector multiplications by AP involve only those unknowns that are associated with the smaller, locally refined region. In addition, since A is sparse every update in Step 3 affects only those unknowns that lie inside the refined region, or immediately next to it.

To establish the accuracy and stability of the above local time-stepping scheme we shall now show how to rewrite it in "leap-frog manner". To do so, we first need the following technical result.

Lemma 3. *For $m \geq 2$, $\tilde{z}_{m/p}$ defined by Algorithm 2 satisfies*

$$\tilde{z}_{m/p} = z_n - \frac{m^2}{2}\left(\frac{\Delta t}{p}\right)^2 Az_n + \sum_{j=1}^{m-1}\left(\frac{\Delta t}{p}\right)^{2(j+1)} \alpha_j^m (AP)^j Az_n, \tag{37}$$

where the constants α_j^m are given by

$$\begin{cases} \alpha_1^2 = \dfrac{1}{2}, \quad \alpha_1^3 = 3, \quad \alpha_2^3 = -\dfrac{1}{2}, \\ \alpha_1^{m+1} = \dfrac{m^2}{2} + 2\alpha_1^m - \alpha_1^{m-1}, \\ \alpha_j^{m+1} = 2\alpha_j^m - \alpha_j^{m-1} - \alpha_{j-1}^m, \qquad j = 2...m-2, \\ \alpha_{m-1}^{m+1} = 2\alpha_{m-1}^m - \alpha_{m-2}^m, \\ \alpha_m^{m+1} = -\alpha_{m-1}^m. \end{cases} \tag{38}$$

Proof. The proof is by induction on m.

We first show that (37) holds for $m = 2$. Since

$$\tilde{z}_{1/p} = z_n - \frac{1}{2}\left(\frac{\Delta t}{p}\right)^2 Az_n,$$

we immediately find that

$$\tilde{z}_{2/p} = z_n - 2\left(\frac{\Delta t}{p}\right)^2 Az_n + \frac{1}{2}\left(\frac{\Delta t}{p}\right)^4 APAz_n.$$

Hence, (37) holds with $\alpha_1^2 = 1/2$.

Next, let (37) hold for m. Then,

$$\begin{aligned} \tilde{z}_{(m+1)/p} &= 2\left(z_n - \frac{m^2}{2}\left(\frac{\Delta t}{p}\right)^2 Az_n + \sum_{j=1}^{m-1}\left(\frac{\Delta t}{p}\right)^{2(j+1)} \alpha_j^m (AP)^j Az_n\right) \\ &- \left(z_n - \frac{(m-1)^2}{2}\left(\frac{\Delta t}{p}\right)^2 Az_n + \sum_{j=1}^{m-2}\left(\frac{\Delta t}{p}\right)^{2(j+1)} \alpha_j^{m-1} (AP)^j Az_n\right) \\ &- \left(\frac{\Delta t}{p}\right)^2 \left(A(I-P)z_n + AP\left[z_n - \frac{m^2}{2}\left(\frac{\Delta t}{p}\right)^2 Az_n + \sum_{j=1}^{m-1}\left(\frac{\Delta t}{p}\right)^{2(j+1)} \alpha_j^m (AP)^j Az_n\right]\right), \end{aligned}$$

which after some algebra simplifies to

$$\begin{aligned} \tilde{z}_{(m+1)/p} &= z_n - \left(m^2 - \frac{(m-1)^2}{2} + 1\right)\left(\frac{\Delta t}{p}\right)^2 Az_n + \frac{m^2}{2}\left(\frac{\Delta t}{p}\right)^4 APAz_n \\ &+ 2\sum_{j=1}^{m-1}\left(\frac{\Delta t}{p}\right)^{2(j+1)} \alpha_j^m (AP)^j Az_n - \sum_{j=1}^{m-2}\left(\frac{\Delta t}{p}\right)^{2(j+1)} \alpha_j^{m-1} (AP)^j Az_n \\ &- \sum_{j=1}^{m-1}\left(\frac{\Delta t}{p}\right)^{2(j+2)} \alpha_j^m (AP)^{j+1} Az_n \ . \end{aligned}$$

Rearranging terms we then find

$$\begin{aligned}\tilde{z}_{(m+1)/p} &= z_n - \frac{(m+1)^2}{2}\left(\frac{\Delta t}{p}\right)^2 Az_n + \left(\frac{\Delta t}{p}\right)^4 \left(\frac{m^2}{2} + 2\alpha_1^m - \alpha_1^{m-1}\right) APAz_n \\ &+ \sum_{j=2}^{m-2} \left(\frac{\Delta t}{p}\right)^{2(j+1)} \left(2\alpha_j^m - \alpha_j^{m-1} - \alpha_{j-1}^m\right) (AP)^j Az_n \\ &+ \left(\frac{\Delta t}{p}\right)^{2m} \left(2\alpha_{m-1}^m - \alpha_{m-2}^m\right) (AP)^{m-1} Az_n - \left(\frac{\Delta t}{p}\right)^{2(m+1)} \alpha_{m-1}^m (AP)^m Az_n \ ,\end{aligned}$$

which yields (37) with α_j^m as in (38). □

As a consequence, we can rewrite the above local time-stepping algorithm in "leap-frog manner".

Proposition 4. *The local time-stepping Algorithm 2 is equivalent to*

$$z_{n+1} = 2z_n - z_{n-1} - \Delta t^2 A_p z_n, \tag{39}$$

where A_p is defined by

$$A_p = A - \frac{2}{p^2} \sum_{j=1}^{p-1} \left(\frac{\Delta t}{p}\right)^{2j} \alpha_j^p (AP)^j A \tag{40}$$

and the constants α_j^p are given by (38). This scheme is second-order accurate. Furthermore, the matrix A_p is symmetric.

Proposition 4 is crucial for the accuracy and stability analysis below. However, the actual implementation of the local time-stepping scheme follows Algorithm 2; in particular, neither A_p nor the constants α_j^p are ever used in practice.

Proof. Recall that $z_{n+1} = -z_{n-1} + 2\tilde{z}_1$. We now use (37) with $m = p$ to replace $\tilde{z}_1$. This yields

$$z_{n+1} = -z_{n-1} + 2z_n - \Delta t^2 Az_n + 2\left(\frac{\Delta t}{p}\right)^2 \sum_{j=1}^{p-1} \left(\frac{\Delta t}{p}\right)^{2j} \alpha_j^p (AP)^j Az_n, \tag{41}$$

which corresponds to (39) with A_p as in (40).

To prove consistency we now rewrite (41) as

$$\frac{z_{n+1} - 2z_n + z_{n-1}}{\Delta t^2} - \frac{2}{p^2} \sum_{j=1}^{p-1} \left(\frac{\Delta t}{p}\right)^{2j} \alpha_j^p (AP)^j Az_n = -Az_n.$$

Since

$$\frac{d^2 z}{dt^2}(t) = \frac{z(t+\Delta t) - 2z(t) + z(t-\Delta t)}{\Delta t^2} + O(\Delta t^2)$$

and

$$\sum_{j=1}^{p-1} \left(\frac{\Delta t}{p}\right)^{2j} \alpha_j^p (AP)^j Az(t) = O(\Delta t^2),$$

we conclude that the local time-stepping scheme is second-order accurate in time.

Finally, as the matrices A and P are symmetric, we have

$$((AP)^j A)^\top = A(PA)^j = (AP)^j A, \qquad j \geq 1.$$

Therefore, the matrix A_p is symmetric, too. □

3.2 Energy conservation

The standard leap-frog scheme (28) conserves the discrete energy,

$$E^{n+\frac{1}{2}} = \frac{1}{2}\left[\left\langle \left(I - \frac{\Delta t^2}{4}A\right)\frac{z_{n+1}-z_n}{\Delta t}, \frac{z_{n+1}-z_n}{\Delta t}\right\rangle + \left\langle A\frac{z_{n+1}+z_n}{2}, \frac{z_{n+1}+z_n}{2}\right\rangle\right]. \quad (42)$$

Here $E^{n+\frac{1}{2}} \simeq E(t_{n+\frac{1}{2}})$ (see Remark 3.2) and the angular brackets denote the standard Euclidean inner product on $\mathbb{R}^N$. Since A is symmetric, so is the quadratic form in (42). Moreover, for sufficiently small Δt it is also positive semidefinite and hence yields a true energy. That restriction on Δt corresponds precisely to the CFL stability condition of the leap-frog method and guarantees its numerical stability.

To prove the numerical stability of the local time-stepping algorithm, and thereby determine a necessary and sufficient condition for stability, we shall now exhibit a conserved discrete energy.

Proposition 5. *The second-order local time-stepping scheme (Algorithm (2)) conserves the following energy:*

$$E^{n+\frac{1}{2}} = \frac{1}{2}\left[\left\langle \left(I - \frac{\Delta t^2}{4}A_p\right)\frac{z_{n+1}-z_n}{\Delta t}, \frac{z_{n+1}-z_n}{\Delta t}\right\rangle + \left\langle A_p\frac{z_{n+1}+z_n}{2}, \frac{z_{n+1}+z_n}{2}\right\rangle\right]. \quad (43)$$

This energy corresponds to the energy conserved by the leap-frog scheme with A replaced by A_p.

Proof. First, we take the inner product of (39) with $z_{n+1} - z_{n-1}$ and divide the resulting expression by Δt^2 to obtain

$$\left\langle \frac{z_{n+1}-2z_n+z_{n-1}}{\Delta t}, \frac{z_{n+1}-z_{n-1}}{\Delta t}\right\rangle + \langle A_p z_n, z_{n+1}-z_{n-1}\rangle = 0.$$

By using

$$z_{n+1} - 2z_n + z_{n-1} = (z_{n+1} - z_n) - (z_n - z_{n-1})$$

and

$$z_{n+1} - z_{n-1} = (z_{n+1} - z_n) + (z_n - z_{n-1})$$

we obtain

$$\left\langle \frac{z_{n+1}-z_n}{\Delta t}, \frac{z_{n+1}-z_n}{\Delta t}\right\rangle - \left\langle \frac{z_n-z_{n-1}}{\Delta t}, \frac{z_n-z_{n-1}}{\Delta t}\right\rangle + \langle A_p z_n, z_{n+1}-z_{n-1}\rangle = 0.$$

Since A_p is symmetric, we find

$$\langle A_p z_n, z_{n+1} - z_{n-1}\rangle = \langle A_p z_{n+1}, z_n\rangle - \langle A_p z_n, z_{n-1}\rangle,$$

which proves that the quantity

$$E^{n+\frac{1}{2}} = \left\langle \frac{z_{n+1}-z_n}{\Delta t}, \frac{z_{n+1}-z_n}{\Delta t} \right\rangle + \langle A_p z_{n+1}, z_n \rangle$$

is conserved. Next, we remark that

$$\langle A_p z_{n+1}, z_n \rangle = \left\langle A_p \frac{z_{n+1}+z_n}{2}, \frac{z_{n+1}+z_n}{2} \right\rangle - \frac{\Delta t^2}{4} \left\langle A_p \frac{z_{n+1}-z_n}{\Delta t}, \frac{z_{n+1}-z_n}{\Delta t} \right\rangle.$$

Thus, we have

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

□

The above local time-stepping algorithm conserves the energy $E^{n+\frac{1}{2}}$ in (43), which guarantees stability if and only if $E^{n+\frac{1}{2}}$ is positive semidefinite or, equivalently, if and only if the matrices $\left(I - \frac{\Delta t^2}{4} A_p\right)$ and A_p are both positive semidefinite. Hence if $\lambda_{\min}$ and $\lambda_{\max}$ denote the smallest and largest eigenvalues of A_p, respectively, the numerical scheme will be stable if and only if

$$0 \le \frac{\Delta t^2}{4} \lambda_{\min} \le \frac{\Delta t^2}{4} \lambda_{\max} \le 1.$$

For $p = 1$ we have $A_p = A$, and thus we recover the well-known CFL condition of the standard leap-frog scheme:

$$\Delta t \le \frac{2}{\sqrt{\lambda_{max}}} = \Delta t_{LF}.$$

For $p > 1$, the matrix A_p explicitly depends on Δt, and so do its eigenvalues. Moreover, as the eigenvectors of A and A_p generally do not coincide, the analytic derivation of a CFL condition is not obvious. Instead, we shall perform a systematic numerical study of the eigenvalues of $\frac{\Delta t^2}{4} A_p$ in the following typical situation.

3.3 Stability and CFL condition: numerical study

We consider the one-dimensional wave equation with constant wave speed $c = 1$ on the interval $\Omega = [0\,;\,6]$ with periodic boundary conditions. Next, we divide Ω into three equal parts. The left and right intervals, $[0\,;\,2]$ and $[4\,;\,6]$, are discretized with an equidistant mesh of size h_{coarse}, whereas the interval $[2\,;\,4]$ is discretized with an equidistant mesh of size $h_{\text{fine}} = h_{\text{coarse}}/p$. Hence, the two outer intervals correspond to the coarse region and the inner interval, $[2\,;\,4]$, to the refined region – see Fig.1.

For every time-step Δt, we shall take p steps of size $\Delta\tau = \Delta t/p$ in the refined region. In the absence of local refinement, i.e. $p = 1$, 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. For $p \ge 2$, we let Δt_p denote the maximal time-step of Algorithm 2. If $\Delta t_p = \Delta t_{LF}$, the local time-stepping algorithm imposes no further restriction on Δt and we then shall call the CFL condition of the new scheme optimal.

First, we consider the IP-DG discretization with $\mathcal{P}^1$–elements and (small) penalization, $\alpha = 2$. We choose $h_{\text{coarse}} = 0.2$, which yields the maximal time-step $\Delta t_{LF} = 0.55\, h_{\text{coarse}} =$

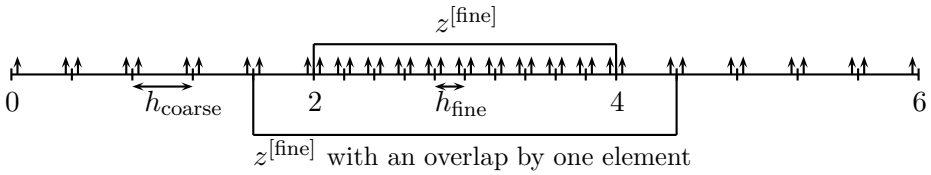

Figure 1: IP-DG $\mathcal{P}^1$–elements. The computational mesh and the associated degrees of freedom, indicated by small arrows.

0.11 for $p = 1$. Now, we refine by a factor $p = 2$ those elements that lie inside the interval $[2, 4]$, that is $h_{\text{fine}} = 0.1$, and set to one all corresponding entries in P. Hence for every time-step Δt, we shall take two steps of size $\Delta\tau = \Delta t/2$ in the refined region.

To determine the range of values Δt for which the local time-stepping scheme is stable, we display the eigenvalues of $(\Delta t^2/4)A_p$ for varying $\Delta t/\Delta t_{LF}$ – recall that A_p also depends on Δt. The numerical scheme is stable for any particular Δt if all corresponding eigenvalues lie between zero and one; otherwise, it is unstable. Since the smallest eigenvalue actually never dips below zero, it does not affect the stability here. As shown in the left frame of Fig. 2, the largest time-step allowed is only about 60% of Δt_{LF}; hence, the gain over a straightforward reduction of the (global) time-step by a factor two is rather modest.

To allow for larger time-steps, we now slightly enlarge the set of unknowns where a local time-step is used by also including those degrees of freedom that are associated with elements directly adjacent to the refined region. By setting the corresponding entries in P to one, we easily realize this overlap by one element in $z^{[\text{fine}]}$. In the right frame of Fig. 2 we observe that all eigenvalues now lie essentially between zero and one. However, a hundred thousand-fold magnification of that same figure, shown in the left frame of Fig. 3, reveals that some eigenvalues still barely transgress the strict stability limit at one. Further extension of the overlap by one additional element removes all unstable values below $0.9\Delta t_{LF}$, as shown in the right frame of Fig. 3, while four 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.

Table 1: IP-DG $\mathcal{P}^1$ elements. The largest eigenvalue of $(\Delta t^2/4)A_p$ with $\Delta t = \Delta t_{opt}$ for an overlap by one element.

	p				
h_{coarse}	2	3	4	10	13
0.5	1.0002	0.9912	0.9983	1.0003	1.0005
0.2	1.0009	0.9999	1.0003	1.0002	1.0002
0.1	1.0006	1.00001	0.9997	0.9999	0.9999
0.05	1.0005	1.0001	0.9999	1.00006	0.9999
0.025	1.0005	1.00009	1.00005	1.00002	1.00002

Instead, we now address the question whether the local time-stepping scheme is stable for the maximal time-step $\Delta t = \Delta t_{LF}$. In Tab. 1 we list the corresponding maximal eigenvalue of $(\Delta t^2/4)A_p$ for an overlap by one element with different mesh sizes $h_{\text{coarse}} = 0.5, 0.2, 0.1, 0.05, 0.025$ and different $p = 2, 3, 4, 10, 13$; here, the maximal eigenvalue typically is greater than one and the local time-stepping scheme therefore unstable for the optimal

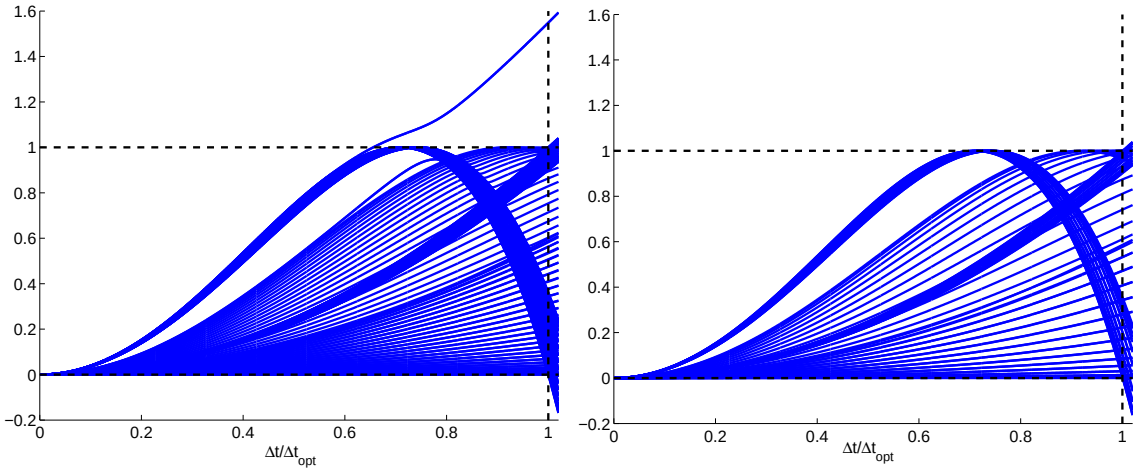

Figure 2: IP-DG with $\mathcal{P}^1$ elements. The eigenvalues of $(\Delta t^2/4)A_p$: without overlap (left); with an overlap by one element (right).

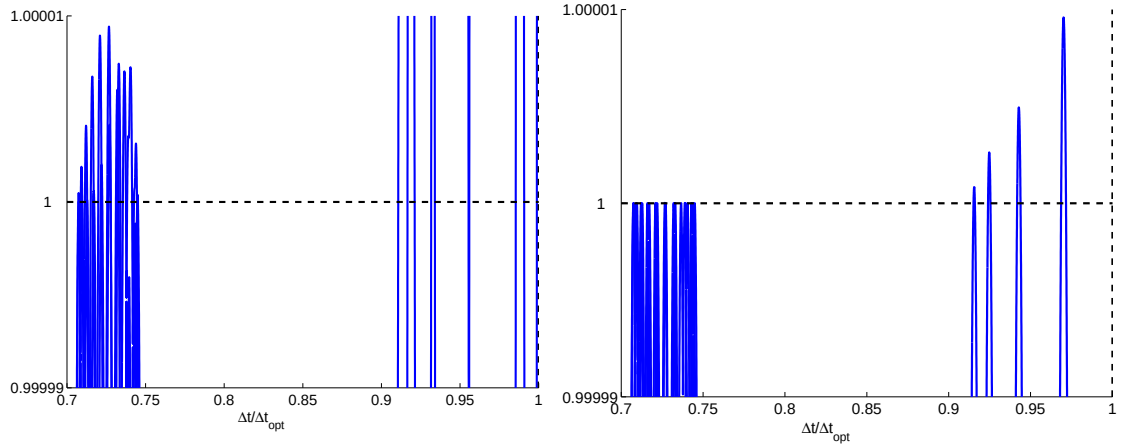

Figure 3: IP-DG $\mathcal{P}^1$ elements. The eigenvalues of $(\Delta t^2/4)A_p$: overlap by one element (left); overlap by two elements (right). The vertical scale is strongly magnified: $0.99999 < \lambda_{\max} < 1.00001$. elements (IP-DG $\mathcal{P}^1$ elements).

time-step. Yet with an overlap by two elements, the maximal eigenvalues listed in Tab. 2 now always lie below one, independently of h_{coarse} and p; hence for an overlap by two elements, Algorithm 2 is always stable for the optimal time-step.

Table 2: IP-DG $\mathcal{P}^1$ elements. The largest eigenvalue of $(\Delta t^2/4)A_p$ with $\Delta t = \Delta t_{LF}$ for an overlap by two elements.

	p				
h_{coarse}	2	3	4	10	13
0.5	0.9981	0.9902	0.9983	0.9997	0.9999
0.2	0.9998	0.9994	0.9999	0.9999	0.9999
0.1	0.9998	0.9999	0.9996	0.9999	0.9999
0.05	0.9999	0.9999	0.9999	0.9999	0.9999
0.025	0.9999	0.9999	0.9999	0.9999	0.9999

Next, we repeat the above experiment with standard $\mathcal{P}^1$ continuous, piecewise linear finite elements with mass lumping. Then, the CFL condition for the classical leap-frog scheme is $\Delta t_{LF} = h_{\text{coarse}} = 0.2$. In contrast to the IP-DG FE method, a degree of freedom can now belong to both a fine and a coarse element; hence, any degree of freedom at the interface between the two sub-regions is automatically included in $z^{[\text{fine}]}$, as shown in Fig. 4.

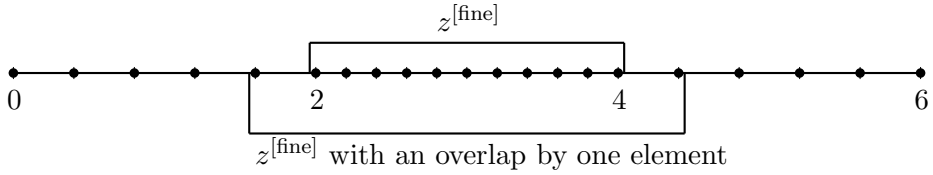

Figure 4: The nodes in $z^{[\text{fine}]}$ for continuous $\mathcal{P}^1$ elements

In Tab. 3 we list the maximal eigenvalue of $\frac{\Delta t^2}{4}A_p$ at $\Delta t = \Delta t_{LF}$ for an overlap by a single element for different mesh sizes $h_{\text{coarse}} = 0.5, 0.2, 0.1, 0.05, 0.025$ and different $p = 2, 3, 4, 10, 13$. Here, we find that an overlap by only one element already leads to a stable scheme for the optimal time-step.

Table 3: Continuous $\mathcal{P}^1$-FE. The largest eigenvalue of $\frac{\Delta t^2}{4}A_p$ at $\Delta t = \Delta t_{LF}$ for an overlap by one element.

	p				
h_{coarse}	2	3	4	10	13
0.5	0.9828	0.9792	0.9993	0.9999	0.9999
0.2	0.9969	0.9962	0.9999	0.9999	0.9999
0.1	0.9992	0.9991	0.9999	0.9999	0.9999
0.05	0.9998	0.9998	0.9999	0.9999	0.9999
0.025	0.9999	0.9999	0.9999	0.9999	0.9999

In summary, a slight extension (overlap) of the region where local time-steps are used into that part of the mesh immediately adjacent to the refined region always improves the stability of the time-stepping scheme. An overlap by one element for the $\mathcal{P}^1$ continuous FE

discretization (with mass lumping), or by two elements for the IP-DG discretization, always permits the use of the maximal (optimal) time-step Δt_{LF}, dictated by the leap-frog method in the coarse region.

4 High order local time-stepping

We shall now show how to extend the second-order local time-stepping method from Section 3 to arbitrarily high accuracy. First, we develop in detail a fourth-order local time-stepping scheme and again exhibit a conserved discrete energy. Then, we consider the general case of arbitrary (even) order. By including a small overlap into the surrounding coarser region we show via numerical experiments that both the fourth- and sixth-order local time-stepping schemes are stable for the optimal time-step dictated by the coarse mesh size.

4.1 Fourth-order local time-stepping

A fourth-order extension of the leap-frog scheme for (25) is provided by the Modified Equation (ME) approach [7, 17, 2, 39]. Thus, we replace $Az(t+\theta\Delta t)$ in (27) by the leading terms in its Taylor expansion,

$$Az(t+\theta\Delta t) \simeq A\left(z(t)+\theta\Delta t z'(t)+\frac{\theta^2\Delta t^2}{2}z''(t)\right).$$

By integrating in (27) the resulting polynomial in θ, the first derivative of z vanishes; since $z''(t)=-Az(t)$, we thus obtain the modified equation scheme:

$$\frac{z_{m+1}-2z_m+z_{m-1}}{\Delta t^2} = -Az_m+\frac{\Delta t^2}{12}A^2 z_m, \qquad z_m \simeq z(t_m). \tag{44}$$

In contrast to standard fourth-order Runge-Kutta schemes, the ME approach requires only two multiplications with A per time-step. Its CFL condition follows from the CFL condition of the leap-frog scheme [12],

$$\Delta t_{ME} \le \frac{2\sqrt{3}}{\sqrt{\lambda_{max}}} = \sqrt{3}\Delta t_{LF}.$$

As the maximal time-step allowed by the modified equation approach is about 70% times *larger* than that of the leap-frog scheme, the additional work needed for the improved accuracy is quite small. Clearly, high-order schemes of arbitrary (even) order can be obtained by using additional terms in the Taylor expansion of $Az(t+\theta\Delta t)$. Recently Gilbert and Joly derived new time-stepping methods that maximize the CFL condition for a given order of accuracy [21].

Following [35], we expand in (29) $z^{[\text{coarse}]}$ in Taylor series as

$$z^{[\text{coarse}]}(t+\theta\Delta t) = z^{[\text{coarse}]}(t)+\theta\Delta t\frac{dz^{[\text{coarse}]}}{dt}(t)+\frac{\theta^2\Delta t^2}{2}\frac{d^2z^{[\text{coarse}]}}{dt^2}(t)+O(\Delta t^3).$$

Since

$$\frac{d^2z^{[\text{coarse}]}}{dt^2}(t) = (I-P)\frac{d^2z}{dt^2}(t) = -(I-P)Az(t),$$

we find that

$$z^{[\text{coarse}]}(t+\theta\Delta t) \simeq (I-P)z(t) + \theta\Delta t(I-P)\frac{dz}{dt}(t) - \frac{\theta^2\Delta t^2}{2}(I-P)Az(t). \tag{45}$$

Next, we use (45) in (29) to replace $z^{[\text{coarse}]}$, which yields

$$\begin{aligned} z(t+\Delta t) - 2z(t) + z(t-\Delta t) \;\simeq\; & -\Delta t^2 A(I-P)z(t) + \frac{\Delta t^4}{12}A(I-P)Az(t) \\ & -\Delta t^2 \int_{-1}^{1} (1-|\theta|)Az^{[\text{fine}]}(t+\theta\Delta t)d\theta. \end{aligned} \tag{46}$$

Hence if $P=0$ we recover the standard modified equation scheme (44).

Similarly to Section 3.1, we shall now approximate the right-hand side of (46) by solving the following differential equation for $\tilde{z}(\tau)$:

$$\begin{cases} \dfrac{d^2\tilde{z}}{d\tau^2} = -A(I-P)z(t) + \dfrac{\tau^2}{2}A(I-P)Az(t) - AP\tilde{z}(\tau) \\ \tilde{z}(0) = z(t), \qquad \dfrac{d\tilde{z}}{d\tau}(0) = 0. \end{cases} \tag{47}$$

Again by symmetry, we have

$$z(t+\Delta t) + z(t-\Delta t) \simeq 2\tilde{z}(\Delta t). \tag{48}$$

Thus, we approximate the right side of (46) by solving (47) on $[0,\Delta t]$, and then use (48) to compute $z(t+\Delta t)$. The last term on the right of (46) explicitly depends on $\tilde{z}(\tau)$, but it involves only the unknowns associated with the nonzero entries in P. If those nonzero entries occupy only a small fraction of all unknowns, the additional effort from solving (47) will be small. Clearly, in doing so we must ensure that the overall numerical scheme remains fourth-order accurate in time.

In summary, the fourth-order local time-stepping algorithm for the solution of (23) computes $z_{n+1} \simeq z(t+\Delta t)$, given z_n and z_{n-1}, as follows:

Algorithm 6. *1. Set* $\tilde{z}_0 = z_n$, $w_1 = A(I-P)z_n$, $w_2 = A(I-P)Az_n$, $u_1 = APz_n$ *and* $u_2 = APAz_n$;

2. Compute $\tilde{z}_{1/p} = \tilde{z}_0 - \frac{1}{2}\left(\frac{\Delta t}{p}\right)^2 (w_1+u_1) + \frac{1}{24}\left(\frac{\Delta t}{p}\right)^4 (w_2+u_2)$;

3. For $m = 1...p-1$, *compute*

- $v_1 = -w_1 + \frac{1}{2}\left(\frac{m\Delta t}{p}\right)^2 w_2 - AP\tilde{z}_{m/p}$;
- $v_2 = w_2 - APv_1$;
- $\tilde{z}_{(m+1)/p} = 2\tilde{z}_{m/p} - \tilde{z}_{(m-1)/p} + \left(\frac{\Delta t}{p}\right)^2 v_1 + \frac{1}{12}\left(\frac{\Delta t}{p}\right)^4 v_2$;

4. Compute $z_{n+1} = -z_{n-1} + 2\tilde{z}_1$.

Here Steps 1–3 correspond to the numerical solution of (47) until $\tau = \Delta t$ with the modified equation approach using the local time-step $\Delta\tau = \Delta t/p$. This algorithm requires two multiplications by $A(I-P)$ in Step 1 and $2p$ further multiplications by AP. No multiplication by A is needed to compute Az_n in Step 1, since

$$Az_n = A(I-P)z_n + APz_n = w_1 + u_1.$$

For $P = 0$, that is without any local time-stepping, this algorithm reduces to the modified equation scheme (44) above.

To establish the accuracy and stability of the above local time-stepping scheme we shall now show how to rewrite it in "leap-frog manner". To do so, we first need the following technical result.

Lemma 7. *For $m \geq 2$, $\tilde{z}_{m/p}$ defined by Algorithm 6 satisfies*

$$\begin{aligned}\tilde{z}_{m/p} = \quad & z_n - \frac{m^2}{2}\left(\frac{\Delta t}{p}\right)^2 Az_n + \frac{m^4}{24}\left(\frac{\Delta t}{p}\right)^4 A^2 z_n \\ & + \sum_{j=1}^{2(m-1)} \left(\frac{\Delta t}{p}\right)^{2(j+2)} \beta_j^m (AP)^j AAz_n \end{aligned} \tag{49}$$

where β_j^m are constant.

Proof. The proof is by induction over m.

We first show that (49) holds for $m = 2$. Starting from the definitions of w_1 and w_2 in Step 1 of Algorithm 6, we find that

$$\begin{aligned} v_1 &= -A(I-P)z_n + \frac{1}{2}\left(\frac{\Delta t}{p}\right)^2 A(I-P)Az_n - AP\tilde{z}_{1/p}, \\ v_2 &= A(I-P)Az_n - APv_1. \end{aligned}$$

Since

$$\tilde{z}_{1/p} = z_n - \frac{1}{2}\left(\frac{\Delta t}{p}\right)^2 Az_n + \frac{1}{24}\left(\frac{\Delta t}{p}\right)^4 A^2 z_n,$$

the two expressions for v_1 and v_2 simplify as

$$v_1 = -Az_n + \frac{1}{2}\left(\frac{\Delta t}{p}\right)^2 A^2 z_n - \frac{1}{24}\left(\frac{\Delta t}{p}\right)^4 APA^2 z_n, \tag{50}$$

$$v_2 = A^2 z_n - \frac{1}{2}\left(\frac{\Delta t}{p}\right)^2 APA^2 z_n + \frac{1}{24}\left(\frac{\Delta t}{p}\right)^4 APAPA^2 z_n. \tag{51}$$

We now replace v_1, v_2 in the expression for $\tilde{z}_{2/p}$ in Step 3. This yields

$$\tilde{z}_{2/p} = \quad z_n - 2\left(\frac{\Delta t}{p}\right)^2 Az_n + \frac{2}{3}\left(\frac{\Delta t}{p}\right)^4 A^2 z_n \tag{52}$$

$$- \quad \frac{1}{12}\left(\frac{\Delta t}{p}\right)^6 APA^2 z_n + \frac{1}{288}\left(\frac{\Delta t}{p}\right)^8 APAPA^2 z_n. \tag{53}$$

Hence, $\beta_1^2 = -1/12$ and $\beta_1^2 = 1/288$, which completes the proof for $m = 2$.

For higher $m \geq 3$ (and $p \geq 3$), the proof iso by induction on m; those straightforward but tedious calculations are omitted here. □

As a consequence, we can rewrite the above local time-stepping algorithm in "leap-frog manner".

Proposition 8. *The local time-stepping Algorithm 6 is equivalent to*

$$z_{n+1} = 2z_n - z_{n-1} - \Delta t^2 A_p z_n, \tag{54}$$

where A_p is defined by

$$A_p = A - \frac{\Delta t^2}{12} A^2 - \frac{2}{p^2} \sum_{j=1}^{2(p-1)} \left(\frac{\Delta t}{p}\right)^{2(j+1)} \beta_j^p (AP)^j A^2. \tag{55}$$

This scheme is fourth-order accurate. Furthermore, the matrix AA_p is symmetric.

Proof. Recall that $z_{n+1} = -z_{n-1} + 2\tilde{z}_1$. We now use (49) with $m = p$ to replace $\tilde{z}_1$. This yields

$$z_{n+1} = -z_{n-1} + 2z_n - \Delta t^2 A z_n + \frac{\Delta t^4}{12} A^2 z_n + \frac{2\Delta t^2}{p^2} \sum_{j=1}^{2(p-1)} \left(\frac{\Delta t}{p}\right)^{2(j+1)} \beta_j^p (AP)^j A A z_n,$$

which corresponds to (54) with A_p as in (55).

To prove consistency, we rewrite the scheme as

$$\frac{z_{n+1} - z_{n-1} + 2z_n}{\Delta t^2} - \frac{\Delta t^2}{12} A^2 z_n - \frac{2}{p^2} \sum_{j=1}^{2(p-1)} \left(\frac{\Delta t}{p}\right)^{2(j+1)} \beta_j^p (AP)^j A A z_n = -A z_n.$$

From

$$\frac{z(t+\Delta t) - 2z(t) + z(t-\Delta t)}{\Delta t^2} - \frac{\Delta t^2}{12} A^2 z(t) = z''(t) + O(\Delta t^4)$$

and

$$\sum_{j=1}^{2(p-1)} \left(\frac{\Delta t}{p}\right)^{2(j+1)} \beta_j^p (AP)^j A^2 z(t) = O(\Delta t^4),$$

we conclude that Algorithm 6 is indeed a fourth-order approximation of (25).

As the terms $(AP)^j A^2$ are obviously not symmetric, the matrix A_p is not symmetric either. However, since

$$(A(AP)^j A)^\top = A(PA)^j A = (AP)^j A, \qquad j \geq 1,$$

the matrix AA_p indeed is symmetric. □

4.2 Energy conservation and stability

We shall now exhibit a discrete energy which is conserved by the fourth-order local time-stepping scheme (Algorithm 6) and thereby determine a necessary and sufficient condition for stability. Next, we perform a systematic numerical study to demonstrate the improvement in the CFL condition achieved by slightly extending the overlap of the fine into the coarse region.

First, using Proposition 8 we rewrite the local time-stepping scheme as (54)–(55). As a consequence, we can prove the conservation of the following energy.

Proposition 9. *The fourth-order local time-stepping scheme (Algorithm (6)) conserves the following energy:*

$$E^{n+\frac{1}{2}} = \frac{1}{2}\left[\left\langle\left(A-\frac{\Delta t^2}{4}AA_p\right)\frac{z_{n+1}-z_n}{\Delta t},\frac{z_{n+1}-z_n}{\Delta t}\right\rangle\right. \tag{56}$$

$$\left.+\left\langle AA_p\frac{z_{n+1}+z_n}{2},\frac{z_{n+1}+z_n}{2}\right\rangle\right]. \tag{57}$$

Proof. As A_p is not symmetric, we first have to pre-multiply (54) by A. Next, we take the inner product of the resulting expression with $z_{n+1}-z_{n-1}$, which yields the conservation of

$$\left\langle A\frac{z_{n+1}-z_n}{\Delta t},\frac{z_{n+1}-z_n}{\Delta t}\right\rangle+\langle AA_pz_{n+1},z_n\rangle\,.$$

That expression is equal to $E^{n+\frac{1}{2}}$ above, since

$$\langle AA_pz_{n+1},z_n\rangle=\left\langle AA_p\frac{z_{n+1}+z_n}{2},\frac{z_{n+1}+z_n}{2}\right\rangle-\frac{\Delta t^2}{4}\left\langle AA_p\frac{z_{n+1}-z_n}{\Delta t},\frac{z_{n+1}-z_n}{\Delta t}\right\rangle.$$

□

Since both A and AA_p are symmetric matrices, $E^{n+\frac{1}{2}}$ is a symmetric quadratic form which is positive for Δt sufficiently small. Hence the local time-stepping scheme will be stable if, and only if, the two matrices $\left(A-\frac{\Delta t^2}{4}AA_p\right)$ and AA_p are positive semidefinite. For $p=1$, we have $AA_p=A\left(A-\frac{\Delta t}{12}A^2\right)$ and thus we recover the CFL condition of the modified equation scheme:

$$\Delta t\le\frac{2\sqrt{3}}{\sqrt{\lambda_{max}}}=\sqrt{3}\Delta t_{LF}=\Delta t_{ME},$$

where λ_{max} is the largest eigenvalue of A.

For $p\ge 2$, the CFL condition cannot be determined analytically. Thus, we shall again provide a systematic numerical study to evaluate the stability of Algorithm 6 in a typical situation.

We consider the same one-dimensional problem as in Section 3.3, but now use an IP-DG discretization with $\mathcal{P}^3-$ elements with (small) penalization $\alpha=7$ to achieve fourth-order accuracy both in space and time. In the absence of local time-stepping, that is with $p=1$, the maximal time-step dictated by the equidistant (coarse) mesh is $\Delta t_{ME}=0.265\,h_{\text{coarse}}$. For $p\ge 2$, we choose the same optimal time-step, $\Delta t=\Delta t_{ME}$ and compute the smallest eigenvalue of $\left(A-\frac{\Delta t^2}{4}AA_p\right)$ and AA_p, respectively – both need to be positive for stability.

From Tab. 4 we conclude that Algorithm 6 without overlap is unstable for the maximal time-step. However, as we extend by a single adjacent element the region where smaller time-steps are used, the fourth-order local time-stepping scheme becomes stable independently of p and h_{coarse}.

Table 4: IP-DG $\mathcal{P}^3$ elements. The smallest eigenvalue of AA_p and $A-(\Delta t^2/4)AA_p$ at $\Delta t = \Delta t_{ME}$ with $p=2$: without overlap (left); with overlap by one element (right).

h_{coarse}	AA_p	$A-(\Delta t^2/4)A_p$
0.5	-5.10^5	1.1
0.2	-2.10^7	1.1
0.1	-3.10^8	1.1
0.05	-5.10^9	1.1
0.025	-7.10^{10}	1.1

h_{coarse}	AA_p	$A-(\Delta t^2/4)AA_p$
0.5	1.2	1.1
0.2	1.2	1.1
0.1	1.2	1.1
0.05	1.2	1.1
0.025	1.2	1.1

Next, we consider continuous piecewise $\mathcal{P}^3$ finite elements with mass lumping. Since the CFL condition of the leap-frog scheme is $\Delta t_{LF} = 0.232\, h_{\text{coarse}}$ [19], the corresponding CFL condition of the modified equation scheme is $\Delta t_{ME} = \sqrt{3}\Delta t_{LF} = 0.401\, h_{\text{coarse}}$. Again, we automatically include those degrees of freedom that belong to both a fine and a coarse cell into $z^{[\text{fine}]}$. Remarkably, no overlap is needed here for the local time-stepping scheme to remain stable, regardless of and h_{coarse}, as shown in Tab. 5 for $p=2$.

Table 5: $\mathcal{P}^3$ FE elements. The smallest eigenvalue of AA_p and $A-(\Delta t^2/4)AA_p$ for $\Delta t = \Delta t_{ME}$ with $p=2$ without overlap.

h_{coarse}	AA_p	$A-(\Delta t^2/4)AA_p$
0.5	1.2	0.15
0.2	1.2	0.9
0.1	1.2	0.53
0.05	1.2	1.1
0.025	1.2	1.1

4.3 Local time-stepping of arbitrary order

The modified equation approach used in Section 4.1 to extend the local time-stepping approach to fourth-order can be generalized to arbitrarily accuracy. Since the procedure parallels that used in Section 4.1, we shall omit details and state only key results.

Starting again from the Taylor expansion

$$z^{[\text{coarse}]}(t+\theta\Delta t) = z^{[\text{coarse}]}(t) + \sum_{i=1}^{2s-2} \frac{\theta^i \Delta t^i}{i!} \frac{d^i z^{[\text{coarse}]}}{d\theta^i}(t) + O(\Delta t^{2s-1})),$$

we use the fact that

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

to obtain

$$\begin{aligned} z^{[\text{coarse}]}(t+\theta\Delta t) &= z^{[\text{coarse}]}(t) + \sum_{i=1}^{s-1} \frac{\theta^{2i}\Delta t^{2i}}{(2i)!}(I-P)(-A)^i z(t) \\ &+ \sum_{i=1}^{s-1} \frac{\theta^{2i-1}\Delta t^{2i-1}}{(2i-1)!} \frac{d^{2i-1} z^{[\text{coarse}]}}{d\theta^{2i-1}}(t) + O(\Delta t^{2s-1})). \end{aligned} \tag{58}$$

Next, we use (58) in (29) to replace $z^{[\text{coarse}]}$, which yields

$$\begin{aligned} z(t+\Delta t) - 2z(t) + z(t-\Delta t) &\simeq -\Delta t^2 A(I-P)z(t) + 2\sum_{i=1}^{s-1} \frac{\Delta t^{2i+2}}{(2i+2)!} A(I-P)(-A)^i z(t) \\ &\quad -\Delta t^2 \int_{-1}^{1} (1-|\theta|) A z^{[\text{fine}]}(t+\theta\Delta t) d\theta. \end{aligned} \tag{59}$$

Similarly to Section 3.1, we now approximate the right-hand side of (59) by solving

$$\begin{cases} \dfrac{d^2\tilde{z}}{d\tau^2} = -A(I-P)z(t) - \displaystyle\sum_{i=1}^{s} \frac{\tau^{2i}}{2i!} A(I-P)(-A)^i z(t) - AP\tilde{z}(\tau), \\ \tilde{z}(0) = z(t), \quad \dfrac{d\tilde{z}}{dt}(0) = 0, \end{cases} \tag{60}$$

and we then shall use the approximation

$$z(t+\Delta t) + z(t-\Delta t) \simeq 2\tilde{z}(\Delta t). \tag{61}$$

Thus, we approximate the right side of (59) by solving (60) for $0 \le \tau \le \Delta t$, and then use (61) to compute $z(t+\Delta t)$. The last term on the right of (60) explicitly depends on $\tilde{z}(\tau)$, but it involves only the unknowns associated with the nonzero entries in P. If those nonzero entries occupy only a small fraction of all unknowns, the additional effort from solving (60) will be small. Clearly, in doing so we must ensure that the overall numerical scheme remains $2s$th-order accurate in time.

In summary, the local time-stepping algorithm of order $2s$ for the solution of (23) computes $z_{n+1} \simeq z(t+\Delta t)$, given z_n and z_{n-1}, as follows:

Algorithm 10. *1. Set* $\tilde{z}_0 = z_n, \quad w_m = A(I-P)A^{m-1}z_n, \quad u_m = APA^{m-1}z_n, \quad m = 1,\dots,s;$

2. Compute

- $v_m = (-1)^m(w_m + u_m\tilde{z}_0), \quad m = 1,\dots,s;$
- $\tilde{z}_{1/p} = \tilde{z}_0 + \displaystyle\sum_{m=1}^{s} \frac{1}{(2m)!}\left(\frac{\Delta t}{p}\right)^{2m} v_m;$

3. For $m = 1,\dots,p-1$, *compute*

- $v_1 = -\left(w_1 + \displaystyle\sum_{l=1}^{s} \frac{(-1)^l}{(2l)!} w_l\right) - AP\tilde{z}_{m/p}, \quad k = 1,\dots,s;$

- $v_k = (-1)^k \left(w_k + \sum_{l=k+1}^{s} \frac{(-1)^{l-k}}{(2(l-k))!} w_l \right) - APv_{k-1}, \quad k = 2, \ldots, s;$
- $\tilde{z}_{(m+1)/p} = 2\tilde{z}_{m/p} - \tilde{z}_{(m-1)/p} + \sum_{k=1}^{s} \frac{1}{(2m)!} \left(\frac{\Delta t}{p} \right)^{2k} a_k,$

4. Compute $z_{n+1} = -z_{n-1} + 2\tilde{z}_1$.

Steps 1–3 compute the numerical solution of (60) at time $\tau = \Delta t$ using the $2s$-th order modified equation scheme with local time-step $\Delta\tau = \Delta t/p$. Note that we recover the standard modified equation scheme by setting $P = 0$. Clearly, this algorithm only requires s multiplications by $A(I-P)$ and ps multiplications by AP. No multiplication by A is needed to compute $A^{m-1}z_n$ in Step 1, since

$$A^{m-1}z_n = A(I-P)A^{m-1}z_n + APA^{m-1}z_n = w_m + u_m.$$

We now rewrite the above alogrithm in "leap-frog manner" to determine the accuracy and establish the stability of the above algorithm.

Proposition 11. *The local time-stepping method (Algorithm 10) of order* $2s$ *is equivalent to*

$$z_{n+1} = 2z_n - z_{n-1} - \Delta t^2 A_p z_n, \tag{62}$$

where A_p *is defined by*

$$A_p = A - 2\sum_{j=2}^{s} \frac{\Delta t^{2(j-1)}}{(2j)!} A^j - \frac{2}{p^2} \sum_{j=1}^{(p-1)s} \left(\frac{\Delta t}{p} \right)^{2(s-1+j)} \gamma_j^p (AP)^j A^s.$$

This scheme is $2s$*-th order accurate. Furthermore the matrix* $A^{s-1}A_p$ *is symmetric.*

Proof. We do not detail the proof of the first part of the proposition, which is similar to that of Prop. 4 and 8. To prove consistency, we rewrite the scheme as

$$\frac{z_{n+1} - 2z_n - z_{n-1}}{\Delta t^2} - 2\sum_{j=2}^{s} \frac{\Delta t^{2(j-1)}}{(2j)!} A^j - \frac{2}{p^2} \sum_{j=1}^{(p-1)s} \left(\frac{\Delta t}{p} \right)^{2(s-1+j)} \gamma_j^p (AP)^j A^s = -Az_n.$$

Since

$$\frac{z(t+\Delta t) - 2z(t) + z(t-\Delta t)}{\Delta t^2} - 2\sum_{j=2}^{s} \frac{\Delta t^{2(j-1)}}{(2j)!} A^j = z''(t) + O(\Delta t^{2s})$$

and

$$\sum_{j=1}^{(p-1)s} \left(\frac{\Delta t}{p} \right)^{2(s-1+j)} \gamma_j^p (AP)^j A^s = O(\Delta t^{2s}),$$

Algorithm 10 is a $2s$-th order approximation of (25).

As the matrices $(AP)^j A^s$ are generally not symmetric, the matrix A_p is not symmetric either. However, since

$$(A^{s-1}(AP)^j A^s)^\top = A^s (PA)^j A^{s-1} = A^{s-1}(AP)^j A^s, \qquad j \geq 1,$$

the matrix $A^{s-1}A_p$ is indeed symmetric . □

We can now prove the conservation of a discrete energy.

Proposition 12. *The local time-stepping scheme (Algorithm 10) of order* $2s$ *conserves the following energy:*

$$E^{n+\frac{1}{2}} = \frac{1}{2}\left[\left\langle\left(A^{s-1}-\frac{\Delta t^2}{4}A^{s-1}A_p\right)\frac{z_{n+1}-z_n}{\Delta t},\frac{z_{n+1}-z_n}{\Delta t}\right\rangle\right. \tag{63}$$

$$\left.+\left\langle A^{s-1}A_p\frac{z_{n+1}+z_n}{2},\frac{z_{n+1}+z_n}{2}\right\rangle\right]. \tag{64}$$

Proof. As A_p is not symmetric, we first pre-multiply (62) by A^{s-1}. Next we take the inner product of the resulting expression by $z_{n+1}-z_{n-1}$, which yields the conservation of

$$\left\langle A^{s-1}\frac{z_{n+1}-z_n}{\Delta t},\frac{z_{n+1}-z_n}{\Delta t}\right\rangle+\left\langle A^{s-1}A_pz_{n+1},z_n\right\rangle.$$

That expression is equal to $E^{n+\frac{1}{2}}$ above, since

$$\left\langle A^{s-1}A_pz_{n+1},z_n\right\rangle = \left\langle A^{s-1}A_p\frac{z_{n+1}+z_n}{2},\frac{z_{n+1}+z_n}{2}\right\rangle \tag{65}$$

$$-\frac{\Delta t^2}{4}\left\langle A^{s-1}A_p\frac{z_{n+1}-z_n}{\Delta t},\frac{z_{n+1}-z_n}{\Delta t}\right\rangle. \tag{66}$$

□

Since both A^{s-1} and $A^{s-1}A_p$ are symmetric matrices, $E^{n+\frac{1}{2}}$ is a symmetric quadratic form which is positive for Δt sufficiently small. Hence the local time-stepping scheme will be stable if, and only if, the two matrices $\left(A^{s-1}-\frac{\Delta t^2}{4}A^{s-1}A_p\right)$ and $A^{s-1}A_p$ are positive definite.

5 Numerical Results

We shall now present numerical experiments that confirm the expected order of convergence and demonstrate the versatility of the above local time-stepping methods. First, we consider a simple one-dimensional test problem to show that the different local time-stepping schemes presented above indeed yield the expected overall rate of convergence when combined with a spatial finite element discretization of comparable accuracy, independently of the number of local time-steps p. Then, we consider wave propagation in two space dimensions with a locally highly refined mesh to illustrate the usefulness of local time-stepping in the presence of complex geometry.

5.1 Convergence study

We consider the one-dimensional wave equation with constant wave speed $c = 1$ on the interval $\Omega = [0\,;\,6]$ with periodic boundary conditions. The initial conditions are chosen to yield the exact solution $u_{ex}(x,t) = \sin(8\pi(x-t)/3)$, which corresponds to a sinusoidal wave propagating to the right. Again, we divide Ω into three equal parts. The left and right intervals, $[0\,;\,2]$ and $[4\,;\,6]$, are discretized with an equidistant mesh of size h_{coarse}, whereas the interval $[2\,;\,4]$ is discretized with an equidistant mesh of size $h_{\text{fine}} = h_{\text{coarse}}/p$. Hence,

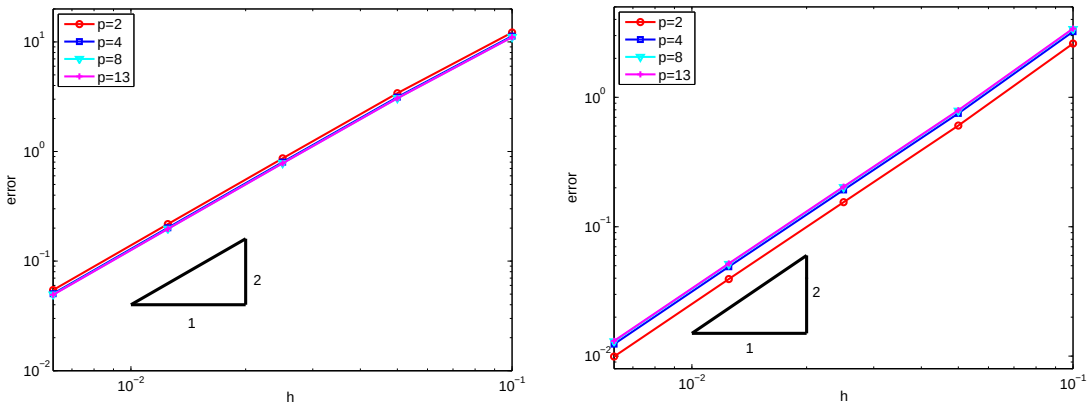

Figure 5: Error vs. $h = h_{\text{coarse}}$ for P^1 finite elements with $p = 2$, 4, 8 and 13: IP-DG (left) and continuous FE (right).

the two outer intervals correspond to the coarse region and the inner interval, $[2\,;\,4]$, to the refined region – refer to Section 3.3 for further details.

For every time-step Δt, we shall take $p \geq 2$ local steps of size $\Delta\tau = \Delta t/p$ in the refined region, with the second-order local time-stepping Algorithm 2. We choose an overlap of two and set in all instances $\Delta t = \Delta t_{LF}$, the largest time-step allowed by the leap-frog method on an equidistant mesh with mesh size $h = h_{\text{coarse}}$.

First, we consider an IP-DG discretization with $\mathcal{P}^1$ elements and (small) penalty parameter $\alpha = 2$, as described in Section 3.3, and the sequence of meshes $\mathcal{T}_h$, $h_{\text{coarse}} = 0.1$, 0.05, 0.025, 0.0125, 0.00625. As we systematically reduce the global mesh size, h_{coarse}, while simultaneously reducing Δt, we monitor the L^2 space-time error in the numerical solution, $\|u - u_{ex}\|_{L^2(0,T;L^2(\Omega))}$ until the final time $T = 60$. In the left frame in Fig. 5, the numerical error is shown vs. the mesh size, $h = h_{\text{coarse}}$. Regardless of the number of local time-steps $p = 2, 4, 8, 13$, the numerical method converges with order two.

We now repeat the same experiment with P^1 continuous finite elements with mass lumping for the same sequence of meshes. As shown in the right frame of Fig. 5, the local time-stepping method again yields overall second-order convergence independently of p.

Next, we consider the fourth-order time-stepping scheme (Algorithm 6) and combine it either with a continuous FE or the IP-DG discretization with $\mathcal{P}^3$ elements. Thus, we expect both numerical schemes to exhibit overall fourth-order convergence with respect to the L^2 norm. 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 $h = h_{\text{coarse}}$. In Fig. 6 we display the space-time L^2-errors of the numerical solutions for the sequence of meshes $h_{\text{coarse}} = 0.2, 0.1, 0.05, 0.025, 0.0125$ and different values of p. Both the continuous FE method with mass lumping and the IP-DG method, here with $\alpha = 7$, yield the expected fourth-order convergence.

Finally, to validate the order of convergence of the sixth order time-stepping scheme (Algorithm 10 with $s = 3$), we consider the IP-DG method with $\mathcal{P}^5$ elements, where we set $\alpha = 16$. As above we choose an overlap of two elements and set the time-step to its (maximal) optimal value, $\Delta t = \Delta t_{ME}$, the largest possible time-step allowed by the modified equation approach of order six on an equidistant mesh with $h = h_{\text{coarse}}$. Again the numerical results shown in Fig. 7 for $p = 2$ corroborate the expected sixth order of convergence.

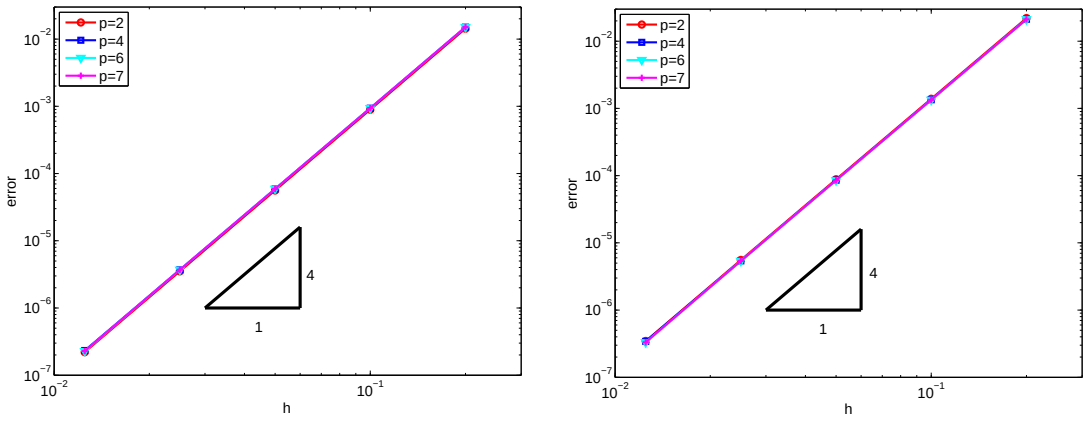

Figure 6: Error vs. $h = h_{\text{coarse}}$ for P^3 finite elements with $p = 2$, 4, 6, 7: IP-DG (left) and continuous FE (right).

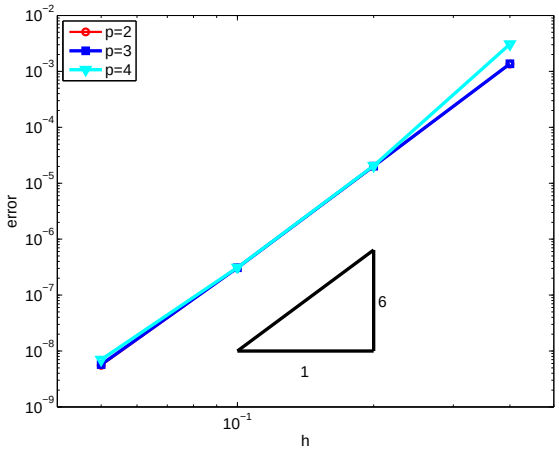

Figure 7: Error vs. $h = h_{\text{coarse}}$ for the IP-DG method with P^5 elements with $p = 2$.

5.2 Two-dimensional example

To illustrate the usefulness of the local time-stepping methods presented above, we consider the wave equation (1)–(4) with constant speed $c = 1$ in a computational domain Ω, that consists of two 1×0.95 rectangles connected by a very narrow 0.004×0.1 channel – see Fig. 8. We impose homogeneous Neumann conditions on the boundary of Ω and choose as initial conditions

$$u_0(x) = \begin{cases} \exp(\|x - x_0\|^2/r^2), & \|x - x_0\| \le \sqrt{2}\, r, \\ 0 & \text{otherwise} \end{cases} \tag{67}$$

$$v_0(x) = 0, \tag{68}$$

where $x_0 = (0, 0.25)$ and $r = 0.025$.

For the spatial discretization we opt for the IP-DG method with P^3 triangular elements and $\alpha = 11$. Hence to resolve the propagating wave with sufficient accuracy, we find that a mesh size $h_{\text{coarse}} = 0.0125$ is approximately needed. However, such triangles do not even fit

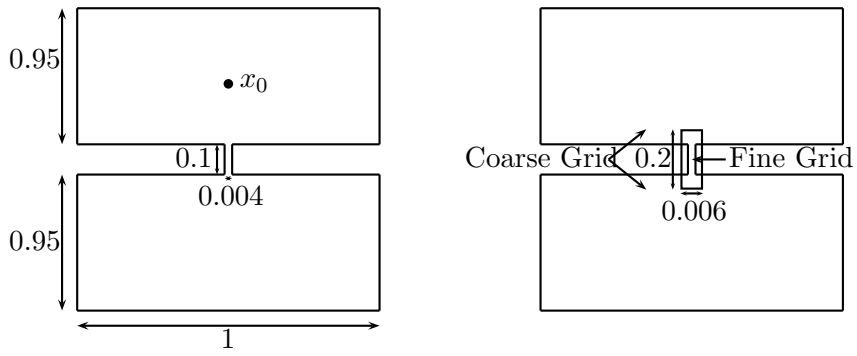

Figure 8: Two-dimensional example: the computational domain Ω.

inside the narrow gap, which requires $h_{\text{fine}} \simeq h_{\text{coarse}}/16.4$ to resolve its geometric features, as shown in Fig. 9.

For the time discretization we choose the fourth-order local time-stepping method from Section 4.1. Thus, the numerical method is fourth-order accurate in both space and time under the CFL restriction $\Delta t \leq 0.14h$, determined experimentally. If the same (global) time-step Δt were used everywhere in Ω, it would need to be about seventeen times smaller than necessary in most of Ω, for stability reasons only. Instead, we shall use the fourth-order local time-stepping with $p = 17$, which for every time step $\Delta t = 0.14h_{\text{coarse}}$ takes seventeen local time steps $\Delta\tau = \Delta t/17$ inside the highly refined region.

In dimension two, or higher, the boundary between the fine and coarse mesh is typically not as well-defined as in one space dimension while the transition between larger and smaller elements is more gradual. Here to define the *fine mesh*, that is the subregion of Ω where local time steps are required, we surround the narrow channel by a small $[-0.042\,;\,0.042] \times [-0.095\,;\,0.095]$ box, as shown in Fig. 8. Those triangles, whose center of gravity lies inside that box, belong to the fine mesh; the corresponding degrees of freedom in the finite element solution are then selected merely by setting the corresponding diagonal entries of the matrix P to one – see Section 3.1.

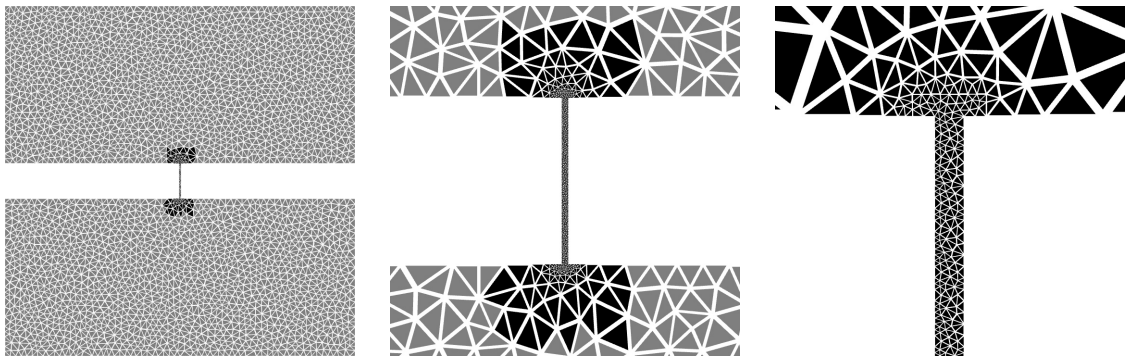

Figure 9: The triangular mesh at various magnification rates: the darker triangles belong to the "fine mesh" and the lighter triangles to the "coarse mesh".

In Fig. 10 snapshots of the numerical solution u are shown at times $t = 0.02$, 0.67, 0.13, 0.17, 0.24, 0.27. A circular wave is initiated by the Gaussian pulse centered about x_0 in the upper region, which propagates outward until it impinges on the lower boundary at $t = 0.67$.

Then, a fraction of the wave penetrates the channel and generates a circular outgoing wave as it reaches the opposite lower region. Further reflections occur as the wave moves back and forth inside the channel, subsequently generating multiple circular waves in the upper and lower domains.

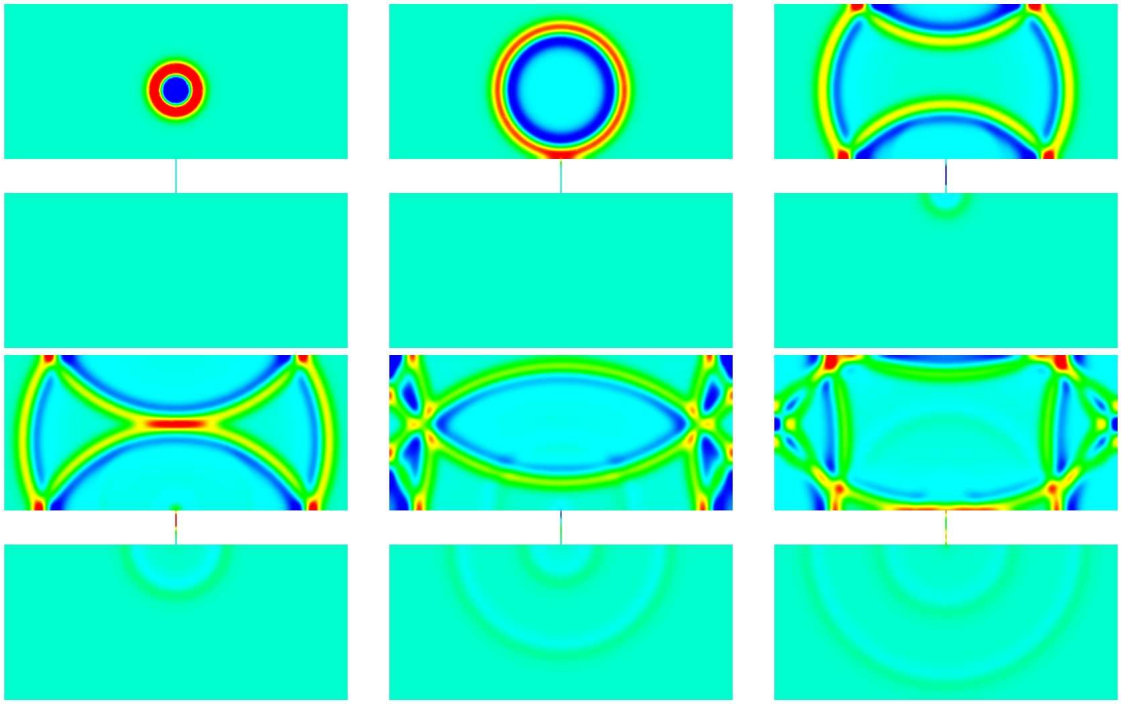

Figure 10: Two-dimensional example: the solution is shown at times t=0.02, 0.67, 0.13, 0.17, 0.24, 0.27.

Finally, we verify that the discrete numerical energy defined in (56) is truly conserved over time. In Fig. 11 we follow the time evolution of the energy and its relative variation, which remains within machine precision for the entire simulation until $T = 8.7$, that is during 5000 time-steps.

6 Conclusion

We have presented explicit local time-stepping methods for the wave equation, which allow arbitrarily small time-steps precisely where the smallest elements in the mesh are located. When combined with a symmetric finite element discretization in space with an essentially diagonal mass matrix, the resulting discrete time-marching scheme remains truly explicit, while it also conserves a discrete energy. Starting from the standard second-order "leap-frog" scheme, we have derived local time integration methods of arbitrary order. When the "fine" region, where local time-steps are used, slightly extends into the surrounding "coarse" region of the mesh, we find that the resulting numerical scheme permits the use of the optimal maximal time-step, dictated by the coarse mesh size.

Since the local time-stepping methods presented here are truly explicit, their parallel implementation is straightforward. Let Δt denote the time-step imposed by the CFL condition in the coarser part of the mesh. Then, during every (global) time-step Δt, each local time-step

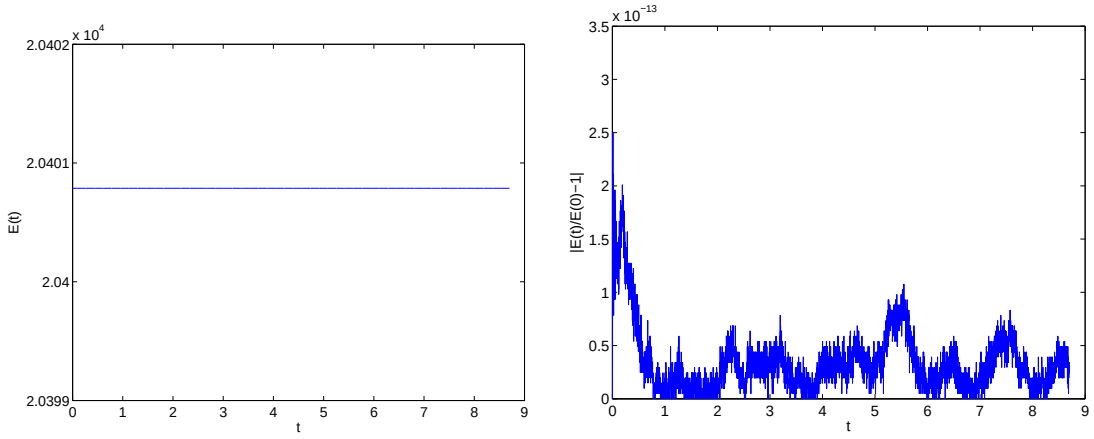

Figure 11: Time evolution of the discrete energy defined in (56) (left) and of its relative variation $|E(t)/E(0) - 1|$ (right).

of size $\Delta t/p$ inside the fine region of the mesh, with $p \geq 2$ any integer, simply corresponds to sparse matrix-vector multiplications that only involve the degrees of freedom associated with the fine region of the mesh. Those "fine" degrees of freedom can be selected individually and without any restriction by setting the corresponding entries in the diagonal projection matrix P to one; in particular, no adjacency or coherence in the numbering of the degrees of freedom is assumed. Hence the implementation is straightforward and requires no special data structures.

The local time-stepping methods derived here for the scalar wave equation immediately apply to more general second-order hyperbolic problems, as in elasticity or electromagnetics, for which either symmetric discontinuous Galerkin [23, 24] or mass lumping techniques are available [18]. They also generalize to the situation of nonzero forcing. Clearly these time-stepping schemes can also be combined with finite difference methods on highly stretched grids, such as body-fitted grids [40], if the underlying finite difference discretization leads to a symmetric stiffness matrix. In the presence of hierarchical mesh refinement, each local time-step in the fine region can itself include further local time-steps inside a smaller subregion with an even higher degree of local mesh refinement.

Acknowledgements We thank Ernst Hairer and Christian Lubich for useful comments and suggestions.

References

[1] M. Ainsworth, P. Monk, and W. Muniz. Dispersive and dissipative properties of discontinuous Galerkin finite element methods for the second-order wave equation. *J. Sci. Comput.*, 27(1-3):5–40, 2006.

[2] Laurent Anné, Patrick Joly, and Quang Huy Tran. Construction and analysis of higher order finite difference schemes for the 1D wave equation. *Comput. Geosci.*, 4(3):207–249, 2000.

[3] Douglas N. Arnold, Franco Brezzi, Bernardo Cockburn, and Luisa Donatella Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39:1749–1779, 2001.

[4] Garth A. Baker. Error estimates for finite element methods for second order hyperbolic equations. *SIAM J. Numer. Anal.*, 13(4):564–576, 1976.

[5] Garth A. Baker and Vassilios A. Dougalis. The effect of quadrature errors on finite element approximations for second order hyperbolic equations. *SIAM J. Numer. Anal.*, 13(4):577–598, 1976.

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

[7] Romuald Carpentier, Armel de La Bourdonnaye, and Bernard Larrouturou. On the derivation of the modified equation for the analysis of linear numerical methods. *RAIRO Modél. Math. Anal. Numér.*, 31(4):459–470, 1997.

[8] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North–Holland, Amsterdam, 1978.

[9] B. Cockburn. Discontinuous Galerkin methods for convection-dominated problems. In T. Barth and H. Deconink, editors, *High-Order Methods for Computational Physics*, volume 9 of *Lect. Notes Comput. Sci. Engrg.*, pages 69–224. Springer–Verlag, 1999.

[10] B. Cockburn, G.E. Karniadakis, and C.-W. Shu. The development of discontinuous Galerkin methods. In B. Cockburn, G.E. Karniadakis, and C.-W. Shu, editors, *Discontinuous Galerkin Methods: Theory, Computation and Applications*, volume 11 of *Lect. Notes Comput. Sci. Engrg.*, pages 3–50. Springer–Verlag, 2000.

[11] B. Cockburn and C.-W. Shu. Runge–Kutta discontinuous Galerkin methods for convection–dominated problems. *J. Sci. Comput.*, 16:173–261, 2001.

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

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

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

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

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

[17] Mark A. Dablain. High order differencing for the scalar wave equation. *SEG Technical Program Expanded Abstracts*, 3(1):854–854, 1984.

[18] A. Elmkies and P. Joly. Finite elements and mass lumping for Maxwell's equations: the 2D case. *C. R. Acad. Sci. Paris, Série I*, 324:1287–1293, 1997.

[19] S. Fauqueux. *Eléments finis mixtes spectraux et couches absorbantes parfaitement adaptées pour la propagation d'ondes élastiques en régime transitoire*. PhD thesis, Université Paris IX, 2003.

[20] Martin J. Gander, Laurence Halpern, and Frédéric Nataf. Optimal Schwarz waveform relaxation for the one dimensional wave equation. *SIAM J. Numer. Anal.*, 41(5):1643–1681 (electronic), 2003.

[21] Jean-Charles Gilbert and Patrick Joly. Higher order time stepping for second order hyperbolic problems and optimal cfl conditions. In *Numerical Analysis and Scientific Computing for PDE's and their Challenging Applications*, 2006.

[22] Marcus J. Grote, Anna Schneebeli, and Dominik Schötzau. Discontinuous Galerkin finite element method for the wave equation. *SIAM J. Numer. Anal.*, 44(6):2408–2431, 2006.

[23] Marcus J. Grote, Anna Schneebeli, and Dominik Schötzau. Interior penalty discontinuous Galerkin method for Maxwell's equations: energy norm error estimates. *J. Comput. Appl. Math.*, 204(2):375–386, 2007.

[24] Marcus J. Grote, Anna Schneebeli, and Dominik Schötzau. Interior penalty discontinuous Galerkin method for Maxwell's equations: optimal L^2-norm error estimates. *IMA J. Numer. Anal.*, 2007. (electronic).

[25] Bertil Gustafsson and Eva Mossberg. Time compact high order difference methods for wave propagation. *SIAM J. Sci. Comput.*, 26(1):259–271 (electronic), 2004.

[26] Bertil Gustafsson and Per Wahlund. Time compact high order difference methods for wave propagation, 2D. *J. Sci. Comput.*, 25(1-2):195–211, 2005.

[27] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric Numerical Integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2006. Structure-preserving algorithms for ordinary differential equations.

[28] Laurence Halpern. Local space-time refinement for the one-dimensional wave equation. *J. Comput. Acoust.*, 13(3):547–568, 2005.

[29] J. P. Hennart. Topics in finite element discretization of parabolic evolution problems. In J. P. Hennart, editor, *Numerical Analysis Proceedings of the Third IIMAS Workshop held at Cocoyoc, Mexico*, volume 909 of *Lecture Notes in Mathematics*, pages 185–199, 1982.

[30] T. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice Hall, 1987.

[31] Patrick Joly and Jerónimo Rodríguez. An error analysis of conservative space-time mesh refinement methods for the one-dimensional wave equation. *SIAM J. Numer. Anal.*, 43(2):825–859 (electronic), 2005.

[32] Heinz-Otto Kreiss, N. Anders Petersson, and Jacob Yström. Difference approximations for the second order wave equation. *SIAM J. Numer. Anal.*, 40(5):1940–1967 (electronic), 2002.

[33] Benedict Leimkuhler and Sebastian Reich. *Simulating Hamiltonian dynamics*, volume 14 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2004.

[34] J.L. Lions and E. Magenes. *Non-homogeneous Boundary Value Problems and Applications, Volume I.* Springer–Verlag, New York, 1972.

[35] Christian Lubich. Private communication.

[36] Yvon Maday and Anthony Patera. Spectral element methods for the incompressible Navier-Stokes equations. In A. Noor and J. T. Oden, editors, *State-of-the-Art Surveys in Computational Mechanics*, pages 71–143. ASME, New York, 1989.

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

[38] Géza Seriani and Enrico Priolo. Spectral element method for acoustic wave simulation in heterogeneous media. *Finite Elem. Anal. Des.*, 16(3-4):337–348, 1994. ICOSAHOM '92 held at Montpellier, France.

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

[40] Vianey Villamizar. Time-dependent numerical method with boundary-conforming curvilinear coordinates applied to wave interactions with prototypical antennas. *J. Comput. Phys.*, 177(1):1–36, 2002.

Unité de recherche INRIA Futurs
Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399