Runge-Kutta Based Explicit Local Time-Stepping Methods for Wave Propagation

Marcus J. Grote, Michaela Mehlin, Teodora Mitkova

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

Preprint No. 2014-05
February, 2014

www.math.unibas.ch
Abstract. Locally refined meshes severely impede the efficiency of explicit Runge-Kutta (RK) methods for the simulation of time-dependent wave phenomena. By taking smaller time-steps precisely where the smallest elements are located, local time-stepping (LTS) methods overcome the bottleneck caused by the stringent stability constraint of but a few small elements in the mesh. Starting from classical or low-storage explicit RK methods, explicit LTS methods of arbitrarily high accuracy are derived. When combined with an essentially diagonal finite element mass matrix, the resulting time-marching schemes retain the high accuracy, stability and efficiency of the original RK methods while circumventing the geometry-induced stiffness. Numerical experiments with continuous and discontinuous Galerkin finite element discretizations corroborate the expected rates of convergence and illustrate the usefulness of these LTS-RK methods.

Key words. finite element methods, discontinuous Galerkin methods, explicit time integration, time-marching, multirate methods, hyperbolic problems

AMS subject classifications. 65N30

1. Introduction. The efficient numerical simulation of time-dependent wave phenomena is of fundamental importance in acoustic, electromagnetic or seismic wave propagation. In the presence of heterogeneous media or complex geometry, finite element methods (FEM), be they continuous or discontinuous, are increasingly popular due to their inherent flexibility: elements can be small precisely where small features are located, and larger elsewhere. Local mesh refinement, however, also imposes severe stability constraints on explicit time integration, as the maximal time-step 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 generally too high a price to pay. Local time-stepping (LTS) methods alleviate that geometry induced stability restriction by dividing the elements 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 [32, 2] and the references therein. In 2006, Piperno [33] combined the explicit leap-frog (LF) with the implicit Crank-Nicolson (CN) scheme for a nodal discontinuous Galerkin (DG) discretization of Maxwell’s equations in a non-conducting medium. Here, a linear system needs to be solved inside the refined region at every time-step. Although each method is time accurate of order two, the implicit-explicit component splitting can reduce by one the order of convergence of the resulting scheme when both the time-step and the mesh size decrease simultaneously [14, 12]. By using the LF/CN-IMEX approach of Verwer [36] instead, Descombes, Lanteri and Moya [12] remedy that unexpected loss in accuracy and hence recover second-order convergence, yet at the
price of a significantly larger albeit sparse linear system. To achieve higher accuracy, Kanevsky et al. [27] apply a fourth-order IMEX Runge-Kutta (RK) method [29] to nodal DG discretizations in fluid flow. The nonlinear system associated with the "fine" elements needs to be solved by a Newton-Krylov method at every time-step and becomes increasingly ill-conditioned as the grid-induced stiffness increases.

In contrast, explicit LTS methods overcome the crippling effect of local refinement by using smaller time-steps inside the refined region, while remaining fully explicit in the entire computational domain. In [16], the energy conserving, second-order LTS approach from Collino et al. [8, 9] was combined with a DG-FE discretization for the numerical solution of symmetric first-order hyperbolic systems. That LTS approach is explicit inside the coarse and the fine mesh; at the interface, however, it nonetheless requires the solution of a linear system at every time-step. In [33], Piperno also proposed a fully explicit second-order Störmer-Verlet LTS scheme for Maxwell’s equations in first-order formulation; while it preserves a discrete form of the electromagnetic energy in a non-conducting medium, it is also limited to second-order accuracy. Starting from the standard LF method, energy conserving fully explicit LTS integrators of arbitrarily high accuracy were proposed for the classical wave equation [13] and Maxwell’s equations [17] in second-order formulation.

Based on the arbitrary high-order derivatives (ADER) DG approach, alternative explicit LTS methods for Maxwell’s equations and for elastic wave equations were proposed in [35, 15]. Constantinescu et al. and later Krivodonova devised multirate explicit methods for hyperbolic conservation laws, which are based on Runge-Kutta [11, 30] schemes; again these schemes are limited to second-order accuracy. Multirate schemes based on first and second order explicit RK methods were analyzed in [26] to achieve arbitrarily high accuracy in the presence of dissipation, while remaining fully explicit, LTS methods for damped wave equations based on Adams-Bashforth multi-step schemes were derived in [18]; they can also be interpreted as particular approximations of exponential-Adams multi-step methods [23]. Recently, Angulo et al. proposed a "causal-path" LTS technique for both the second-order LF and a fourth-order low-storage (LS) explicit RK method [7], which assorts elements in tiers according to their size and computes required intermediate values recursively [1].

Here, we propose explicit LTS methods based on standard explicit RK schemes of arbitrarily high accuracy. Since RK methods are one-step methods, they do not require a starting procedure and easily accommodate adaptivity in time. Clearly, the idea of using different time-steps for different components in the context of explicit RK methods is not new [20, 34]. However, RK methods achieve higher accuracy not by extrapolating farther from the (known) past, but instead by adding intermediate values from the current time-step only. Hence for partial differential equations, the derivation of high-order explicit LTS methods that are based on RK schemes is generally more difficult, and only little work on explicit LTS-RK methods is available.

The rest of our paper is organized as follows. Following the method of lines approach, we consider in Section 2 the wave equation as a model problem and rewrite its standard semi-discrete Galerkin FE formulation as a general first-order system of differential equations. In Section 3, we derive explicit LTS methods of arbitrarily high accuracy by starting from standard or low-storage RK methods. Next, we prove in Section 4 that those LTS-RK methods indeed preserve the accuracy of their original single time-step counterparts. Finally, numerical experiments that illustrate the expected convergence and stability properties of our LTS schemes are presented in Section 5.
2. Finite element discretizations of wave equations. We consider the wave equation

\[
\begin{align*}
    u_{tt} + \sigma u_t - \nabla \cdot (c^2 \nabla u) &= f & \text{in } \Omega \times (0, T), \\
    u(\cdot, t) &= 0 & \text{on } \partial \Omega \times (0, T), \\
    u(\cdot, 0) &= u_0, \quad u_t(\cdot, 0) = v_0 & \text{in } \Omega,
\end{align*}
\]  

(2.1)

a standard model problem for wave phenomena. Here \( \Omega \) is a bounded domain in \( \mathbb{R}^d \), \( f \in L^2(0, T; L^2(\Omega)) \) is a (known) source term, while \( u_0 \in H^1_0(\Omega) \) and \( v_0 \in L^2(\Omega) \) are prescribed initial conditions. At the boundary, \( \partial \Omega \), we impose a homogeneous Dirichlet boundary condition, for simplicity. The damping coefficient, \( \sigma = \sigma(x) \), is assumed non-negative \((\sigma \geq 0)\) whereas the speed of propagation, \( c = c(x) \), is piecewise smooth and strictly positive \((c(x) \geq c_0 > 0)\). If \( \sigma \) is identically zero throughout \( \Omega \), then (2.1) coincides with the classical (undamped) wave equation.

Various finite element methods (FEM) are available for the spatial discretization of (2.1). For instance, the standard \( H^1 \)-conforming FEM with mass-lumping starts from the weak formulation: Find \( u : [0, T] \to H^1_0(\Omega) \) such that

\[
\begin{align*}
    (u_{tt}, v) + (\sigma u_t, v) + (c \nabla u, c \nabla v) &= (f, v) & \forall v \in H^1_0(\Omega), \quad t \in (0, T), \\
    u_{|t=0} &= u_0 & \text{in } \Omega, \\
    u_t_{|t=0} &= v_0 & \text{in } \Omega,
\end{align*}
\]

(2.2)

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 \( \Omega \) into disjoint elements \( K \), i.e. \( \Omega = \cup_{K \in \mathcal{T}_h} K \); for simplicity, we assume that \( \Omega \) is polygonal. The diameter of element \( K \), a triangle or a quadrilateral in two space dimension, and a tetrahedron or hexahedron in three dimensions, is denoted by \( h_K \); hence, the mesh size, \( h \), is given by \( h = \max_{K \in \mathcal{T}_h} h_K \). We also let \( V_h \subset H^1_0(\Omega) \) denote the finite dimensional subspace

\[
V_h = \{ v \in H^1_0(\Omega) : v|_K \in \mathcal{P}_\ell(K), \quad \forall K \in \mathcal{T}_h \},
\]

where \( \mathcal{P}_\ell(K) \) corresponds to the space \( \mathcal{P}_\ell(K) \) of polynomials of total degree at most \( \ell \), if \( K \) is a triangle or tetrahedron, or the space \( \mathcal{Q}_\ell(K) \) of polynomials of maximal degree \( \ell \) 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 \( 0 \leq t < T \) by the restriction of (2.2) to \( V_h \). Let \( U(t) \in \mathbb{R}^N \) denote the coefficients of \( u^h(t) \) with respect to the standard Lagrangian basis \( \{ \phi_i \}_{i=1,...,N} \) of \( V_h \). Then, \( U(t) \) satisfies

\[
\begin{align*}
    \mathbf{M} \frac{d^2 U}{dt^2}(t) + \mathbf{M}_\sigma \frac{dU}{dt}(t) + \mathbf{K} U(t) &= \mathbf{R}(t), & t \in (0, T), \\
    \mathbf{M} U(0) &= u_0^h, \quad \mathbf{M} \frac{dU}{dt}(0) = v_0^h,
\end{align*}
\]

(2.3)

where \( u_0^h, v_0^h \) are suitable approximations to the initial conditions. Moreover, the mass matrix, \( \mathbf{M} \), and the stiffness matrix, \( \mathbf{K} \), are given by

\[
\mathbf{M}_{ij} = (\phi_j, \phi_i), \quad \mathbf{K}_{ij} = (c \nabla \phi_j, c \nabla \phi_i);
\]

the matrix \( \mathbf{M}_\sigma \) also corresponds to a mass matrix with weight \( \sigma \). The matrix \( \mathbf{M} \) is sparse, symmetric and positive definite, whereas the matrices \( \mathbf{K} \) and \( \mathbf{M}_\sigma \) are sparse, symmetric but, in general, only positive semi-definite.
Even though explicit numerical time integration may be applied directly to (2.3), 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 [10, 31], which replace $M$ by a diagonal approximation without spoiling the accuracy [3]. Alternatively, the spectral element method [6, 28] and the symmetric interior penalty discontinuous Galerkin (DG) method [19] 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 (2.3) with an essentially diagonal mass matrix $M$.

To apply a Runge-Kutta method to (2.3), we first need to rewrite it as a first-order system. Since $M$ in (2.3) is symmetric, positive definite and assumed diagonal, the matrix $M^{-\frac{1}{2}}$ is immediately available. Then, multiplication of (2.3) by $M^{-\frac{1}{2}}$ yields

\[
\frac{d^2z}{dt^2}(t) + D\frac{dz}{dt}(t) + Az(t) = \tilde{R}(t),
\]

with

\[
z(t) = M^{\frac{1}{2}}U(t), \quad D = M^{-\frac{1}{2}}M_sM^{-\frac{1}{2}}, \quad A = M^{-\frac{1}{2}}M_sM^{-\frac{1}{2}}, \quad \tilde{R}(t) = M^{-\frac{1}{2}}R(t).
\]

Again, the matrix $A$ is sparse, symmetric and positive semi-definite. We now rewrite (2.4) as a first-order system

\[
\frac{dy}{dt}(t) = By(t) + F(t), \quad t \in (0, T),
\]

\[
y(0) = y_0,
\]

where we have introduced

\[
y(t) = \left( z(t), \frac{dz}{dt}(t) \right)^T, \quad B = \begin{pmatrix} 0 & I \\ -A & -D \end{pmatrix}, \quad F(t) = \begin{pmatrix} 0 \\ \tilde{R}(t) \end{pmatrix}.
\]

If none of the above but instead the nodal DG approach from [22] is used, the semi-discrete Galerkin formulation immediately leads to (2.5), as its derivation starts from (2.1) reformulated as a first-order hyperbolic system.

3. Runge-Kutta based LTS. We shall now derive high-order explicit LTS methods for (2.5), which are based on explicit RK schemes. Although we concentrate on classical and low-storage RK schemes, probably the most popular methods for the time integration of partial differential equations, the derivation below also applies to any other explicit RK variant [7, 21, 37].

3.1. Explicit RK based LTS methods. Starting from an explicit $s$-stage RK (RKs) method of order $k$ [21], we now derive an explicit LTS scheme of the same accuracy for (2.5). Let $y_n$ denote the numerical approximation to the exact solution $y(t_n)$ at $t_n = n\Delta t$ and $F_n = F(t_n)$. A general explicit RKs method applied to (2.5)
then yields
\begin{align*}
    k_1 &= B y_n + F(t_n), \\
    k_2 &= B(y_n + \Delta t a_{21} k_1) + F(t_n + c_2 \Delta t), \\
    & \vdots \\
    k_s &= B \left( y_n + \Delta t \sum_{i=1}^{s-1} a_{si} k_i \right) + F(t_n + c_s \Delta t), \\
    y_{n+1} &= y_n + \Delta t \sum_{i=1}^{s} b_i k_i.
\end{align*}

(3.1)

The constants \(a_{ij}, b_i, c_i\), with \(c_1 = 0\), uniquely identify the RK method and are typically listed in a Butcher-tableau, as in Table 3.1.

\begin{table}[h]
\centering
\caption{Butcher-tableau of an explicit RK scheme.}
\begin{tabular}{cccc}
\hline
0  &  &  &  \\
\hline
\v1  & a_{21}  &  &  \\
\v2  & a_{31}  & a_{32}  &  \\
\vdots  & \vdots  & \vdots  & \ddots  \\
\v{s}  & a_{s1}  & a_{s2}  & \ldots  & a_{s,s-1}  \\
\hline
b_1  & b_2  & \ldots  & b_{s-1}  & b_s  \\
\hline
\end{tabular}
\end{table}

If the RKs method has order \(k\), with \(s \geq k\), then \(b_i, c_i, i = 1, \ldots, s\) must satisfy
\[ \sum_{i=1}^{s} b_i c_i^{q-1} = \frac{1}{q}, \quad q = 1, \ldots, k. \]

(3.2)

Hence the underlying quadrature formula with weights \(b_1, \ldots, b_s\) and nodes \(c_1, \ldots, c_s\) also has at least order \(k\).

We now wish to derive an RKs based LTS scheme for (2.5), which permits arbitrarily small time-steps precisely where the smaller elements in the FE mesh are located. To do so, we first partition the vectors \(y(t)\) and \(F(t)\) as
\begin{align*}
    y(t) &= (I - P)y(t) + Py(t) = y^{c}[t] + y^{f}[t], \\
    F(t) &= (I - P)F(t) + PF(t) = F^{c}[t] + F^{f}[t],
\end{align*}

(3.3)

where the matrix \(P\) is diagonal. Its diagonal entries, equal to zero or one, identify the unknowns associated with the locally refined region, where smaller time-steps are needed. The exact solution of (2.5) is given by
\begin{align*}
    y(t_n + \xi \Delta t) &= y(t_n) + \int_{t_n}^{t_n + \xi \Delta t} B y^{c}[t] + F^{c}[t] \, dt \\
                        &+ \int_{t_n}^{t_n + \xi \Delta t} B y^{f}[t] + F^{f}[t] \, dt, \quad 0 \leq \xi \leq 1.
\end{align*}

(3.4)

Hence, any numerical method for the time integration of (2.5) in fact approximately evaluates the right side of (3.4).
Inside the coarse region, the LTS method we seek ought to coincide with the initial \( k \)th-order RKs method. Therefore, we approximate in (3.4) the term involving \( y^{[c]}(t) \) by the corresponding quadrature formula
\[
\int_{t_n}^{t_n + \xi \Delta t} B y^{[c]}(t) \, dt \simeq \xi \Delta t B \left( \sum_{i=1}^{s} b_i y^{[c]}(t_n + c_i \xi \Delta t) \right). \tag{3.5}
\]
Next, we approximate the (unknown) values of \( y \) at the quadrature points by Taylor expansion:
\[
\int_{t_n}^{t_n + \xi \Delta t} B(I - P)y(t) \, dt \simeq \xi \Delta t B(I - P) \left( \sum_{i=1}^{s} b_i \sum_{j=0}^{s-1} \frac{c_i^j(\xi \Delta t)^j}{j!} y^{(j)}(t_n) \right). \tag{3.6}
\]
Repeated use of (2.5) to evaluate the derivatives \( y^{(j)} \) of \( y \) in (3.6) then leads to
\[
\int_{t_n}^{t_n + \xi \Delta t} B(I - P)y(t) \, dt \simeq \xi \Delta t B(I - P) \left( \sum_{i=1}^{s} b_i \sum_{j=0}^{s-1} \frac{c_i^j(\xi \Delta t)^j}{j!} \left( B y_n + \sum_{\ell=1}^{j} B^{j-\ell} F^{(\ell-1)}(t_n) \right) \right)
+ \sum_{\ell=1}^{j} B^{j-\ell} F^{(\ell-1)}(t_n)). \tag{3.7}
\]
To avoid the derivatives of \( F(t) \) in (3.7), we now interpolate \( F(t) \) by a polynomial \( q(t) \) through the points \((t_n + c_i \Delta t, F(t_n + c_i \Delta t)), i = 1, \ldots, s\). Since the nodes \( c_i, \quad i = 1, \ldots, s \) need not be distinct (e.g. \( c_2 = c_3 = 1/2 \) for RK4, see Table A.1 in the Appendix), the degree of \( q \) may be strictly less then \( s - 1 \). We now replace the derivatives of \( F \) in (3.7) by the corresponding derivatives of \( q \) to obtain
\[
y(t_n + \xi \Delta t) \simeq y_n + B(I - P) \left( \sum_{i=1}^{s} b_i \sum_{j=0}^{s-1} \frac{c_i^j(\xi \Delta t)^j}{j!} \left( B y_n + \sum_{\ell=1}^{j} B^{j-\ell} q^{(\ell-1)}(t_n) \right) \right)
+ (I - P)(q(t_n + \xi \Delta t) - \hat{q}(t_n)) + \int_{t_n}^{t_n + \xi \Delta t} (B P y(t) + P F(t)) \, dt, \tag{3.8}
\]
where \( \hat{q}'(t) = q(t) \). Since \( F \) is known, so are \( q \) and \( \hat{q} \), and thus all terms needed in (3.8) to advance the solution inside the coarse region, that is those involving \( (I - P) \), are also explicitly known.

As we seek to compute \( y(t_n + \Delta t) \), we need to evaluate the entire right side of (3.8). To circumvent the severe stability restriction on \( \Delta t \) dictated by the smallest elements in the mesh, we shall treat \( y^{[0]}(t) \) and \( F^{[0]}(t) \) differently from \( y^{[c]}(t) \) and \( F^{[c]}(t) \). Hence, we now approximate the remaining integral in (3.8) as
\[
\int_{t_n}^{t_n + \xi \Delta t} B P y(t) + P F(t) \, dt \simeq \int_{0}^{\xi \Delta t} B P \tilde{y}(\tau) + P F(t_n + \tau) \, d\tau,
\]
where \( \tilde{y}(\tau) \) solves the following differential equation for \( 0 < \tau \leq \Delta t \):
\[
\frac{d\tilde{y}}{d\tau} = B(I - P) \left[ \sum_{i=1}^{s} b_i \sum_{j=0}^{s-1} \frac{(j + 1)c_i^j \tau^j}{j!} \left( B y_n + \sum_{\ell=1}^{j} B^{j-\ell} q^{(\ell-1)}(t_n) \right) \right]
+ (I - P)q(t_n + \tau) + BP \tilde{y}(\tau) + PF(t_n + \tau), \tag{3.9}
\]
\( \tilde{y}(0) = y_n \).
Its exact solution is given by
\[
\tilde{y}(\xi \Delta t) = y_n + B(I - P) \left( \sum_{i=1}^{s} b_i \sum_{j=0}^{s-1} \frac{c_i^j (\xi \Delta t)^{j+1}}{j!} \left( B^j y_n + \sum_{\ell=1}^{j} B^{j-\ell} q^{(\ell-1)}(t_n) \right) \right) + (I - P) (q(t_n + \xi \Delta t) - q(t_n)) + \int_0^{\xi \Delta t} B P \tilde{y}(\tau) + P F(t_n + \tau) d\tau.
\] (3.10)

Since the right side of (3.10) coincides with that of (3.8) with \(P y(t)\) replaced by \(P \tilde{y}(t)\) inside the integral, we immediately infer that \(y(t_n + \xi \Delta t) \simeq \tilde{y}(\xi \Delta t)\).

To advance \(y\) from \(t_n\) to \(t_n + \Delta t\), we shall therefore evaluate \(\tilde{y}(\Delta t)\) by solving (3.9) on \([0, \Delta t]\) numerically. Here, we again use the RKs method of order \(k\), though with a smaller time-step \(\Delta \tau = \Delta t/p\), where \(p\) denotes the ratio of local mesh refinement. Clearly, in doing so we must ensure that the overall numerical scheme remains truly \(k\)-th-order accurate, as we shall show in Section 4. In summary, given \(y_n\) the \(k\)-th-order RKs based LTS algorithm for the solution of (2.5) computes \(y_{n+1} \simeq y(t_n + \Delta t)\), as follows:

**Algorithm 1. LTS-RKs(\(p\))**

1. Set \(\tilde{y}_0 := y_n\).
2. For \(j = 0, \ldots, s - 1\) compute

\[
w_{n,j} := \frac{(j + 1)}{j!} B(I - P) \sum_{i=1}^{s} b_i c_i^j \left( B^j y_n + \sum_{\ell=1}^{j} B^{j-\ell} q^{(\ell-1)}(t_n) \right).
\] (3.11)

3. For \(m = 0, \ldots, p - 1\) compute

\[
k_{r, \frac{m+1}{p}} := \sum_{j=0}^{s-1} ((m + c_r) \Delta \tau)^j w_{n,j} + (I - P) q(t_n + (m + c_r) \Delta \tau)
+ B P \left( \tilde{y}_{\frac{m}{p}} + \Delta \tau \sum_{i=1}^{r-1} a_r k_{r, \frac{m+1}{p}} \right) + P F_{n,m+c_r}, \quad r = 1, \ldots, s,
\]

\[
\tilde{y}_{\frac{m+1}{p}} := \tilde{y}_{\frac{m}{p}} + \Delta \tau \sum_{i=1}^{s} b_i k_{i, \frac{m+1}{p}}.
\]

4. Set \(y_{n+1} := \tilde{y}_1\).

Here, we have introduced the notation \(F_{n,m+c_r} = F(t_n + \tau_{m+c_r})\), where \(\tau_{m+c_r} = (m + c_r) \Delta \tau\); note that \(F_{n,0} = F(t_n + \tau_0) = F(t_n) = F_n\). Steps 1–3 correspond to the numerical solution of (3.9) until \(\tau = \Delta t\) by using the \(k\)-th-order RKs scheme with local time-step \(\Delta \tau = \Delta t/p\). In Step 2, we precompute all \(s\) multiplications with \(B(I - P)\). The remaining \(s \times p\) multiplications with \(BP\) in Step 3 only affect those unknowns located inside the fine part of the mesh, or immediately next to it; hence, their computational cost is proportional to the number of unknowns associated with the locally refined region only. In that sense, Algorithm 1 corresponds to a local time-stepping method. In Step 3, the term \((I - P)q(t_n + (m + c_r)\Delta \tau)\) appears inside the inner for-loops only for ease of notation. In fact given any specific RK-method, we can explicitly calculate \(q\) and then add at once its \(j\)-th order term involving \((m + c_r)\Delta \tau)^j\) to \(w_{n,j}\) in Step 2 – e.g. Algorithm 3 for LTS-RK4 in Section 3.3.
3.2. Low-storage RK based LTS methods. Following the derivation in the previous section, we now delineate the main steps in devising an LTS method based on an s-stage LSRK (LSRKs) scheme of order $k$. A general explicit $2N$-storage LSRKs method [7, 37] applied to (2.5) advances the solution from $t_n$ to $t_{n+1}$ as

$$z_0 = y_n, \quad k_r = A_r k_{r-1} + \Delta t (B z_{r-1} + F(t_n + C_r \Delta t)), \quad r = 1, \ldots, s,$$

$$z_r = z_{r-1} + B_r k_r, \quad r = 1, \ldots, s,$$

$$y_{n+1} = z_s,$$  \hspace{1cm} (3.12)

with $A_1 = 0$ for a self-starting algorithm. Note that $k_r$ and $z_r$ only depend on quantities from the previous stage $r - 1$. Every LSRK scheme can be put into the standard RK form (3.1) through variable substitutions

$$A_r = (b_{r-1} - B_{r-1})/b_r, \quad r = 2, \ldots, s, \quad b_r \neq 0,$$

$$A_r = (a_{r+1,r-1} - c_r)/B_r, \quad r = 2, \ldots, s, \quad b_r = 0,$$

$$B_r = a_{r+1,r}, \quad r = 1, \ldots, s - 1,$$

$$B_s = b_s,$$

$$C_r = c_r, \quad r = 1, \ldots, s,$$  \hspace{1cm} (3.13)

but not vice-versa – see [7, 37] for various sets of values for $A_r$, $B_r$ and $C_r$ with $s \leq 5$.

Starting from the LSRKs scheme (3.12), we thus first calculate the corresponding standard RK constants $a_{ij}$, $b_i$ and $c_i$ from (3.13). Following the derivation in Section 3.1, we then evaluate $\tilde{y}(\Delta t)$ by solving (3.9) until $\tau = \Delta t$ to advance $y$ from $t_n$ to $t_{n+1}$, though now by applying the LSRKs scheme with smaller time-step $\Delta \tau = \Delta t/p$.

In summary, given $y_n$, the $k$th-order LSRKs based LTS algorithm for the solution of (2.5) computes $y_{n+1} \simeq y(t_n + \Delta t)$, as follows:

**Algorithm 2. LTS-LSRKs($p$)**

1. Set $\tilde{y}_0 := y_n$.
2. For $j = 0, \ldots, s - 1$ compute $w_{n,j}$ as in (3.11).
3. For $m = 0, \ldots, p - 1$ compute
   $$z_{0,\frac{m+1}{p}} := \tilde{y}_{\frac{m+1}{p}};$$
   for $r = 1, \ldots, s$ compute
   $$k_{\frac{r-1}{p}, \frac{m+1}{p}} := A_r k_{\frac{r-1}{p}, \frac{m+1}{p}} + \Delta \tau \left[ \sum_{j=0}^{s-1} ((m + C_r) \Delta \tau)^j w_{n,j} + (I - P)q(t_n + (m + C_r) \Delta \tau) \right. \left. + BP z_{\frac{r-1}{p}, \frac{m+1}{p}} + PF_{n,m+C_r} \right];$$
   $$z_{\frac{r-1}{p}, \frac{m+1}{p}} := z_{\frac{r-1}{p}, \frac{m+1}{p}} + B_r k_{\frac{r-1}{p}, \frac{m+1}{p}};$$
   $$\tilde{y}_{\frac{m+1}{p}} := z_{\frac{m+1}{p}};$$
4. Set $y_{n+1} := \tilde{y}_1$.

Here, Steps 1–3 correspond to the numerical solution of (3.9) until $\tau = \Delta t$ by using the LSRKs scheme of order $k$ with the smaller time-step $\Delta \tau = \Delta t/p$. 
For $k = s$, we observe that (3.11) reduces to

\[ w_{n,j} := \frac{1}{j!} B(I - P) \left( B^j y_n + \sum_{\ell=1}^{j} B^{j-\ell} q^{(\ell-1)}(t_n) \right) \]

because of (3.2); then, the constants $b_r$ and $c_r$ are not needed and the LTS-LSRKs($p$) algorithm only uses the coefficients $A_r$, $B_r$ and $C_r$ of the underlying LSRKs scheme. In general, however, Step 2 of Algorithm 2 also requires the values for $b_r$ and $c_r$ of the corresponding standard RKs scheme, given by (3.13).

**Remark 1.** If the fraction of nonzero entries in $P$ is small, the overall cost of the LTS-RKs($p$) and LTS-LSRKs($p$) algorithms is dominated by the computation of the $s$ vectors $w_{n,j}$, $j = 0, \ldots, s - 1$, in Step 2 which requires $s$ multiplications by $B(I - P)$ per time-step $\Delta t$. All further $s \times p$ matrix-vector multiplications by $BP$ only affect those unknowns that lie inside the refined region, or immediately next to it; hence, their computational cost remains negligible as long as the locally refined region contains a small part of the computational domain.

### 3.3. The RK4 based LTS method

In Section 3.1, we have shown how to derive an LTS-RKs($p$) starting from an arbitrary $k$-th order explicit RK method. Because of its popularity, we shall now present in detail the LTS method based on the classical explicit RK4 scheme with coefficients listed in Table A.1 in the Appendix.

Following the previous derivation, we first split the vectors $y(t)$ and $F(t)$ as in (3.3) and approximate the term in (3.4) involving $y^{[c]}(t)$ by Simpson quadrature. Next, we approximate in (3.5) the values of $y_{n+\frac{1}{2}}$ and $y_{n+\xi}$, still unknown at time $t = t_n$, by their Taylor expansions up to $O(\Delta t^4)$. We also interpolate the points $(t_n, F_n)$, $(t_{n+\frac{1}{2}}, F_{n+\frac{1}{2}})$ and $(t_{n+1}, F_{n+1})$ by the quadratic polynomial,

\[ q(t_n + \tau) = F_n + \frac{\tau}{\Delta t} \left( -3F_n + 4F_{n+\frac{1}{2}} - F_{n+1} \right) + \frac{\tau^2}{2\Delta t^2} \left( 4F_n - 8F_{n+\frac{1}{2}} + 4F_{n+1} \right). \]

Thus, we can integrate exactly the term in (3.4) involving $F^{[c]}(t)$ and also explicitly evaluate the derivatives of $q$ to approximate $F'(t_n)$ and $F''(t_n)$. The differential equation (3.9) for $\vec{y}(\tau)$ is now given by

\[ \frac{dy}{d\tau}(\tau) = B(I - P) \left[ y_n + \tau (By_n + F_n) \right. \]

\[ + \frac{\tau^2}{2} \left( B^2 y_n + BF_n + \frac{-3F_n + 4F_{n+\frac{1}{2}} - F_{n+1}}{\Delta t} \right) \]

\[ + \frac{\tau^3}{6} \left( B^3 y_n + B^2 F_n + B \frac{-3F_n + 4F_{n+\frac{1}{2}} - F_{n+1}}{\Delta t} + \frac{4F_n - 8F_{n+\frac{1}{2}} + 4F_{n+1}}{\Delta t^2} \right) \]

\[ \left. + (I - P) \left[ F_n + \tau \frac{-3F_n + 4F_{n+\frac{1}{2}} - F_{n+1}}{\Delta t} + \frac{\tau^2}{2} \frac{4F_n - 8F_{n+\frac{1}{2}} + 4F_{n+1}}{\Delta t^2} \right] \right] \]

\[ + BP \vec{y}(\tau) + PF(t_n + \tau), \]

\[ \vec{y}(0) = y_n. \]

Again to advance $y$ from $t_n$ to $t_n + \Delta t$, we shall solve (3.14) by using the RK4 scheme with a smaller time-step $\Delta \tau = \Delta t/p$. In summary, given $y_n$, the LTS algorithm based on the classical explicit RK4 method for the solution of (2.5) computes $y_{n+1} \approx y(t_n + \Delta t)$ as follows:
Algorithm 3. LTS-RK₄(p)
1. Set $\tilde{y}_0 := y_n$.
2. Compute

$$w_{n,0} := B(I - P) y_n + (I - P) F_n,$$

$$w_{n,1} := B(I - P)((By_n + F_n) + (I - P) \left( -\frac{3F_n + 4F_{n+\frac{1}{2}} - F_{n+1}}{\Delta t} \right),$$

$$w_{n,2} := B(I - P) \left( B^2 y_n + BF_n + \frac{-3F_n + 4F_{n+\frac{1}{2}} - F_{n+1}}{\Delta t} \right)$$

$$+ (I - P) \left( \frac{4F_n - 8F_{n+\frac{1}{2}} + 4F_{n+1}}{\Delta t^2} \right),$$

$$w_{n,3} := B(I - P) \left( B^4 y_n + B^3 F_n + B \left( \frac{-3F_n + 4F_{n+\frac{1}{2}} - F_{n+1}}{\Delta t} \right)$$

$$+ \frac{4F_n - 8F_{n+\frac{1}{2}} + 4F_{n+1}}{\Delta t^2} \right).$$

3. For $m = 0, \ldots, p - 1$, compute

$$k_{1, \frac{m+1}{p}} := w_{n,0} + m \Delta \tau \ w_{n,1} + \frac{m^2}{2} \Delta \tau^2 w_{n,2} + \frac{m^3}{6} \Delta \tau^3 w_{n,3}$$

$$+ BP \tilde{y}_{\frac{m}{p}} + PF_{n,m},$$

$$k_{2, \frac{m+1}{p}} := w_{n,0} + \left( m + \frac{1}{2} \right) \Delta \tau \ w_{n,1} + \frac{1}{2} \left( m + \frac{1}{2} \right)^2 \Delta \tau^2 w_{n,2}$$

$$+ \frac{1}{6} \left( m + \frac{1}{2} \right)^3 \Delta \tau^3 w_{n,3} + BP \left( \tilde{y}_{\frac{m}{p}} + \frac{\Delta \tau}{2} k_{1, \frac{m+1}{p}} \right) + PF_{n,m+\frac{1}{2}},$$

$$k_{3, \frac{m+1}{p}} := w_{n,0} + \left( m + \frac{1}{2} \right) \Delta \tau \ w_{n,1} + \frac{1}{2} \left( m + \frac{1}{2} \right)^2 \Delta \tau^2 w_{n,2}$$

$$+ \frac{1}{6} \left( m + \frac{1}{2} \right)^3 \Delta \tau^3 w_{n,3} + BP \left( \tilde{y}_{\frac{m}{p}} + \frac{\Delta \tau}{2} k_{2, \frac{m+1}{p}} \right) + PF_{n,m+\frac{1}{2}},$$

$$k_{4, \frac{m+1}{p}} := w_{n,0} + (m + 1) \Delta \tau \ w_{n,1} + \frac{1}{2} (m + 1)^2 \Delta \tau^2 w_{n,2}$$

$$+ \frac{1}{6} (m + 1)^3 \Delta \tau^3 w_{n,3} + BP \left( \tilde{y}_{\frac{m}{p}} + \Delta \tau k_{3, \frac{m+1}{p}} \right) + PF_{n,m+1},$$

$$\tilde{y}_{\frac{m+1}{p}} := \tilde{y}_{\frac{m}{p}} + \frac{1}{6} \Delta \tau \left( k_{1, \frac{m+1}{p}} + 2k_{2, \frac{m+1}{p}} + 2k_{3, \frac{m+1}{p}} + k_{4, \frac{m+1}{p}} \right).$$

4. Set $y_{n+1} := \tilde{y}_1$.

Steps 1–3 compute the numerical solution of (3.14) at time $\tau = \Delta t$ by using the classical RK4 scheme with local time-step $\Delta \tau = \Delta t/p$. In contrast to Algorithm 1 for a general RKs method, we have precomputed the $(I - P)q$ values at intermediate times and already added them to $w_{n,j}$.

4. Accuracy and Convergence. Starting from an arbitrary explicit RK scheme of order $k$, we have shown in Section 3 how to derive an explicit LTS method from it. We shall now prove that the resulting LTS-RK method indeed preserves the accuracy of the original RK scheme and is also convergent of order $k$. Hence we consider a general explicit RKs method (3.1) of order $k$ and denote by $\{c_1, \ldots, c_{k_0}\} \subset \{c_1, \ldots, c_s\}$
the maximal subset of all coefficients $c_i$ such that no two are identical, i.e. $\bar{c}_i \neq \bar{c}_j$ if $i \neq j$. We assume that $s_0 \geq k - 1$, a condition fulfilled by standard explicit RK methods. For the RK4 method, for instance, we have $c_1 = 0$, $c_2 = c_3 = 1/2$, $c_4 = 1$ and $\bar{c}_1 = 0$, $\bar{c}_2 = 1/2$ and $\bar{c}_3 = 1$; hence, $s_0 = 3$, $k = 4$ and the condition is indeed satisfied. We now prove the following technical lemma.

**Lemma 4.1.** Let $y$ be the solution of (2.5), $\bar{y}$ the solution of (3.9), $q(t)$ the interpolation polynomial through the points $(t_n + \bar{c}_i \Delta t, F(t_n + \bar{c}_i \Delta t))$, $i = 1, \ldots, s_0$ and $F(t)$ satisfy $F \in C^{s_0}([0, T])$. Then,

$$
\bar{y}'(0) = y'(t_n), \quad \|\bar{y}^{(\nu)}(0) - y^{(\nu)}(t_n)\|_\infty \leq \mathcal{O}(\Delta t^{s_0 - \nu + 1}), \quad \nu = 2, \ldots, k - 1. \quad (4.1)
$$

**Proof.** Since $\bar{y}(0) = y(t_n)$, $q(t_n) = F(t_n)$ and $\sum_{s=1}^{s_0} b_s = 1$, we immediately find from (3.9) with $\tau = 0$ that

$$
\bar{y}'(0) = B(I - P) \sum_{s=1}^{s_0} b_s y(t_n) + (I - P) q(t_n) + B P \bar{y}(0) + P F(t_n)
$$

$$
= B y(t_n) + F(t_n) = y'(t_n),
$$

which yields the equality in (4.1).

To prove the inequality for $\nu = 2, \ldots, k - 1$ in (4.1), we first note that straightforward differentiation of (2.5) yields the identity

$$
y^{(\nu)}(t) = B^n y(t) + \sum_{m=1}^{\nu} B^{\nu-m} F^{(m-1)}(t), \quad \nu \geq 1. \quad (4.2)
$$

Next, we derive an upper bound on the error in approximating $F^{(j)}$, the $j$-th derivative of $F$, by the corresponding derivative of $q$. From [24], we recall that

$$
\|F^{(j)} - q^{(j)}\|_\infty \leq \|\omega^{(j)}\|_\infty \|F^{(s_0)}\|_\infty \frac{1}{j! (s_0 - j)!}, \quad 1 \leq j \leq s_0 - 1, \quad (4.3)
$$

where $\omega(\eta) := \prod_{i=1}^{s_0} (\eta - t_n - \bar{c}_i \Delta t)$ for $\eta \in [t_n, t_{n+1}]$. Since

$$
\omega^{(j)}(\eta) = \sum_{m_1=1}^{s_0} \sum_{m_2=1}^{s_0} \cdots \sum_{m_j=1}^{s_0} \prod_{i=1}^{s_0} (\eta - t_n - \bar{c}_i \Delta t)
$$

and $\eta \in [t_n, t_{n+1}]$, we have $\|\omega^{(j)}\|_\infty = \mathcal{O}(\Delta t^{s_0-j})$ and therefore

$$
\|F^{(j)}(t_n) - q^{(j)}(t_n)\|_\infty \leq \mathcal{O}(\Delta t^{s_0-j}), \quad 1 \leq j \leq s_0 - 1. \quad (4.4)
$$

We shall now prove the inequality in (4.1) by induction over $\nu$. First, let $\nu = 2$. We differentiate (3.9) and set $\tau = 0$ in the resulting expression, which yields

$$
\bar{y}''(0) = B(I - P) \left[ \sum_{s=1}^{s_0} 2 b_s c_s (B y_n + q(t_n)) \right] + (I - P) q'(t_n) + B P \bar{y}'(0) + P F'(t_n).
$$

Then, we use (3.2) with $q = 2$, the fact that $q(t_n) = F(t_n)$, and the equality in (4.1) to obtain

$$
\bar{y}''(0) = B(I - P) (By(t_n) + F(t_n)) + (I - P) q'(t_n) + B P y'(t_n) + P F'(t_n). \quad (4.5)
$$
By using (2.5) and reordering terms, we rewrite (4.5) as
\[
\tilde{y}''(0) = By'(t_n) + (I - P)q'(t_n) + Pf'(t_n),
\]
or equivalently as
\[
\tilde{y}''(0) = y''(t_n) + q'(t_n) - F'(t_n) + P(F'(t_n) - q'(t_n)),
\]
where we have used (2.5) differentiated once. Finally, we apply (4.4) with \( j = 1 \) and use \( \|P\|_\infty \leq 1 \) to obtain
\[
\|\tilde{y}''(0) - y''(t_n)\|_\infty \leq (1 + \|P\|_\infty)\|F'(t_n) - q'(t_n)\|_\infty \leq \mathcal{O}(\Delta t^{\nu - 1}),
\]
which yields (4.1) with \( \nu = 2 \).

Next, we proceed with the induction step and assume that (4.1) holds for \( \nu - 1 \). We differentiate (3.9) \( \nu - 1 \) times and set \( \tau = 0 \) in the resulting expression to obtain
\[
\begin{align*}
\tilde{y}^{(\nu)}(0) &= B(I - P) \left[ \frac{\nu!}{(\nu - 1)!} \sum_{i=1}^{s} b_i c_i^{\nu - 1} \left( B^{\nu - 1} y(t_n) + \sum_{\ell=1}^{\nu - 1} B^{\nu - \ell - 1} q^{(\ell - 1)}(t_n) \right) \right] \\
&\quad + (I - P) q^{(\nu - 1)}(t_n) + BP\tilde{y}^{(\nu - 1)}(0) + Pf^{(\nu - 1)}(t_n). \\
\end{align*}
\]
(4.7)

By using (3.2) and reordering terms, we rewrite (4.7) as
\[
\begin{align*}
\tilde{y}^{(\nu)}(0) &= B^\nu y(t_n) + \sum_{\ell=1}^{\nu - 1} B^{\nu - \ell} q^{(\ell - 1)}(t_n) + q^{(\nu - 1)}(t_n) - BP \left( B^{\nu - 1} y(t_n) \\
&\quad + \sum_{\ell=1}^{\nu - 1} B^{\nu - \ell - 1} q^{(\ell - 1)}(t_n) \right) + BP\tilde{y}^{(\nu - 1)}(0) + P \left( F^{(\nu - 1)}(t_n) - q^{(\nu - 1)}(t_n) \right). \\
\end{align*}
\]
(4.8)

We now use (4.2) to replace the first term on the right of (4.8), which yields
\[
\begin{align*}
\tilde{y}^{(\nu)}(0) &= y^{(\nu)}(t_n) + \sum_{\ell=1}^{\nu - 1} B^{\nu - \ell} \left( q^{(\ell - 1)}(t_n) - F^{(\ell - 1)}(t_n) \right) + BP\tilde{y}^{(\nu - 1)}(0) \\
&\quad - BP \left[ B^{\nu - 1} y(t_n) + \sum_{\ell=1}^{\nu - 1} B^{\nu - \ell - 1} F^{(\ell - 1)}(t_n) + \sum_{\ell=1}^{\nu - 1} B^{\nu - \ell - 1} \left( q^{(\ell - 1)}(t_n) - F^{(\ell - 1)}(t_n) \right) \right] \\
&\quad + P \left( F^{(\nu - 1)}(t_n) - q^{(\nu - 1)}(t_n) \right). \\
\end{align*}
\]
(4.9)

Again we use (4.2) to replace the first two terms in square brackets in (4.9) by \( y^{(\nu - 1)} \). Then we repeatedly apply the triangle inequality and use that \( \|P\|_\infty \leq 1 \) to derive the upper bound
\[
\begin{align*}
\|\tilde{y}^{(\nu)}(0) - y^{(\nu)}(t_n)\|_\infty &\leq \sum_{\ell=1}^{\nu} \|B^{\nu - \ell}\|_\infty \|q^{(\ell - 1)}(t_n) - F^{(\ell - 1)}(t_n)\|_\infty \\
&\quad + \|B\|_\infty \|y^{(\nu - 1)}(0) - y^{(\nu - 1)}(t_n)\|_\infty + \|F^{(\nu - 1)}(t_n) - q^{(\nu - 1)}(t_n)\|_\infty \\
&\quad + \|B\|_\infty \sum_{\ell=1}^{\nu - 1} \|B^{\nu - \ell - 1}\|_\infty \|q^{(\ell - 1)}(t_n) - F^{(\ell - 1)}(t_n)\|_\infty.
\end{align*}
\]
(4.10)
Finally we use the induction hypothesis to estimate the second and (4.4) to estimate the remaining terms on the right side of (4.10), which yields (4.1). □

We are now ready to establish the accuracy of the LTS-RKs methods from Section 3.

Proposition 4.2. Let \( y \in C^{k+1}(0,T) \) be the solution of (2.5) and \( y_{n+1} \) defined by Algorithm 1 with \( y_n = y(t_n) \). Then \( \|y(t_{n+1}) - y_{n+1}\|_\infty = O(\Delta t^{k+1}) \), i.e. the LTS-RKs(\( p \)) method is \( k \)-th order accurate.

Proof. Let \( \tilde{y}(\tau) \) be the solution of (3.9) for \( 0 \leq \tau \leq \Delta t \). We split the local truncation error as

\[
\|y(t_{n+1}) - y_{n+1}\|_\infty \leq \|y(t_{n+1}) - \tilde{y}(\Delta t)\|_\infty + \|\tilde{y}(\Delta t) - y_{n+1}\|_\infty.
\]

Since \( y_{n+1} \) corresponds to the numerical solution of (3.9) with the initial \( k \)-th order RKs method (3.1) with time-step \( \Delta t = \Delta t/p \), we immediately have

\[
\|\tilde{y}(\Delta t) - y_{n+1}\|_\infty = O(\Delta t^{k+1}).
\]

To complete the proof we thus need to show that \( \|y(t_{n+1}) - \tilde{y}(\Delta t)\|_\infty = O(\Delta t^{k+1}) \).

First, we integrate (3.9) from 0 until \( \Delta t \) to obtain

\[
\tilde{y}(\Delta t) = y(t_n) + B(I - P) \left[ \sum_{j=0}^{s-1} \frac{1}{j!} \sum_{i=1}^{s} b_i c_i^j \Delta t^{j+1} \left( B^j y(t_n) + \sum_{\ell=1}^{j} B^{j-\ell} q^{(\ell-1)}(t_n) \right) \right] + \int_0^{\Delta t} (I - P) q(t_n + \tau) \, d\tau + \int_0^{\Delta t} BP \tilde{y}(\tau) \, d\tau + \int_0^{\Delta t} PF(t_n + \tau) \, d\tau. \tag{4.11}
\]

We now approximate the first and last integral on the right of (4.11) using the \( k \)-th order quadrature formula defined by the weights \( b_i \) and the nodes \( c_i \). Since \( q \) interpolates \( F \) at the nodes \( t_n + c_i \Delta t \), we conclude that

\[
\int_0^{\Delta t} (I - P) q(t_n + \tau) \, d\tau + \int_0^{\Delta t} PF(t_n + \tau) \, d\tau = \int_{t_n}^{t_{n+1}} F(t) \, dt + O(\Delta t^{k+1}). \tag{4.12}
\]

Next, truncate in (4.11) the sum in square brackets at \( j = k - 1 \) and use (3.2) to replace \( \sum_{i=1}^{s} b_i c_i^j \) by \( 1/(j+1) \). By using (4.2) in the resulting expression, we thus rewrite the second term on the right of (4.11) as

\[
B(I - P) \left[ \sum_{j=0}^{k-1} \frac{1}{j!} \sum_{i=1}^{s} b_i c_i^j \Delta t^{j+1} \left( B^j y(t_n) + \sum_{\ell=1}^{j} B^{j-\ell} q^{(\ell-1)}(t_n) \right) \right] + B(I - P) \left[ \sum_{j=0}^{k-1} \frac{\Delta t^{j+1}}{(j+1)!} y^{(j)}(t_n) \right] + \sum_{j=0}^{k-1} \frac{\Delta t^{j+1}}{(j+1)!} \sum_{\ell=1}^{j} B^{j-\ell} q^{(\ell-1)}(t_n) - F^{(k-1)}(t_n) + O(\Delta t^{k+1}). \tag{4.13}
\]

On the right of (4.13), we identify the first term as the Taylor expansion of \( B y(t) \), integrated up to \( \Delta t \), and hence rewrite it as

\[
\int_{t_n}^{t_{n+1}} B y(t) \, dt = B \int_{t_n}^{t_{n+1}} y(t) \, dt + O(\Delta t^{k+1}).
\]
Similarly, we approximate \( \tilde{y}(\tau) \) in the remaining integral of (4.11) by Taylor expansion up to order \( k - 1 \). By combining the above approximations of the various terms on the right of (4.11), we conclude that

\[
\tilde{y}(\Delta t) = y(t_n) + \int_{t_n}^{t_{n+1}} B y(t) \, dt + \int_{t_n}^{t_{n+1}} F(t) \, dt - BP \sum_{j=0}^{k-1} \frac{\Delta t^{j+1}}{(j+1)!} y^{(j)}(t_n) \\
+ BP \sum_{j=0}^{k-1} \frac{\Delta t^{j+1}}{(j+1)!} y^{(j)}(0) + B(I - P) \left[ \sum_{j=0}^{k-1} \frac{\Delta t^{j+1}}{(j+1)!} \sum_{\ell=1}^{j} B^{j-\ell} \left( q^{(\ell-1)}(t_n) \right) \\
- F^{(\ell-1)}(t_n) \right] + O(\Delta t^{k+1}).
\]

(4.14)

The sum of the first three terms on the right of (4.14) equals \( y(t_{n+1}) \). By using the triangle inequality and the fact that \( \|P\|_\infty \leq 1 \), we therefore obtain

\[
\|\tilde{y}(\Delta t) - y(t_{n+1})\|_\infty \leq \|B\|_\infty \sum_{j=0}^{k-1} \frac{\Delta t^{j+1}}{(j+1)!} \|y^{(j)}(0) - y^{(j)}(t_n)\|_\infty \\
+ \|B(I - P)\|_\infty \sum_{j=0}^{k-1} \frac{\Delta t^{j+1}}{(j+1)!} \|B^{(j)}\|_\infty \|q^{(\ell-1)}(t_n) - F^{(\ell-1)}(t_n)\|_\infty + O(\Delta t^{k+1}),
\]

where the first two terms on the right can be estimated by using (4.4) and Lemma 4.1. This concludes the proof. \( \square \)

We shall now show that the LTS-RK methods with \( s = k = 2, 3, 4 \) converge as \( O(\Delta t^k) \). For simplicity, we shall restrict ourselves to the homogeneous case with \( F(t) = 0 \). In the following Lemma, we first rewrite the LTS-RKs(\( p \)) scheme for \( s = k = 2, 3, 4 \) as a one-step method.

**Lemma 4.3.** Let \( \tilde{y}_{m+1}^n \) be defined by Algorithm 1 for \( s = k = 2, 3 \) or 4, \( m \geq 0 \), and assume that \( F(t) = 0 \). Then we have for \( 0 \leq m \leq p - 1 \):

\[
\tilde{y}_{m+1}^n = y_n + (m + 1)\Delta \tau B y_n + \frac{(m + 1)^2}{2} \Delta \tau^2 B^2 y_n \\
+ \sum_{i=3}^{2m+2} \Delta \tau^i \alpha_{i,m+1}(BP)^{i-2}B^2 y_n
\]

(4.15)

for \( s = k = 2 \),

\[
\tilde{y}_{m+1}^n = y_n + (m + 1)\Delta \tau B y_n + \frac{(m + 1)^2}{2} \Delta \tau^2 B^2 y_n \frac{(m + 1)^3}{6} \Delta \tau^3 B^3 y_n \\
+ \sum_{i=3}^{3m+4} \Delta \tau^i \left( \alpha_{i,m+1}(BP)^{i-2}B^2 y_n + \beta_{i,m+1}(BP)^{i-3}B^3 y_n \right)
\]

(4.16)

for \( s = k = 3 \), and
$\bar{y}_{n+1} = y_n + (m+1)\Delta \tau B y_n + \frac{(m+1)^2}{2} \Delta \tau^2 B^2 y_n \frac{(m+1)^3}{6} \Delta \tau^3 B^3 y_n$

$$+ \frac{(m+1)^4}{24} \Delta \tau^4 B^4 y_n + \sum_{i=5}^{4m+6} \Delta \tau^i \left( \alpha_{i,m+1}(BP)^{i-2} B^2 y_n, \right.$$  

$+ \beta_{i,m+1}(BP)^{i-3} B^3 y_n + \delta_{i,m+1}(BP)^{i-4} B^4 y_n \bigg)$  

(4.17)

for $s = k = 4$. The constants $\alpha_{i,m+1}$, $\beta_{i,m+1}$ and $\delta_{i,m+1}$ are determined recursively.

Proof. We shall now prove (4.15) by induction on $m$. The proofs of (4.16) and (4.17) are similar but more cumbersome; hence, they are omitted here.

We let $s = k = 2$ and consider Algorithm 1. Since $F(t)$ is identically zero, so is the interpolation polynomial $q(t)$. Therefore, Step 2 of Algorithm 1 reduces to

$$w_{n,0} = B(I - P) \sum_{i=1}^{2} b_i y_n, \quad w_{n,1} = B(I - P) \sum_{i=1}^{2} b_ic_i y_n.$$

(4.18)

The coefficients of the underlying RK2 method satisfy the order conditions (3.2) with $q = 1, 2$ and $c_1 = 0$ – see also ([4], p. 156):

$$a_{21} = c_2, \quad b_1 + b_2 = 1, \quad b_2 c_2 = 1/2.$$

(4.19)

Thus, we rewrite (4.18) as

$$w_{n,0} = B(I - P)y_n, \quad w_{n,1} = B(I - P)By_n.$$

(4.20)

In Step 3 of Algorithm 1, we now use (4.20), the fact that $\bar{y}_0 = y_n$ and (4.19) to calculate

$$k_{1,1} = w_{n,0} + BP\bar{y}_0 = By_n,$$

$$k_{2,1} = w_{n,0} + c_2 \Delta \tau w_{n,1} + BP \left( \bar{y}_0 + \Delta \tau a_{21} k_{1,1} \right) = By_n + c_2 \Delta \tau B^2 y_n.$$

Then we complete the first local time-step as

$$\bar{y}_{1/2} = \bar{y}_0 + \Delta \tau \left( b_1 k_{1,1} + b_2 k_{2,1} \right) = y_n + \Delta \tau By_n + \frac{\Delta \tau^2}{2} B^2 y_n,$$

(4.21)

which yields (4.15) with $m = 0$.

We now proceed with the induction step and assume that (4.15) holds for $m \geq 0$.

Following Algorithm 1, we again explicitly calculate

$$k_{1, m+1} = w_{n,0} + m \Delta \tau w_{n,1} + BP\bar{y}_{m},$$

$$k_{2, m+1} = w_{n,0} + (m + c_2) \Delta \tau w_{n,1} + BP \left( \bar{y}_m + \Delta \tau a_{21} k_{1, m+1} \right).$$

(4.22)

Next, we use the induction hypothesis to replace $\bar{y}_m$ and rewrite (4.22) as

$$k_{1, m+1} = By_n + m \Delta \tau B^2 y_n + \frac{m^2 \Delta \tau^2}{2} B^2 y_n + \sum_{i=3}^{2m} \alpha_{i,m} \Delta \tau^i (BP)^{i-2} B^2 y_n,$$

$$k_{2, m+1} = By_n + (m + c_2) \Delta \tau B^2 y_n + (m^2 + 2mc_2) \frac{\Delta \tau^2}{2} B^2 y_n$$

$$+ \sum_{i=3}^{2m} \alpha_{i,m} \Delta \tau^i (BP)^{i-1} B^2 y_n + \sum_{i=3}^{2m} c_2 \alpha_{i,m} \Delta \tau^{i+1} (BP)^{i-2} B^2 y_n.$$
Moreover, Proposition 4.2 implies that the truncation error satisfies
\[ \alpha \text{ coefficients} \]
By collecting like powers in \( \Delta \) and again using the induction hypothesis to replace \( \tilde{LTS-RK} \) the one-step formulation from Lemma 4.3, we shall now prove the convergence of the \( \alpha \) order.

Lemma 4.3, we then infer that

To apply standard convergence theory for one-step methods (e.g. [21]) and thus conclude the proof, we must show that the increment function \( \Phi \) is Lipschitz-continuous in its first argument. Since \( \|P\| \leq 1 \) and also \( \Delta t \|B\| \leq 1 \) for \( \Delta t \) sufficiently
small, we conclude that
\[
\|\Phi_{RK2}(\mathbf{u}, \Delta t) - \Phi_{RK2}(\mathbf{v}, \Delta t)\| \\
\leq \left\| \mathbf{B} + \frac{\Delta t}{2} \mathbf{B}^2 + \sum_{i=3}^{2p} \alpha_{i,p} \frac{\Delta t^{i-1}}{p^i} (\mathbf{BP})^{i-2} \mathbf{B}^2 \right\| \cdot \| \mathbf{u} - \mathbf{v} \| \leq L_{RK2} \| \mathbf{u} - \mathbf{v} \|.
\]

Here,
\[
L_{RK2} = \| \mathbf{B} \| + \frac{\Delta t}{2} \| \mathbf{B} \|^2 + \sum_{i=3}^{2p} |\alpha_{i,p}| \frac{\Delta t^{i-1}}{p^i} \| \mathbf{B} \|^i \leq C \| \mathbf{B} \|,
\]

where \( C \) independent of \( \Delta t, p \) and \( \| \mathbf{B} \| \) for \( \Delta t \) sufficiently small. Therefore, \( \Phi_{RK2}(\mathbf{y}, \Delta t) \) is Lipschitz-continuous and the LTS-RK2 method convergent of order two.

For \( s = k = 3 \) or \( s = k = 4 \), the proof is similar with (4.15) replaced by (4.16) or (4.17), respectively. \( \square \)

The proof of Theorem 4.4 is straightforward and immediately extends to arbitrarily high order of accuracy. It relies, however, on the one-step formulation provided by Lemma 4.3, which cannot be stated for arbitrary \( k \) because no generic form of the order conditions, such as (4.19), is available for \( k \geq 5 \) – see [21].

**Remark 2.** For \( s = k \), the one-step formulations (4.15)–(4.17) of the LTS-RK methods with \( m = p - 1 \) and \( \mathbf{P} = 0 \) reduce to
\[
\mathbf{y}_{n+1} = \tilde{y}_1 = \tilde{R}(\Delta t \mathbf{B}) \mathbf{y}_n,
\]
where \( \tilde{R}(z) = 1 + z + \ldots + z^s/s! \) denotes the (unique) stability function of the corresponding standard RKs method. Thus for \( \mathbf{F} = 0 \), the LTS-RKs scheme coincides with the standard RKs scheme inside the coarse region, independently of the number of local time-steps \( p \).

For \( s > k \), however, the one-step formulation of the underlying RKs method applied to (2.5) yields
\[
\mathbf{y}_{n+1} = \sum_{j=0}^{k} \frac{(\Delta t \mathbf{B})^j}{j!} \mathbf{y}_n + \sum_{j=k+1}^{s} \sigma_j (\Delta t \mathbf{B})^j \mathbf{y}_n, \tag{4.26}
\]
where the additional parameters \( \sigma_j, j \geq k+1 \) are no longer uniquely determined by the order of accuracy, but instead depend on the coefficients \( b_i, a_{ij}, \) and \( c_i, 1 \leq i, j \leq s \), of the particular RKs method. Similarly in the derivation of the LTS-RKs method with \( s > k \), the coefficients \( j \geq k \) in the Taylor expansion (3.6) can be modified without affecting the order of accuracy. Then, Step 2 of Algorithm 1 becomes:

2. For \( j = 0, \ldots, s - 1 \) compute
\[
\mathbf{w}_{n,j} = \begin{cases} \frac{1}{j!} \mathbf{B}(\mathbf{I} - \mathbf{P}) \left( \mathbf{B}^j \mathbf{y}_n + \sum_{\ell=1}^{j} \mathbf{B}^{j-\ell} \mathbf{q}^{(\ell-1)}(t_n) \right), & j \leq k \\ \gamma_j \mathbf{B}(\mathbf{I} - \mathbf{P}) \left( \mathbf{B}^j \mathbf{y}_n + \sum_{\ell=1}^{j} \mathbf{B}^{j-\ell} \mathbf{q}^{(\ell-1)}(t_n) \right), & j > k \end{cases}.
\]

Although the coefficients \( \gamma_j \) do not affect the accuracy of the scheme, they do affect its stability. To recover the stability properties of the original RKs method, we also rewrite the LTS-RKs method for \( p = 1, \mathbf{P} = 0, \) and \( \mathbf{F} = 0 \) as a one-step method
\[
\mathbf{y}_{n+1} = \sum_{j=0}^{k} \frac{(\Delta t \mathbf{B})^j}{j!} \mathbf{y}_n + \sum_{j=k+1}^{s} \gamma_j \sum_{i=1}^{s} b_i c_i^{j-1} (\Delta t \mathbf{B})^j \mathbf{y}_n. \tag{4.27}
\]
The two stability functions in (4.26) and (4.27) are identical, if

\[ \gamma_j = \frac{\sigma_j}{\sum_{i=1}^{s} b_i c_i^{j-1}}, \quad k + 1 \leq j \leq s, \]

which we shall henceforth assume.

For the LSRK5 scheme with \( s = 5 \) and \( k = 4 \), for instance, the choice of

\[ \sigma_5 = \sum_{i,j,k,l} b_i a_{i,j} a_{j,k} a_{k,l} c_l \]

is crucial for the stability of the scheme; the optimal value, \( \sigma_5 = 1/200 \), indeed corresponds to the values in Table A.2 [7]. To retain that optimal stability for the LTS-LSRK5 method, we infer from the LTS-LSRK5(1) algorithm with \( F = 0 \) and \( P = 0 \) that \( \gamma_5 \) must equal

\[ \gamma_5 = \frac{\sigma_5}{\sum_{i=1}^{5} B_i C_i^4 + \sum_{i=2}^{5} B_i A_i C_i^4 + \sum_{i=3}^{5} B_i A_i A_i-1 C_i^4 + B_5 A_5 A_4 A_3 C_2^4}. \]

5. Numerical experiments. We shall now present numerical experiments which validate the expected order of convergence of the LTS-RK methods derived in Section 3, illustrate their stability properties and demonstrate their usefulness in the presence of complex geometry. First, we consider a simple one-dimensional test problem to show that the \( k \)th order LTS-RK methods always yield the optimal space-time rate of convergence when combined with a spatial FE discretization of comparable accuracy. Next, we study the stability properties of the various LTS-RK schemes. Finally, we illustrate the versatility of our LTS schemes by simulating the propagation of a plane wave across a narrow gap.

5.1. Convergence. We consider (2.1) with \( c = 1 \) and \( \sigma = 0.1 \) on the interval \( \Omega = [0, 6] \). The initial conditions \( u_0, v_0 \) and the source \( f \) yield the exact solution

\[ u(x, t) = \cos(t) \sin(\pi x). \]

Next, we divide \( \Omega \) into three equal parts. The left and right intervals, \([0, 2]\) and \([4, 6]\), respectively, are discretized with an equidistant mesh of size \( h^{\text{coarse}} \), whereas the interval \( \Omega_f = [2, 4] \) is discretized with an equidistant mesh of size \( h^{\text{fine}} = h^{\text{coarse}} / \rho \) – see Fig. 5.1. Hence, the two outer intervals correspond to the coarse region and the inner interval \([2, 4]\) to the refined region.

![Fig. 5.1. One-dimensional example: the computational domain \( \Omega = [0, 6] \) with the refined region \( \Omega_f = [2, 4] \).](image)

We begin with the LTS schemes based on the popular RK4 and LSRK5 methods from Sections 3.2 and 3.3. On a sequence of increasingly finer meshes with mesh size
RUNGE-KUTTA BASED EXPLICIT LOCAL TIME-STEPPING

19

$h_{\text{coarse}}$ and fixed ratio $h_{\text{fine}} = h_{\text{coarse}} / p$, we consider a continuous $\mathcal{P}^3$-FE discretization of (2.1) with mass lumping. For every time-step, $\Delta t$, we take $p$ local steps of size $\Delta \tau = \Delta t / p$ inside $\Omega_f$, either with the LTS-RK4($p$) or the LTS-LSRK5($p$) algorithms from Sections 3.2 – see also Tables A.1 and A.2 in the Appendix. Due to stability, every reduction of the mesh size implies a corresponding reduction of the time-step, which depends linearly on $h_{\text{coarse}}$. As we simultaneously reduce $h_{\text{coarse}}$ and $\Delta t$, we monitor the $L^2$ space-time error $\|u(\cdot, T) - u_{h,\Delta t}(\cdot, T)\|_{L^2(\Omega)}$ at the final time $T = 10$.

Regardless of the rate of local refinement $p = 2, 5$ or 11, the LTS-RK4 and LTS-LSRK5 methods yield fourth-order convergence, as shown in Fig. 5.2(a) and Fig. 5.3(a).

Next, we repeat the numerical experiment with a $\mathcal{P}^3$-FE nodal DG discretization with upwinding flux [22]. As shown in Fig. 5.2(b) and Fig. 5.3(b), both the LTS-RK4($p$) and LTS-LSRK5($p$) methods again yield overall fourth-order convergence independently of $p$.

Similar numerical experiments with $\mathcal{P}^{s-1}$ continuous and nodal DG FE also corroborate the space-time $s$-th order rate of convergence of both the LTS-RK$s$ and the LTS-LSRK$s$ methods for $s = 2, 3$; these results are omitted here. Note that for the
LTS-RK2 method the local time-step $\Delta \tau$ must also comply with the suboptimal CFL-condition $\Delta t \leq Ch^{4/3}$ of the RK2 (or Heun’s) method [5], and thus satisfy $\Delta \tau = \Delta t/\bar{p}$ with $\bar{p} = [p^{4/3}]$.

5.2. Stability. We consider again (2.1) with $c = 1$, $\sigma = 0$, but $f = 0$ on the interval $\Omega = [0, 6]$, where $[0, 2]$ and $[4, 6]$ are discretized with an equidistant mesh of size $h^{\text{coarse}}$, whereas $\Omega_f = [2, 4]$ is discretized with an equidistant mesh of size $h^{\text{fine}} = h^{\text{coarse}}/p$ – see Fig. 5.1. For $p = 1$, the mesh is equidistant throughout $\Omega$ and we denote by $\Delta t_{\text{RK}s}$ or $\Delta t_{\text{LSRK}s}$ the largest time-step permitted by the standard RKs or LSRKs method, respectively; note that $\Delta t_{\text{RK}s} = \Delta t_{\text{LSRK}s}$ for $s = 2, 3$.

For $p \geq 2$, we use either the LTS-RK$s$($p$) or LTS-LSRK$s$($p$) method and take $p$ substeps of size $\Delta \tau = \Delta t/p$ inside $\Omega_f$. Clearly, the maximal permissible time-step, denoted by $\Delta t_p$, satisfies either $\Delta t_p \leq \Delta t_{\text{RK}s}$ or $\Delta t_p \leq \Delta t_{\text{LSRK}s}$, respectively. If $\Delta t_p = \Delta t_{\text{RK}s}$ or $\Delta t_p = \Delta t_{\text{LSRK}s}$, the corresponding LTS method imposes no further constraint on $\Delta t$ and its CFL-condition is therefore optimal.

To determine the stability range of any LTS scheme from Section 3, we proceed as follows:

1. Determine $\Delta t_{\text{RK}s}$, respectively $\Delta t_{\text{LSRK}s}$, the maximal $\Delta t$ allowed by the corresponding standard or low-storage RKs method (without LTS) for the equidistant mesh of mesh size $h^{\text{coarse}}$;
2. refine the mesh $p$ times inside $\Omega_f$;
3. determine the maximal time-step $\Delta t_p$ allowed by the LTS method for the locally refined mesh and compare $\Delta t_p$ with $\Delta t_{\text{RK}s}$ or $\Delta t_{\text{LSRK}s}$, respectively.

Again, we begin with the popular RK4 scheme and consider a continuous $P^3$ FE discretization with mass lumping on an equidistant mesh of size $h^{\text{coarse}} = 0.2$. The RK4 method, which we rewrite as

$$y_{n+1} = C_{\text{RK}4} y_n, \quad C_{\text{RK}4} = I + \Delta t B + \frac{\Delta t^2}{2} B^2 + \frac{\Delta t^3}{3!} B^3 + \frac{\Delta t^4}{4!} B^4,$$

is stable if $\rho(C_{\text{RK}4}) \leq 1$, where $\rho(C_{\text{RK}4})$ denotes the spectral radius of the matrix $C_{\text{RK}4}$ [21]. Progressively increasing $\Delta t$ while monitoring $\rho(C_{\text{RK}4})$, we find that the maximal time-step is $\Delta t_{\text{RK}4} = 0.0656$.

Next, 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 the matrix $P$. To
determine the stability range of the LTS-RK4(2) method (Algorithm 3, Sect. 3.3), we use Lemma 4.3 with \( s = k = 4 \) to rewrite it as

\[
y_{n+1} = C_{LTS-RK4} y_n,
\]

where the matrix \( C_{LTS-RK4} \) is explicitly given by (4.17) with \( p = 2 \) and \( m = p - 1 \). Again, the LTS-RK4 method is stable when \( C_{LTS-RK4} \leq 1 \). In Fig. 5.4(a), we observe that the spectral radius of \( C_{LTS-RK4} \) lies below one for all time-steps \( \Delta t \leq \Delta t_{RK4} \). Therefore, the LTS-RK4(2) scheme, when combined with standard \( P^3 \)-elements, is stable up to the maximal time-step and hence its CFL-condition optimal. Next, we repeat the numerical experiment for the \( P^3 \) nodal DG-FE discretization with upwinding flux. As shown in Fig. 5.4(b), the range of stable time-steps again is maximal and the CLF condition thus optimal.

We perform similar numerical experiments with different values of \( p \), but also \( P^{s-1} \) continuous and nodal DG FE for the LTS-RKs methods with \( s = 2, 3 \). These results are summarized in Table 5.1. Since the CFL-condition of the RK2 method is not linear but instead constrained by \( \Delta t \leq Ch^{4/3} [5] \), we again set \( \Delta \tau = \Delta t / \bar{p} \) with \( \bar{p} = [p^{4/3}] \) for stability.

<table>
<thead>
<tr>
<th>( p )</th>
<th>FE</th>
<th>DG</th>
<th>FE</th>
<th>DG</th>
<th>FE</th>
<th>DG</th>
<th>FE</th>
<th>DG</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTS-RK2(( p ))</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>LTS-RK3(( p ))</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>LTS-RK4(( p ))</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>0.98</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Table 5.1**

Stability of the LTS-RKs(2) schemes for \( s = 2, 3, 4 \) when combined with a continuous FE or nodal DG-FE discretization: the ratio \( \Delta t_p / \Delta t_{RKs} \) is shown for varying \( p \).

Next, we repeat the above numerical stability analysis for the LTS-LSRKs methods. For \( s = 2, 3 \), we obtain identical results, as expected, since all methods with \( k = s \) have the same stability regions; these results are omitted here. To study the

![Fig. 5.5](image-url)  
**Fig. 5.5.** The spectral radius of \( C_{LTS-LSRK5} \) for different \( P^3 \) FEM.
and also Remark 2 – we consider either a continuous or a nodal DG $P^3$-FE discretization. Again, we set $h_{\text{coarse}} = 0.2$ and refine by a factor $p = 2$ those elements that lie inside the interval $[0.2, 4]$. Hence for every time-step $\Delta t$, we use the fourth-order LTS-LSRK5(2) method with $\Delta \tau = \Delta t/2$ inside the refined region. After reformulating the LTS-LSRK5(2) scheme as one-step method, $y_{n+1} = C_{\text{LTS-LSRK5}} y_n$, we compute the spectral radius of $C_{\text{LTS-LSRK5}}$ for varying $\Delta t/\Delta t_{\text{LSRK5}}$. As shown in Fig. 5.5(a), the spectral radius of $C_{\text{LTS-LSRK5}}$ lies below one for all time-steps $\Delta t \leq \Delta t_{\text{LSRK5}}$. Hence the LTS-LSRK5(2) scheme, when combined with standard $P^3$-elements, is stable up to the optimal time-step. Similarly, we observe in Fig. 5.5(b) that the LTS-LSRK5(2) scheme also yields optimal stability when combined with a $P^3$ nodal DG-FE discretization.

Finally to illustrate the effect of damping on the stability, we display in Table 5.2 the (maximal) time-step ratio $\Delta t_2/\Delta t_{\text{RK}s}$ ($s = 2, 3, 4$) for varying $\sigma$, either with a conforming or a nodal DG FEM. Most of the LTS schemes yield an optimal CFL-condition, independently of $\sigma > 0$. Identical results were obtained with the LTS-LSRKs ($s = 2, 3, 5$) methods. For $\sigma = 0$, the LTS-RK3 method remains stable regardless of the spatial discretization, whereas the LTS-RK$s$ ($s = 2, 4$) and LTS-LSRKs ($s = 2, 5$) methods are stable when combined with the nodal DG-FE discretization with an upwinding flux.

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>LTS-RK2(2)</th>
<th>LTS-RK3(2)</th>
<th>LTS-RK4(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cont. FE</td>
<td>nodal DG</td>
<td>cont. FE</td>
</tr>
<tr>
<td>0.001</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>0.01</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>0.1</td>
<td>1.0</td>
<td>1.0</td>
<td>0.97</td>
</tr>
<tr>
<td>1</td>
<td>1.0</td>
<td>1.0</td>
<td>0.91</td>
</tr>
<tr>
<td>10</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2: The LTS-RK$s$ (2) scheme for $s = 2, 3, 4$ when combined with a continuous FE or nodal DG-FE discretization: the ratio $\Delta t_2/\Delta t_{\text{RK}s}$ is shown for varying $\sigma$.

5.3. Two-dimensional example. To illustrate the usefulness of the LTS-RK methods, we now consider (2.1) in $\Omega$, a rectangular domain of size $[0, 2] \times [0, 1]$ with two rectangular barriers inside forming a narrow gap, as shown in Fig. 5.6. We set the constant wave speed to $c = 1$ and the constant damping coefficient to $\sigma = 0.1$. At the boundary of $\Omega$, we impose homogeneous Neumann conditions and choose as initial conditions the vertical Gaussian plane wave

$$u_0(x, y) = \exp\left(-\frac{(x-x_0)^2}{\delta^2}\right), \quad v_0(x, y) = 0, \quad (x, y) \in \Omega,$$

centered about $x_0 = 0.8$ and of width $\delta = 0.009$.

For the spatial discretization we opt for $P^2$ continuous finite elements with mass lumping. First, $\Omega$ is discretized with triangles of maximal size $h_{\text{coarse}} = 0.03$. However, such coarse triangles do not resolve the small geometric features of the gap, which require $h_{\text{fine}} \approx h_{\text{coarse}}/7$, as shown in Fig. 5.6. Then, we successively refine the entire mesh three times, each time splitting every triangle into four. Since the initial mesh in $\Omega$ is unstructured, the boundary between the fine and the coarse mesh is not well-defined. Here we let the fine mesh correspond to all triangles with $h < 0.7 h_{\text{coarse}}$ in size, i.e. the darker (green) triangles in Fig. 5.6. The corresponding degrees of
freedom in the finite element solution are then selected merely by setting to one the corresponding diagonal entries of the matrix $P$ — see Section 3.

For the time discretization, we choose the third-order LTS-LSRK3(7) scheme with $p = 7$, which for every time-step $\Delta t$ takes seven local time-steps inside the refined region that surrounds the gap. Thus, the numerical method is third-order accurate in both space and time with respect to the $L^2$-norm. If instead the same (global) time-step $\Delta t$ was used everywhere inside $\Omega$, it would have to be about seven times smaller than necessary in most of $\Omega$. In Fig. 5.7, snapshots of the numerical solution are shown at different times. The vertical Gaussian pulse initiates two Gaussian plane waves propagating in opposite directions. As the right-moving wave proceeds, it impinges upon the obstacle; then, a fraction of the wave penetrates the gap and generates a circular wave, which further interacts with the propagating wave field.
6. Concluding remarks. Starting from standard explicit Runge-Kutta (RK) methods, we have derived explicit local time-stepping (LTS) methods which permit arbitrarily small time-steps precisely where the smallest elements in the underlying mesh are located. When combined with a finite element discretization with an essentially diagonal mass matrix, the resulting discrete time-marching schemes remain fully explicit and thus highly parallel. The LTS methods based on classical RK schemes are given in Algorithm 1, the LTS methods based on low-storage (LS) RK methods are given in Algorithm 2, and the LTS method based on the popular RK4 method is detailed in Algorithm 3.

Let $\Delta t$ denote the time-step dictated by the CFL-condition in the coarser part of the mesh. Then, during every (global) time step $\Delta t$, each local time step of size $\Delta t/p$ inside the locally refined region of the mesh, with $p \geq 2$ any integer, simply corresponds to sparse matrix-vector multiplications that involve only 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 partition matrix $P$ to one; in particular, no adjacency or coherence in the numbering is assumed. Hence, the implementation is straightforward and requires no special data structures.

The intermediate values needed at the coarse/fine mesh interface during sub-steps are obtained through a judicious combination of interpolation and Taylor expansion, which preserves the accuracy of the original scheme without resorting to the solution of a linear systems or recursive dependencies. If the underlying $s$-stage RK method has order $k$, we have proved that the corresponding LTS-RKs method retains the same accuracy, independently of $s \geq k$. For $s = k = 2, 3, 4$ we have proved that the LTS-RKs methods converge as $\Delta t \to 0$ to the semi-discrete solution. Our numerical experiments indicate that if an LTS-RK method of order $k$ is combined with a $P_{k-1}$ FE spatial discretization, the numerical solution will converge to the true solution with optimal rate $O(h^k, \Delta t^k)$, as $h, \Delta t \to 0$. Moreover, our numerical experiments suggest that the LTS-RK methods preserve the optimal CFL-condition imposed on $\Delta t$ by the coarser region of the mesh, independently of $p$. Hence our LTS-RK methods overcome the bottleneck caused by the geometry induced stiffness without sacrificing the explicitness, accuracy or efficiency of the original RK method.

Our derivation of explicit LTS-RK methods applies to a general explicit RK method of arbitrary order. Hence, it not only applies to low-storage RK methods, but also to any other explicit RK method, such as the low-dispersion low-dissipation RK methods from [25]. Although the LTS-RK methods are particularly appropriate for computational wave propagation, where the use of explicit time integration prevails, they are likely to prove useful for other partial differential equations.

Appendix A. Runge-Kutta methods.

REFERENCES


Table A.1
Coefficients of the classical RK methods.

<table>
<thead>
<tr>
<th></th>
<th>a (RK2 order 2)</th>
<th>b (RK3 order 3)</th>
<th>c (RK4 order 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1/2</td>
<td>1/2</td>
<td>1/6</td>
</tr>
</tbody>
</table>

(a) RK2 (order 2) (b) RK3 (order 3) (c) RK4 (order 4)

Table A.2
Coefficients of the LSRK methods.

<table>
<thead>
<tr>
<th></th>
<th>a (LSRK2 order 2)</th>
<th>b (LSRK3 order 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1/2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1/3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-3/4</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>-153/128</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>-1/16</td>
</tr>
<tr>
<td>6</td>
<td>1/3</td>
<td>15/16</td>
</tr>
<tr>
<td>7</td>
<td>1/4</td>
<td>8/15</td>
</tr>
</tbody>
</table>

(a) LSRK2 (order 2) (b) LSRK3 (order 3)

<table>
<thead>
<tr>
<th>0</th>
<th>A_i</th>
<th>B_i</th>
<th>C_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1432997174477</td>
<td>9575080441755</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>567301805773</td>
<td>13612068292357</td>
<td>1432997174477</td>
</tr>
<tr>
<td>3</td>
<td>2016746695238</td>
<td>5126769341429</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3529016866466</td>
<td>6820363962896</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1275806237668</td>
<td>2802321613138</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>842570457699</td>
<td>2924317926251</td>
<td></td>
</tr>
</tbody>
</table>

(c) LSRK5 (order 4)


<table>
<thead>
<tr>
<th>No.</th>
<th>Author</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-23</td>
<td>Helmut Harbrecht, Johannes Tausch</td>
<td>On Shape Optimization with Parabolic State Equation</td>
</tr>
<tr>
<td>2013-24</td>
<td>Zoé Chatzidakis, Dragos Ghioca, David Masser, Guillaume Maurin</td>
<td>Unlikely, Likely and Impossible Intersections without Algebraic Groups</td>
</tr>
<tr>
<td>2013-26</td>
<td>Luigi Ambrosio and Gianluca Crippa</td>
<td>Continuity Equations and ODE Flows with Non-Smooth Velocity</td>
</tr>
<tr>
<td>2013-27</td>
<td>Andriy Regeta</td>
<td>Lie Subalgebras of Vector Fields and the Jacobian Conjecture</td>
</tr>
<tr>
<td>2013-28</td>
<td>Helmut Harbrecht, Michael Peters, Markus Siebenmorgen</td>
<td>Tractability of the Quasi-Monte Carlo Quadrature with Halton Points for Elliptic Pdes with Random Diffusion</td>
</tr>
<tr>
<td>2013-29</td>
<td>Helmut Harbrecht, Giannoula Mitrou</td>
<td>Stabilization of the Triangular Method for the Bernoulli Problem in Case of Prescribed Dirichlet Data</td>
</tr>
<tr>
<td>2014-01</td>
<td>Helmut Harbrecht, Michael Peters, Markus Siebenmorgen</td>
<td>Efficient Approximation of Random Fields for Numerical Applications</td>
</tr>
<tr>
<td>2014-02</td>
<td>Ali Hyder, Luca Martinazzi</td>
<td>Conformal Metrics on $R^{2m}$ with Constant $Q$-Curvature, Prescribed Volume and Asymptotic Behavior</td>
</tr>
<tr>
<td>2014-03</td>
<td>Jürgen Dölz, Helmut Harbrecht, and Michael Peters</td>
<td>$H$-Matrix Accelerated Second Moment Analysis for Potentials with Rough Correlation</td>
</tr>
<tr>
<td>2014-04</td>
<td>Tianling Jin, Ali Maalaoui, Luca Martinazzi, Jingang Xiong</td>
<td>Existence and Asymptotics for Solutions of a Non-Local $Q$-Curvature Equation in Dimension Three</td>
</tr>
<tr>
<td>2014-05</td>
<td>Marcus J. Grote, Michaela Mehlin, Teodora Mitkova</td>
<td>Runge-Kutta Based Explicit Local Time-Stepping Methods for Wave Propagation</td>
</tr>
</tbody>
</table>

Preprints are available under http://math.unibas.ch/research/publications/