Efficient Explicit Time Integration for the Simulation of Acoustic and Electromagnetic Waves

Inauguraldissertation

zur
Erlangung der Würde eines Doktors der Philosophie
vorgelegt der
Philosophisch-Naturwissenschaftlichen Fakultät
der Universität Basel

von

Michaela Mehlin

aus Rheinfelden, Deutschland

Basel, 2015

Originaldokument gespeichert auf dem Dokumentenserver der Universität Basel
edoc.unibas.ch

Dieses Werk ist unter dem Vertrag "Creative Commons Namensnennung-Keine kommerzielle Nutzung-Keine Bearbeitung 3.0 Schweiz" (CC BY-NC-ND 3.0 CH) lizenziert. Die vollständige Lizenz kann unter creativecommons.org/licenses/by-nc-nd/3.0/ch eingesehen werden.

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät
auf Antrag von

Prof. Dr. Marcus J. Grote
Dr. Stéphane Lanteri

Basel, den 21. April 2015

Prof. Dr. Jörg Schibler
Dekan

ABSTRACT

The efficient and accurate numerical simulation of time-dependent wave phenomena is of fundamental importance in acoustic, electromagnetic or seismic wave propagation. Model problems describing wave propagation include the wave equation and Maxwell's equations, which we study in this work. Both models are partial differential equations in space and time. Following the method-of-lines approach we first discretize the two model problems in space using finite element methods (FEM) in their continuous or discontinuous form. FEM are increasingly popular in the presence of heterogeneous media or complex geometry due to their inherent flexibility: elements can be small precisely where small features are located, and larger elsewhere. Such a 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.

Here we first present LTS schemes based on explicit Runge-Kutta (RK) methods. Starting from classical or low-storage explicit RK methods, we derive explicit LTS methods of arbitrarily high accuracy. We prove that the LTS-RK$s(p)$ methods yield the same rate of convergence as the underlying RKs scheme. 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.

As a second method we propose local exponential Adams-Bashforth (LexpAB) schemes. Unlike LTS schemes, LexpAB methods overcome the severe stability restrictions caused by local mesh refinement not by integrating with a smaller time-step but by using the exact matrix exponential in the fine region. Thus, they present an interesting alternative to the LTS schemes. Numerical experiments in 1D and 2D confirm the expected order of convergence and demonstrate the versatility of the approach in cases of extreme refinement.

ACKNOWLEDGEMENTS

This thesis would not have been possible without the help and support of so many people. My sincere thanks go to Prof. Dr. Marcus Grote for giving me the opportunity of doing a PhD in his group and for his guidance and support throughout the time.
I would like to thank Dr. Stéphane Lanteri for giving me the possibility of presenting my work in his Minisymposium at ACOMEN 2014 and for kindly agreeing to be co-referee in the thesis committee.
My thanks go also to Prof. Dr. Marlis Hochbruck and Prof. Dr. Alexander Ostermann for the helpful discussions and their advice on exponential integrators.
I would like to thank all my colleagues and the administrative staff at the Mathematical Institute, all the former and current members of the Numerical Analysis group, in particular Uri Nahum and Dr. Marie Kray for their moral support, many helpful discussions, lots of tea and many hours of laughter. Further I would like to thank Dr. Teodora Mitkova for making the start into the PhD world as smooth as possible and for always having an open ear and many useful advises.
A big thank you goes to Dr. Loredana Gaudio for being the best office mate anyone could wish for, for the constant supply of chocolate and for carefully proof-reading parts of this thesis.
I wish to thank my friends for all the encouragement and love, especially Manuela Utzinger for being at my side since day one of the Bachelor, Meinrad Egger for always believing in my abilities even at times when I didn't and Justine Heusser for being my lighthouse in times when the sea was rough and I couldn't see the coast. You are the best friend I could wish for.
My deepest gratitude goes to my wonderful family for all their support. Words cannot express how much you mean to me.
This thesis was partly supported by the Swiss National Science Foundation.

CONTENTS

1 INTRODUCTION 1
  1.1 Model problems 1
  1.2 Multirate methods 4
  1.3 Local time-stepping schemes 5
    1.3.1 Locally implicit methods 5
    1.3.2 Locally explicit schemes 6
    1.3.3 Local exponential integrators 7
  1.4 Outline of the thesis 8

i MODEL PROBLEMS AND SPATIAL DISCRETIZATION 9
2 INTRODUCTION 11
3 FINITE ELEMENT DISCRETIZATIONS OF WAVE EQUATIONS 13
  3.1 Continuous FEM 13
  3.2 Nodal Discontinuous Galerkin Methods 16
4 MAXWELL'S EQUATIONS 21
  4.1 Spatial Discretization 21

ii RUNGE-KUTTA BASED EXPLICIT LOCAL TIME-STEPPING METHODS 25
5 INTRODUCTION 27
6 RUNGE-KUTTA BASED LTS 29
  6.1 Runge-Kutta methods 29
  6.2 Explicit RK based LTS methods 30
  6.3 Low-storage RK based LTS methods 34
  6.4 The RK4 based LTS method 36
7 ACCURACY AND CONVERGENCE 39
8 NUMERICAL EXPERIMENTS 49
  8.1 Numerical results for the wave equation 49
  8.2 Numerical results for Maxwell's Equation 59

iii LOCAL EXPONENTIAL ADAMS BASHFORTH SCHEMES 63
9 INTRODUCTION 65
10 LOCAL EXPONENTIAL ADAMS-METHODS 67
  10.1 Exponential methods of Adams-type 67
  10.2 Application to the damped wave equation 69
  10.3 Efficient Implementation 70

11 NUMERICAL EXPERIMENTS 73
  11.1 Convergence study 73
  11.2 Stability 75
  11.3 Two-dimensional example 81
  11.4 Comparison 83
  11.5 Krylov-Methods 85

iv CONCLUSION AND FUTURE WORK 93
12 CONCLUSION AND FUTURE WORK 95
  12.1 Concluding Remarks 95
  12.2 Future Work 96

v APPENDIX 97
A APPENDIX 99
References 101
B CURRICULUM VITAE 111

LIST OF FIGURES

Figure 1 Element K with neighbors. 18
Figure 2 The theoretical speed-up Q_{eff} vs. r for $p = 2, 5, 10, 100$ (a) and Q_{eff} vs. p for $r = 0.5, 0.1, 0.05, 0.01$ (b). 34
Figure 3 One-dimensional example: the computational domain $\Omega = [0, 6]$ with the refined region $\Omega_f = [2, 4]$. 50
Figure 4 LTS-RK4(p) L^2-error at $T = 10$ vs. $H = h^{\text{coarse}} = 0.2, 0.1, 0.05, 0.025$ for different $\mathcal{P}^3$ FEM with $p = 2, 5, 11$. The errors for different p coincide at this scale - see also Table 2. 51
Figure 5 LTS-LSRK5(p) L^2-error at $T = 10$ vs. $H = h^{\text{coarse}} = 0.2, 0.1, 0.05, 0.025$ for different $\mathcal{P}^3$ FEM with $p = 2, 5, 11$. The errors for different p coincide at this scale. 51
Figure 6 Stability domains of RKs methods for $k = s = 2, 3, 4$ and RK5 of order 4. 52
Figure 7 The spectral radius of $\mathbf{B}$ for different $\mathcal{P}^3$ FEM, with $\sigma = 0$ (first row) and $\sigma = 0.1$ (second row). 53
Figure 8 The spectral radius of $\mathbf{C}_{LTS-RK4}$ vs. $\Delta t/\Delta t_{RK4}$ for different $\mathcal{P}^3$ FEM. 54
Figure 9 The spectral radius of $\mathbf{C}_{LTS-LSRK5}$ for different $\mathcal{P}^3$ FEM. 56
Figure 10 The initial triangular mesh (left); zoom on the "fine" mesh indicated by the darker (green) triangles (right). 57
Figure 11 Two-dimensional example: the solution is shown at times t =0.1, 0.3, 0.45, 0.55, 0.7 and 0.9. 58
Figure 12 LTS-RK2(p) error vs. $H = h^{\text{coarse}} = 0.02, 0.01, 0.005, 0.0025$ for $\mathcal{P}^1$ finite elements with $p = 3, 7, 13$. 60
Figure 13 LTS-RK4(p) error (a) and LTS-LSRK5(p) (b) vs. $H = h^{\text{coarse}} = 0.04, 0.02, 0.01, 0.005$ for $\mathcal{P}^3$ finite elements with $p = 3, 7, 13$. 60
Figure 14 The initial triangular mesh : $h_{fine} \approx h_{coarse}/7$ 61
Figure 15 Snapshots of $\mathbf{E}_h^z$ at times $t = 0.2, 0.35, 0.5, 0.75$ 62
Figure 16 Exact and numerical solution at time $T = 1.5$ (a) and Lexp-AB2(p) error vs. $H = h^{\text{coarse}} = 0.04, 0.02, 0.01, 0.005$ for $\mathcal{P}^1$ finite elements with $p = 2, 5, 13$(b). 74
Figure 17 Lexp-ABk(p) error vs. $H = h^{\text{coarse}} = 0.04, 0.02, 0.01, 0.005$ for $\mathcal{P}^{k-1}$ finite elements with $p = 2, 5, 13$ for $k = 3$ (a) and $k = 4$ (b). 75
Figure 18 Eigenvalues of $\mathbf{C}_{AB2}$ (top) and $\mathbf{C}_{LexpAB2}$ (bottom) for $\sigma = 0.1$ (left), 1 (middle) and 20 (right) and $\Delta t = \Delta t_{\text{opt}}$. 77

Figure 19 Maximal norm of the eigenvalues of $\mathbf{C}_{AB2}$ (top) and $\mathbf{C}_{LexpAB2}$ respectively with respect to Δt for $\sigma = 0.1$ (left), 1 (middle) and 20 (right) 78

Figure 20 Eigenvalues of $\mathbf{C}_{AB3}$ (top) and $\mathbf{C}_{LexpAB3}$ (bottom) for $\sigma = 0.1$ (left), 1 (middle) and 20 (right) and $\Delta t = \Delta t_{\text{opt}}$. 79

Figure 21 Maximal norm of the eigenvalues of $\mathbf{C}_{AB3}$ and $\mathbf{C}_{LexpAB3}$ respectively for $\sigma = 0.1$ (left), 1 (middle) and 20 (right) 80

Figure 22 Eigenvalues of $\mathbf{C}_{AB4}$ (top) and $\mathbf{C}_{LexpAB4}$ (bottom) for $\sigma = 0.1$ (left), 1 (middle) and 20 (right) and $\Delta t = \Delta t_{\text{opt}}$. 81

Figure 23 Maximal norm of the eigenvalues of $\mathbf{C}_{AB4}$ (top) and $\mathbf{C}_{LexpAB4}$ respectively for $\sigma = 0.1$ (left), 1 (middle) and 20 (right) 82

Figure 24 Initial triangular mesh with local refinement ratio $p = 6$, i.e. $h^{\text{fine}} \approx h^{\text{coarse}}/6$ 82

Figure 25 Numerical solution at times $t = 0.1$, 0.3, 0.5 and 0.7 83

Figure 26 Ω with one fine element. 84

Figure 27 Ω with three fine element. 84

Figure 28 Reference solution (blue) and solution calculated using an Arnoldi method with $m = 10$ (green) at time $T = 3.5$ 88

Figure 29 Eigenvalues of $\mathbf{B}$ (left), of $\mathbf{H}_{10,0}$ and of a Hessenberg matrix $\mathbf{H}$ generated with a random vector $\mathbf{v}$. 89

Figure 30 Eigenvalues of $\mathbf{H}_{10,1}$ (left), $e^{\frac{T}{3}\mathbf{H}_{10,1}}$ (middle) and $\mathbf{V}_{10,1}e^{\frac{T}{3}\mathbf{H}_{10,1}}\mathbf{V}_{10,1}^{\top}$ (right). 89

LIST OF TABLES

Table 1 Butcher-tableau of an explicit RKs scheme. 30
Table 2 L^2-error at $T = 10$ for $h^{\text{coarse}} = 0.05$ for varying p. 50
Table 3 Stability of the LTS-RK$s(p)$ schemes for $s = 2, 3, 4$ when combined with a continuous FE or nodal DG-FE discretization with $h^{\text{coarse}} = 0.2$: the ratio $\Delta t_p / \Delta t_{RKs}$ is shown for varying p. 55
Table 4 Stability of the LTS-RK$s(p)$ schemes for $s = 2, 3, 4$ when combined with a continuous FE or nodal DG-FE discretization with $h^{\text{coarse}} = 0.05$: the ratio $\Delta t_p / \Delta t_{RKs}$ is shown for varying p. 56
Table 5 The LTS-RK$s(2)$ scheme for $s = 2, 3, 4$ when combined with a continuous FE or nodal DG-FE discretization: the ratio $\Delta t_2 / \Delta t_{RKs}$ is shown for varying σ. 57
Table 6 Coefficients for the Adams-Bashforth methods 69
Table 7 Δt_{AB2} for linear FE combined with AB2 on equidistant coarse mesh for different σ. 76
Table 8 Δt_{AB3} for quadratic FE combined with AB3 on equidistant coarse mesh for different σ. 78
Table 9 Δt_{AB4} for cubic FE combined with AB4 on equidistant coarse mesh for different σ. 80
Table 10 Runtime results for $\mu = 1$ and $h^{\text{coarse}} = 0.2$ for different p 84
Table 11 Runtime results for $\mu = 1$ and $h^{\text{coarse}} = 0.05$ for different p 85
Table 12 Runtime results for $\mu = 3$ and $h^{\text{coarse}} = 0.2$ for different p 86
Table 13 Runtime results for $\mu = 3$ and $h^{\text{coarse}} = 0.05$ for different p 87
Table 14 Case A with $h = 0.1$ and $h^{\text{fine}} = 0.001$. 90
Table 15 Case B with $h = 0.1$ and $h^{\text{fine}} = 0.001$. 90
Table 16 Case A with $h = 0.01$ and $h^{\text{fine}} = 0.0001$. 91
Table 17 Case B with $h = 0.01$ and $h^{\text{fine}} = 0.0001$. 91
Table 18 Coefficients of the classical RKs methods. 99
Table 19 Coefficients of the LSRKs methods. 100

1

INTRODUCTION

Wave phenomena appear in many fields of science, for example in seismology, acoustics and electromagnetics.

We can distinguish between two main type of waves. *Mechanical waves*, the first type, need a medium to propagate and are characterized by transferring energy through a medium. Mechanical waves can only occur in media which possess elasticity and inertia. Examples are sound, seismic and water waves. The medium further characterizes the properties of the wave (e.g. speed). The second type of waves are *electromagnetic waves*. They require no medium to travel within and can propagate in vacuum, but they can still exist in a medium. Typical examples are light (visible, infrared and ultraviolet), radio waves, microwaves, X-rays and many more.

Equations that model wave propagation can be assigned to three physical categories. The acoustics equation and the elastic system model mechanical waves in fluids and solids, respectively, and Maxwell's equations, which describe the propagation of electromagnetic waves. As these equations so widely appear in science it is of importance to have accurate and efficient numerical solvers.

We focus in this thesis on two types of equations: on acoustic wave equations and Maxwell's equations. We consider both in time domain rather than frequency domain. Both approaches have their advantages, but time dependent equations appear to have a wider range of applications. Especially when dealing with pulse sources or radar sources covering a large range of frequencies time dependent equations model the phenomena better.

1.1 MODEL PROBLEMS

Wave equation

As a first model problem we consider the damped wave equation

$$\frac{\partial^2 u}{\partial t^2}(\mathbf{x},t) + \sigma \frac{\partial u}{\partial t}(\mathbf{x},t) - \nabla \cdot (c^2 \nabla u(\mathbf{x},t)) = f(\mathbf{x},t) \quad \text{in } \Omega \times (0,T), \tag{1.1}$$

a standard model for wave phenomena. Here Ω is a bounded domain in $\mathbb{R}^d$, $d = 1,2,3$, $\mathbf{x} \in \mathbb{R}^d$, f is a (known) source term. The damping coefficient, $\sigma = \sigma(\mathbf{x})$, is assumed non-negative ($\sigma(\mathbf{x}) \geq 0$) whereas the speed of propagation, $c = c(\mathbf{x})$, is piecewise smooth and strictly positive ($c(\mathbf{x}) \geq c_0 > 0$). If σ is identically zero throughout Ω, then (1.1) coincides with the classical (undamped) wave equation. Given initial and boundary conditions complete the model. For a more detailed description see Chapter 3.

Maxwell's equations

The second problem we want to study in this work is the propagation of a time-dependent electromagnetic field through a linear isotropic medium. The set of equations

$$\frac{\partial \mathbf{D}}{\partial t}(\mathbf{x},t) - \nabla \times \mathbf{H}(\mathbf{x},t) = -\mathbf{J}(\mathbf{x},t) \quad \text{in } \Omega \times (0,T), \tag{1.2}$$

$$\frac{\partial \mathbf{B}}{\partial t}(\mathbf{x},t) + \nabla \times \mathbf{E}(\mathbf{x},t) = 0 \quad \text{in } \Omega \times (0,T), \tag{1.3}$$

$$\nabla \cdot \mathbf{B}(\mathbf{x},t) = 0 \quad \text{in } \Omega \times (0,T), \tag{1.4}$$

$$\nabla \cdot \mathbf{D}(\mathbf{x},t) = \rho(\mathbf{x},t) \quad \text{in } \Omega \times (0,T), \tag{1.5}$$

known as Maxwell's equations, forms the basis of classical electromagnetic phenomena and describes the interaction between $\mathbf{E}$, $\mathbf{H}$, $\mathbf{D}$ and $\mathbf{B}$, the electric and magnetic fields and the electric and magnetic inductions, respectively. The vector $\mathbf{E}$ contains x-, y- and z-component of the electric field, i.e. $\mathbf{E} = (E^x, E^y, E^z)$. $\mathbf{H}$, $\mathbf{D}$ and $\mathbf{B}$ are defined likewise. $\mathbf{J}$ denotes here the current density and ρ a given charge density. For this model we assume that $\Omega \subset \mathbb{R}^d$, $d = 1,2,3$, is a bounded polygon/polyhedron.

For linear and isotropic material we have additionally the following relations

$$\mathbf{D}(\mathbf{x},t) = \varepsilon(\mathbf{x})\mathbf{E}(\mathbf{x},t), \tag{1.6}$$

$$\mathbf{B}(\mathbf{x},t) = \mu(\mathbf{x})\mathbf{H}(\mathbf{x},t), \tag{1.7}$$

which we can use to eliminate $\mathbf{D}$ from (1.2) and $\mathbf{B}$ from (1.3). Here μ and ε denote the relative magnetic permeability and the relative electric permittivity, respectively, and are positive, bounded scalar functions of position.

Moreover, if the fields are not too strong, we can assume that Ohm's law holds and thus

$$\mathbf{J}(\mathbf{x},t) = \sigma\mathbf{E}(\mathbf{x},t) + \mathbf{J}_a(\mathbf{x},t), \tag{1.8}$$

where σ is a non-negative function of position, the so-called conductivity of the medium, while $\mathbf{J}_a$ describes the applied current density.

In the following we shall generally refer to the set of equations

$$\varepsilon\frac{\partial}{\partial t}\mathbf{E}(\mathbf{x},t) - \nabla \times \mathbf{H}(\mathbf{x},t) + \sigma\mathbf{E}(\mathbf{x},t) = -\mathbf{J}_a(\mathbf{x},t) \quad \text{in } \Omega \times (0,T), \tag{1.9}$$

$$\mu\frac{\partial}{\partial t}\mathbf{H}(\mathbf{x},t) + \nabla \times \mathbf{E}(\mathbf{x},t) = 0 \quad \text{in } \Omega \times (0,T), \tag{1.10}$$

as Maxwell's equations, which we get from inserting (1.6) and (1.8) into (1.2) and (1.7) into (1.3). Regions where σ is positive are called *conductors*. Where $\sigma = 0$ and $\varepsilon \neq \varepsilon_0$, the material is termed a dielectric, and ε is referred to as the dielectric constant. In vacuum (or air at low field strength) we find that $\sigma = 0$, $\varepsilon = \varepsilon_0$ and $\mu = \mu_0$. We again need initial and boundary conditions to complete the model. It can be shown that if (1.4) and (1.5) hold at one time, they hold for all time. Thus, if the initial conditions satisfy (1.4) and (1.5), the solutions of (1.9) and (1.10) fulfill the divergence conditions and we can omit them in our set of equations.

Both the damped wave equation (1.1) and Maxwell's equations (1.9)–(1.10) are examples of partial differential equations (PDEs) in space and time. Following the method-of-lines approach we discretize them first in space and then in time. Different spatial discretizations are available; among the most popular choices are finite differences (FD). FD methods often reach their limit when dealing with locally refined meshes. These meshes are very useful to capture local behavior originating from small geometric features without introducing too many unknowns and are thus computationally efficient. In this work we are using finite element methods (FEM). These methods, in their continuous and discontinuous form, are well able to handle locally refined unstructured meshes and their extension to higher orders is straightforward. For more details we refer to Part I of this work. The spatial discretization of our model problems leads to ordinary differential equations (ODEs) in time, which we subsequently need to solve.

Locally refined meshes, however, impose severe stability restrictions on explicit time integration as the maximal time-step allowed by the CFL condition is dictated by the smallest element 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.

Over the last decades various schemes, such as multirate and local time-stepping (LTS) methods, have been developed to overcome this geometry induced stiffness.

In this work we propose high-order local time-stepping (LTS) schemes based on explicit Runge-Kutta (RK) methods and local exponential Adams-Bashforth (Lexp-AB) methods. By using smaller time-steps or the exact matrix exponential precisely where the small elements in the mesh are located, these methods overcome the severe stability restrictions caused by local mesh refinement without sacrificing the explicitness, accuracy or efficiency of their underlying explicit time integration scheme.

Before we describe the two methods in detail we give an overview on existing multirate and LTS methods in Sections 1.2 and 1.3. A short survey on some of the methods mentioned here can also be found in [44].

1.2 MULTIRATE METHODS

Multirate schemes were developed in the ODE community for first order systems, where different components need different time scales. Already in 1960 Rice [90] derived split RK methods for equations of the form

$$x'(t) = F(x, y, t), \quad x(0) = x_0, \tag{1.11}$$

$$y'(t) = G(x, y, t), \quad y(0) = y_0, \tag{1.12}$$

where the component $y(t)$ does not depend strongly on $x(t)$ or varies much more rapidly than $x(t)$. The scheme proposed in [90] allows (1.11) and (1.12) to be solved with different time-steps, while keeping the self-starting property of the classical RK method and the ability to change the integration step easily.
Gomm and later Andrus presented a stability analysis of multirate versions of linear multistep methods [46] and fourth- and third-order RK methods [2]. Gear and Wells [45] described multirate schemes based on linear multistep methods with automatic stepsize and order control. In [94] Skelboe and Andersen studied the stability for multirate schemes based on backward Euler.
In [52] Günther, Kværnø and Rentrop introduced multirate partitioned RK methods, which can be viewed as a generalization of both partitioned RK [89] and multirate Rosenbrock-Wanner methods [53]. Starting from a first order ODE the unknowns are divided into an active component, which is rapidly changing, and a latent but stiff component. The computation of the active (non-stiff) part using an explicit scheme with a small time-step is coupled with the linearly implicit computation of the latent (stiff) part.
The multirate methods mentioned so far all require an a priori knowledge of the splitting into active and latent components. To overcome this, Engstler and Lubich [39] proposed a multirate method based on Richardson extrapolation. This approach allows for a dynamic partitioning into several classes of slow to fast components.
A finite element based approach is presented by Logg in [74, 75]. The multi-adaptive Galerkin method for ODEs even lets each individual component have its own time-step, but it also leads to complicated implicit relations which need to be solved iteratively across each time-slab.
More recently, Savcenco, Hundsdorfer and Verwer [92] introduced a self-adjusting multirate time-stepping strategy for stiff ODE's starting from a second-order (linearly implicit) Rosenbrock method with an embedded first-order method. This approach allows large time steps for inactive components, without sacrificing accuracy.
In their recent work [29] Demirel et al. presented multiple time-stepping (MTS) algorithms which allow any choice of explicit Adams type or predictor-corrector

scheme for the non-stiff integration and any choice of ODE solver for the stiff part. Furthermore, they constructed MTS schemes with optimized stability contours for both stiff and non-stiff integration.

1.3 LOCAL TIME-STEPPING SCHEMES

So-called local time-stepping methods present another approach. They were developed in the PDE community to overcome the bottleneck in explicit time-stepping caused by locally refined meshes by dividing the mesh into two parts: ″fine″ and ″coarse″. While they integrate in the coarse region with an explicit scheme, they resolve the stability restriction on the fine part by using either an implicit scheme (locally implicit methods) or an explicit method with a finer time-step (locally explicit methods).

1.3.1 *Locally implicit methods*

Locally implicit methods build on the long tradition of hybrid implicit-explicit (IMEX) algorithms for operator splitting for advection-diffusion-reaction equations. The different terms appearing in the ODE after spatial discretization possess different characteristics and IMEX schemes allow to treat the stiff term (e.g. diffusion term) implicitly while using an explicit scheme for the non-stiff part (e.g. advection) – see [4, 67] and the references therein. In 2006, Piperno [87] presented a combination of 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 [34, 32]. By using the LF/CN-IMEX approach of Verwer [98] instead, Descombes, Lanteri and Moya [32] resolved 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. [68] applied a fourth-order IMEX RK method [70] 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. This can however be corrected with the use of suitable preconditioners.

1.3.2 *Locally explicit schemes*

Unlike locally implicit schemes, 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. One of the first local time-stepping methods based on the forward Euler scheme was presented by Osher and Sanders in [85] for one-dimensional conservation laws using finite volume schemes in space. In the mid-eighties Berger and Oliger [12] and Berger and Colella [10] proposed a local adaptive mesh refinement (AMR) strategy for first-order nonlinear hyperbolic conservation laws, based on a hierarchy of rectangular finite-difference grids. In [11] Berger and LeVeque extended this strategy to hyperbolic equations not necessarily in conservation form.

More recently, Dawson and Kirby [28] developed local time-stepping for conservation laws based on forward Euler and second order time discretizations, which satisfy, as an additional property, a maximum principle.

To achieve higher-order schemes Baeza and Mulet [5] and Dumbser et al. [36] combined the AMR strategy with weighted essentially non-oscillatory (WENO) reconstruction techniques.

As already mentioned, finite element based approaches are a natural choice in the presence of complex geometry or adaptive mesh refinement due to their flexibility.

In [42] Flaherty et al. presented the first combination of DG FE on unstructured meshes and LTS. The results, however, were restricted to first order.

In [41] Ezziani and Joly combined the energy conserving, second-order LTS approach from Collino et al. [23, 24, 25] 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 [87], 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. In [79] Montseny et al. presented a local time-stepping strategy defined by a recursive multi-class method based upon a leap-frog scheme, which they combined with discontinuous hexahedral elements. Starting from the standard LF method, energy conserving fully explicit LTS integrators of arbitrarily high even accuracy were proposed for the classical wave equation [33] by Diaz and Grote. In [48] Grote and Mitkova extended this approach to Maxwell's equation for non-conductive media in second-order formulation. Furthermore, an explicit LTS scheme is derived for conducting media by combining LF and Crank-Nicolson methods. This is, however, limited to second-order.

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 by Taube et al. [97] and Dumbser et. al [35]. Constantinescu et al. and later Krivodonova devised multirate explicit methods for hyperbolic conservation laws , which are based on Runge-Kutta [26, 71] and Adams-Bashforth [27] schemes; again these schemes are limited to second-order accuracy. Monotonicity conditions of multirate schemes based on first and second order explicit RK methods [85, 96, 26, 92] were analyzed by Hundsdorfer et al. in [66].

To achieve arbitrarily high accuracy in the presence of dissipation, while remaining fully explicit, Grote and Mitkova [49] derived LTS methods for damped wave equations based on Adams-Bashforth (AB) multi-step schemes; they can also be interpreted as particular approximations of exponential-Adams multi-step methods [63]. Recently, Angulo et al. [3] proposed a "causal-path" LTS technique for both the second-order LF and a fourth-order low-storage (LS) explicit RK method [15], which assorts elements in tiers according to their size and computes required intermediate values recursively.

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.

1.3.3 *Local exponential integrators*

The second type of methods we investigate in this work are local exponential Adams-Bashforth (Lexp-AB) schemes which are based on exponential Adams methods.

In [63] Hochbruck and Ostermann presented exponential multistep methods for semilinear problems of the form

$$u'(t) = -Au(t) + g\left(t, u(t)\right), \qquad u(t_0) = u_0,$$

which, for $A = 0$ reduce to well-known AB methods. Beside the construction, implementation and numerical analysis of exponential Adams methods, the article further provides a stiff error analysis performed in the abstract framework of linear semigroups. An error analysis based on order conditions for exponential general linear methods is presented by Ostermann et al. in [86]. Exponential Adams methods were first introduced by Certaine in [16] and by Nørsett in [84]. They are part of a long tradition of exponential integrators dating back to the 60th and regaining a stronger interest in recent years. For an overview on different methods, applications and extensive references we refer to Hochbruck and Ostermann [62].

Instead of choosing a smaller time-step in the fine region Lexp-AB methods calculate the matrix exponential and can thus keep one global time-step. The idea was first presented in [63]. Especially when facing problems with very few, but very small, fine elements, these methods present an interesting alternative to LTS schemes.

1.4 OUTLINE OF THE THESIS

The rest of the thesis is structured in three parts.
In Part I we introduce the model problems (1.1) and (1.9)-(1.10) in more details as to function spaces, boundary and initial conditions, etc. Furthermore, we introduce variational formulations, which build the basis for our finite element discretizations, both continuous and DG. Additionally we present some convergence results for semi-discrete schemes.

Part II describes the derivation of RK based LTS schemes. We further present and prove theoretical convergence results and conclude by giving numerical results, both in one dimension, which confirm our convergence results and show stability results, and two dimensions, for both wave equations and Maxwell's equations.

In Part III we propose Lexp-AB methods. Numerical results in one spatial dimension show the expected high rates of convergence and optimal stability properties. We explain with the help of examples why the use of Krylov methods to evaluate matrix exponential times vector does usually not give the desired results when applied to ODEs coming from the FE discretization of wave equations. Additionally we present numerical results in two spatial dimensions.

We end this work with concluding remarks and an outlook to possible future work.

Part I

MODEL PROBLEMS AND SPATIAL DISCRETIZATION

2

INTRODUCTION

A variety of different spatial discretizations for model problems describing wave phenomena such as (1.1) or (1.9)–(1.10) is available when following the method-of-lines strategy. As mentioned in Section 1.1 FD present an intuitive and easy approach, as they directly start from the PDEs (1.1) or (1.9)–(1.10) and replace the appearing spatial derivatives by difference quotients. Due to their simplicity they are still popular and often present the method of choice in engineering. The construction of higher-order schemes, which are essential to achieve the necessary high accuracy, is however more involved. [1] is one of the earliest work on a fourth-order FD scheme for the acoustic wave equation. It presents a 9-point scheme in space for the wave equation in two dimensions. Extensive work on higher-order schemes in space began in geophysics around 1985. A nice overview and extensive references can be found in [21], Part II.

In electromagnetics the research on FD methods for time dependent problems was less active. However, among the most popular methods to advance time-dependent electromagnetic fields we find the Yee-scheme [100], which uses second order FD in space and time on a staggered grid and is still regarded as a reference method. For a detailed overview on FD methods for Maxwell's equations see [95].

The main drawback of FD methods is that they are restricted to quadrilateral and hexahedral meshes in two and three dimensions, respectively. Hence, they are not well suited for complex geometries and local mesh refinement.

FEM appear to be a natural solution to this problems as they are able to handle locally refined and unstructured grids. However, such methods introduce a mass-matrix $\mathbf{M}$, which needs to be inverted even when applying explicit time integration schemes, thus compromising the efficiency of explicit methods. To avoid that computational work, various mass-lumping techniques have been developed [20, 80], which replace $\mathbf{M}$ by a diagonal approximation without spoiling the accuracy [9].

Approximating Maxwell's equations with standard (H^1- conforming) finite elements would lead to a numerical solution without singularities, even in the presence of re-entrant corners. With this approach we would consequently loose actual physical solutions. The appropriate space to consider is $H(\mathrm{curl}, \Omega)$ which only asks continuity of the tangential component of a field, but allows its normal component to jump across element boundaries. The first finite elements based on this principle can be found in [81, 82]. [78] gives an overview on these so-called *edge elements*. In [22] Cohen and Monk presented mass-lumping for Maxwell's equations and mixed finite elements on quadrilateral meshes, which can be extended to triangular meshes [37, 38].

When considering problems in 3D, mass-lumping introduces additional degrees of freedom (23 dofs instead of 10 for third-order methods). This drastically influences their efficiency. Furthermore, mass-lumped finite elements in 3D are only known up to order four [17]. Spectral element methods (SEM) [14, 69] present an alternative. They can be interpreted as extensions of mass-lumped finite elements in 1D to higher dimensions [21] and provide a diagonal mass matrix. However, they are again restricted to quadrilateral and hexahedral meshes.

DG methods automatically lead to a block-diagonal mass matrix with block size equal to the number of degrees of freedom per element; therefore $\mathbf{M}$ is computationally cheap to invert. Furthermore DG schemes allow for locally refined and even irregular nonmatching meshes without enforcing continuity across element boundaries. The first DG method was presented by Reed and Hill in [88] for the solution of the steady-state neutron transport equation. Today various different DG schemes are available [19, 91, 50, 51]. For an extensive overview on the development of DG schemes see [18, 57, 31] and references therein.

In this work we focus on two different discretizations of our model problems: continuous FEM with mass-lumping [20, 80] (for the wave equation) and nodal DG methods [56, 57] (for both wave equation and Maxwell's equations).

The rest of this part on model problems and spatial discretizations is structured as follows. In Chapter 3 we present the wave equation in more detail. We recall both continuous FEM with mass-lumping and nodal DG FEM as discretization methods to attain ODEs in time, which we can rewrite as a general first-order system of differential equations. In Chapter 4 we extend the nodal DG formulation to Maxwell's equations. This again leads to first-order system of ODEs in time.

3

FINITE ELEMENT DISCRETIZATIONS OF WAVE EQUATIONS

In this chapter we present the wave equation (1.1) in more detail. We state some classical existence, uniqueness and regularity results for the weak solution of the wave equation and present spatial discretizations based on standard H^1-conforming FEM with mass-lumping and nodal DG methods.
We remind the reader of the wave equation presented in (1.1)

$$\begin{aligned} u_{tt} + \sigma u_t - \nabla \cdot (c^2 \nabla u) &= f \quad \text{in } \Omega \times (0,T), \\ u(\cdot,t) &= 0 \quad \text{on } \partial\Omega \times (0,T), \\ u(\cdot,0) = u_0\,,\ u_t(\cdot,0) &= v_0 \quad \text{in } \Omega, \end{aligned} \tag{3.1}$$

which serves as a standard model problem for wave phenomena. As introduced before, Ω is a bounded domain in $\mathbb{R}^d$, f is a (known) source term, while u_0 and v_0 are prescribed initial conditions. At the boundary, $\partial\Omega$, we impose a homogeneous Dirichlet boundary condition, for simplicity. The damping coefficient, $\sigma = \sigma(\mathbf{x})$, is assumed non-negative ($\sigma(\mathbf{x}) \geq 0$) whereas the speed of propagation, $c = c(\mathbf{x})$, is piecewise smooth and strictly positive ($c(\mathbf{x}) \geq c_0 > 0$). If σ is identically zero throughout Ω, then (3.1) coincides with the classical (undamped) wave equation.

3.1 CONTINUOUS FEM

Various FEM are available for the spatial discretization of (3.1). For instance, the standard H^1-conforming FEM with mass-lumping starts from the weak formulation: Find $u : [0,T] \to H_0^1(\Omega)$ such that

$$\begin{aligned} (u_{tt}, v) + (\sigma u_t, v) + (c\nabla u, c\nabla v) &= (f, v) \qquad \forall\, v \in H_0^1(\Omega),\ t \in (0,T), \\ u|_{t=0} &= u_0 \qquad \text{in } \Omega, \\ u_t|_{t=0} &= v_0 \qquad \text{in } \Omega, \end{aligned} \tag{3.2}$$

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

$$(w, v) = \int_\Omega w(\mathbf{x}) v(\mathbf{x})\, d\mathbf{x}.$$

For $\sigma = 0$ (3.2) reduces to the variational formulation of the classical wave equation. After several assumptions on initial conditions and source term we get the following result on the existence and uniqueness of a weak solution of (3.2).

Theorem 3.1. *Let* $u_0 \in H_0^1(\Omega)$*,* $v_0 \in L^2(\Omega)$ *and* $f \in L^2(0,T;L^2(\Omega))$*. Then there exists a unique weak solution of (3.2)* $u \in L^2(0,T;H_0^1(\Omega))$ *with* $u_t \in L^2(0,T;L^2(\Omega))$*.*

The proof can be found in [40]. Furthermore one can proof that the solution has actually even more regularity, i.e. $u \in C(0,T;H_0^1(\Omega)) \cap C^1(0,T;L^2(\Omega))$. For $\sigma(\mathbf{x}) > 0$ similar results can be found in [73].

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

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

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

The semi-discrete Galerkin approximation, $u^h(t) \in V_h$, is then defined for $0 \leq t < T$ by the restriction of (3.2) to V_h: Find $u^h : [0,T] \to V^h$ such that

$$\begin{aligned} (u_{tt}^h, v) + (\sigma u_t^h, v) + (c\nabla u^h, c\nabla v) = (f, v) \qquad &\forall\, v \in V^h\,, \quad t \in (0,T)\,, \\ u^h(\cdot, 0) = \Pi_h u_0\,,\ u_t^h(\cdot,0) = \Pi_h v_0\,. & \end{aligned} \tag{3.3}$$

Here, Π_h denotes the L^2-projection onto V^h. In [7] semi-discrete convergence results for (3.3) as well as fully-discrete convergence proofs are presented. This was extended in [8] to a first-order formulation of (3.3).

Let $\mathbf{U}(t) \in \mathbb{R}^N$ denote the coefficients of $u^h(t)$ with respect to the standard Lagrangian basis $\{\phi_i\}_{i=1,\dots,N}$ of V_h. Then (3.3) is equivalent to the second-order system of ODEs

$$\begin{aligned} &\mathbf{M}\,\frac{d^2\mathbf{U}}{dt^2}(t) + \mathbf{M}_\sigma\,\frac{d\mathbf{U}}{dt}(t) + \mathbf{K}\,\mathbf{U}(t) = \mathbf{R}(t)\,, \qquad t \in (0,T), \\ &\mathbf{M}\mathbf{U}(0) = u_0^h, \quad \mathbf{M}\frac{d\mathbf{U}}{dt}(0) = v_0^h, \end{aligned} \tag{3.4}$$

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

$$\mathbf{M}_{ij} = (\phi_j, \phi_i), \qquad \mathbf{K}_{ij} = (c\nabla\phi_j, c\nabla\phi_i);$$

the matrix $\mathbf{M}_\sigma$ also corresponds to a mass matrix with weight σ. 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 (3.4), every time-step then requires the solution of a linear system involving $\mathbf{M}$ or multiplication with the inverse of $\mathbf{M}$, which need no longer be sparse. To avoid that computational work, various mass-lumping techniques have been developed [20, 80], which replace $\mathbf{M}$ by a diagonal approximation without spoiling the accuracy [9]. Alternatively, the spectral element method [14, 69] and the symmetric interior penalty DG method [50] 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 mentioned alternative discretizations also lead to (3.4) with an essentially diagonal mass matrix $\mathbf{M}$.

Since $\mathbf{M}$ in (3.4) is symmetric, positive definite and (block) diagonal, the matrix $\mathbf{M}^{-\frac{1}{2}}$ is immediately available. Then, multiplication of (3.4) by $\mathbf{M}^{-\frac{1}{2}}$ yields

$$\frac{d^2\mathbf{z}}{dt^2}(t) + \mathbf{D}\frac{d\mathbf{z}}{dt}(t) + \mathbf{A}\mathbf{z}(t) = \widetilde{\mathbf{R}}(t), \tag{3.5}$$

with

$$\mathbf{z}(t) = \mathbf{M}^{\frac{1}{2}}\mathbf{U}(t), \quad \mathbf{D} = \mathbf{M}^{-\frac{1}{2}}\mathbf{M}_\sigma\mathbf{M}^{-\frac{1}{2}}, \quad \mathbf{A} = \mathbf{M}^{-\frac{1}{2}}\mathbf{K}\mathbf{M}^{-\frac{1}{2}}, \quad \widetilde{\mathbf{R}}(t) = \mathbf{M}^{-\frac{1}{2}}\mathbf{R}(t).$$

Again, the matrix $\mathbf{A}$ is sparse, symmetric and positive semi-definite.

LTS techniques based on leap-frog type methods start from the second-order system (3.5). To apply a RK type or AB method, however, we first need to rewrite (3.5) as a first-order system

$$\begin{aligned} &\frac{d\mathbf{y}}{dt}(t) = \mathbf{B}\mathbf{y}(t) + \mathbf{F}(t)\,, \qquad t \in (0,T)\,, \\ &\mathbf{y}(0) = \mathbf{y}_0\,, \end{aligned} \tag{3.6}$$

where we have introduced

$$\mathbf{y}(t) = \left(\mathbf{z}(t), \frac{d\mathbf{z}}{dt}(t)\right)^T, \qquad \mathbf{B} = \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{A} & -\mathbf{D} \end{pmatrix}, \qquad \mathbf{F}(t) = \begin{pmatrix} \mathbf{0} \\ \widetilde{\mathbf{R}}(t) \end{pmatrix}.$$

3.2 NODAL DISCONTINUOUS GALERKIN METHODS

As an alternative to mass-lumped FEM we consider now the discretization of (3.1) using nodal DG elements [56, 57]. We start by introducing new variables $v := u_t$, $\mathbf{w} := -\nabla u$, and thus rewriting (3.1) as a first-order hyperbolic system:

$$\begin{aligned} v_t + \sigma v + \nabla \cdot (c^2 \mathbf{w}) &= f && \text{in } \Omega \times (0,T)\,, \\ \mathbf{w}_t + \nabla v &= \mathbf{0} && \text{in } \Omega \times (0,T)\,, \\ v(\cdot,t) &= 0 && \text{on } \partial\Omega \times (0,T)\,, \\ v(\cdot,0) = v_0\,,\ \mathbf{w}(\cdot,0) &= -\nabla u_0 && \text{in } \Omega\,, \end{aligned} \tag{3.7}$$

or in more compact notation as

$$\mathbf{q}_t + \mathbf{\Sigma}\,\mathbf{q} + \nabla \cdot \mathcal{F}(\mathbf{q}) = \mathbf{S}\,, \tag{3.8}$$

with

$$\mathbf{q} = \begin{bmatrix} v \\ \mathbf{w} \end{bmatrix}, \quad \mathbf{\Sigma} = \begin{bmatrix} \sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathcal{F}(\mathbf{q}) = \begin{bmatrix} c^2 \mathbf{w}^\top \\ v \mathbf{I}_{d\times d} \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} f \\ \mathbf{0} \end{bmatrix}.$$

Similar to the case of continuous FEM we search for solutions of (3.8) in the weak sense. The weak formulation is however posed on a local instead of a global level. We start again by approximating Ω by nonoverlapping elements.

To take discontinuous solutions into account, it is further necessary to double the inner points.

On each element K we express the local numerical solution as a polynomial of degree N

$$\mathbf{q}_h^K(\mathbf{x},t) = \sum_{i=1}^{N_p} \mathbf{q}_h^K(\mathbf{x}_i^K,t)\ell_i^K(\mathbf{x}), \tag{3.9}$$

where $\mathbf{x}_i^k$ represent the $N_p = N+1$ grid points based within the element K and $\ell_i^K(\mathbf{x})$ is the multidimensional Lagrange polynomial based on these grid points. The grid points are chosen to lead to well-behaved interpolation. They can be interpreted as extensions of the Legendre-Gauss-Lobatto points in 1D [57]. The global solution is given by the direct sum of the K local polynomials

$$\mathbf{q}(\mathbf{x},t) \simeq \mathbf{q}_h(x,t) = \bigoplus \mathbf{q}_h^K(x,t). \tag{3.10}$$

To calculate the local solutions we insert $\mathbf{q}_h^K(\mathbf{x},t)$ into (3.8), multiply with $\ell_j^K(\mathbf{x})$ and integrate over K to get

$$\int_K \left(\frac{\partial}{\partial t}\mathbf{q}_h^K + \mathbf{\Sigma}\mathbf{q}_h^K + \nabla \cdot \mathcal{F}_h^K - \mathbf{S}_h^k \right) \ell_j^K \, d\mathbf{x} = \int_{\partial K} \hat{\mathbf{n}} \cdot \left(\mathcal{F}_h^K - \mathcal{F}^* \right) \ell_j^K \, ds, \tag{3.11}$$

where $\hat{\mathbf{n}}$ is the local outward pointing normal. The here introduced numerical flux $\mathcal{F}^*$ plays a crucial role in the design of a DG method. Without the numerical flux the expression in (3.11) would be fully local and the solution at the element interfaces multiple defined. The choice of numerical flux determines which solution, or which combination of solutions, is chosen. Additionally, the boundary conditions are imposed weakly through the numerical fluxes. This further highlights the importance of the choice. Two popular examples are upwind and central flux. While the energy conserving central flux is the average of the different solutions at the interface, the upwind flux takes information were it comes from. This, however, introduces an artificial dissipation to the numerical solution. It depends on the problem we want to study, which numerical flux is adequate. As an example we comment on the upwind flux for the wave equation in 1D with constant c. With $v = u_t$, $w = -u_x$ (3.8) reduces then to

$$\mathbf{q}_t + \mathbf{\Sigma}\mathbf{q} + \mathcal{A}\mathbf{q}_x = \mathbf{S}, \tag{3.12}$$

where

$$\mathbf{q} = (v, w)^\top, \quad \mathbf{S} = (f, 0)^\top, \quad \mathbf{\Sigma} = \begin{bmatrix} \sigma & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathcal{A} = \begin{bmatrix} 0 & c^2 \\ 1 & 0 \end{bmatrix}.$$

The flux of the first-order hyperbolic system (3.12) is then given by $\mathcal{A}\mathbf{q}$. The eigenvalues of $\mathcal{A}$ are $\pm c$ and represent two waves with speed c travelling in opposite directions. In the following, interior information of an element will be marked with a superscript "−" and "+" is the symbol for exterior information. By applying the Riemann jump conditions [72] we recover the following equations

$$c\left(\mathbf{q}^* - \mathbf{q}^-\right) + (\Pi\mathbf{q})^* - (\Pi\mathbf{q})^- = 0, \tag{3.13}$$

$$-c\left(\mathbf{q}^* - \mathbf{q}^+\right) + (\Pi\mathbf{q})^* - (\Pi\mathbf{q})^+ = 0, \tag{3.14}$$

where $\mathbf{q}^*$ refers to the intermediate state, $(\Pi\mathbf{q})^*$ is the numerical flux along the outward normal $\hat{\mathbf{n}}$ with operator Π defined as

$$\Pi\mathbf{q} = \hat{\mathbf{n}} \cdot (\mathcal{A}\mathbf{q}) = \hat{\mathbf{n}} \cdot \begin{bmatrix} c^2 w \\ v \end{bmatrix}. \tag{3.15}$$

We can derive an equation for the unknown numerical flux $(\Pi\mathbf{q})^*$ by combining (3.13) and (3.14) and thus arriving at

$$(\Pi\mathbf{q})^* = \frac{1}{2}\left(c(\mathbf{q}^- - \mathbf{q}^+) + (\Pi\mathbf{q})^- + (\Pi\mathbf{q})^+\right). \tag{3.16}$$

With (3.15) and (3.16) we finally get expressions involving v^* and w^*

$$(\Pi\mathbf{q})^* = \hat{\mathbf{n}} \cdot \begin{bmatrix} c^2 w^* \\ v^* \end{bmatrix} = \frac{1}{2}\left(c\begin{bmatrix} v^- - v^+ \\ w^- - w^+ \end{bmatrix} + \hat{\mathbf{n}} \cdot \begin{bmatrix} c^2(w^- + w^+) \\ v^- + v^+ \end{bmatrix}\right). \tag{3.17}$$

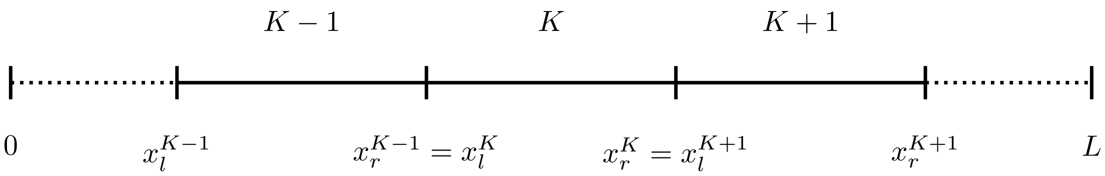


Figure 1: Element K with neighbors.

We are now ready to formulate the equations of the local solutions in matrix-vector form. For simplicity we restrict ourselves again to the one-dimensional case, see Fig. 1 for the grid. (3.9) corresponds then to the two equations

$$v_h^K(x,t)=\sum_{i=1}^{N_p} v_h^K(x_i^K,t)\ell_i^K(x), \quad w_h^K(x,t)=\sum_{i=1}^{N_p} w_h^K(x_i^K,t)\ell_i^K(x),$$

which we can now insert into (3.11). This gives us the following local semi-discrete scheme

$$\mathcal{M}^K\frac{d}{dt}\mathbf{v}_h^K+\sigma\mathcal{M}^K\mathbf{v}_h^K+c^2\mathcal{S}^K\mathbf{w}_h^K-\mathcal{M}^K\mathbf{f}_h^K=c^2\oint_{x_l^K}^{x_r^K}\hat{\mathbf{n}}\cdot(w_h^K-w^*)\ell_j^K\,ds, \tag{3.18}$$

$$\mathcal{M}^K\frac{d}{dt}\mathbf{w}_h^K+\mathcal{S}^K\mathbf{v}_h^K=\oint_{x_l^K}^{x_r^K}\hat{\mathbf{n}}\cdot(v_h^K-v^*)\ell_j^K\,ds, \tag{3.19}$$

where $\mathbf{v}_h^K\in\mathbb{R}^{N_p}$ denotes the vector of the coefficients $v_h^K(x_i^K,t)$, $i=1,\dots,N_p$, and $\mathcal{M}^K$ and $\mathcal{S}^K$ are the local mass and stiffness matrices, respectively. Their entries can be calculated as follows

$$\mathcal{M}_{ij}^K=\int_{x_l^K}^{x_r^K}\ell_j^K(x)\ell_i^K(x)\,dx=\int_{-1}^{1}\ell_j(r)\ell_i(r)\,dr=\frac{h^K}{2}\mathcal{M}_{ij}^I,$$

$$\mathcal{S}_{ij}^K=\int_{x_l^K}^{x_r^K}\ell_i^K(x)\frac{\ell_j^K(x)}{dx}\,dx=\int_{-1}^{1}\ell_i(r)\frac{\ell_j(r)}{dr}\,dr=\mathcal{S}_{ij}^I.$$

with $h_K=x_r^K-x_l^K$ the length of element K. Thus, we only have to calculate the local matrices once on the reference element $I=[-1,1]$.

As the above equations (3.18) and (3.19) are only local, inverting the arising local mass matrix $\mathcal{M}^K$ is cheap. Using further the expression for the numerical flux (3.17), we can represent the global solution with the following global system of ODEs

$$\frac{d}{dt}\mathbf{v}_h(t) = \Big(-\sigma\mathbf{v}_h(t) - c^2\mathcal{M}^{-1}\mathcal{S}\mathbf{w}_h(t) + \mathcal{M}^{-1}\mathcal{C}\mathbf{w}_h(t) + \mathbf{f}_h(t)\Big)\,, \tag{3.20}$$

$$\frac{d}{dt}\mathbf{w}_h(t) = \Big(-\mathcal{M}^{-1}\mathcal{S}\mathbf{v}_h(t) + \mathcal{M}^{-1}\mathcal{C}\mathbf{v}_h(t)\Big)\,, \tag{3.21}$$

with global mass and stiffness matrix $\mathcal{M}$ and $\mathcal{S}$ and interface matrix $\mathcal{C}$.
(3.20) and (3.21) can be directly written as (3.6)

$$\begin{aligned} \frac{d\mathbf{y}}{dt}(t) &= \mathbf{B}\mathbf{y}(t) + \mathbf{F}(t)\,, \qquad t \in (0,T)\,, \\ \mathbf{y}(0) &= \mathbf{y}_0\,, \end{aligned}$$

with

$$\mathbf{y}(t) = \begin{pmatrix} \mathbf{v}_h \\ \mathbf{w}_h \end{pmatrix}, \; \mathbf{B} = \begin{pmatrix} -\sigma\mathbf{I} & -c^2\mathcal{M}^{-1}\mathcal{S} + \mathcal{M}^{-1}\mathcal{C} \\ -\mathcal{M}^{-1}\mathcal{S} + \mathcal{M}^{-1}\mathcal{C} & \mathbf{0} \end{pmatrix}, \; \mathbf{F}(t) = \begin{pmatrix} \mathbf{f}_h \\ \mathbf{0} \end{pmatrix}.$$

Note that we in general never assemble the global matrices as displayed in (3.20) and (3.21), but only compute them on a local level.
In a similar way as presented here for the 1D wave equation we can discretize any first-order hyperbolic system (3.8). We likewise end up with $d+1$ local equations corresponding to (3.18)-(3.19) which can be rewritten in global form as (3.6).

4

MAXWELL'S EQUATIONS

In this chapter we concentrate on Maxwell's equations and their spatial discretization schemes. We restrict ourselves here to nodal DG approach [56, 57]. We start by repeating our model problem as seen in Chapter 1

$$\varepsilon \frac{\partial}{\partial t}\mathbf{E} - \nabla \times \mathbf{H} + \sigma \mathbf{E} = -\mathbf{J}_a, \qquad \text{in } \Omega \times (0,T), \tag{4.1}$$

$$\mu \frac{\partial}{\partial t}\mathbf{H} + \nabla \times \mathbf{E} = 0 \qquad \text{in } \Omega \times (0,T), \tag{4.2}$$

where $\Omega \subset \mathbb{R}^d$, $d = 1,2,3$, is a bounded polygon/polyhedron. Here μ, ε and σ denote the relative magnetic permeability, the relative electric permittivity and the conductivity of the medium, respectively, while the source term j corresponds to the applied current density. $\sigma(\mathbf{x})$ is a non-negative function of position. Regions where σ is positive are called *conductors*. Where $\sigma = 0$ and $\varepsilon \neq \varepsilon_0$, the material is termed a dielectric, and ε is referred to as the dielectric constant. In vacuum (or air at low field strength) $\sigma = 0$, $\varepsilon = \varepsilon_0$ and $\mu = \mu_0$. For simplicity, we assume that materials are piecewise constant. Furthermore, let $\mathbf{E}(\cdot,t) = 0$ on $\partial\Omega \times (0,T)$ corresponding to a physical situation where the computational domain is enclosed in a metallic cavity (also called perfectly electrically conducting (PEC) boundaries). Given initial conditions will complete our model.

4.1 SPATIAL DISCRETIZATION

We extend the nodal DG discretization seen in Chapter 3 for damped wave equations to electromagnetic problems described by Maxwell's equations. Following [56] we rewrite (4.1)-(4.2) as a system of the form

$$\mathbf{Q}(\mathbf{x})\mathbf{q}_t + \mathbf{\Sigma}\,\mathbf{q} + \nabla \cdot \mathcal{F}(\mathbf{q}) = \mathbf{S}\,, \tag{4.3}$$

with

$$\mathbf{Q} = \begin{bmatrix} \varepsilon \mathbf{I}_{d\times d} & \mathbf{0} \\ \mathbf{0} & \mu \mathbf{I}_{d\times d} \end{bmatrix}, \mathbf{q} = \begin{bmatrix} \mathbf{E} \\ \mathbf{H} \end{bmatrix}, \mathbf{\Sigma} = \begin{bmatrix} \sigma \mathbf{I}_{d\times d} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \mathcal{F}_i(\mathbf{q}) = \begin{bmatrix} -\mathbf{e}_i \times \mathbf{H} \\ \mathbf{e}_i \times \mathbf{E} \end{bmatrix}, \mathbf{S} = \begin{bmatrix} -\mathbf{J}_a \\ \mathbf{0} \end{bmatrix}.$$

Here $\mathbf{e}_i$, $i = 1, 2, 3$, denote the three Cartesian unit vectors. (4.3) can be directly written into the form of (3.8) by inverting $\mathbf{Q}$. Thus, the derivation of the local solution (3.11) also holds for Maxwell's equations (4.1) - (4.2). Again the most crucial part in designing the DG method is the choice of numerical flux. As for the wave equation the numerical flux along a normal $\hat{\mathbf{n}}$ can be obtained with the Riemann jump conditions, also known as the Rankine-Hugoniot conditions. First we recall that the normal component of the flux in (4.3) is given by

$$\hat{\mathbf{n}} \cdot \mathcal{F} = \begin{bmatrix} -\hat{\mathbf{n}} \times \mathbf{H} \\ \hat{\mathbf{n}} \times \mathbf{E} \end{bmatrix}$$

and thus represents the tangential field components. This leads to the following equations for the numerical fluxes

$$\hat{\mathbf{n}} \cdot (\mathcal{F} - \mathcal{F}^*) = \begin{cases} -\dfrac{1}{2\{Z\}} \hat{\mathbf{n}} \times \left(Z^+ \left(\mathbf{H}^- - \mathbf{H}^+\right) - \alpha \hat{\mathbf{n}} \times \left(\mathbf{E}^- - \mathbf{E}^+\right)\right) \\ \\ \dfrac{1}{2\{Y\}} \hat{\mathbf{n}} \times \left(Y^+ \left(\mathbf{E}^- - \mathbf{E}^+\right) + \alpha \hat{\mathbf{n}} \times \left(\mathbf{H}^- - \mathbf{H}^+\right)\right) \end{cases} \tag{4.4}$$

for electric and magnetic fields, respectively – see [56] for derivation. Here, $\{\cdot\}$ denotes the average, i.e.

$$\{Z\} = \frac{Z^+ + Z^-}{2}.$$

For simplicity we assume that ε and μ are piecewise constant. Thus,

$$Z^\pm = \frac{1}{Y^\pm} = \sqrt{\frac{\mu^\pm}{\varepsilon^\pm}},$$

gives us the relation between local impedance Z and conductance Y.
The choice of $\alpha = 1$ results in a classical upwind flux. For $\alpha = 0$, (4.4) reduces to the nondissipative central flux. Applying a central flux gives an energy-conserving method for $\sigma = 0$. However, we do not always get optimal convergence rates, but an even-odd pattern where we get accuracy of $\mathcal{O}(h^{N+1})$ for N even and $\mathcal{O}(h^N)$ for N odd, see [21], Chapter 6.5 (presentation of phenomena for Maxwell) and Chapter 7.1 (theoretical results, for Heat equation). Here, N is the degree of the local polynomial on each element. Furthermore, when studying the stability of explicit RK methods, we observe another advantage of an upwind flux as it gives a system matrix with a spectrum better suited for the stability regions of RK schemes than that of system matrices originating from continuous FEM or DG with central flux discretizations, see Section 8.1.
For illustration we present now more detailed calculations for Maxwell's equations in 2 spatial dimensions. If the electric and magnetic fields only depend on

the two spatial components x and y, but not on z, then the system of equations (4.1) and (4.2), containing 6 unknowns, decouples and we get 2 independent sets of equations, one involving H^x, H^y, E^z (known as transverse magnetic form (TM)) and a second one containing E^x, E^y, H^z (transverse electric form (TE)). The two sets can then be solved in parallel. Here we only consider the TM form, which is given by

$$\varepsilon \frac{\partial E^z}{\partial t} - \left(\frac{\partial H^y}{\partial x} - \frac{\partial H^x}{\partial y} \right) + \sigma E^z = -J_a \qquad \text{in } \Omega \times (0,T), \tag{4.5}$$

$$\mu \frac{\partial H^x}{\partial t} + \frac{\partial E^z}{\partial y} = 0 \qquad \text{in } \Omega \times (0,T), \tag{4.6}$$

$$\mu \frac{\partial H^y}{\partial t} - \frac{\partial E^z}{\partial x} = 0 \qquad \text{in } \Omega \times (0,T). \tag{4.7}$$

Following what we have seen in Section 3.2 for the wave equation in 1D we start discretizing (4.5)-(4.7) in space by deriving local weak formulations for each element K. This gives us again a set of local equations

$$\frac{d\mathbf{E}_h^{z,K}}{dt} = \frac{1}{\varepsilon^K} \left(\mathcal{D}_x^K \mathbf{H}_h^{y,K} - \mathcal{D}_y^K \mathbf{H}_h^{x,K} - \sigma^k \mathbf{E}_h^{z,K} - \mathbf{J}_{a,h}^K + \mathcal{M}^{-K} \mathcal{C}^K \mathbf{P}_{E^z} \right) \tag{4.8}$$

$$\frac{d\mathbf{H}_h^{x,K}}{dt} = -\frac{1}{\mu^K} \left(\mathcal{D}_y^K \mathbf{E}_h^{z,K} + \mathcal{M}^{-K} \mathcal{C}^K \mathbf{P}_{H^x} \right) \tag{4.9}$$

$$\frac{d\mathbf{H}_h^{y,K}}{dt} = \frac{1}{\mu^K} \left(\mathcal{D}_x^K \mathbf{E}_h^{z,K} + \mathcal{M}^{-K} \mathcal{C}^K \mathbf{P}_{H^y} \right) \tag{4.10}$$

Here, $\mathcal{M}^K$ is again the local mass-matrix, $\mathcal{C}^K$ is the interface matrix, while $\mathcal{D}_m^K$, $m = x, y, z$, is given by

$$\mathcal{D}_m^K = \mathcal{M}^{-K} \mathcal{S}_m^K, \quad \text{with } \left(\mathcal{S}_m^K \right)_{ij} = \int_K \ell_i^K(\mathbf{x}) \frac{\ell_j^K(\mathbf{x})}{dm} \, d\mathbf{x}$$

The terms $\mathbf{P}_{H^x}$, $\mathbf{P}_{H^y}$ and $\mathbf{P}_{E^z}$ stand for the fluxes and are chosen as

$$\mathbf{P}_{E^z} = \frac{1}{2} \left(\hat{n}_y [H_h^x] - \hat{n}_x [H_h^y] - \alpha [E_h^z] \right)$$

$$\mathbf{P}_{H^x} = \frac{1}{2} \left(\hat{n}_y [E_h^z] + \alpha \left(\hat{n}_x [\![\mathbf{H}^k]\!] - [H_h^x] \right) \right)$$

$$\mathbf{P}_{H^y} = \frac{1}{2} \left(-\hat{n}_x [E_h^z] + \alpha \left(\hat{n}_y [\![\mathbf{H}^k]\!] - [H_h^y] \right) \right)$$

where $[q]$, $[\![q]\!]$ and $[\![\mathbf{q}]\!]$, the jumps along the normal $\hat{\mathbf{n}}$ are defined as

$$[q] = q^- - q^+ = \hat{\mathbf{n}} \cdot [\![q]\!], \quad [\![q]\!] = \hat{\mathbf{n}}^- q^- + \hat{\mathbf{n}}^+ q^+, \quad [\![\mathbf{q}]\!] = \hat{\mathbf{n}}^- \cdot \mathbf{q}^- + \hat{\mathbf{n}}^+ \cdot \mathbf{q}^+.$$

Note the difference between the definitions for scalar functions q and vector valued functions $\mathbf{q}$. For the detailed derivation of the fluxes see [21].

In the same way as in the one-dimensional damped wave equation we can again rewrite the local expressions (4.8)-(4.10) as global equations and end up with a system of the form (3.6):

$$\begin{aligned}\frac{d\mathbf{y}}{dt}(t) &= \mathbf{B}\mathbf{y}(t) + \mathbf{F}(t)\,, \qquad t \in (0,T)\,,\\ \mathbf{y}(0) &= \mathbf{y}_0\,,\end{aligned}$$

with

$$\mathbf{y}(t) = \left(\mathbf{H}_h^x, \mathbf{H}_h^y, \mathbf{E}_h^z\right)^\top, \qquad \mathbf{F}(t) = (-\mathbf{J}_{a,h}(t), 0, 0)^\top$$

and

$$\mathbf{B} = \begin{bmatrix} -\sigma\mathbf{I} & -\frac{1}{\varepsilon}\mathcal{D}_y & \frac{1}{\varepsilon}\mathcal{D}_x \\ -\frac{1}{\mu}\mathcal{D}_y & \mathbf{0} & \mathbf{0} \\ \frac{1}{\mu}\mathcal{D}_x & \mathbf{0} & \mathbf{0} \end{bmatrix} + \mathcal{M}^{-1}\mathcal{C}.$$

Part II

RUNGE-KUTTA BASED EXPLICIT LOCAL TIME-STEPPING METHODS

5

INTRODUCTION

Following the method of lines approach we have, in Part I, discretized our model problems in space using FE methods, both continuous and discontinuous. We have seen that either discretization leads to a system of first order ODEs (3.6) of the form

$$\begin{aligned} \frac{d\mathbf{y}}{dt}(t) &= \mathbf{B}\mathbf{y}(t) + \mathbf{F}(t)\,, \qquad t \in (0,T)\,, \\ \mathbf{y}(0) &= \mathbf{y}_0\,. \end{aligned}$$

Especially the spatial discretizations we consider in this work, i.e. FE based schemes, can easily handle locally refined meshes, which appear when dealing with complex geometries or adaptivity; however at a high price for explicit time integration as the admissible time-step is dictated by the smallest elements in the mesh. Calculating with an implicit scheme or with an explicit scheme with the small time-step dominated by the refined region everywhere, is computationally too expensive if the fine region is small compared to the whole domain.

Here, we propose LTS methods based on explicit RK schemes of arbitrarily high accuracy.

The rest of this part is organized as follows. In Chapter 6, we derive explicit LTS methods of arbitrarily high accuracy by starting from standard or low-storage RK methods. Next, we prove in Chapter 7 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-RK schemes are given in Chapter 8 for both damped wave equation and Maxwell's equations. The work presented here is based on [47].

6

RUNGE-KUTTA BASED LTS

We shall now derive high-order explicit LTS methods for (3.6), 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 [15, 55, 83, 99].

Explicit Runge-Kutta methods, generalizations of the famous Euler method (1768), have been around for many years. Due to their one-step nature and their availability in higher order they are still among the most popular methods for the integration of time dependent problems today.

In the following chapter we introduce RK methods and their notation, derive LTS schemes based on explicit RK methods, both classical and low-storage and present the LTS algorithm based on the popular classical RK4 scheme.

6.1 RUNGE-KUTTA METHODS

Starting from an explicit s-stage RK (RKs) method of order k [55], we now derive an explicit LTS scheme of the same accuracy for (3.6). Let $\mathbf{y}_n$ denote the numerical approximation to the exact solution $\mathbf{y}(t_n)$ at $t_n = n\Delta t$ and $\mathbf{F}_n = \mathbf{F}(t_n)$. A general explicit RK$s$ method applied to (3.6) then yields

$$\begin{aligned}
\mathbf{k}_1 &= \mathbf{B}\mathbf{y}_n + \mathbf{F}(t_n)\,, \\
\mathbf{k}_2 &= \mathbf{B}(\mathbf{y}_n + \Delta t\, a_{21}\mathbf{k}_1) + \mathbf{F}(t_n + c_2\Delta t)\,, \\
&\;\vdots \\
\mathbf{k}_s &= \mathbf{B}\left(\mathbf{y}_n + \Delta t \sum_{i=1}^{s-1} a_{si}\mathbf{k}_i\right) + \mathbf{F}(t_n + c_s\Delta t)\,, \\
\mathbf{y}_{n+1} &= \mathbf{y}_n + \Delta t \sum_{i=1}^{s} b_i\mathbf{k}_i\,.
\end{aligned} \tag{6.1}$$

The constants a_{ij}, b_i and c_i, with $c_1 = 0$, uniquely identify the RK method and are typically listed in a Butcher-tableau, as in Table 1.

Table 1: Butcher-tableau of an explicit RKs scheme.

0					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
$\vdots$	$\vdots$	$\vdots$	$\ddots$		
c_s	a_{s1}	a_{s2}	$\dots$	$a_{s,s-1}$	
	b_1	b_2	$\dots$	b_{s-1}	b_s

If the RKs method has order k, with $s \geq k$, then b_i, c_i, $i = 1, \dots, s$ must satisfy [55]

$$\sum_{i=1}^{s} b_i c_i^{q-1} = \frac{1}{q}, \qquad q = 1, \dots, k\,. \tag{6.2}$$

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

6.2 EXPLICIT RK BASED LTS METHODS

We now wish to derive an RKs based LTS scheme for (3.6), 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 $\mathbf{y}(t)$ and $\mathbf{F}(t)$ as

$$\begin{aligned} \mathbf{y}(t) &= (\mathbf{I} - \mathbf{P})\mathbf{y}(t) + \mathbf{P}\mathbf{y}(t) = \mathbf{y}^{[c]}(t) + \mathbf{y}^{[f]}(t), \\ \mathbf{F}(t) &= (\mathbf{I} - \mathbf{P})\mathbf{F}(t) + \mathbf{P}\mathbf{F}(t) = \mathbf{F}^{[c]}(t) + \mathbf{F}^{[f]}(t), \end{aligned} \tag{6.3}$$

where the matrix $\mathbf{P}$ 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 (3.6) is given by

$$\begin{aligned} \mathbf{y}(t_n + \xi\Delta t) = \mathbf{y}(t_n) &+ \int_{t_n}^{t_n+\xi\Delta t} \mathbf{B}\mathbf{y}^{[c]}(t) + \mathbf{F}^{[c]}(t)\, dt \\ &+ \int_{t_n}^{t_n+\xi\Delta t} \mathbf{B}\mathbf{y}^{[f]}(t) + \mathbf{F}^{[f]}(t)\, dt, \qquad 0 \leq \xi \leq 1\,. \end{aligned} \tag{6.4}$$

Hence, any numerical method for the time integration of (3.6) in fact approximately evaluates the right side of (6.4).

Inside the coarse region, the LTS method we seek ought to coincide with the initial kth-order RKs method. Therefore, we approximate in (6.4) the term involving $\mathbf{y}^{[c]}(t)$ by the corresponding quadrature formula

$$\int_{t_n}^{t_n+\xi\Delta t} \mathbf{B}\mathbf{y}^{[c]}(t)\, dt \simeq \xi\Delta t\mathbf{B}\left(\sum_{i=1}^{s} b_i \mathbf{y}^{[c]}(t_n + c_i\xi\Delta t)\right). \tag{6.5}$$

Next, we approximate the (unknown) values of $\mathbf{y}^{[c]}$ at the quadrature points by Taylor expansion:

$$\int_{t_n}^{t_n+\xi\Delta t} \mathbf{B}(\mathbf{I}-\mathbf{P})\mathbf{y}(t)\,dt \simeq \xi\Delta t\mathbf{B}(\mathbf{I}-\mathbf{P})\left(\sum_{i=1}^{s} b_i \sum_{j=0}^{s-1} \frac{c_i^j(\xi\Delta t)^j}{j!}\mathbf{y}^{(j)}(t_n)\right). \tag{6.6}$$

Repeated use of (3.6) to evaluate the derivatives $\mathbf{y}^{(j)}$ of $\mathbf{y}$ in (6.6) then leads to

$$\begin{aligned}\int_{t_n}^{t_n+\xi\Delta t} \mathbf{B}(\mathbf{I}-\mathbf{P})\mathbf{y}(t)\,dt \simeq \xi\Delta t\mathbf{B}(\mathbf{I}-\mathbf{P})\left(\sum_{i=1}^{s} b_i \sum_{j=0}^{s-1} \frac{c_i^j(\xi\Delta t)^j}{j!}\left(\mathbf{B}^j\mathbf{y}_n\right.\right.\\ \left.\left.+\sum_{\ell=1}^{j}\mathbf{B}^{j-\ell}\mathbf{F}^{(\ell-1)}(t_n)\right)\right).\end{aligned} \tag{6.7}$$

To avoid the derivatives of $\mathbf{F}(t)$ in (6.7), we now interpolate $\mathbf{F}(t)$ by a polynomial $\mathbf{q}(t)$ through the points $(t_n + c_i\Delta t, \mathbf{F}(t_n + c_i\Delta t))$, $i = 1, \ldots, s$. Since the nodes c_i, $i = 1, \ldots, s$ need not be distinct (e.g. $c_2 = c_3 = 1/2$ for RK4, see Table 18 in the Appendix), the degree of $\mathbf{q}$ may be strictly less than $s-1$. We now replace the derivatives of $\mathbf{F}$ in (6.7) by the corresponding derivatives of $\mathbf{q}$ to obtain

$$\begin{aligned}\mathbf{y}(t_n &+ \xi\Delta t) \simeq \mathbf{y}_n\\ &+ \mathbf{B}(\mathbf{I}-\mathbf{P})\left(\sum_{i=1}^{s} b_i \sum_{j=0}^{s-1} \frac{c_i^j(\xi\Delta t)^{j+1}}{j!}\left(\mathbf{B}^j\mathbf{y}_n + \sum_{\ell=1}^{j}\mathbf{B}^{j-\ell}\mathbf{q}^{(\ell-1)}(t_n)\right)\right)\\ &+ (\mathbf{I}-\mathbf{P})\left(\hat{\mathbf{q}}(t_n+\xi\Delta t) - \hat{\mathbf{q}}(t_n)\right) + \int_{t_n}^{t_n+\xi\Delta t}\left(\mathbf{B}\mathbf{P}\mathbf{y}(t) + \mathbf{P}\mathbf{F}(t)\right)\,dt,\end{aligned} \tag{6.8}$$

where $\hat{\mathbf{q}}'(t) = \mathbf{q}(t)$. Since $\mathbf{F}$ is known, so are $\mathbf{q}$ and $\hat{\mathbf{q}}$, and thus all terms needed in (6.8) to advance the solution inside the coarse region, that is those involving $(\mathbf{I}-\mathbf{P})$, are also explicitly known.

As we seek to compute $\mathbf{y}(t_n + \Delta t)$, we need to evaluate the entire right side of (6.8). To circumvent the severe stability restriction on Δt dictated by the smallest elements in the mesh, we shall treat $\mathbf{y}^{[f]}(t)$ and $\mathbf{F}^{[f]}(t)$ differently from $\mathbf{y}^{[c]}(t)$ and $\mathbf{F}^{[c]}(t)$. Hence, we now approximate the remaining integral in (6.8) as

$$\int_{t_n}^{t_n+\xi\Delta t} \mathbf{B}\mathbf{P}\mathbf{y}(t) + \mathbf{P}\mathbf{F}(t)\,dt \simeq \int_{0}^{\xi\Delta t} \mathbf{B}\mathbf{P}\widetilde{\mathbf{y}}(\tau) + \mathbf{P}\mathbf{F}(t_n+\tau)\,d\tau\,,$$

where $\widetilde{\mathbf{y}}(\tau)$ solves the following differential equation for $0 < \tau \le \Delta t$:

$$\begin{aligned}\frac{d\widetilde{\mathbf{y}}}{d\tau}(\tau) &= \mathbf{B}(\mathbf{I}-\mathbf{P})\left[\sum_{i=1}^{s} b_i \sum_{j=0}^{s-1} \frac{(j+1)c_i^j\tau^j}{j!}\left(\mathbf{B}^j\mathbf{y}_n + \sum_{\ell=1}^{j}\mathbf{B}^{j-\ell}\mathbf{q}^{(\ell-1)}(t_n)\right)\right]\\ &\quad + (\mathbf{I}-\mathbf{P})\mathbf{q}(t_n+\tau) + \mathbf{B}\mathbf{P}\,\widetilde{\mathbf{y}}(\tau) + \mathbf{P}\mathbf{F}(t_n+\tau)\,,\\ \widetilde{\mathbf{y}}(0) &= \mathbf{y}_n\,.\end{aligned} \tag{6.9}$$

Its exact solution is given by

$$\begin{aligned} \widetilde{\mathbf{y}}(\xi\Delta t) &= \mathbf{y}_n \\ &+ \mathbf{B}(\mathbf{I}-\mathbf{P})\left(\sum_{i=1}^{s} b_i \sum_{j=0}^{s-1} \frac{c_i^j(\xi\Delta t)^{j+1}}{j!}\left(\mathbf{B}^j\mathbf{y}_n + \sum_{\ell=1}^{j}\mathbf{B}^{j-\ell}\mathbf{q}^{(\ell-1)}(t_n)\right)\right) \\ &+ (\mathbf{I}-\mathbf{P})\left(\hat{\mathbf{q}}(t_n+\xi\Delta t)-\hat{\mathbf{q}}(t_n)\right) + \int_0^{\xi\Delta t} \mathbf{B}\mathbf{P}\widetilde{\mathbf{y}}(\tau) + \mathbf{P}\mathbf{F}(t_n+\tau)\,d\tau\,. \end{aligned} \tag{6.10}$$

Since the right side of (6.10) coincides with that of (6.8) with $\mathbf{Py}(t)$ replaced by $\mathbf{P}\widetilde{\mathbf{y}}(t)$ inside the integral, we immediately infer that $\mathbf{y}(t_n+\xi\Delta t) \simeq \widetilde{\mathbf{y}}(\xi\Delta t)$.
To advance $\mathbf{y}$ from t_n to $t_n+\Delta t$, we shall therefore evaluate $\widetilde{\mathbf{y}}(\Delta t)$ by solving (6.9) 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 kth-order accurate, as we shall show in Chapter 7. In summary, given $\mathbf{y}_n$ the kth-order RKs based LTS algorithm for the solution of (3.6) computes $\mathbf{y}_{n+1} \simeq \mathbf{y}(t_n+\Delta t)$, as follows:

Algorithm 6.1. *LTS-RKs(p)*

1. *Set* $\widetilde{\mathbf{y}}_0 := \mathbf{y}_n$.
2. *For* $j = 0,\ldots,s-1$ *compute*

$$\mathbf{w}_{n,j} := \frac{(j+1)}{j!}\mathbf{B}(\mathbf{I}-\mathbf{P})\sum_{i=1}^{s} b_i c_i^j\left(\mathbf{B}^j\mathbf{y}_n + \sum_{\ell=1}^{j}\mathbf{B}^{j-\ell}\mathbf{q}^{(\ell-1)}(t_n)\right). \tag{6.11}$$

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

$$\begin{aligned} \mathbf{k}_{r,\frac{m+1}{p}} &:= \sum_{j=0}^{s-1}((m+c_r)\Delta\tau)^j\,\mathbf{w}_{n,j} + (\mathbf{I}-\mathbf{P})\mathbf{q}(t_n+(m+c_r)\Delta\tau) \\ &\quad +\mathbf{BP}\left(\widetilde{\mathbf{y}}_{\frac{m}{p}} + \Delta\tau\sum_{i=1}^{r-1} a_{ri}\mathbf{k}_{i,\frac{m+1}{p}}\right) + \mathbf{PF}_{n,m+c_r}\,, \qquad r = 1,\ldots,s\,, \\ \widetilde{\mathbf{y}}_{\frac{m+1}{p}} &:= \widetilde{\mathbf{y}}_{\frac{m}{p}} + \Delta\tau\sum_{i=1}^{s} b_i\mathbf{k}_{i,\frac{m+1}{p}}\,. \end{aligned}$$

4. *Set* $\mathbf{y}_{n+1} := \widetilde{\mathbf{y}}_1$.

Here, we have introduced the notation $\mathbf{F}_{n,m+c_r} = \mathbf{F}(t_n+\tau_{m+c_r})$, where $\tau_{m+c_r} = (m+c_r)\Delta\tau$; note that $\mathbf{F}_{n,0} = \mathbf{F}(t_n+\tau_0) = \mathbf{F}(t_n) = \mathbf{F}_n$.

In Algorithm 6.1, steps 1–3 correspond to the numerical solution of (6.9) until $\tau = \Delta t$ by using the kth-order RKs scheme with local time-step $\Delta\tau = \Delta t/p$. In step 2, we precompute all s multiplications with $\mathbf{B}(\mathbf{I}-\mathbf{P})\mathbf{B}^j$, $j = 1, \ldots, s$. Those higher powers of $\mathbf{B}$ are never explicitly computed but simply successively applied to the vectors $\mathbf{y}_n$ and $\mathbf{q}^{(\ell-1)}(t_n)$; hence, it suffices to implement a single matrix-vector multiplication with nearest-neighbor communications in a parallel environment. The remaining $s \times p$ multiplications with $\mathbf{BP}$ 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 6.1 corresponds to a local time-stepping method. In step 3, the term $(\mathbf{I}-\mathbf{P})\mathbf{q}(t_n + (m + c_r)\Delta\tau)$ appears inside the inner for-loops only for ease of notation. In fact given any specific RK-method, we can explicitly calculate $\mathbf{q}$ and then add at once its j-th order term involving $((m + c_r)\Delta\tau)^j$ to $\mathbf{w}_{n,j}$ in step 2 – e.g. Algorithm 6.5 for LTS-RK4 in Section 6.4.

To investigate the theoretical speed-up of the LTS-RKs method over a standard RKs scheme with a small time-step in the entire computational domain, we consider (3.6) with $\mathbf{F} = 0$. Now let r denote the percentage of "fine" elements, $0 < r < 1$. Thus rN degrees of freedom belong to the locally refined mesh and $(1-r)N$ to the remaining coarse region, $0 < r < 1$. To leading order, the computational cost is determined by matrix-vector products with $\mathbf{B}$ and we denote by qN the cost of multiplying $\mathbf{B}$ with any vector. Hence, the cost to advance the solution from t_n to t_{n+1} by performing p small time-steps of size $\Delta\tau = \Delta t/p$ everywhere with a standard RKs method is

$$C_{\text{RK}} \simeq psqN. \tag{6.12}$$

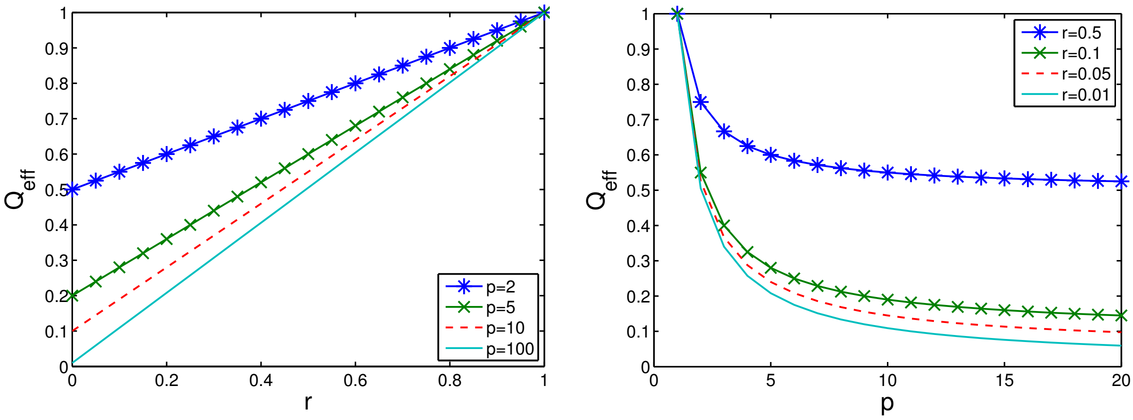
For the LTS-RK$s(p)$ method, Algorithm 6.1 reduces to the underlying standard RKs scheme in the coarse region – see also Remark 4 in Section 4. Again to leading order, the computational cost here is $sq(1-r)N$, since the product of $\mathbf{B}$ with a vector of the form $(\mathbf{I}-\mathbf{P})\mathbf{v}$ only affects entries which are associated with the coarse part of the mesh. In the fine region, we have sp multiplications with $\mathbf{B}$, which only affect unknowns located in the fine part – see step 3 of Algorithm 6.1. Therefore, the total cost of the LTS-RKs scheme is

$$C_{\text{LTS-RK}} \simeq sq(1-r)N + psqrN. \tag{6.13}$$

The ratio of $C_{\text{LTS-RK}}$ over C_{RK} thus yields the theoretical speed-up Q_{eff}:

$$Q_{\text{eff}}(r,p) = \frac{C_{\text{LTS-RK}}}{C_{\text{RK}}} \simeq r + \frac{1-r}{p}. \tag{6.14}$$

In Fig. 2 we illustrate the dependence of Q_{eff} for varying r or p. The smaller r, the larger the gain in using the LTS-RKs method for increasing values of p. In



(a) Q_{eff} for fixed p (b) Q_{eff} for fixed r

Figure 2: The theoretical speed-up Q_{eff} vs. r for $p = 2, 5, 10, 100$ (a) and Q_{eff} vs. p for $r = 0.5, 0.1, 0.05, 0.01$ (b).

addition, for larger values of p, we observe in Fig. 2(b) that Q_{eff} quickly reaches its asymptotic value at r.

Remark 6.2. *Conservation of linear invariants is an important property of all explicit and implicit Runge-Kutta methods (see [54], Chapter IV, Theorem 1.5). To verify the conservation of linear invariants we consider (3.6) with* $\mathbf{F} = 0$ *and let* $\mathbf{d} \in \mathbb{R}^N$ *satisfy* $\mathbf{d}^\top \mathbf{B} = 0$. *Hence,* $\mathbf{d}^\top \mathbf{y}(t)$ *is a conserved quantity. To show that any LTS-RKs scheme also conserves linear invariants, we must show that* $\mathbf{d}^\top \mathbf{y}_{n+1}$ *remains constant. This easily follows by induction, since* $\mathbf{d}^\top \mathbf{k}_{r,\frac{m+1}{p}} = 0$, *for* $1 \le r \le s$, $m = 0, \dots, p-1$ *in step 3 of Algorithm 6.1. Unlike partitioned RK methods, where conservation gives rise to restrictions on the choice of coefficients (see [66]), the LTS-RK methods therefore always conserve linear invariants.*

6.3 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 [15, 99] applied to (3.6) advances the solution from t_n to t_{n+1} as

$$\begin{aligned} \mathbf{z}_0 &= \mathbf{y}_n\,, \\ \mathbf{k}_r &= A_r\,\mathbf{k}_{r-1} + \Delta t\left(\mathbf{B}\,\mathbf{z}_{r-1} + \mathbf{F}(t_n + C_r\Delta t)\right), \quad r = 1, \dots, s\,, \\ \mathbf{z}_r &= \mathbf{z}_{r-1} + B_r \mathbf{k}_r\,, \quad r = 1, \dots, s\,, \\ \mathbf{y}_{n+1} &= \mathbf{z}_s\,, \end{aligned} \tag{6.15}$$

with $A_1 = 0$ for a self-starting algorithm. Note that $\mathbf{k}_r$ and $\mathbf{z}_r$ only depend on quantities from the previous stage $r-1$. Every LSRK scheme can be put into the standard RK form (6.1) through variable substitutions

$$\begin{aligned}
&A_r = (b_{r-1} - B_{r-1})/b_r\,, \qquad r = 2,\dots,s, \qquad b_r \neq 0\,,\\
&A_r = (a_{r+1,r-1} - c_r)/B_r\,, \qquad r = 2,\dots,s, \qquad b_r = 0\,,\\
&B_r = a_{r+1,r}\,, \qquad r = 1,\dots,s-1\,,\\
&B_s = b_s\,,\\
&C_r = c_r\,, \qquad r = 1,\dots,s\,,
\end{aligned} \tag{6.16}$$

but not vice-versa – see [15, 99] for various sets of values for A_r, B_r and C_r with $s \leq 5$.
Starting from the LSRKs scheme (6.15), we thus first calculate the corresponding standard RK constants a_{ij}, b_i and c_i from (6.16). Following the derivation in Section 6.2, we then evaluate $\widetilde{\mathbf{y}}(\Delta t)$ by solving (6.9) until $\tau = \Delta t$ to advance $\mathbf{y}$ 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 $\mathbf{y}_n$ the kth-order LSRKs based LTS algorithm for the solution of (3.6) computes $\mathbf{y}_{n+1} \simeq \mathbf{y}(t_n + \Delta t)$, as follows:

Algorithm 6.3. *LTS-LSRKs(p)*

1. *Set* $\widetilde{\mathbf{y}}_0 := \mathbf{y}_n$.

2. *For* $j = 0,\dots,s-1$ *compute* $\mathbf{w}_{n,j}$ *as in (6.11).*

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

   $\mathbf{z}_{0,\frac{m+1}{p}} := \widetilde{\mathbf{y}}_{\frac{m}{p}}\,;$

   for $r = 1,\dots,s$ *compute*

   $$\begin{aligned}
   \mathbf{k}_{r,\frac{m+1}{p}} := {} & A_r \mathbf{k}_{r-1,\frac{m+1}{p}}\\
   & + \Delta\tau \Bigg[\sum_{j=0}^{s-1} \left((m+C_r)\Delta\tau\right)^j \mathbf{w}_{n,j} + (\mathbf{I}-\mathbf{P})\mathbf{q}(t_n + (m+C_r)\Delta\tau)\\
   & \qquad + \mathbf{B}\mathbf{P}\mathbf{z}_{r-1,\frac{m+1}{p}} + \mathbf{P}\mathbf{F}_{n,m+C_r} \Bigg]\,,
   \end{aligned}$$

   $$\mathbf{z}_{r,\frac{m+1}{p}} := \mathbf{z}_{r-1,\frac{m+1}{p}} + B_r \mathbf{k}_{r,\frac{m+1}{p}}\,;$$

   $\widetilde{\mathbf{y}}_{\frac{m+1}{p}} := \mathbf{z}_{s,\frac{m+1}{p}}\,.$

4. *Set* $\mathbf{y}_{n+1} := \widetilde{\mathbf{y}}_1$.

Here, Steps 1–3 correspond to the numerical solution of (6.9) 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 (6.11) reduces to

$$\mathbf{w}_{n,j} := \frac{1}{j!}\mathbf{B}(\mathbf{I} - \mathbf{P})\left(\mathbf{B}^j\mathbf{y}_n + \sum_{\ell=1}^{j}\mathbf{B}^{j-\ell}\mathbf{q}^{(\ell-1)}(t_n)\right)$$

because of (6.2); then, the constants b_r and c_r are not needed and the LTS-LSRK$s(p)$ algorithm only uses the coefficients A_r, B_r and C_r of the underlying LSRKs scheme. In general, however, Step 2 of Algorithm 6.3 also requires the values for b_r and c_r of the corresponding standard RKs scheme, given by (6.16).

Remark 6.4. *If the fraction of nonzero entries in* $\mathbf{P}$ *is small, the overall cost of the LTS-RKs(p) and LTS-LSRKs(p) algorithms is dominated by the computation of the s vectors* $\mathbf{w}_{n,j}$, $j = 0, \ldots, s-1$, *in Step 2 which requires s multiplications by* $\mathbf{B}(\mathbf{I} - \mathbf{P})$ *per time-step* Δt. *All further* $s \times p$ *matrix-vector multiplications by* $\mathbf{BP}$ *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.*

6.4 THE RK4 BASED LTS METHOD

In Section 6.2, we have shown how to derive an LTS-RK$s(p)$ 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 18 in the Appendix.

Following the previous derivation, we first split the vectors $\mathbf{y}(t)$ and $\mathbf{F}(t)$ as in (6.3) and approximate the term in (6.4) involving $\mathbf{y}^{[c]}(t)$ by Simpson quadrature. Next, we approximate in (6.5) the values of $\mathbf{y}_{n+\frac{\xi}{2}}$ and $\mathbf{y}_{n+\xi}$, still unknown at time $t = t_n$, by their Taylor expansions up to $\mathcal{O}(\Delta t^4)$. We also interpolate the points $(t_n, \mathbf{F}_n)$, $(t_{n+\frac{1}{2}}, \mathbf{F}_{n+\frac{1}{2}})$ and $(t_{n+1}, \mathbf{F}_{n+1})$ by the quadratic polynomial,

$$\mathbf{q}(t_n + \tau) = \mathbf{F}_n + \frac{\tau}{\Delta t}\left(-3\mathbf{F}_n + 4\mathbf{F}_{n+\frac{1}{2}} - \mathbf{F}_{n+1}\right) + \frac{\tau^2}{2\Delta t^2}\left(4\mathbf{F}_n - 8\mathbf{F}_{n+\frac{1}{2}} + 4\mathbf{F}_{n+1}\right).$$

Thus, we can integrate exactly the term in (6.4) involving $\mathbf{F}^{[c]}(t)$ and also explicitly evaluate the derivatives of $\mathbf{q}$ to approximate $\mathbf{F}^{'}(t_n)$ and $\mathbf{F}^{''}(t_n)$. The differential equation (6.9) for $\widetilde{\mathbf{y}}(\tau)$ is now given by

$$\begin{aligned}
\frac{d\widetilde{\mathbf{y}}}{d\tau}(\tau) &= \mathbf{B}(\mathbf{I}-\mathbf{P})\Bigg[\mathbf{y}_n + \tau\left(\mathbf{B}\mathbf{y}_n + \mathbf{F}_n\right) \\
&+ \frac{\tau^2}{2}\left(\mathbf{B}^2\mathbf{y}_n + \mathbf{B}\mathbf{F}_n + \frac{-3\mathbf{F}_n + 4\mathbf{F}_{n+\frac{1}{2}} - \mathbf{F}_{n+1}}{\Delta t}\right) \\
&+ \frac{\tau^3}{6}\left(\mathbf{B}^3\mathbf{y}_n + \mathbf{B}^2\mathbf{F}_n + \mathbf{B}\,\frac{-3\mathbf{F}_n + 4\mathbf{F}_{n+\frac{1}{2}} - \mathbf{F}_{n+1}}{\Delta t} + \frac{4\mathbf{F}_n - 8\mathbf{F}_{n+\frac{1}{2}} + 4\mathbf{F}_{n+1}}{\Delta t^2}\right)\Bigg] \\
&+ (\mathbf{I}-\mathbf{P})\left[\mathbf{F}_n + \tau\,\frac{-3\mathbf{F}_n + 4\mathbf{F}_{n+\frac{1}{2}} - \mathbf{F}_{n+1}}{\Delta t} + \frac{\tau^2}{2}\frac{4\mathbf{F}_n - 8\mathbf{F}_{n+\frac{1}{2}} + 4\mathbf{F}_{n+1}}{\Delta t^2}\right] \\
&+ \mathbf{B}\mathbf{P}\,\widetilde{\mathbf{y}}(\tau) + \mathbf{P}\mathbf{F}(t_n+\tau)\,, \\
\widetilde{\mathbf{y}}(0) &= \mathbf{y}_n\,.
\end{aligned} \tag{6.17}$$

Again to advance $\mathbf{y}$ from t_n to $t_n + \Delta t$, we shall solve (6.17) by using the RK4 scheme with a smaller time-step $\Delta\tau = \Delta t/p$. In summary, given $\mathbf{y}_n$, the LTS algorithm based on the classical explicit RK4 method for the solution of (3.6) computes $\mathbf{y}_{n+1} \simeq \mathbf{y}(t_n + \Delta t)$ as follows:

Algorithm 6.5. *LTS-RK4(p)*

1. *Set* $\widetilde{\mathbf{y}}_0 := \mathbf{y}_n$.

2. *Compute*

$$\begin{aligned}
\mathbf{w}_{n,0} &:= \mathbf{B}(\mathbf{I}-\mathbf{P})\mathbf{y}_n + (\mathbf{I}-\mathbf{P})\mathbf{F}_n\,, \\
\mathbf{w}_{n,1} &:= \mathbf{B}(\mathbf{I}-\mathbf{P})(\mathbf{B}\mathbf{y}_n + \mathbf{F}_n) + (\mathbf{I}-\mathbf{P})\,\frac{-3\mathbf{F}_n + 4\mathbf{F}_{n+\frac{1}{2}} - \mathbf{F}_{n+1}}{\Delta t}\,, \\
\mathbf{w}_{n,2} &:= \mathbf{B}(\mathbf{I}-\mathbf{P})\left(\mathbf{B}^2\mathbf{y}_n + \mathbf{B}\,\mathbf{F}_n + \frac{-3\mathbf{F}_n + 4\mathbf{F}_{n+\frac{1}{2}} - \mathbf{F}_{n+1}}{\Delta t}\right) \\
&\quad + (\mathbf{I}-\mathbf{P})\frac{4\mathbf{F}_n - 8\mathbf{F}_{n+\frac{1}{2}} + 4\mathbf{F}_{n+1}}{\Delta t^2}\,, \\
\mathbf{w}_{n,3} &:= \mathbf{B}(\mathbf{I}-\mathbf{P})\left(\mathbf{B}^3\mathbf{y}_n + \mathbf{B}^2\,\mathbf{F}_n + \mathbf{B}\,\frac{-3\mathbf{F}_n + 4\mathbf{F}_{n+\frac{1}{2}} - \mathbf{F}_{n+1}}{\Delta t}\right. \\
&\qquad\qquad \left. + \frac{4\mathbf{F}_n - 8\mathbf{F}_{n+\frac{1}{2}} + 4\mathbf{F}_{n+1}}{\Delta t^2}\right)\,.
\end{aligned}$$

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

$$
\begin{aligned}
\mathbf{k}_{1,\frac{m+1}{p}} &:= \mathbf{w}_{n,0} + m\,\Delta\tau\,\mathbf{w}_{n,1} + \frac{m^2}{2}\,\Delta\tau^2\,\mathbf{w}_{n,2} + \frac{m^3}{6}\,\Delta\tau^3\,\mathbf{w}_{n,3} \\
&+ \mathbf{BP}\,\widetilde{\mathbf{y}}_{\frac{m}{p}} + \mathbf{PF}_{n,m}\,, \\
\mathbf{k}_{2,\frac{m+1}{p}} &:= \mathbf{w}_{n,0} + \left(m+\frac{1}{2}\right)\,\Delta\tau\,\mathbf{w}_{n,1} + \frac{1}{2}\left(m+\frac{1}{2}\right)^2\,\Delta\tau^2\,\mathbf{w}_{n,2} \\
&+ \frac{1}{6}\left(m+\frac{1}{2}\right)^3\,\Delta\tau^3\,\mathbf{w}_{n,3} + \mathbf{BP}\,\left(\widetilde{\mathbf{y}}_{\frac{m}{p}} + \frac{\Delta\tau}{2}\,\mathbf{k}_{1,\frac{m+1}{p}}\right) + \mathbf{PF}_{n,m+\frac{1}{2}}\,, \\
\mathbf{k}_{3,\frac{m+1}{p}} &:= \mathbf{w}_{n,0} + \left(m+\frac{1}{2}\right)\,\Delta\tau\,\mathbf{w}_{n,1} + \frac{1}{2}\left(m+\frac{1}{2}\right)^2\,\Delta\tau^2\,\mathbf{w}_{n,2} \\
&+ \frac{1}{6}\left(m+\frac{1}{2}\right)^3\,\Delta\tau^3\,\mathbf{w}_{n,3} + \mathbf{BP}\,\left(\widetilde{\mathbf{y}}_{\frac{m}{p}} + \frac{\Delta\tau}{2}\,\mathbf{k}_{2,\frac{m+1}{p}}\right) + \mathbf{PF}_{n,m+\frac{1}{2}}\,, \\
\mathbf{k}_{4,\frac{m+1}{p}} &:= \mathbf{w}_{n,0} + (m+1)\,\Delta\tau\,\mathbf{w}_{n,1} + \frac{1}{2}\,(m+1)^2\,\Delta\tau^2\,\mathbf{w}_{n,2} \\
&+ \frac{1}{6}\,(m+1)^3\,\Delta\tau^3\,\mathbf{w}_{n,3} + \mathbf{BP}\,\left(\widetilde{\mathbf{y}}_{\frac{m}{p}} + \Delta\tau\,\mathbf{k}_{3,\frac{m+1}{p}}\right) + \mathbf{PF}_{n,m+1}\,, \\
\widetilde{\mathbf{y}}_{\frac{m+1}{p}} &:= \widetilde{\mathbf{y}}_{\frac{m}{p}} + \frac{1}{6}\,\Delta\tau\,\left(\mathbf{k}_{1,\frac{m+1}{p}} + 2\mathbf{k}_{2,\frac{m+1}{p}} + 2\mathbf{k}_{3,\frac{m+1}{p}} + \mathbf{k}_{4,\frac{m+1}{p}}\right)\,.
\end{aligned}
$$

4. *Set* $\mathbf{y}_{n+1} := \widetilde{\mathbf{y}}_1$.

Steps 1–3 compute the numerical solution of (6.17) at time $\tau = \Delta t$ by using the classical RK4 scheme with local time-step $\Delta\tau = \Delta t/p$. In contrast to Algorithm 6.1 for a general RK*s* method, we have precomputed the $(\mathbf{I}-\mathbf{P})\mathbf{q}$ values at intermediate times and already added them to $\mathbf{w}_{n,j}$.

7

ACCURACY AND CONVERGENCE

Starting from an arbitrary explicit RK scheme of order k, we have shown in Chapter 6 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 (6.1) of order k and denote by $\{\widetilde{c}_1, \ldots, \widetilde{c}_{s_0}\} \subset \{c_1, \ldots, c_s\}$ the maximal subset of all coefficients c_i such that no two are identical, i.e. $\widetilde{c}_i \neq \widetilde{c}_j$ 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 $\widetilde{c}_1 = 0$, $\widetilde{c}_2 = 1/2$ and $\widetilde{c}_3 = 1$; hence, $s_0 = 3$, $k = 4$ and the condition is indeed satisfied. We now prove the following technical lemma.

Lemma 7.1. *Let* $\mathbf{y}$ *be the solution of (3.6),* $\widetilde{\mathbf{y}}$ *the solution of (6.9),* $\mathbf{q}(\eta)$ *the interpolation polynomial through the points* $(t_n + \widetilde{c}_i \Delta t, \mathbf{F}(t_n + \widetilde{c}_i \Delta t))$, $i = 1, \ldots, s_0$ *and* $\mathbf{F}(t)$ *satisfy* $\mathbf{F} \in C^{s_0}([0, T])$. *Then,*

$$\widetilde{\mathbf{y}}'(0) = \mathbf{y}'(t_n), \quad \|\widetilde{\mathbf{y}}^{(\nu)}(0) - \mathbf{y}^{(\nu)}(t_n)\|_\infty \leq \mathcal{O}(\Delta t^{s_0 - \nu + 1}), \quad \nu = 2, \ldots, k-1. \quad (7.1)$$

Proof. Since $\widetilde{\mathbf{y}}(0) = \mathbf{y}(t_n)$, $\mathbf{q}(t_n) = \mathbf{F}(t_n)$ and $\sum_{i=1}^{s} b_i = 1$, we immediately find from (6.9) with $\tau = 0$ that

$$\begin{aligned} \widetilde{\mathbf{y}}'(0) &= \mathbf{B}(\mathbf{I} - \mathbf{P}) \sum_{i=1}^{s} b_i \mathbf{y}(t_n) + (\mathbf{I} - \mathbf{P})\mathbf{q}(t_n) + \mathbf{B}\mathbf{P}\widetilde{\mathbf{y}}(0) + \mathbf{P}\mathbf{F}(t_n) \\ &= \mathbf{B}\mathbf{y}(t_n) + \mathbf{F}(t_n) = \mathbf{y}'(t_n), \end{aligned}$$

which yields the equality in (7.1).

To prove the inequality for $\nu = 2, \ldots, k-1$ in (7.1), we first note that straightforward differentiation of (3.6) yields the identity

$$\mathbf{y}^{(\nu)}(t) = \mathbf{B}^\nu \mathbf{y}(t) + \sum_{m=1}^{\nu} \mathbf{B}^{\nu - m} \mathbf{F}^{(m-1)}(t), \qquad \nu \geq 1. \quad (7.2)$$

Next, we derive an upper bound on the error in approximating $\mathbf{F}^{(j)}$, the j-th derivative of $\mathbf{F}$, by the corresponding derivative of $\mathbf{q}$. From [64], we recall that

$$\|\mathbf{F}^{(j)} - \mathbf{q}^{(j)}\|_\infty \leq \|\omega^{(j)}\|_\infty \frac{\|\mathbf{F}^{(s_0)}\|_\infty}{j!\,(s_0 - j)!}, \qquad 1 \leq j \leq s_0 - 1, \tag{7.3}$$

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

$$\omega^{(j)}(\eta) = \sum_{m_1=1}^{s_0} \sum_{\substack{m_2=1\\ m_2\neq m_1}}^{s_0} \cdots \sum_{\substack{m_j=1\\ m_j\neq m_1,\dots,m_{j-1}}}^{s_0} \prod_{\substack{i=1\\ i\neq m_1,\dots,m_j}}^{s_0} (\eta - t_n - \widetilde{c}_i \Delta t)$$

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

$$\|\mathbf{F}^{(j)}(t_n) - \mathbf{q}^{(j)}(t_n)\|_\infty \leq \mathcal{O}(\Delta t^{s_0 - j}), \qquad 1 \leq j \leq s_0 - 1. \tag{7.4}$$

We shall now prove the inequality in (7.1) by induction over ν. First, let $\nu = 2$. We differentiate (6.9) and set $\tau = 0$ in the resulting expression, which yields

$$\widetilde{\mathbf{y}}''(0) = \mathbf{B}(\mathbf{I} - \mathbf{P}) \left[\sum_{i=1}^{s} 2 b_i c_i \left(\mathbf{B}\mathbf{y}_n + \mathbf{q}(t_n)\right)\right] + (\mathbf{I} - \mathbf{P})\mathbf{q}'(t_n) + \mathbf{B}\mathbf{P}\widetilde{\mathbf{y}}'(0) + \mathbf{P}\mathbf{F}'(t_n).$$

Then, we use (6.2) with $q = 2$, the fact that $\mathbf{q}(t_n) = \mathbf{F}(t_n)$, and the equality in (7.1) to obtain

$$\widetilde{\mathbf{y}}''(0) = \mathbf{B}(\mathbf{I} - \mathbf{P}) \left(\mathbf{B}\mathbf{y}(t_n) + \mathbf{F}(t_n)\right) + (\mathbf{I} - \mathbf{P})\mathbf{q}'(t_n) + \mathbf{B}\mathbf{P}\mathbf{y}'(t_n) + \mathbf{P}\mathbf{F}'(t_n). \tag{7.5}$$

By using (3.6) and reordering terms, we rewrite (7.5) as

$$\widetilde{\mathbf{y}}''(0) = \mathbf{B}\mathbf{y}'(t_n) + (\mathbf{I} - \mathbf{P})\mathbf{q}'(t_n) + \mathbf{P}\mathbf{F}'(t_n),$$

or equivalently as

$$\widetilde{\mathbf{y}}''(0) = \mathbf{y}''(t_n) + \mathbf{q}'(t_n) - \mathbf{F}'(t_n) + \mathbf{P}\left(\mathbf{F}'(t_n) - \mathbf{q}'(t_n)\right),$$

where we have used (3.6) differentiated once. Finally, we apply (7.4) with $j = 1$ and use $\|\mathbf{P}\|_\infty \leq 1$ to obtain

$$\|\widetilde{\mathbf{y}}''(0) - \mathbf{y}''(t_n)\|_\infty \leq (1 + \|\mathbf{P}\|_\infty)\|\mathbf{F}'(t_n) - \mathbf{q}'(t_n)\|_\infty \leq \mathcal{O}(\Delta t^{s_0 - 1}), \tag{7.6}$$

which yields (7.1) with $\nu = 2$.

Next, we proceed with the induction step and assume that (7.1) holds for $\nu - 1$. We differentiate (6.9) $\nu - 1$ times and set $\tau = 0$ in the resulting expression to obtain

$$\begin{aligned}\widetilde{\mathbf{y}}^{(\nu)}(0) = \mathbf{B}\left(\mathbf{I} - \mathbf{P}\right) &\left[\frac{\nu!}{(\nu-1)!} \sum_{i=1}^{s} b_i c_i^{\nu-1} \left(\mathbf{B}^{\nu-1}\mathbf{y}(t_n) + \sum_{\ell=1}^{\nu-1} \mathbf{B}^{\nu-\ell-1}\mathbf{q}^{(\ell-1)}(t_n)\right)\right] \\ &+ \left(\mathbf{I} - \mathbf{P}\right)\mathbf{q}^{(\nu-1)}(t_n) + \mathbf{B}\mathbf{P}\widetilde{\mathbf{y}}^{(\nu-1)}(0) + \mathbf{P}\mathbf{F}^{(\nu-1)}(t_n).\end{aligned} \tag{7.7}$$

By using (6.2) and reordering terms, we rewrite (7.7) as

$$\widetilde{\mathbf{y}}^{(\nu)}(0) = \mathbf{B}^{\nu}\mathbf{y}(t_n) + \sum_{\ell=1}^{\nu-1} \mathbf{B}^{\nu-\ell}\mathbf{q}^{(\ell-1)}(t_n) + \mathbf{q}^{(\nu-1)}(t_n) - \mathbf{BP}\Bigg(\mathbf{B}^{\nu-1}\mathbf{y}(t_n) + \sum_{\ell=1}^{\nu-1} \mathbf{B}^{\nu-\ell-1}\mathbf{q}^{(\ell-1)}(t_n)\Bigg) + \mathbf{BP}\widetilde{\mathbf{y}}^{(\nu-1)}(0) + \mathbf{P}\left(\mathbf{F}^{(\nu-1)}(t_n) - \mathbf{q}^{(\nu-1)}(t_n)\right). \tag{7.8}$$

We now use (7.2) to replace the first term on the right of (7.8), which yields

$$\begin{aligned}\widetilde{\mathbf{y}}^{(\nu)}(0) = {} & \mathbf{y}^{(\nu)}(t_n) + \sum_{\ell=1}^{\nu} \mathbf{B}^{\nu-\ell}\left(\mathbf{q}^{(\ell-1)}(t_n) - \mathbf{F}^{(\ell-1)}(t_n)\right) + \mathbf{BP}\widetilde{\mathbf{y}}^{(\nu-1)}(0) \\ & - \mathbf{BP}\Bigg[\mathbf{B}^{\nu-1}\mathbf{y}(t_n) + \sum_{\ell=1}^{\nu-1} \mathbf{B}^{\nu-\ell-1}\mathbf{F}^{(\ell-1)}(t_n) + \sum_{\ell=1}^{\nu-1} \mathbf{B}^{\nu-\ell-1}\Big(\mathbf{q}^{(\ell-1)}(t_n) \\ & - \mathbf{F}^{(\ell-1)}(t_n)\Big)\Bigg] + \mathbf{P}\left(\mathbf{F}^{(\nu-1)}(t_n) - \mathbf{q}^{(\nu-1)}(t_n)\right).\end{aligned} \tag{7.9}$$

Again we use (7.2) to replace the first two terms in square brackets in (7.9) by $\mathbf{y}^{(\nu-1)}$. Then we repeatedly apply the triangle inequality and use that $\|\mathbf{P}\|_\infty \leq 1$ to derive the upper bound

$$\begin{aligned}\|\widetilde{\mathbf{y}}^{(\nu)}(0) - \mathbf{y}^{(\nu)}(t_n)\|_\infty \leq {} & \sum_{\ell=1}^{\nu} \|\mathbf{B}^{\nu-\ell}\|_\infty \|\mathbf{q}^{(\ell-1)}(t_n) - \mathbf{F}^{(\ell-1)}(t_n)\|_\infty \\ & + \|\mathbf{B}\|_\infty \|\widetilde{\mathbf{y}}^{(\nu-1)}(0) - \mathbf{y}^{(\nu-1)}(t_n)\|_\infty + \|\mathbf{F}^{(\nu-1)}(t_n) - \mathbf{q}^{(\nu-1)}(t_n)\|_\infty \\ & + \|\mathbf{B}\|_\infty \sum_{\ell=1}^{\nu-1} \|\mathbf{B}^{\nu-\ell-1}\|_\infty \|\mathbf{q}^{(\ell-1)}(t_n) - \mathbf{F}^{(\ell-1)}(t_n)\|_\infty.\end{aligned} \tag{7.10}$$

Finally we use the induction hypothesis to estimate the second and (7.4) to estimate the remaining terms on the right side of (7.10), which yields (7.1). ☐

We are now ready to establish the accuracy of the LTS-RK*s* methods from Chapter 6.

Proposition 7.2. *Let* $\mathbf{y} \in C^{k+1}(0,T)$ *be the solution of (3.6) and* $\mathbf{y}_{n+1}$ *defined by Algorithm 1 with* $\mathbf{y}_n = \mathbf{y}(t_n)$*. Then* $\|\mathbf{y}(t_{n+1}) - \mathbf{y}_{n+1}\|_\infty = \mathcal{O}(\Delta t^{k+1})$*, i.e. the LTS-RKs(p) method is kth-order accurate.*

Proof. Let $\widetilde{\mathbf{y}}(\tau)$ be the solution of (6.9) for $0 \leq \tau \leq \Delta t$. We split the local truncation error as

$$\|\mathbf{y}(t_{n+1}) - \mathbf{y}_{n+1}\|_\infty \leq \|\mathbf{y}(t_{n+1}) - \widetilde{\mathbf{y}}(\Delta t)\|_\infty + \|\widetilde{\mathbf{y}}(\Delta t) - \mathbf{y}_{n+1}\|_\infty.$$

Since $\mathbf{y}_{n+1}$ corresponds to the numerical solution of (6.9) with the initial k-th order RKs method (6.1) with time-step $\Delta\tau = \Delta t/p$, we immediately have

$$\|\widetilde{\mathbf{y}}(\Delta t) - \mathbf{y}_{n+1}\|_\infty = \mathcal{O}(\Delta t^{k+1}).$$

To complete the proof we thus need to show that $\|\mathbf{y}(t_{n+1}) - \widetilde{\mathbf{y}}(\Delta t)\|_\infty = \mathcal{O}(\Delta t^{k+1})$. First, we integrate (6.9) from 0 until Δt to obtain

$$\begin{aligned} \widetilde{\mathbf{y}}(\Delta t) \;&=\; \mathbf{y}(t_n) + \mathbf{B}(\mathbf{I}-\mathbf{P})\left[\sum_{j=0}^{s-1}\frac{1}{j!}\sum_{i=1}^{s} b_i c_i^j \Delta t^{j+1}\left(\mathbf{B}^j\mathbf{y}(t_n) + \sum_{\ell=1}^{j}\mathbf{B}^{j-\ell}\mathbf{q}^{(\ell-1)}(t_n)\right)\right] \\ &\quad + \int_0^{\Delta t}(\mathbf{I}-\mathbf{P})\mathbf{q}(t_n+\tau)\,d\tau + \int_0^{\Delta t}\mathbf{BP}\widetilde{\mathbf{y}}(\tau)\,d\tau + \int_0^{\Delta t}\mathbf{PF}(t_n+\tau)\,d\tau. \end{aligned} \tag{7.11}$$

We now approximate the first and last integral on the right of (7.11) using the k-th order quadrature formula defined by the weights b_i and the nodes c_i. Since $\mathbf{q}$ interpolates $\mathbf{F}$ at the nodes $t_n + c_i\Delta t$, we conclude that

$$\int_0^{\Delta t}(\mathbf{I}-\mathbf{P})\mathbf{q}(t_n+\tau)\,d\tau + \int_0^{\Delta t}\mathbf{PF}(t_n+\tau)\,d\tau = \int_{t_n}^{t_{n+1}}\mathbf{F}(t)\,dt + \mathcal{O}(\Delta t^{k+1}). \tag{7.12}$$

Next, we truncate in (7.11) the sum in square brackets at $j = k-1$ and use (6.2) to replace $\sum_{i=1}^{s} b_i c_i^j$ by $1/(j+1)$. By using (7.2) in the resulting expression, we thus rewrite the second term on the right of (7.11) as

$$\begin{aligned} &\mathbf{B}(\mathbf{I}-\mathbf{P})\left[\sum_{j=0}^{s-1}\frac{1}{j!}\sum_{i=1}^{s} b_i c_i^j \Delta t^{j+1}\left(\mathbf{B}^j\mathbf{y}(t_n) + \sum_{\ell=1}^{j}\mathbf{B}^{j-\ell}\mathbf{q}^{(\ell-1)}(t_n)\right)\right] \\ &\quad = \mathbf{B}(\mathbf{I}-\mathbf{P})\left[\sum_{j=0}^{k-1}\frac{\Delta t^{j+1}}{(j+1)!}\mathbf{y}^{(j)}(t_n)\right] + \mathbf{B}(\mathbf{I}-\mathbf{P})\left[\sum_{j=0}^{k-1}\frac{\Delta t^{j+1}}{(j+1)!}\sum_{\ell=1}^{j}\mathbf{B}^{j-\ell}\left(\mathbf{q}^{(\ell-1)}(t_n)\right.\right. \\ &\quad \left.\left. - \mathbf{F}^{(\ell-1)}(t_n)\right)\right] + \mathcal{O}(\Delta t^{k+1}). \end{aligned} \tag{7.13}$$

On the right of (7.13), we identify the first term as the Taylor expansion of $\mathbf{By}(t)$, integrated up to Δt, and hence rewrite it as

$$\int_{t_n}^{t_{n+1}}\mathbf{By}(t)\,dt - \mathbf{BP}\sum_{j=0}^{k-1}\frac{\Delta t^{j+1}}{(j+1)!}\mathbf{y}^{(j)}(t_n) + \mathcal{O}(\Delta t^{k+1}).$$

Similarly, we approximate $\widetilde{\mathbf{y}}(\tau)$ in the remaining integral of (7.11) by Taylor expansion up to order $k-1$. By combining the above approximations of the various terms on the right of (7.11), we conclude that

$$\begin{aligned}\widetilde{\mathbf{y}}(\Delta t) &= \mathbf{y}(t_n) + \int_{t_n}^{t_{n+1}} \mathbf{B}\mathbf{y}(t)\,dt + \int_{t_n}^{t_{n+1}} \mathbf{F}(t)\,dt - \mathbf{BP}\sum_{j=0}^{k-1} \frac{\Delta t^{j+1}}{(j+1)!}\mathbf{y}^{(j)}(t_n) \\ &\quad + \mathbf{BP}\sum_{j=0}^{k-1} \frac{\Delta t^{j+1}}{(j+1)!}\widetilde{\mathbf{y}}^{(j)}(0) + \mathbf{B}(\mathbf{I}-\mathbf{P})\left[\sum_{j=0}^{k-1} \frac{\Delta t^{j+1}}{(j+1)!}\sum_{\ell=1}^{j}\mathbf{B}^{j-\ell}\left(\mathbf{q}^{(\ell-1)}(t_n)\right.\right. \\ &\quad \left.\left. - \mathbf{F}^{(\ell-1)}(t_n)\right)\right] + \mathcal{O}(\Delta t^{k+1}).\end{aligned} \tag{7.14}$$

The sum of the first three terms on the right of (7.14) equals $\mathbf{y}(t_{n+1})$. By using the triangle inequality and the fact that $\|\mathbf{P}\|_\infty \le 1$, we therefore obtain

$$\begin{aligned}\|\widetilde{\mathbf{y}}(\Delta t) - \mathbf{y}(t_{n+1})\|_\infty &\le \|\mathbf{B}\|_\infty \sum_{j=0}^{k-1} \frac{\Delta t^{j+1}}{(j+1)!}\|\widetilde{\mathbf{y}}^{(j)}(0) - \mathbf{y}^{(j)}(t_n)\|_\infty \\ &+ \|\mathbf{B}(\mathbf{I}-\mathbf{P})\|_\infty \sum_{j=0}^{k-1} \frac{\Delta t^{j+1}}{(j+1)!}\sum_{\ell=1}^{j}\|\mathbf{B}^{j-\ell}\|_\infty\|\mathbf{q}^{(\ell-1)}(t_n) - \mathbf{F}^{(\ell-1)}(t_n)\|_\infty + \mathcal{O}(\Delta t^{k+1}),\end{aligned}$$

where the first two terms on the right can be estimated by using (7.4) and Lemma 7.1. This concludes the proof. □

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

Lemma 7.3. *Let $\widetilde{\mathbf{y}}_{\frac{m+1}{p}}$ be defined by Algorithm 6.1 for $s = k = 2$, 3 or 4, $m \ge 0$, and assume that $\mathbf{F}(t) = 0$. Then we have for $0 \le m \le p-1$:*

$$\begin{aligned}\widetilde{\mathbf{y}}_{\frac{m+1}{p}} = \mathbf{y}_n + (m+1)\Delta\tau\mathbf{B}\mathbf{y}_n + \frac{(m+1)^2}{2}\Delta\tau^2\mathbf{B}^2\mathbf{y}_n \\ + \sum_{i=3}^{2m+2}\Delta\tau^i\alpha_{i,m+1}(\mathbf{BP})^{i-2}\mathbf{B}^2\mathbf{y}_n\end{aligned} \tag{7.15}$$

for $s = k = 2$,

$$\begin{aligned}\widetilde{\mathbf{y}}_{\frac{m+1}{p}} &= \mathbf{y}_n + (m+1)\Delta\tau\mathbf{B}\mathbf{y}_n + \frac{(m+1)^2}{2}\Delta\tau^2\mathbf{B}^2\mathbf{y}_n + \frac{(m+1)^3}{6}\Delta\tau^3\mathbf{B}^3\mathbf{y}_n \\ &+ \sum_{i=4}^{3m+4}\Delta\tau^i\left(\alpha_{i,m+1}(\mathbf{BP})^{i-2}\mathbf{B}^2\mathbf{y}_n + \beta_{i,m+1}(\mathbf{BP})^{i-3}\mathbf{B}^3\mathbf{y}_n\right)\end{aligned} \tag{7.16}$$

for $s = k = 3$, and

$$\begin{aligned}\widetilde{\mathbf{y}}_{\frac{m+1}{p}} &= \mathbf{y}_n + (m+1)\Delta\tau \mathbf{B}\mathbf{y}_n + \frac{(m+1)^2}{2}\Delta\tau^2\mathbf{B}^2\mathbf{y}_n + \frac{(m+1)^3}{6}\Delta\tau^3\mathbf{B}^3\mathbf{y}_n \\ &+ \frac{(m+1)^4}{24}\Delta\tau^4\mathbf{B}^4\mathbf{y}_n + \sum_{i=5}^{4m+6}\Delta\tau^i\Big(\alpha_{i,m+1}(\mathbf{BP})^{i-2}\mathbf{B}^2\mathbf{y}_n \\ &+ \beta_{i,m+1}(\mathbf{BP})^{i-3}\mathbf{B}^3\mathbf{y}_n + \delta_{i,m+1}(\mathbf{BP})^{i-4}\mathbf{B}^4\mathbf{y}_n\Big)\end{aligned} \tag{7.17}$$

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 (7.15) by induction on m. The proofs of (7.16) and (7.17) are similar but more cumbersome; hence, they are omitted here.

We let $s = k = 2$ and consider Algorithm 6.1. Since $\mathbf{F}(t)$ is identically zero, so is the interpolation polynomial $\mathbf{q}(t)$. Therefore, Step 2 of Algorithm 6.1 reduces to

$$\mathbf{w}_{n,0} = \mathbf{B}(\mathbf{I}-\mathbf{P})\sum_{i=1}^{2} b_i\mathbf{y}_n, \qquad \mathbf{w}_{n,1} = \mathbf{B}(\mathbf{I}-\mathbf{P})\sum_{i=1}^{2} b_i c_i \mathbf{B}\mathbf{y}_n. \tag{7.18}$$

The coefficients of the underlying RK2 method satisfy the order conditions (6.2) with $q = 1, 2$ and $c_1 = 0$ – see also ([13], p. 156):

$$a_{21} = c_2, \quad b_1 + b_2 = 1, \quad b_2c_2 = 1/2. \tag{7.19}$$

Thus, we rewrite (7.18) as

$$\mathbf{w}_{n,0} = \mathbf{B}(\mathbf{I}-\mathbf{P})\mathbf{y}_n, \qquad \mathbf{w}_{n,1} = \mathbf{B}(\mathbf{I}-\mathbf{P})\mathbf{B}\mathbf{y}_n. \tag{7.20}$$

In Step 3 of Algorithm 6.1, we now use (7.20), the fact that $\widetilde{\mathbf{y}}_0 = \mathbf{y}_n$ and (7.19) to calculate

$$\begin{aligned}\mathbf{k}_{1,\frac{1}{p}} &= \mathbf{w}_{n,0} + \mathbf{BP}\widetilde{\mathbf{y}}_0 = \mathbf{B}\mathbf{y}_n, \\ \mathbf{k}_{2,\frac{1}{p}} &= \mathbf{w}_{n,0} + c_2\Delta\tau\mathbf{w}_{n,1} + \mathbf{BP}\left(\widetilde{\mathbf{y}}_0 + \Delta\tau a_{21}\mathbf{k}_{1,\frac{1}{p}}\right) = \mathbf{B}\mathbf{y}_n + c_2\Delta\tau\mathbf{B}^2\mathbf{y}_n.\end{aligned}$$

Then we complete the first local time-step as

$$\widetilde{\mathbf{y}}_{\frac{1}{p}} = \widetilde{\mathbf{y}}_0 + \Delta\tau\left(b_1\mathbf{k}_{1,\frac{1}{p}} + b_2\mathbf{k}_{2,\frac{1}{p}}\right) = \mathbf{y}_n + \Delta\tau\mathbf{B}\mathbf{y}_n + \frac{\Delta\tau^2}{2}\mathbf{B}^2\mathbf{y}_n, \tag{7.21}$$

which yields (7.15) with $m = 0$.

We now proceed with the induction step and assume that (7.15) holds for $m \geq 0$. Following Algorithm 6.1, we again explicitly calculate

$$\begin{aligned}\mathbf{k}_{1,\frac{m+1}{p}} &= \mathbf{w}_{n,0} + m\Delta\tau\,\mathbf{w}_{n,1} + \mathbf{BP}\widetilde{\mathbf{y}}_{\frac{m}{p}}, \\ \mathbf{k}_{2,\frac{m+1}{p}} &= \mathbf{w}_{n,0} + (m+c_2)\Delta\tau\,\mathbf{w}_{n,1} + \mathbf{BP}\left(\widetilde{\mathbf{y}}_{\frac{m}{p}} + \Delta\tau a_{21}\mathbf{k}_{1,\frac{m+1}{p}}\right).\end{aligned} \tag{7.22}$$

Next, we use the induction hypothesis to replace $\widetilde{\mathbf{y}}_{\frac{m}{p}}$ and rewrite (7.22) as

$$\begin{aligned}
\mathbf{k}_{1,\frac{m+1}{p}} &= \mathbf{B}\mathbf{y}_n + m\Delta\tau\mathbf{B}^2\mathbf{y}_n + \frac{m^2\Delta\tau^2}{2}\mathbf{B}\mathbf{P}\mathbf{B}^2\mathbf{y}_n + \sum_{i=3}^{2m}\alpha_{i,m}\Delta\tau^i(\mathbf{B}\mathbf{P})^{i-1}\mathbf{B}^2\mathbf{y}_n, \\
\mathbf{k}_{2,\frac{m+1}{p}} &= \mathbf{B}\mathbf{y}_n + (m+c_2)\Delta\tau\mathbf{B}^2\mathbf{y}_n + (m^2+2mc_2)\frac{\Delta\tau^2}{2}\mathbf{B}\mathbf{P}\mathbf{B}^2\mathbf{y}_n \\
&\quad + \sum_{i=3}^{2m}\alpha_{i,m}\Delta\tau^i(\mathbf{B}\mathbf{P})^{i-1}\mathbf{B}^2\mathbf{y}_n + \sum_{i=3}^{2m}c_2\alpha_{i,m}\Delta\tau^{i+1}(\mathbf{B}\mathbf{P})^{i-2}\mathbf{B}^2\mathbf{y}_n.
\end{aligned}$$

We now insert these expressions for the two stages $\mathbf{k}_{1,\frac{m+1}{p}}$, $\mathbf{k}_{2,\frac{m+1}{p}}$ into the update of Step 3 in Algorithm 6.1,

$$\widetilde{\mathbf{y}}_{\frac{m+1}{p}} = \widetilde{\mathbf{y}}_{\frac{m}{p}} + \Delta\tau\left(b_1\mathbf{k}_{1,\frac{m+1}{p}} + b_2\mathbf{k}_{2,\frac{m+1}{p}}\right),$$

and again use the induction hypothesis to replace $\widetilde{\mathbf{y}}_{\frac{m}{p}}$. These calculations lead to

$$\begin{aligned}
\widetilde{\mathbf{y}}_{\frac{m+1}{p}} &= \mathbf{y}_n + m\Delta\tau\mathbf{B}\mathbf{y}_n + \frac{m^2\Delta\tau^2}{2}\mathbf{B}^2\mathbf{y}_n + \sum_{i=3}^{2m}\alpha_{i,m}\Delta\tau^i(\mathbf{B}\mathbf{P})^{i-2}\mathbf{B}^2\mathbf{y}_n \\
&\quad + b_1\Delta\tau\left(\mathbf{B}\mathbf{y}_n + m\Delta\tau\mathbf{B}^2\mathbf{y}_n + \frac{m^2\Delta\tau^2}{2}\mathbf{B}\mathbf{P}\mathbf{B}^2\mathbf{y}_n + \sum_{i=3}^{2m}\alpha_{i,m}\Delta\tau^i(\mathbf{B}\mathbf{P})^{i-1}\mathbf{B}^2\mathbf{y}\right) \\
&\quad + b_2\Delta\tau\Bigg(\mathbf{B}\mathbf{y}_n + (m+c_2)\Delta\tau\mathbf{B}^2\mathbf{y}_n + (m^2+2mc_2)\frac{\Delta\tau^2}{2}\mathbf{B}\mathbf{P}\mathbf{B}^2\mathbf{y}_n \\
&\quad + \sum_{i=3}^{2m}\alpha_{i,m}\Delta\tau^i(\mathbf{B}\mathbf{P})^{i-1}\mathbf{B}^2\mathbf{y}_n + \sum_{i=3}^{2m}c_2\alpha_{i,m}\Delta\tau^{i+1}(\mathbf{B}\mathbf{P})^{i-2}\mathbf{B}^2\mathbf{y}_n\Bigg).
\end{aligned}$$

By collecting like powers in $\Delta\tau$ and again using (7.19), we obtain (7.15), where the coefficients $\alpha_{i,m+1}$, $i \geq 3$, are recursively determined by $\alpha_{i,m}$ as

$$\begin{aligned}
\alpha_{3,m+1} &= \alpha_{3,m} + \frac{m^2+m}{2}, \qquad \alpha_{4,m+1} = \alpha_{3,m} + \alpha_{4,m} + \frac{m^2}{4}, \\
\alpha_{i,m+1} &= \alpha_{i,m} + \alpha_{i-1,m} + \frac{\alpha_{i-2,m}}{2}, \qquad \text{for } 5 \leq i \leq 2m, \\
\alpha_{2m+1,m+1} &= \alpha_{2m,m} + \frac{\alpha_{2m-1,m}}{2}, \qquad \alpha_{2m+2,m+1} = \frac{\alpha_{2m,m}}{2},
\end{aligned}$$

with starting values $\alpha_{i,0} = 0$. ☐

Remark 7.4. *Multirate partitioned RK methods [52, 92, 66], originally developed for the integration of multi-component dynamical systems, can also yield explicit LTS strategies* *in the PDE context if the coarse/refined regions correspond to the latent/active components. If we apply the explicit second-order SH2 method by Hundsdorfer et al. [92], for instance, to (3.6) with* $\mathbf{F} = 0$ *and* $p = 2$, *it reduces to*

$$\mathbf{y}_{n+1}^{SH2} = \mathbf{y}_n + \Delta t \mathbf{B}\mathbf{y}_n + \frac{\Delta t^2}{2}\mathbf{B}^2\mathbf{y}_n + \Delta t^3\left(\frac{3}{16}\mathbf{P}\mathbf{B}^3\mathbf{y}_n - \frac{3}{32}(\mathbf{P}\mathbf{B})^2\mathbf{B}\mathbf{y}_n\right) + \Delta t^4\left(\frac{1}{32}(\mathbf{P}\mathbf{B})^2\mathbf{B}^2\mathbf{y}_n - \frac{1}{64}\mathbf{P}(\mathbf{B}\mathbf{P})^2\mathbf{B}^2\mathbf{y}_n\right) + \frac{\Delta t^5}{256}(\mathbf{P}\mathbf{B})^3(\mathbf{I}-\mathbf{P})\mathbf{B}^2\mathbf{y}_n. \tag{7.23}$$

Similarly from (7.15) with $p = 2$, *we immediately obtain the one-step formulation of the LTS-RK2 method:*

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \mathbf{B}\mathbf{y}_n + \frac{\Delta t^2}{2}\mathbf{B}^2\mathbf{y}_n + \frac{\Delta t^3}{8}\mathbf{B}\mathbf{P}\mathbf{B}^2\mathbf{y}_n + \frac{5\Delta t^4}{4}(\mathbf{B}\mathbf{P})^2\mathbf{B}^2\mathbf{y}_n. \tag{7.24}$$

Clearly, the two methods differ – recall that $\mathbf{P}$ *and* $\mathbf{B}$ *do not commute; moreover, the SH2 method involves additional terms and higher powers of* $\mathbf{B}$.

By combining the upper bound on the truncation error from Proposition 7.2 with the one-step formulation from Lemma 7.3, we shall now prove the convergence of the LTS-RKs methods from Chapter 6, which we state as the following theorem.

Theorem 7.5. *For* $s = k = 2, 3, 4$ *the LTS-RKs(p) method is convergent of order k.*

Proof. First, we let $s = k = 2$, $\Delta\tau = \Delta t/p$, and set $m = p - 1$ in (7.15). From Lemma 7.3, we then infer that

$$\mathbf{y}_{n+1} = \widetilde{\mathbf{y}}_1 = \mathbf{y}_n + \Delta t\, \Phi_{RK2}(\mathbf{y}_n, \Delta t), \tag{7.25}$$

with

$$\Phi_{RK2}(\mathbf{y}_n, \Delta t) = \mathbf{B}\mathbf{y}_n + \frac{\Delta t}{2}\mathbf{B}^2\mathbf{y}_n + \sum_{i=3}^{2p} \alpha_{i,p}\frac{\Delta t^{i-1}}{p^i}(\mathbf{B}\mathbf{P})^{i-2}\mathbf{B}^2\mathbf{y}_n. \tag{7.26}$$

Moreover, Proposition 7.2 implies that the truncation error satisfies

$$\|\mathbf{y}(t+\Delta t) - \mathbf{y}(t) - \Delta t\, \Phi_{RK2}(\mathbf{y}(t), \Delta t)\| \le C\Delta t^3. \tag{7.27}$$

To apply standard convergence theory for one-step methods (e.g. [55]) and thus conclude the proof, we must show that the increment function Φ_{RK2} is Lipschitz-continuous in its first argument. Since $\|\mathbf{P}\| \le 1$ and also $\Delta t\|\mathbf{B}\| \le 1$ for Δt sufficiently small, we conclude that

$$\begin{aligned}&\|\Phi_{RK2}(\mathbf{u}, \Delta t) - \Phi_{RK2}(\mathbf{v}, \Delta t)\| \\ &\le \left\|\mathbf{B} + \frac{\Delta t}{2}\mathbf{B}^2 + \sum_{i=3}^{2p}\alpha_{i,p}\frac{\Delta t^{i-1}}{p^i}(\mathbf{B}\mathbf{P})^{i-2}\mathbf{B}^2\right\| \cdot \|\mathbf{u}-\mathbf{v}\| \le L_{RK2}\|\mathbf{u}-\mathbf{v}\|.\end{aligned}$$

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 Δt, p and $\|\mathbf{B}\|$ for Δt 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 (7.15) replaced by (7.16) or (7.17), respectively. ☐

The proof of Theorem 7.5 is straightforward and immediately extends to arbitrarily high order of accuracy. It relies, however, on the one-step formulation provided by Lemma 7.3, which cannot be stated for arbitrary k because no generic form of the order conditions, such as (7.19), is available for $k \geq 5$ – see [55].

Remark 7.6. *For $s = k$, the one-step formulations (7.15)–(7.17) of the LTS-RKs methods with $m = p - 1$ and $\mathbf{P} = 0$ reduce to*

$$\mathbf{y}_{n+1} = \widetilde{\mathbf{y}}_1 = R(\Delta t\,\mathbf{B})\,\mathbf{y}_n,$$

where $R(z) = 1 + z + \dots z^s/s!$ 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 (3.6) yields

$$\mathbf{y}_{n+1} = \sum_{j=0}^{k} \frac{(\Delta t\mathbf{B})^j \mathbf{y}_n}{j!} + \sum_{j=k+1}^{s} \sigma_j (\Delta t \mathbf{B})^j \mathbf{y}_n, \tag{7.28}$$

where the additional parameters σ_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 (6.6) can be modified without affecting the order of accuracy. Then, Step 2 of Algorithm 6.1 becomes:

2. *For $j = 0, \dots, s-1$ compute*

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

Although the coefficients γ_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

$$\mathbf{y}_{n+1} = \sum_{j=0}^{k} \frac{(\Delta t \mathbf{B})^j \mathbf{y}_n}{j!} + \sum_{j=k+1}^{s} \gamma_{j-1} \sum_{i=1}^{s} b_i c_i^{j-1} (\Delta t \mathbf{B})^j \mathbf{y}_n. \tag{7.29}$$

The two stability functions in (7.28) and (7.29) are identical, if

$$\gamma_{j-1} = \frac{\sigma_j}{\sum_{i=1}^{s} b_i c_i^{j-1}}, \qquad k \leq j \leq s,$$

which we shall henceforth assume.

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

$$\sigma_5 = \sum_{i,j,k,l=1}^{5} b_i a_{i,j} a_{j,k} a_{k,l} c_l$$

is crucial for the stability of the scheme; the optimal value, $\sigma_5 = 1/200$, indeed corresponds to the values in Table 19 [15]. To retain that optimal stability for the LTS-LSRK5 method, we infer from the LTS-LSRK5(1) algorithm with $\mathbf{F} = 0$ and $\mathbf{P} = 0$ that γ_4 must equal

$$\gamma_4 = \frac{\sigma_5}{\sum_{i=1}^{5} B_i C_i^4 + \sum_{i=2}^{5} B_i A_i C_{i-1}^4 + \sum_{i=3}^{5} B_i A_i A_{i-1} C_{i-2}^4 + B_5 A_5 A_4 A_3 C_2^4}.$$

8

NUMERICAL EXPERIMENTS

We shall now present numerical experiments, for both wave equation (3.1) and Maxwell's equations (4.1) and (4.2), which validate the expected order of convergence of the LTS-RK methods derived in Chapter 6, illustrate their stability properties and demonstrate their usefulness in the presence of complex geometry. Initially we analyse the numerical properties of different LTS-RK schemes for the simulation of acoustic waves. First, we consider a simple one-dimensional test problem to show that the kth 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. We then illustrate the versatility of our LTS schemes by simulating the propagation of a plane wave across a narrow gap. Finally, we present the convergence results seen for the wave equation also for Maxwell's equations in 1D and further demonstrate the usefulness of the LTS approach by studying electromagnetic waves on a complex 2D example.

8.1 NUMERICAL RESULTS FOR THE WAVE EQUATION

In this section we choose the damped wave equation as our model problem to study convergence and stability properties of the proposed LTS-RK schemes. We additionally present a two-dimensional example, where we combine $\mathcal{P}^2$ continuous mass-lumped FE with a third-order LTS-LSRK3 scheme to simulate a propagating wave field.

Convergence:

We consider the damped wave equation (3.1) in 1D with $c = 1$ and $\sigma = 0.1$ on the interval $\Omega = [0\,,\,6]$. The initial conditions $u_0 = \sin(\pi x)$, $v_0 = 0$ and the source

$$f(x,t) = \sin(\pi x)\left(\left(\pi^2 - 1\right)\cos(t) - \sigma\sin(t)\right)$$

yield the exact solution

$$u(x,t) \quad = \quad \cos(t)\sin(\pi x)\,.$$

Next, we divide Ω into three equal parts. The left and right intervals, $[0,2]$ and $[4,6]$, respectively, are discretized with an equidistant mesh of size h^{coarse}, whereas the interval $\Omega_f = [2,4]$ is discretized with an equidistant mesh of size $h^{\text{fine}} = h^{\text{coarse}}/p$ – see Fig. 3. Hence, the two outer intervals correspond to the coarse region and the inner interval $[2, 4]$ to the refined region.

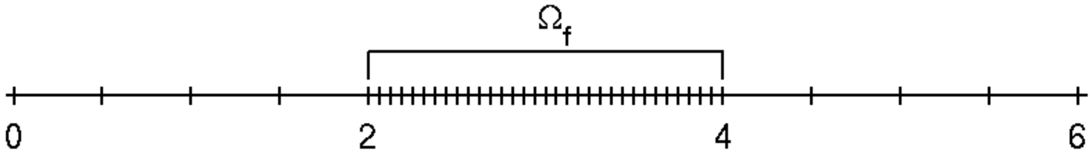


Figure 3: One-dimensional example: the computational domain $\Omega = [0,6]$ with the refined region $\Omega_f = [2,4]$.

We begin with the LTS schemes based on the popular RK4 and LSRK5 methods from Sections 6.2 and 6.3. On a sequence of increasingly finer meshes with mesh size h^{coarse} and fixed ratio $h^{\text{fine}} = h^{\text{coarse}}/p$, we consider a continuous $\mathcal{P}^3$-FE discretization of (3.1) with mass lumping. For every time-step, Δt, we take p local steps of size $\Delta\tau = \Delta t/p$ inside Ω_f, either with the LTS-RK4(p) or the LTS-LSRK5(p) algorithms from Sections 6.2 – see also Tables 18 and 19 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^{coarse}. As we simultaneously reduce h^{coarse} and Δt, 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. 4(a) and Fig. 5(a).

p	RK4	LTS-RK4
2	$2.4014e-07$	$2.4395e-07$
5	$2.4003e-07$	$2.3997e-07$
11	$2.3994e-07$	$2.3997e-07$

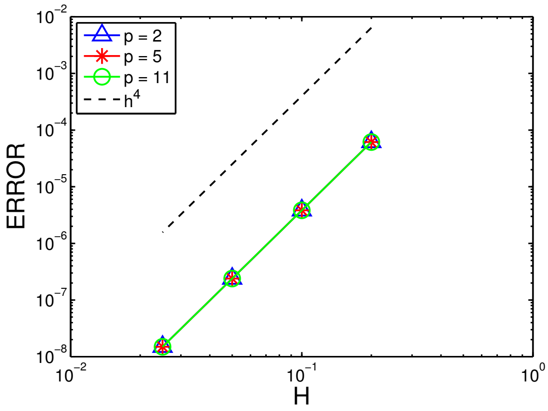
(a) continuous FE

p	RK4	LTS-RK4
2	$1.1407e-06$	$1.1925e-06$
5	$1.1395e-06$	$1.1401e-06$
11	$1.1395e-06$	$1.1396e-06$

(b) nodal DG-FE

Table 2: L^2-error at $T = 10$ for $h^{\text{coarse}} = 0.05$ for varying p.

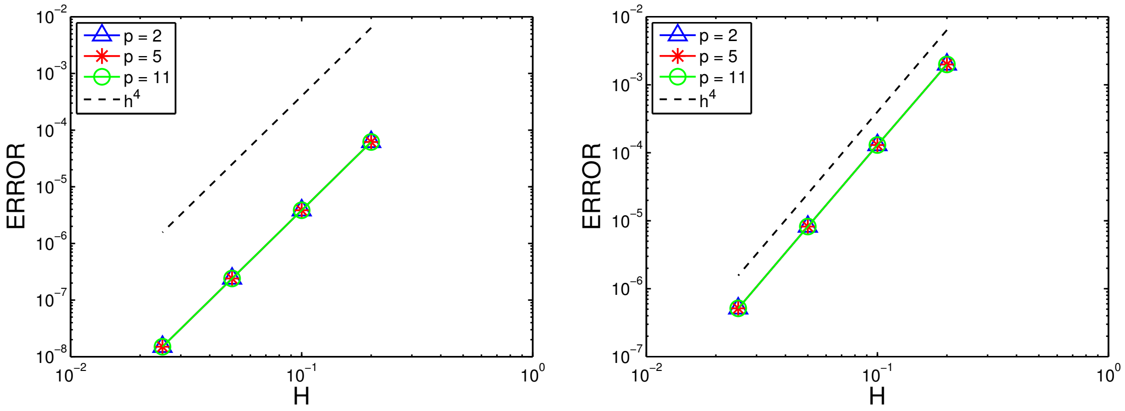
Next, we repeat the numerical experiment with a $\mathcal{P}^3$-FE nodal DG discretization with upwinding flux [57]. As shown in Fig. 4(b) and Fig. 5(b), both the LTS-RK4(p) and LTS-LSRK5(p) methods again yield overall fourth-order convergence



(a) continuous FE (b) nodal DG-FE

Figure 4: LTS-RK4(p) L^2-error at $T = 10$ vs. $H = h^{\text{coarse}} = 0.2, 0.1, 0.05, 0.025$ for different $\mathcal{P}^3$ FEM with $p = 2, 5, 11$. The errors for different p coincide at this scale - see also Table 2.

independently of p. Additionally we compare our results with the classical RK4 method with time-step $\Delta\tau = \Delta t/p$ on the whole domain. For this we fix $h^{\text{coarse}} = 0.05$ and vary p. We again monitor the L^2-error at final time $T = 10$ and obtain comparable results, with or without LTS for both continuous and nodal DG FE.



(a) continuous FE (b) nodal DG-FE

Figure 5: LTS-LSRK5(p) L^2-error at $T = 10$ vs. $H = h^{\text{coarse}} = 0.2, 0.1, 0.05, 0.025$ for different $\mathcal{P}^3$ FEM with $p = 2, 5, 11$. The errors for different p coincide at this scale.

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-RKs and the LTS-LSRKs 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 [30], and thus satisfy $\Delta\tau = \Delta t/\widetilde{p}$ with $\widetilde{p} = \lceil p^{4/3} \rceil$ when combined with a continuous FEM discretization.

Stability:

Before discussing the stability properties of the proposed LTS-RK methods, we first want to recall the stability regions of the underlying explicit RK schemes – see [55] for details.

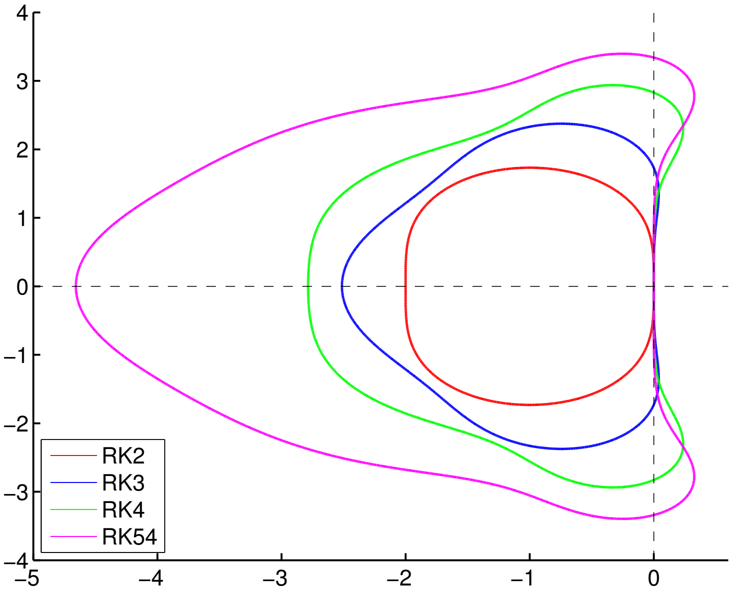
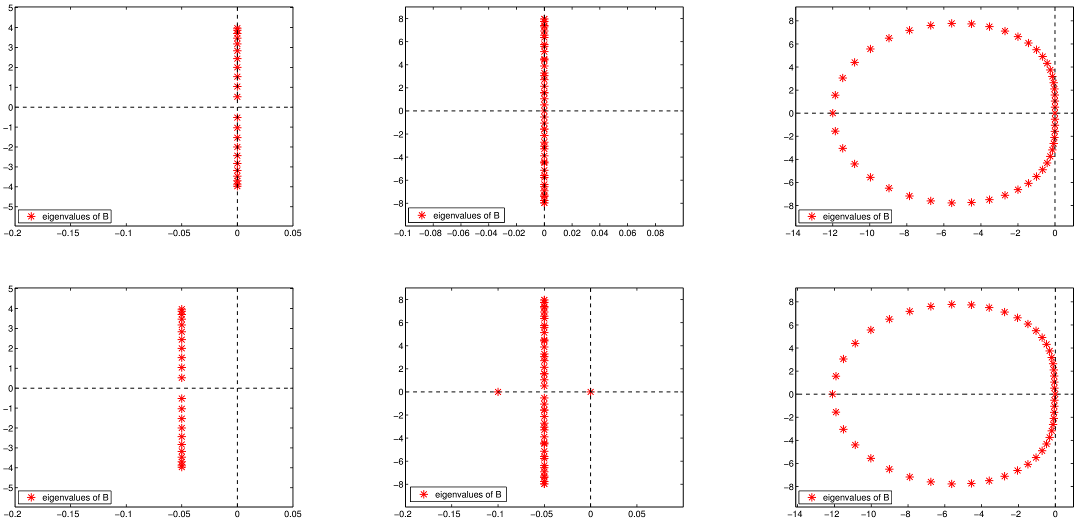


Figure 6: Stability domains of RKs methods for $k = s = 2, 3, 4$ and RK5 of order 4.

In Fig. 6 the stability domains of RKs methods with $k = s = 2, 3, 4$ are displayed. Additionally we can also observe how the additional stage increases the stability contour of the fourth-order LSRK5 scheme significantly when compared to the classical RK4 method.

With Fig. 6 in mind we want to motivate our choice of numerical flux, when opting for a nodal DG spatial discretization. As mentioned in Part I, both proposed fluxes, upwind and central, have their advantages. While the non-dissipative central flux gives us a numerical scheme, which is energy conserving, the choice of an upwind flux results in a system matrix with a spectrum better suited for a time integration with RK type methods. In Fig. 7 we can see the eigenvalues of the system matrix $\mathbf{B}$ for three different $\mathcal{P}^1$ finite elements: continuous FEM with mass lumping (left) and nodal DG FE with both central (middle) and upwind flux (right), respectively. For all three we consider (3.1) with $c = 1$, $\sigma = 0, 0.1$

and $f = 0$ on the interval $\Omega = [0\,,\,6]$ on an equidistant mesh with mesh size $h = 0.5$.



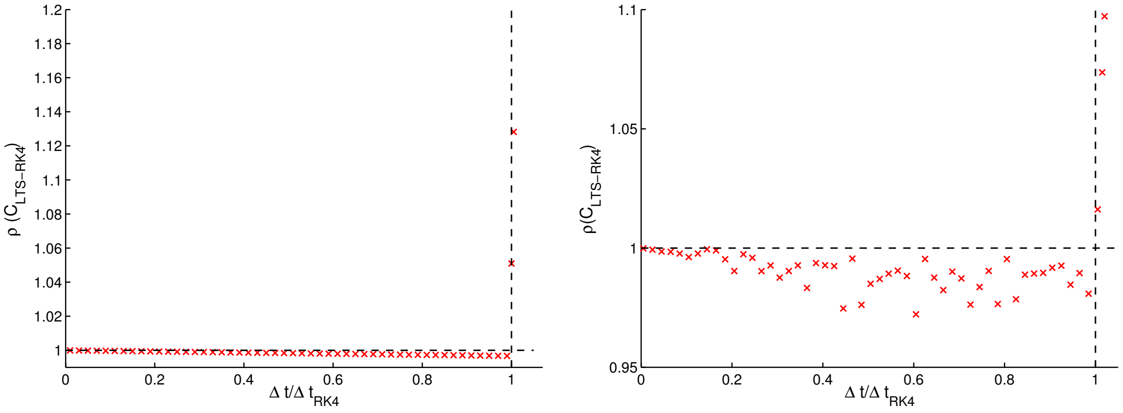
(a) cont. FEM (b) DG with centr. flux (c) DG with upw. flux

Figure 7: The spectral radius of **B** for different $\mathcal{P}^3$ FEM, with $\sigma = 0$ (first row) and $\sigma = 0.1$ (second row).

Note that the larger number of eigenvalues for the DG discretization originates from duplicating inner points in order to allow for discontinuous solutions. We can observe, that the spectrum of **B** coming from a DG discretization with upwind flux has a similar shape as the stability contours of the RK schemes presented in Fig. 6. This improves the stability behaviour when combining the two discretization methods. It also gives a possible explanation, why we don't observe the suboptimal stability condition for RK2 when discretizing in space with nodal DG FE. In [83] Niegemann et al. go even a step further by designing low-storage RK schemes with a stability region adapted to the spectral shape of a given physical problem. Those methods are found to yield significant performance improvements over previously known LSRK schemes.

We now focus on the stability of our LTS schemes. To do so, we consider again (3.1) with $c = 1$, $\sigma = 0.1$, but $f = 0$ on the interval $\Omega = [0\,,\,6]$, where $[0\,,\,2]$ and $[4\,,\,6]$ are discretized with an equidistant mesh of size h^{coarse}, whereas $\Omega_f = [2\,,\,4]$ is discretized with an equidistant mesh of size $h^{\text{fine}} = h^{\text{coarse}}/p$ – see Fig. 3. For $p = 1$, the mesh is equidistant throughout Ω and we denote by Δt_{RKs} or Δt_{LSRKs} the largest time-step permitted by the standard RKs or LSRKs method, respectively; note that $\Delta t_{RKs} = \Delta t_{LSRKs}$ for $s = 2, 3$.

For $p \geq 2$, we use either the LTS-RKs(p) or LTS-LSRKs(p) method and take p substeps of size $\Delta\tau = \Delta t/p$ inside Ω_f. Clearly, the maximal permissible time-



(a) continuous FE

(b) nodal DG-FE

Figure 8: The spectral radius of $\mathbf{C}_{LTS-RK4}$ vs. $\Delta t/\Delta t_{RK4}$ for different $\mathcal{P}^3$ FEM.

step, denoted by Δt_p, satisfies either $\Delta t_p \leq \Delta t_{RKs}$ or $\Delta t_p \leq \Delta t_{LSRKs}$, respectively. If $\Delta t_p = \Delta t_{RKs}$ or $\Delta t_p = \Delta t_{LSRKs}$, the corresponding LTS method imposes no further constraint on Δt and its CFL-condition is therefore optimal.

To determine the stability range of any LTS scheme from Chapter 6, we proceed as follows:

1. Determine Δt_{RKs}, respectively Δt_{LSRKs}, the maximal Δt allowed by the corresponding standard or low-storage RKs method (without LTS) for the equidistant mesh of mesh size h^{coarse};
2. refine the mesh p times inside Ω_f;
3. determine the maximal time-step Δt_p allowed by the LTS method for the locally refined mesh and compare Δt_p with Δt_{RKs} or Δt_{LSRKs}, respectively.

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

$$\mathbf{y}_{n+1} = \mathbf{C}_{RK4}\,\mathbf{y}_n\,, \qquad \mathbf{C}_{RK4} = \mathbf{I} + \Delta t\,\mathbf{B} + \frac{\Delta t^2}{2}\,\mathbf{B}^2 + \frac{\Delta t^3}{3!}\,\mathbf{B}^3 + \frac{\Delta t^4}{4!}\,\mathbf{B}^4\,,$$

is stable if $\rho(\mathbf{C}_{RK4}) \leq 1$, where $\rho(\mathbf{C}_{RK4})$ denotes the spectral radius of the matrix $\mathbf{C}_{RK4}$ [55]. Progressively increasing Δt while monitoring $\rho(\mathbf{C}_{RK4})$, we find that the maximal time-step is $\Delta t_{RK4} = 0.0656$.

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 $\mathbf{P}$. To determine the stability range of the LTS-RK4(2) method (Algorithm 6.5, Sect. 6.4), we use Lemma 7.3 with $s = k = 4$ to rewrite it as

$$\mathbf{y}_{n+1} = \mathbf{C}_{LTS-RK4}\,\mathbf{y}_n\,,$$

where the matrix $\mathbf{C}_{LTS-RK4}$ is explicitly given by (7.17) with $p = 2$ and $m = p - 1$. Again, the LTS-RK4 method is stable when $\mathbf{C}_{LTS-RK4} \leq 1$. In Fig. 8(a), we observe that the spectral radius of $\mathbf{C}_{LTS-RK4}$ lies below one for all time-steps $\Delta t \leq \Delta t_{RK4}$. Therefore, the LTS-RK4(2) scheme, when combined with standard $\mathcal{P}^3$-elements, is stable up to the maximal time-step and hence its CFL-condition optimal. Next, we repeat the numerical experiment for the $\mathcal{P}^3$ nodal DG-FE discretization with upwinding flux. As shown in Fig. 8(b), the range of stable time-steps again is maximal and the CLF condition thus optimal. The non-monotone dependence of the eigenvalues in Fig. 8(b) and Fig. 9(b) is due to the limited accuracy in the numerical computation of the eigenvalues. As standard methods such as `eig` or `eigs` in Matlab, proved even less accurate for this problem, we have used here the augmented block Householder Arnoldi method [6] instead.

We perform similar numerical experiments with different values of p, but also $\mathcal{P}^{s-1}$ continuous and nodal DG FE for the LTS-RKs methods with $s = 2, 3$. These results are summarized in Table 3. Since the CFL-condition of the RK2 method when combined with continuous FEM is not linear but instead constrained by $\Delta t \leq Ch^{4/3}$ [30], we again set $\Delta\tau = \Delta t/\widetilde{p}$ with $\widetilde{p} = \lceil p^{4/3} \rceil$ for stability.

	$p = 2$		$p = 3$		$p = 5$		$p = 11$	
	FE	DG	FE	DG	FE	DG	FE	DG
LTS-RK2(p)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
LTS-RK3(p)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
LTS-RK4(p)	1.0	1.0	1.0	1.0	1.0	1.0	0.98	1.0

Table 3: Stability of the LTS-RKs(p) schemes for $s = 2, 3, 4$ when combined with a continuous FE or nodal DG-FE discretization with $h^{\text{coarse}} = 0.2$: the ratio $\Delta t_p / \Delta t_{RKs}$ is shown for varying p.

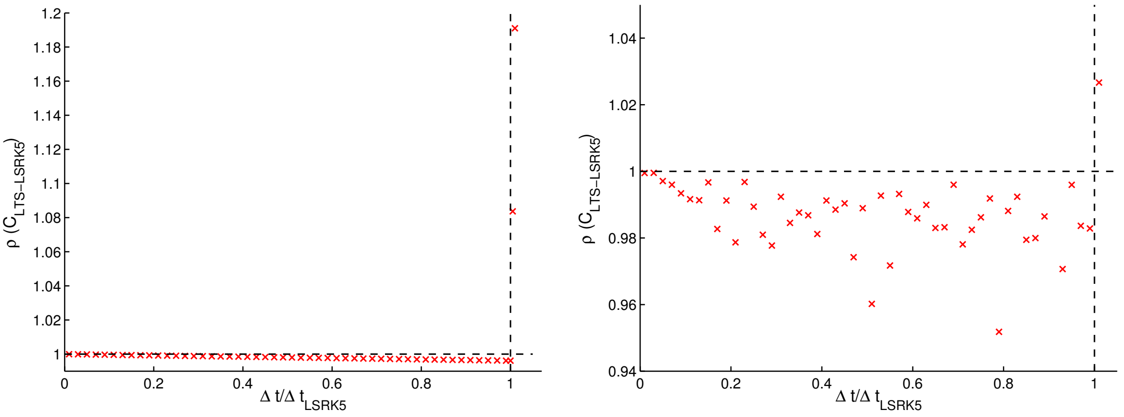
Next, we repeat the above numerical stability analysis for the LTS-LSRKs methods. For $s = 2, 3$, we obtain identical results, as expected, since all methods with $k = s$ have the same stability regions; these results are omitted here.

Different values of h^{coarse} lead to similar results, as shown in Table 4 for $h^{\text{coarse}} = 0.05$.

To study the stability of the fourth-order LTS scheme based on the LSRK5 method – see Section 6.3 and also Remark 7.6 – we consider either a continuous or a nodal DG $\mathcal{P}^3$-FE discretization. Again, we set $h^{\text{coarse}} = 0.2$ and refine by a factor $p = 2$ those elements that lie inside the interval $[2, 4]$. Hence for every time-step Δt, we use the fourth-order LTS-LSRK5(2) method with $\Delta\tau = \Delta t/2$ inside the refined region. After refomulating the LTS-LSRK5(2) scheme as one-step method,

	$p=2$		$p=3$		$p=5$		$p=11$	
	FE	DG	FE	DG	FE	DG	FE	DG
LTS-RK2(p)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
LTS-RK3(p)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
LTS-RK4(p)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 4: Stability of the LTS-RKs(p) schemes for $s = 2,3,4$ when combined with a continuous FE or nodal DG-FE discretization with $h^{\text{coarse}} = 0.05$: the ratio $\Delta t_p / \Delta t_{RKs}$ is shown for varying p.



(a) continuous FE

(b) nodal DG

Figure 9: The spectral radius of $\mathbf{C}_{LTS-LSRK5}$ for different $\mathcal{P}^3$ FEM.

$\mathbf{y}_{n+1} = \mathbf{C}_{LTS-LSRK5}\mathbf{y}_n$, we compute the spectral radius of $\mathbf{C}_{LTS-LSRK5}$ for varying $\Delta t / \Delta t_{LSRK5}$. As shown in Fig. 9(a), the spectral radius of $\mathbf{C}_{LTS-LSRK5}$ lies below one for all time-steps $\Delta t \leq \Delta t_{LSRK5}$. Hence the LTS-LSRK5(2) scheme, when combined with standard $\mathcal{P}^3$-elements, is stable up to the optimal time-step. Similarly, we observe in Fig. 9(b) that the LTS-LSRK5(2) scheme also yields optimal stability when combined with a $\mathcal{P}^3$ nodal DG-FE discretization.

Finally to illustrate the effect of damping on the stability, we display in Table 5 the (maximal) time-step ratio $\Delta t_2 / \Delta t_{RKs}$ ($s = 2,3,4$) for varying σ, 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-RKs ($s =$ 2, 4) and LTS-LSRKs ($s =$ 2, 5) methods are stable when combined with the nodal DG-FE discretization with an upwinding flux.

	LTS-RK2(2)		LTS-RK3(2)		LTS-RK4(2)	
σ	cont. FE	nodal DG	cont. FE	nodal DG	cont. FE	nodal DG
0.001	1.0	1.0	1.0	1.0	1.0	1.0
0.01	1.0	1.0	1.0	1.0	1.0	1.0
0.1	1.0	1.0	1.0	1.0	1.0	1.0
1	1.0	1.0	0.97	1.0	1.0	1.0
10	1.0	1.0	0.91	1.0	0.99	1.0

Table 5: The LTS-RKs(2) scheme for $s = 2, 3, 4$ when combined with a continuous FE or nodal DG-FE discretization: the ratio $\Delta t_2/\Delta t_{RKs}$ is shown for varying σ.

Two-dimensional example:

To illustrate the usefulness of the LTS-RK methods, we now consider (3.1) in Ω, a rectangular domain of size $[0, 2] \times [0, 1]$ with two rectangular barriers inside forming a narrow gap, as shown in Fig. 10. We set the constant wave speed to $c = 1$ and the constant damping coefficient to $\sigma = 0.1$. At the boundary of Ω, we impose homogeneous Neumann conditions and choose as initial conditions the vertical Gaussian plane wave

$$u_0(x, y) = \exp\left(-(x - x_0)^2/\delta^2\right), \qquad v_0(x, y) = 0, \qquad (x, y) \in \Omega,$$

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

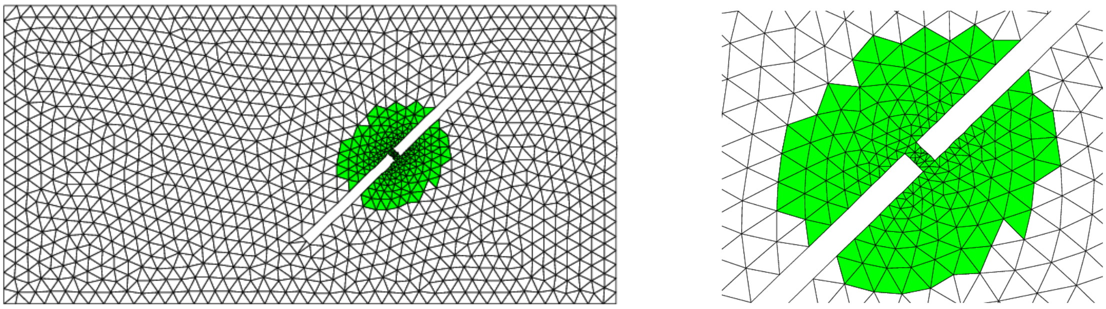


Figure 10: The initial triangular mesh (left); zoom on the "fine" mesh indicated by the darker (green) triangles (right).

For the spatial discretization we opt for $\mathcal{P}^2$ continuous finite elements with mass lumping. The homogeneous Neumann conditions are weakly imposed within the variational formulation. First, Ω is discretized with triangles of maximal size $h^{\text{coarse}} = 0.03$, where the mesh size h denotes the radius of the inscribed circle. However, such coarse triangles do not resolve the small geometric features of the gap, which require $h^{\text{fine}} \simeq h^{\text{coarse}}/7$, as shown in Fig. 10. Then, we successively

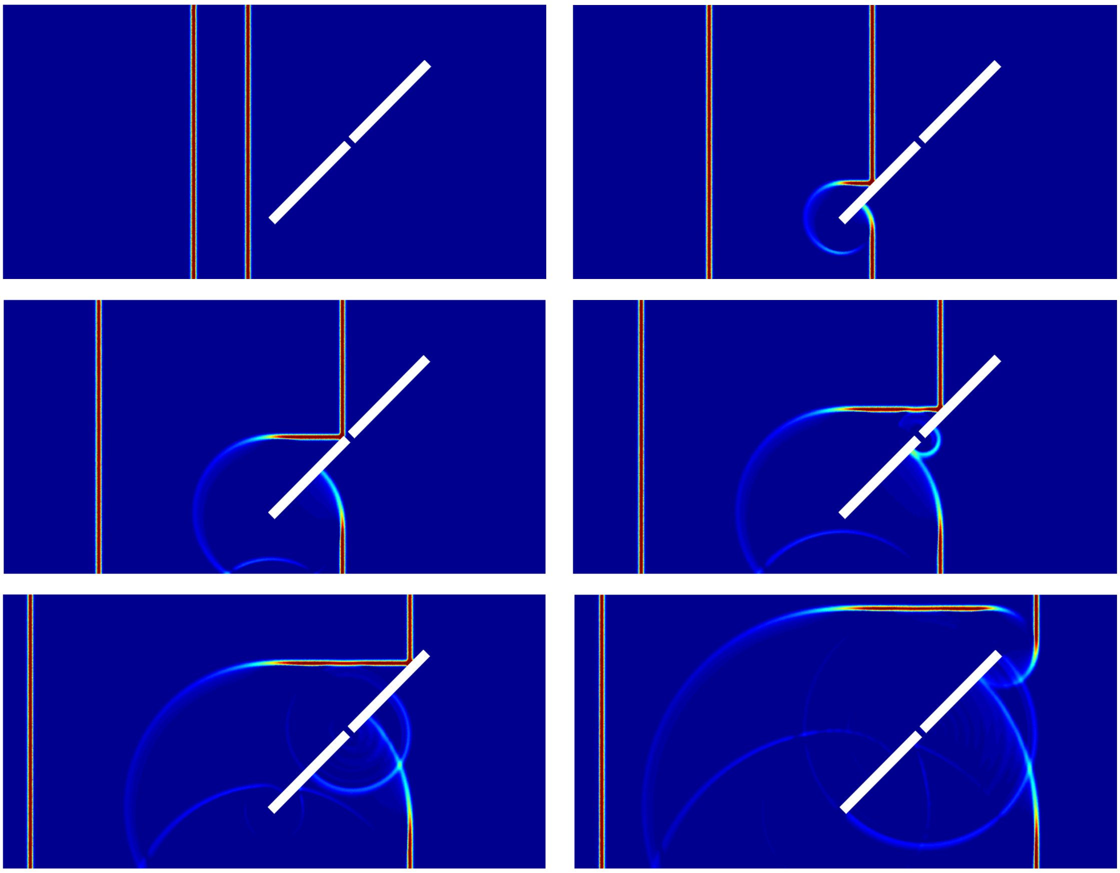


Figure 11: Two-dimensional example: the solution is shown at times t =0.1, 0.3, 0.45, 0.55, 0.7 and 0.9.

refine the entire mesh three times, each time splitting every triangle into four. Since the initial mesh in Ω 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. 10. 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 $\mathbf{P}$ – see Chapter 6.

For the time discretization, we choose the third-order LTS-LSRK3(7) scheme with $p = 7$, which for every time-step Δt 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 Δt was used everywhere inside Ω, it would have to be about seven times smaller than necessary in most of Ω. In Fig. 11, 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.

8.2 NUMERICAL RESULTS FOR MAXWELL'S EQUATION

In this section we study our LTS-RK schemes for electromagnetic problems. First we consider a simple one-dimensional test problem and show that kth-order LTS-RK methods always yield the optimal space-time rate of convergence when combined with a spatial finite element discretization of comparable accuracy, independently of the number of local time-steps p used in the fine region. Second, we apply our LTS-RK method to a more complex two-dimensional electromagnetic scattering problem.

Convergence:

We consider the one-dimensional homogeneous model problem, where (4.1) and (4.2) reduce to

$$\varepsilon\frac{\partial}{\partial t}E(x,t) = -\frac{\partial}{\partial x}H(x,t) - \sigma E(x,t) - J_a(x,t), \tag{8.1}$$

$$\mu\frac{\partial}{\partial t}H(x,t) = -\frac{\partial}{\partial x}E(x,t). \tag{8.2}$$

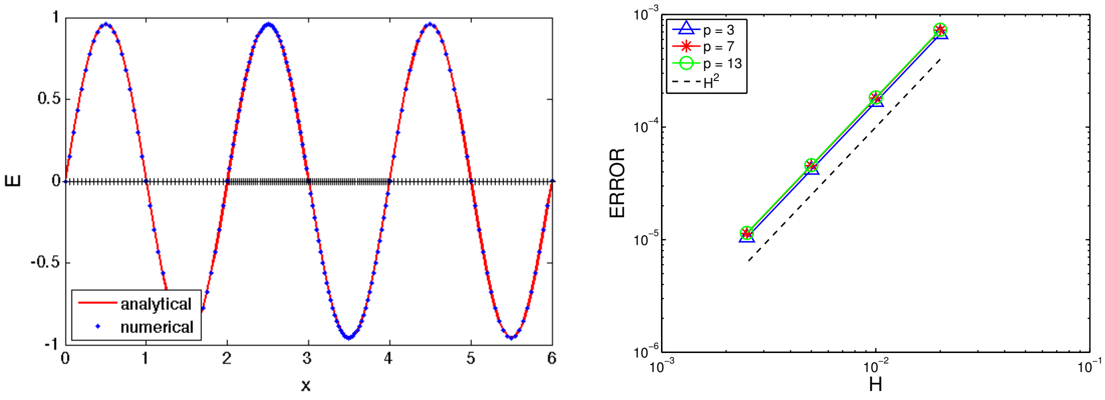
For simplicity we further assume constant material properties $\mu = \varepsilon = 1$ and $\sigma = 0.1$ on the interval $\Omega = [0,6]$. The initial conditions and right-hand side are chosen to yield the exact solutions

$$E(x,t) = -\cos(t)\sin(\pi x), \tag{8.3}$$

$$H(x,t) = -\pi\cos(t)\cos(\pi x). \tag{8.4}$$

Next we divide Ω into three equal parts. The left and the right intervals, $[0,2]$ and $[4,6]$, respectively, are discretized with an equidistant mesh of size h^{coarse}, whereas on the interval $[2,4]$ the mesh size is chosen as $h^{\text{fine}} = h^{\text{coarse}}/p$. Hence, the two outer intervals correspond to the coarse region and the inner interval to the refined region.

First we discretize in space using nodal DG with $\mathcal{P}^1$- elements with upwind flux on a sequence of increasingly finer meshes. 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 time-stepping scheme LTS-RK2(p). 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 $\|E(\cdot,T) - E_{h,\Delta t}(\cdot,T)\|_{L^2(\Omega)}$ at the final time $T = 1$.

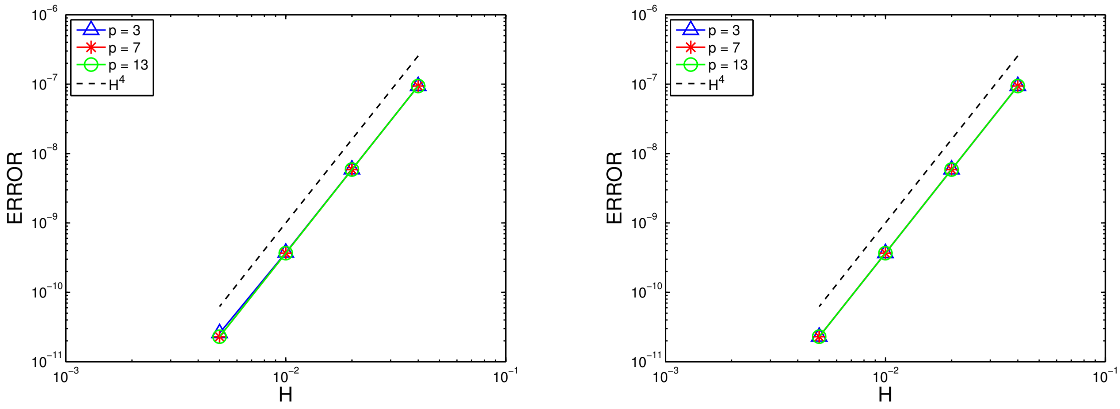


(a) Electric field (b) L^2−error

Figure 12: LTS-RK2(p) error vs. $H = h^{\text{coarse}} = 0.02, 0.01, 0.005, 0.0025$ for $\mathcal{P}^1$ finite elements with $p = 3, 7, 13$.

In Fig. 12 we show the numerical solution for $h^{\text{coarse}} = 0.1$ and $p = 2$ (a) and the L^2 error (b). We observe that regardless of the number of local time-steps $p = 3$, 7 or 13, the numerical method converges with order two.

Next, we combine the fourth-order LTS-RK4(p) and LTS-LSRK5(p) scheme with $\mathcal{P}^3$- elements. Thus, we expect an overall fourth-order convergence with respect to the L^2- norm. In Fig. 13 we display the numerical error vs. the mesh size $H = h^{\text{coarse}}$ for a sequence of meshes and different values of p. The results confirm the expected fourth-order convergence.



(a) RK4 (b) LSRK5

Figure 13: LTS-RK4(p) error (a) and LTS-LSRK5(p) (b) vs. $H = h^{\text{coarse}} = 0.04, 0.02, 0.01, 0.005$ for $\mathcal{P}^3$ finite elements with $p = 3, 7, 13$.

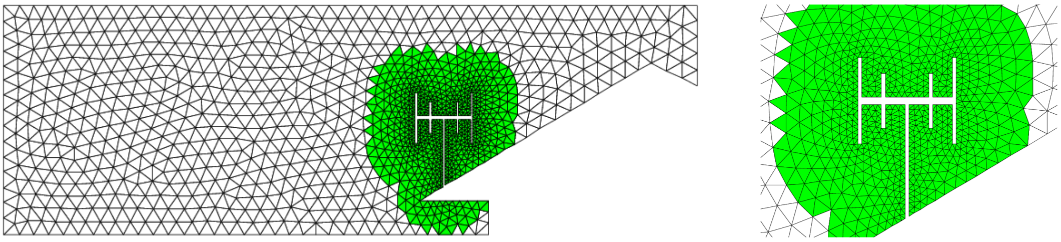


Figure 14: The initial triangular mesh : $h_{fine} \approx h_{coarse}/7$

Two-dimensional example:

To further illustrate the usefulness of the LTS-RK methods, we now consider Maxwell's equations in two space dimensions in transverse magnetic form. The computational domain Ω corresponds to a rectangle of size $[0,3] \times [0,1]$ from which the shape of a roof mounted antenna of thickness 0.01 has been cut out, as shown in Fig. 14. We set $\sigma = 0$ and $\mu = \varepsilon = 1$, $J_a = 0$, and impose PEC boundary conditions. A circular Gaussian wave is initiated through the initial conditions.

We choose $\mathcal{P}^3$ nodal DG elements in space. First, Ω is discretized with triangles of maximal size $h^{\text{coarse}} \approx 0.023$, where the mesh size h denotes the radius of the inscribed circle. However, such coarse triangles do not resolve the small geometric features of the antenna, which require $h^{\text{fine}} \simeq h^{\text{coarse}}/7$, as shown in Fig. 14. Then, we successively refine the entire mesh two times, each time splitting every triangle into four. Again, 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. 14. 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 $\mathbf{P}$.

For the time discretization, we choose the fourth-order LTS-RK4(7) scheme with $p = 7$, which for every time-step Δt takes seven local time-steps inside the refined region that surrounds the antenna. Thus, the numerical method is fourth-order accurate in both space and time with respect to the L^2-norm. If instead the same (global) time-step Δt was used everywhere inside Ω, it would have to be about seven times smaller than necessary in most of Ω.

In Fig. 15, snapshots of the numerical solution $\mathbf{E}_h^z$ at different times show how the circular wave impinges on the antenna and then subsequent reflections interact with the propagating wave front.

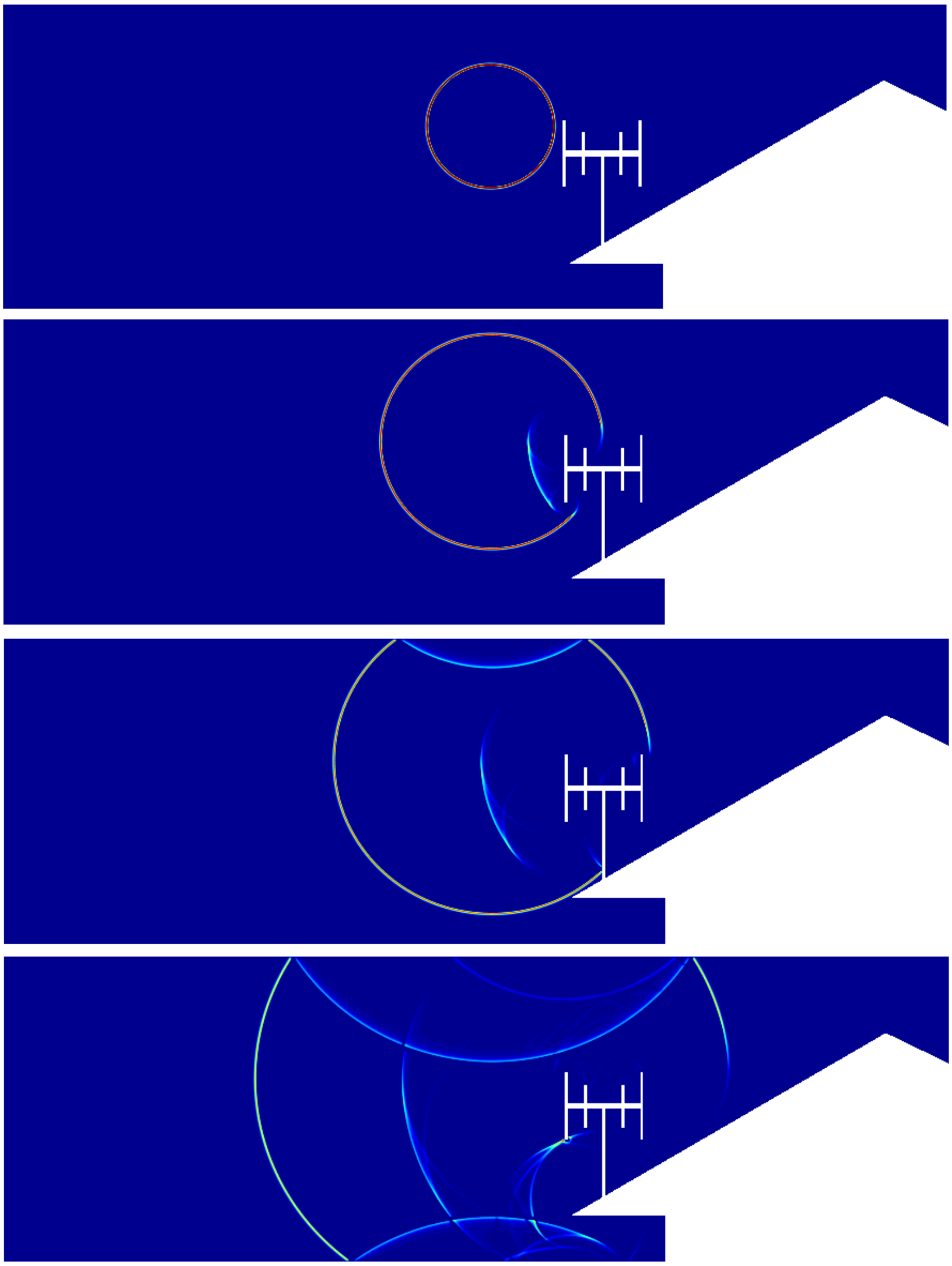


Figure 15: Snapshots of $\mathbf{E}_h^z$ at times $t = 0.2,\ 0.35,\ 0.5,\ 0.75$

Part III

LOCAL EXPONENTIAL ADAMS BASHFORTH SCHEMES

9

INTRODUCTION

As an alternative to the presented LTS-RK schemes we propose here local exponential integrators based on Adams methods for the time discretization of damped wave equations.
Exponential integrators have been around since the late 1950th – see [77] for an historical overview. But as the evaluation of matrix exponentials is very expensive they have only regained interest with the development of Krylov methods, which allow a more efficient computation of matrix function times vector products. For an extensive overview on existing methods we refer to [62].
Starting from the ODE (3.6) we focus here on exponential Adams-methods, which were presented in [63] for semilinear parabolic problems. Instead of splitting into linear and nonlinear part, we distinguish between fine and coarse part. We thus have the advantage of one global time-step, whose size is determined by the mesh size of the coarse region.
In this part we briefly discuss exponential Adams methods (for more details see [63]) and adapt them to our model problem. Numerical results in 1D confirm the expected space-time convergence rate and optimal stability properties. We give some comments on why Krylov methods fail to work in our case. Numerical results, however, suggest that local exponential integrators still present a good alternative in special cases, where we consider problems with only a few fine elements with a very high refinement rate.
We conclude with the familiar narrow-gap example from Part II.

10

LOCAL EXPONENTIAL ADAMS-METHODS

Following the approach presented in [63], we introduce exponential Adams methods in this chapter. We start by giving a derivation of the method for a general semilinear problem. We then apply the scheme to our model problem (3.6) using a splitting into coarse and fine elements and give some comments on efficient implementation.

10.1 EXPONENTIAL METHODS OF ADAMS-TYPE

We start by considering an abstract semilinear initial value problem

$$\mathbf{y}'(t) = -\mathbf{A}\mathbf{y}(t) + \mathbf{g}(t, \mathbf{y}(t)), \qquad \mathbf{y}(t_0) = \mathbf{y}_0 \tag{10.1}$$

where $\mathbf{A}$ is a square matrix and $\mathbf{g}$ is assumed to be Lipschitz continuous.
We formally get an exact solution of (10.1) at time $t_{n+1} = t_n + \Delta t$ by the variation-of-constants formula

$$\mathbf{y}(t_{n+1}) = e^{-\Delta t \mathbf{A}}\mathbf{y}(t_n) + \int_0^{\Delta t} e^{-(\Delta t - \tau)\mathbf{A}}\mathbf{g}(t_n + \tau, \mathbf{y}(t_n + \tau))\, d\tau. \tag{10.2}$$

To derive a numerical method from (10.2) we proceed as for the classical Adams-Bashforth schemes and replace $\mathbf{g}$ by its interpolation polynomial $\mathbf{p}_n$ through the k points

$$\big(t_{n-k+1}, \mathbf{g}(t_{n-k+1}, \mathbf{y}_{n-k+1})\big), \dots, \big(t_n, \mathbf{g}(t_n, \mathbf{y}_n)\big),$$

where $\mathbf{y}_j \approx \mathbf{y}(t_j)$ is a suitable approximation.
Thus, the polynomial $\mathbf{p}_n$ is given by

$$\mathbf{p}_n(t_n + \theta\Delta t) = \mathbf{G}_n + \sum_{j=1}^{k-1} (-1)^j \binom{-\theta}{j} \nabla^j \mathbf{G}_n, \qquad \mathbf{G}_j = \mathbf{g}(t_j, \mathbf{y}_j), \tag{10.3}$$

where $\binom{-\theta}{j}$ is a generalization of binomial coefficients, i.e.

$$\binom{\alpha}{j} = \frac{\alpha(\alpha-1)(\alpha-2)\cdots(\alpha-j+1)}{j!} \qquad \text{for } j \in \mathbb{N},\ \alpha \in \mathbb{R}.$$

Here, $\nabla^j \mathbf{G}_n$ denotes the jth backward difference, which is recursively defined as

$$\nabla^0 \mathbf{G}_n = \mathbf{G}_n, \qquad \nabla^j \mathbf{G}_n = \nabla^{j-1}\mathbf{G}_n - \nabla^{j-1}\mathbf{G}_{n-1}, \qquad j = 1, 2, \dots .$$

We now replace the exact solution at time t_n by its approximation $\mathbf{y}_n$ and the nonlinearity $\mathbf{g}(t_n, \mathbf{y}_n)$ by the interpolation polynomial $\mathbf{p}_n$ in (10.2), which yields the numerical method

$$\mathbf{y}_{n+1} = e^{-\Delta t \mathbf{A}}\mathbf{y}_n + \int_0^{\Delta t} e^{-(\Delta t - \tau)\mathbf{A}} \mathbf{p}_n(t_n + \tau)\, d\tau. \tag{10.4}$$

Inserting the interpolation polynomial (10.3) into (10.4) gives

$$\mathbf{y}_{n+1} = e^{-\Delta t \mathbf{A}}\mathbf{y}_n + \Delta t \sum_{j=0}^{k-1} \gamma_j(-\Delta t \mathbf{A}) \nabla^j \mathbf{G}_n \tag{10.5}$$

with weights

$$\gamma_j(z) = (-1)^j \int_0^1 e^{(1-\theta)z} \binom{-\theta}{j}\, d\theta, \qquad j \geq 0, \tag{10.6}$$

which we henceforth refer to as *exponential Adams-methods*.
The first 4 weights γ_j for $j \in \{0, 1, 2, 3\}$ are given by

$$\begin{aligned} \gamma_0(z) &= \frac{e^z - 1}{z}, \\ \gamma_1(z) &= \frac{e^z - z - 1}{z^2}, \\ \gamma_2(z) &= \frac{1}{2}\frac{(2+z)e^z - 2z^2 - 3z - 2}{z^3}, \\ \gamma_3(z) &= \frac{1}{3!}\frac{(6 + 6z + 2z^2)e^z - 6z^3 - 11z^2 - 12z - 6}{z^4}. \end{aligned} \tag{10.7}$$

Furthermore, it follows directly from (10.6) that for $z = 0$ the γ_j's reduce to the well-known coefficients of the classical Adams-Bashforth (AB) methods

$$\gamma_j(0) = (-1)^j \int_0^1 \binom{-\theta}{j}\, d\theta, \qquad j \geq 0.$$

Thus, for $\mathbf{A} = 0$ the exponential Adams methods reduce to the classical AB methods

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \sum_{j=0}^{k-1} \gamma_j(0) \nabla^j \mathbf{G}_n. \tag{10.8}$$

j	0	1	2	3
$\gamma_j(0)$	1	1/2	5/12	3/8

Table 6: Coefficients for the Adams-Bashforth methods

In Table 6 we present the values of $\gamma_j(0)$ for $j = 0, 1, 2, 3$. For more details on AB methods see [55].

We are now ready to apply (10.5) to our model problem (3.6). Instead of splitting into linear and nonlinear part, however, we will distinguish between coarse and fine region

10.2 APPLICATION TO THE DAMPED WAVE EQUATION

In this section we want to apply the exponential Adams methods derived in Section 10.1 to the first order system (3.6)

$$\begin{aligned}\frac{d\mathbf{y}}{dt}(t) &= \mathbf{B}\mathbf{y}(t) + \mathbf{F}(t)\,, \qquad t \in (0, T)\,,\\ \mathbf{y}(0) &= \mathbf{y}_0\,.\end{aligned}$$

We have in mind that our first-order system (3.6) originates from the spatial discretization of the wave equation. We further restrict ourselves in this part to the spatial discretization with continuous FEM. The matrix $\mathbf{B}$ is potentially large and directly applying an exponential method not feasible. If, however the fine region is small compared to the whole computational domain, calculating with an explicit one-step (e.g. RK) or multi-step (e.g. AB) method and a small time-step is too expensive. As an alternative to the LTS-RK presented in Part II schemes we want to adapt (10.5) to make use of the matrix functions γ_j only in the fine part. To do so, we follow the ideas of Part II and split our unknowns into coarse and fine

$$\mathbf{y}(t) = \mathbf{P}\mathbf{y}(t) + (\mathbf{I} - \mathbf{P})\mathbf{y}(t),$$

where $\mathbf{P}$ is the diagonal projector matrix described in Chapter 10.1.

For simplicity we further assume that $\mathbf{F} = 0$. Thus, (3.6) reduces to

$$\mathbf{y}'(t) = \mathbf{B}\mathbf{P}\mathbf{y}(t) + \mathbf{B}(\mathbf{I} - \mathbf{P})\mathbf{y}(t). \tag{10.9}$$

Let $-\mathbf{A} = \mathbf{B}\mathbf{P}$ and $\mathbf{g}(t, \mathbf{y}(t)) = \mathbf{B}(\mathbf{I} - \mathbf{P})\mathbf{y}(t)$. Then, (10.5) is equivalent to

$$\mathbf{y}_{n+1} = e^{\Delta t \mathbf{B}\mathbf{P}}\mathbf{y}_n + \Delta t \sum_{j=0}^{k-1} \gamma_j(\Delta t \mathbf{B}\mathbf{P})\mathbf{B}(\mathbf{I} - \mathbf{P})\nabla^j \mathbf{y}_n. \tag{10.10}$$

To achieve a more efficient implementation (10.10) should be rewritten as

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \gamma_0(\Delta t \mathbf{B}\mathbf{P})\mathbf{B}\mathbf{y}_n + \Delta t \sum_{j=1}^{k-1} \gamma_j(\Delta t \mathbf{B}\mathbf{P})\mathbf{B}(\mathbf{I} - \mathbf{P})\nabla^j \mathbf{y}_n, \tag{10.11}$$

which we can interpret as a corrected exponential Euler step. This reduces the number of matrix function evaluations by one.

Note that for $\mathbf{P} = 0$, i.e. for an equidistant coarse mesh, these methods reduce to the well known AB methods. Therefore the stability constraints are at best those of the underlying AB method on a grid, where no refinement took place.

10.3 EFFICIENT IMPLEMENTATION

Even with the corrected exponential Euler step (10.11) the exponential Adams method requires the multiplication with full matrices as the exponential of a sparse matrix is usually full. We want to exploit our special situation, i.e. the fact that we assume that the region of mesh refinement is small in comparison to the entire domain. Therefore, we rewrite our system

$$\mathbf{y}'(t) = \mathbf{B}\mathbf{y}(t),$$

following the idea proposed in [63] as

$$\begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix}' = \begin{bmatrix} \mathbf{B}_\mathrm{f} & \mathbf{Z} \\ \mathbf{Y} & \mathbf{B}_\mathrm{c} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix}. \tag{10.12}$$

Here we have sorted the unknowns into fine, $\mathbf{v}$, and coarse, $\mathbf{w}$, which we get from $\mathbf{y}$ via permutation, i.e. $\begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} = \mathbf{Q}\mathbf{y}$, where $\mathbf{Q}$ is a permutation matrix with

$$\begin{bmatrix} \mathbf{B}_\mathrm{f} & \mathbf{Z} \\ \mathbf{Y} & \mathbf{B}_\mathrm{c} \end{bmatrix} = \mathbf{Q}\mathbf{B}\mathbf{Q}^\top.$$

Note that $\mathbf{B}_\mathrm{f}$ and $\mathbf{B}_\mathrm{c}$ have the same structure as the original matrix $\mathbf{B}$, but are smaller in dimension as they only correspond either to the fine or the coarse unknowns. In particular, we find that $\mathbf{B}_f$ is invertible.

We can now apply an exponential Adams method (10.5) to (10.12) with

$$-\mathbf{A} = \begin{bmatrix} \mathbf{B}_\mathrm{f} & \mathbf{0} \\ \mathbf{Y} & \mathbf{0} \end{bmatrix}, \qquad \mathbf{g}(t, \mathbf{v}, \mathbf{w}) = \begin{bmatrix} \mathbf{0} & \mathbf{Z} \\ \mathbf{0} & \mathbf{B}_\mathrm{c} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix}. \tag{10.13}$$

This representation allows us to derive an efficient method for time integration based on exponential Adams methods for the splitting into coarse and fine part. To do so, we have to verify the following Proposition, which allows us to easily evaluate the arising matrix functions for splittings of the form (10.13).

Proposition 10.1. *Let ϕ be an analytic function and $\mathbf{A}$ as defined in* (10.13)*. Then*

$$\phi(-\Delta t\mathbf{A}) = \begin{bmatrix} \phi(\Delta t\mathbf{B}_\mathrm{f}) & \mathbf{0} \\ \Delta t\mathbf{Y}\phi^{[1]}(\Delta t\mathbf{B}_\mathrm{f}) & \phi(0)\mathbf{I} \end{bmatrix}, \tag{10.14}$$

where

$$\phi^{[1]}(z) = \frac{\phi(z)-\phi(0)}{z}. \tag{10.15}$$

Proof. If ϕ is an analytic function, we can make use of the series expansion to calculate $\phi(-\Delta t\mathbf{A})$. Let $\phi(x) = \sum_{n=0}^{\infty} a_n x^n$ be the Taylor expansion of ϕ. We can show by induction that

$$\begin{bmatrix} \mathbf{B}_\mathrm{f} & \mathbf{0} \\ \mathbf{Y} & \mathbf{0} \end{bmatrix}^n = \begin{bmatrix} \mathbf{B}_\mathrm{f}^n & 0 \\ \mathbf{Y}\mathbf{B}_\mathrm{f}^{n-1} & 0 \end{bmatrix} \qquad \text{for } n \geq 1. \tag{10.16}$$

Thus we get

$$\begin{aligned} \phi(-\Delta t\mathbf{A}) &= \sum_{n=0}^{\infty} a_n \left(\Delta t \begin{bmatrix} \mathbf{B}_\mathrm{f} & \mathbf{0} \\ \mathbf{Y} & \mathbf{0} \end{bmatrix}\right)^n = a_0 \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} + \sum_{n=1}^{\infty} a_n \Delta t^n \begin{bmatrix} \mathbf{B}_f & \mathbf{0} \\ \mathbf{Y} & \mathbf{0} \end{bmatrix}^n \\ &= \begin{bmatrix} \phi(\Delta t\mathbf{B}_\mathrm{f}) & \mathbf{0} \\ \alpha & \phi(0)\mathbf{I} \end{bmatrix}, \end{aligned}$$

with $a_0 = \phi(0)$. We now need to determine α. With (10.16) we find that

$$\alpha = \sum_{n=1}^{\infty} a_n \Delta t^n \mathbf{Y}\mathbf{B}_\mathrm{f}^{n-1} \tag{10.17}$$

or equivalently

$$\alpha = \mathbf{Y}\left(\sum_{n=1}^{\infty} a_n \Delta t^n \mathbf{B}_\mathrm{f}^n \mathbf{B}_\mathrm{f}^{-1} + a_0 \mathbf{B}_\mathrm{f}^{-1} - a_0 \mathbf{B}_\mathrm{f}^{-1}\right). \tag{10.18}$$

We thus get

$$\alpha = \mathbf{Y}\Delta t(\phi(\Delta t\mathbf{B}_\mathrm{f}) - \phi(0)\mathbf{I})\,(\Delta t\mathbf{B}_\mathrm{f})^{-1}. \tag{10.19}$$

Now (10.14) follows from (10.19) with (10.15). □

By remarking that $\begin{bmatrix} \mathbf{B}_\mathrm{f} & \mathbf{0} \\ \mathbf{Y} & \mathbf{0} \end{bmatrix}$ corresponds to $\mathbf{BP}$ and $\begin{bmatrix} \mathbf{0} & \mathbf{Z} \\ \mathbf{0} & \mathbf{B}_\mathrm{c} \end{bmatrix}$ to $\mathbf{B}(\mathbf{I}-\mathbf{P})$ we thus get the following k-step method from (10.11)

$$\begin{aligned} \begin{bmatrix} \mathbf{v}_{n+1} \\ \mathbf{w}_{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_n \\ \mathbf{w}_n \end{bmatrix} &+ \Delta t \begin{bmatrix} \gamma_0(\Delta t\mathbf{B}_\mathrm{f}) & \mathbf{0} \\ \Delta t\mathbf{Y}\gamma_0^{[1]}(\Delta t\mathbf{B}_\mathrm{f}) & \gamma_0(0)\mathbf{I} \end{bmatrix} \left(\begin{bmatrix} \mathbf{B}_\mathrm{f} & \mathbf{Z} \\ \mathbf{Y} & \mathbf{B}_\mathrm{c} \end{bmatrix} \begin{bmatrix} \mathbf{v}_n \\ \mathbf{w}_n \end{bmatrix} \right) \\ &+ \Delta t \sum_{j=1}^{k-1} \begin{bmatrix} \gamma_j(\Delta t\mathbf{B}_\mathrm{f}) & \mathbf{0} \\ \Delta t\mathbf{Y}\gamma_j^{[1]}(\Delta t\mathbf{B}_\mathrm{f}) & \gamma_j(0)\mathbf{I} \end{bmatrix} \nabla^j \begin{bmatrix} \mathbf{Z}\mathbf{w}_n \\ \mathbf{B}_\mathrm{c}\mathbf{w}_n \end{bmatrix}, \end{aligned}$$

which we can restate as

$$\begin{aligned}
\mathbf{v}_{n+1} &= \mathbf{v}_n + \Delta t \gamma_0(\Delta t \mathbf{B}_\mathrm{f}) \left(\mathbf{B}_\mathrm{f}\mathbf{v}_n + \mathbf{Z}\mathbf{w}_n\right) + \Delta t \sum_{j=1}^{k-1} \gamma_j(\Delta t \mathbf{B}_\mathrm{f}) \nabla^j \mathbf{Z}\mathbf{w}_n, \\
\mathbf{w}_{n+1} &= \mathbf{w}_n + \Delta t \mathbf{Y} \left(\gamma_0(\Delta t \mathbf{B}_\mathrm{f})\mathbf{v}_n + \Delta t \gamma_1(\Delta t \mathbf{B}_\mathrm{f}) \mathbf{Z}\mathbf{w}_n + \Delta t \sum_{j=1}^{k-1} \gamma_j^{[1]}(\Delta t \mathbf{B}_\mathrm{f}) \nabla^j \mathbf{Z}\mathbf{w}_n \right) \\
&\qquad + \Delta t \sum_{j=0}^{k-1} \gamma_j(0) \nabla^j \mathbf{B}_\mathrm{c} \mathbf{w}_n.
\end{aligned} \tag{10.20}$$

If the refined region in our mesh is small in comparison to the coarse part, the arising matrix functions $\gamma_i(\Delta t \mathbf{B}_\mathrm{f})$ will be small in comparison to the exponential of the whole matrix **BP**. We still have to multiply with full matrices in (10.20), however of much smaller dimensions. Thus we can reduce the effort noticeably by the described approach. We are going to call (10.20) *local exponential Adams method* denoted by LexpABk.

11

NUMERICAL EXPERIMENTS

After deriving LexpAB schemes based on kth order AB methods in Chapter 10 we now show the results of numerical experiments done for the damped wave equation (3.1) in one and two dimensions. First we consider problems in 1D that confirm the expected order of convergence for different refinement factors p. Afterwards we simulate a plane wave and its reflections in a two dimensional domain with a barrier. We then present performance results where we compare LexpABk with their classical AB counterpart on a mesh in 1D with a small fixed number of fine degrees of freedom, but with an increasing rate of refinement p. Finally, we show why for locally refined meshes where the number of fine gridpoints is not very small compared to the whole computational domain, exponential integrators seem to be an unfortunate choice for the simulation of acoustic waves as Krylov methods fail to give an accurate approximation of the involved matrix function times vector.

11.1 CONVERGENCE STUDY

To study the convergence rates of LexpAB schemes we consider (3.1) in 1D on the interval $\Omega = [0,4]$ for $t \in [0,T]$, where $T = 1.5$, with constant wave speed $c = 1$, damping factor $\sigma = 0.1$ and homogeneous Dirichlet boundary conditions. The initial conditions are set to yield the exact solution

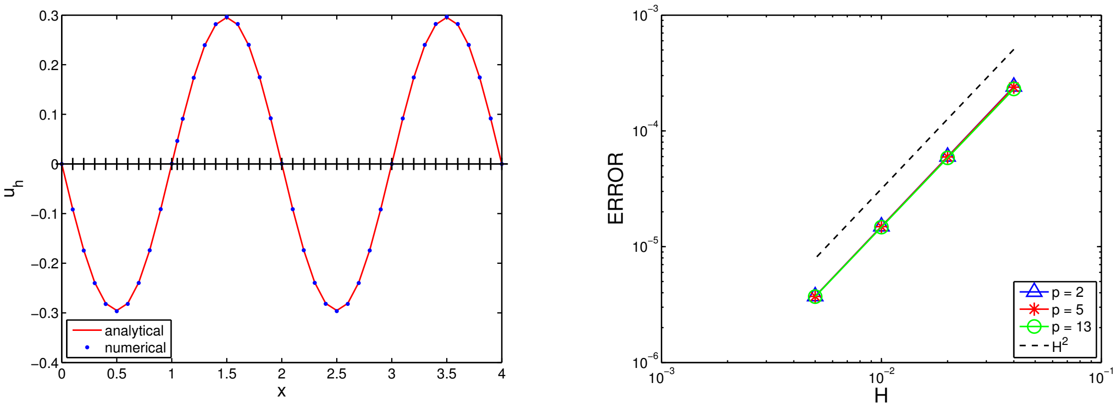
$$u(x,t) = \frac{2e^{-\frac{\sigma t}{2}}}{\sqrt{4\pi^2 - \sigma^2}} \sin(\pi x) \sin\left(\frac{t}{2}\sqrt{4\pi^2 - \sigma^2}\right).$$

First Ω is discretized with an equidistant mesh of size h^{coarse}. Then we refine the interval $[1, 1 + h^{\text{coarse}}]$ by a factor $p = 2$, 5 or 13, i.e. the size of the mesh in the fine part is $h^{\text{fine}} = h^{\text{coarse}}/p$. Hence the two intervals $[0,1)$ and $(1 + h^{\text{coarse}}, 4]$ correspond to the coarse region Ω_c whereas $\Omega_f = [1, 1 + h^{\text{coarse}}]$.

First we discretize in space using a continuous $\mathcal{P}^1$-FE discretization of (3.1) with mass lumping on a sequence of increasingly finer meshes . We combined this spatial discretization with the second-order LexpAB2 scheme, which for $k = 2$ reduces to

$$\mathbf{y}_{n+1} = e^{\Delta t \mathbf{BP}} \mathbf{y}_n + \Delta t \big(\gamma_0(\Delta t \mathbf{BP}) \mathbf{B}(\mathbf{I}-\mathbf{P}) \mathbf{y}_n + \gamma_1(\Delta t \mathbf{BP}) \mathbf{B}(\mathbf{I}-\mathbf{P})(\mathbf{y}_n - \mathbf{y}_{n-1}) \big),$$

with γ_0 and γ_1 as calculated in Section 10.1.
As we systematically reduce the global mesh size $H = h^{\text{coarse}}$, while simultaneously reducing Δt, we monitor the L^2 space-time error in the numerical solution $\|u(\cdot, T) - u_{h,\Delta t}(\cdot, T)\|_{L^2(\Omega)}$ at the final time $T = 1.5$.



(a) solution at time $T = 1.5$

(b) L^2-error

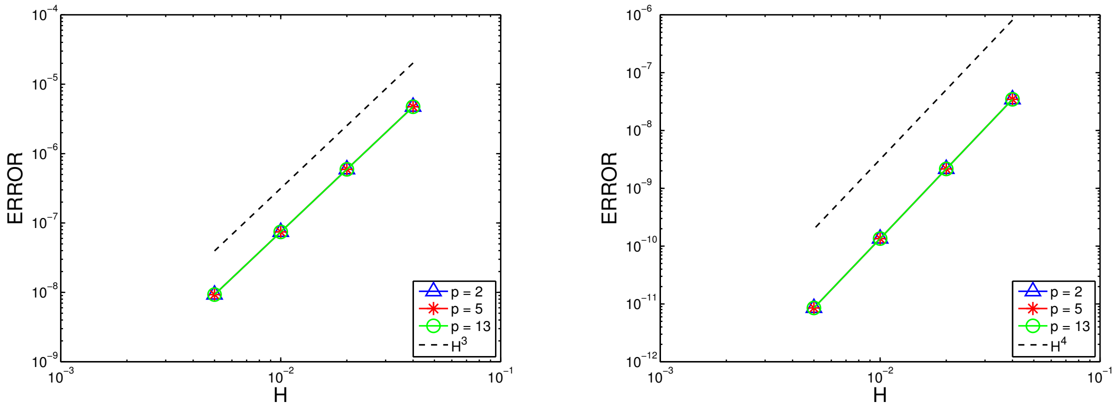
Figure 16: Exact and numerical solution at time $T = 1.5$ (a) and Lexp-AB2(p) error vs. $H = h^{\text{coarse}} = 0.04, 0.02, 0.01, 0.005$ for $\mathcal{P}^1$ finite elements with $p = 2, 5, 13$(b).

In Fig. 16 we display the numerical solution for $h^{\text{coarse}} = 0.1$ and $p = 2$ (a) and the L^2-errors at time $T = 1.5$ for $h^{\text{coarse}} = 0.04, 0.02, 0.01, 0.005$ and different local refinement factors p (b). The numerical results thus confirm the expected second-order accuracy independently of the refinement factor p.
Next, we repeat this numerical experiment with a continuous $\mathcal{P}^2$-FE discretization with mass lumping. In Fig. 17 (a) we can observe that the LexpAB3 method

$$\mathbf{y}_{n+1} = e^{\Delta t \mathbf{BP}} \mathbf{y}_n + \Delta t \sum_{j=0}^{2} \gamma_j(\Delta t \mathbf{BP}) \mathbf{B}(\mathbf{I}-\mathbf{P}) \nabla^j \mathbf{y}_n,$$

yields overall third-order convergence independently of p.
Finally we combine a continuous $\mathcal{P}^3$-FE discretization with mass lumping with a fourth-order LexpAB4 scheme

$$\mathbf{y}_{n+1} = e^{\Delta t \mathbf{BP}} \mathbf{y}_n + \Delta t \sum_{j=0}^{3} \gamma_j(\Delta t \mathbf{BP}) \mathbf{B}(\mathbf{I}-\mathbf{P}) \nabla^j \mathbf{y}_n,$$



(a) expAB3

(b) expAB4

Figure 17: Lexp-AB$k(p)$ error vs. $H = h^{\text{coarse}} = 0.04, 0.02, 0.01, 0.005$ for $\mathcal{P}^{k-1}$ finite elements with $p = 2,\ 5, 13$ for $k = 3$ (a) and $k = 4$ (b).

and monitor the L^2 space-time error. Again, the numerical results confirm the expected fourth-order convergence for varying p – see Figure 17 (b).

The schemes in (10.10) are clearly multistep methods. We calculate the necessary starting values with a classical RK4 method.

11.2 STABILITY

In this section the stability of our methods (10.10) applied to the model problem (3.1) will be discussed. Again we consider the case where $\Omega = [0, 4]$, $T = 1.5$, with constant wave speed $c = 1$, damping factor $\sigma = 0.1$ and homogeneous Dirichlet boundary conditions. As the exponential Adams methods reduce to Adams-Bashforth methods for $p = 1$, i.e. in the case of an equidistant coarse mesh, we have to at least fulfill the stability constraints arising from those underlying methods. Therefore we rewrite our k-step Adams-Bashforth method into a one-step method and choose Δt_{ABk} to be the largest Δt such that the eigenvalues of the matrix arising from the transformation have an absolute value strictly less than 1.

We want to demonstrate that the locally refined mesh imposes no further stability constraint, i.e. the exponential Adams-methods are stable for $\Delta t = \Delta t_{\text{ABk}}$. To do so we likewise transform our local exponential k-step method into an one-step method and examine the eigenvalues of the matrix arising.

In the following experiments we choose $h^{\text{coarse}} = 0.1$ and $p = 13$. Instead of refining only on one element, we consider now refinement on the intervall $[1, 2]$.

For $k = 2$ we first consider the underlying AB2 method

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \left(\frac{3}{2} \mathbf{B} \mathbf{y}_n - \frac{1}{2} \mathbf{B} \mathbf{y}_{n-1} \right). \tag{11.1}$$

We proceed by rewriting (11.1) as a one-step method

$$\begin{bmatrix} \mathbf{y}_{n+1} \\ \mathbf{y}_n \end{bmatrix} = \mathbf{C}_{AB2} \begin{bmatrix} \mathbf{y}_n \\ \mathbf{y}_{n-1} \end{bmatrix}, \tag{11.2}$$

with

$$\mathbf{C}_{AB2} = \begin{bmatrix} \mathbf{I} + \Delta t \frac{3}{2} \mathbf{B} & -\Delta t \frac{1}{2} \mathbf{B} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}$$

To guarantee stability we have to ensure that all the eigenvalues of $\mathbf{C}_{AB2}$ are in absolute value strictly less than 1. Δt_{AB2} is the maximum Δt such that this is fulfilled.

In Table 7 we can see Δt_{AB2} for different σ. An increasing σ seems to have a positive effect on the stability behavior as Δt_{AB2} gets larger as σ gets bigger.

σ	0.1	1	20
Δt_{opt}	0.01065	0.022	0.0454

Table 7: Δt_{AB2} for linear FE combined with AB2 on equidistant coarse mesh for different σ.

For the discretization in time on the locally refined mesh the exponential Adams method is the following

$$\mathbf{y}_{n+1} = e^{\Delta t \mathbf{BP}} \mathbf{y}_n + \Delta t \left(\gamma_0(\Delta t \mathbf{BP}) \mathbf{B}(\mathbf{I} - \mathbf{P}) \mathbf{y}_n + \gamma_1(\Delta t \mathbf{BP}) \mathbf{B}(\mathbf{I} - \mathbf{P})(\mathbf{y}_n - \mathbf{y}_{n-1}) \right),$$

which can again be written as a one-step method

$$\begin{bmatrix} \mathbf{y}_{n+1} \\ \mathbf{y}_n \end{bmatrix} = \mathbf{C}_{LexpAB2} \begin{bmatrix} \mathbf{y}_n \\ \mathbf{y}_{n-1} \end{bmatrix},$$

with

$$\mathbf{C}_{LexpAB2} = \begin{bmatrix} \alpha_1 & \alpha_2 \\ \mathbf{I} & \mathbf{0} \end{bmatrix},$$

where α_1 and α_2 are given by

$$\begin{aligned} \alpha_1 &= e^{\Delta t \mathbf{BP}} + \Delta t \left(\gamma_0(\Delta t \mathbf{BP}) \mathbf{B}(\mathbf{I} - \mathbf{P}) + \gamma_1(\Delta t \mathbf{BP}) \mathbf{B}(\mathbf{I} - \mathbf{P}) \right), \\ \alpha_2 &= -\Delta t \gamma_1(\Delta t \mathbf{BP}) \mathbf{B}(\mathbf{I} - \mathbf{P}). \end{aligned}$$

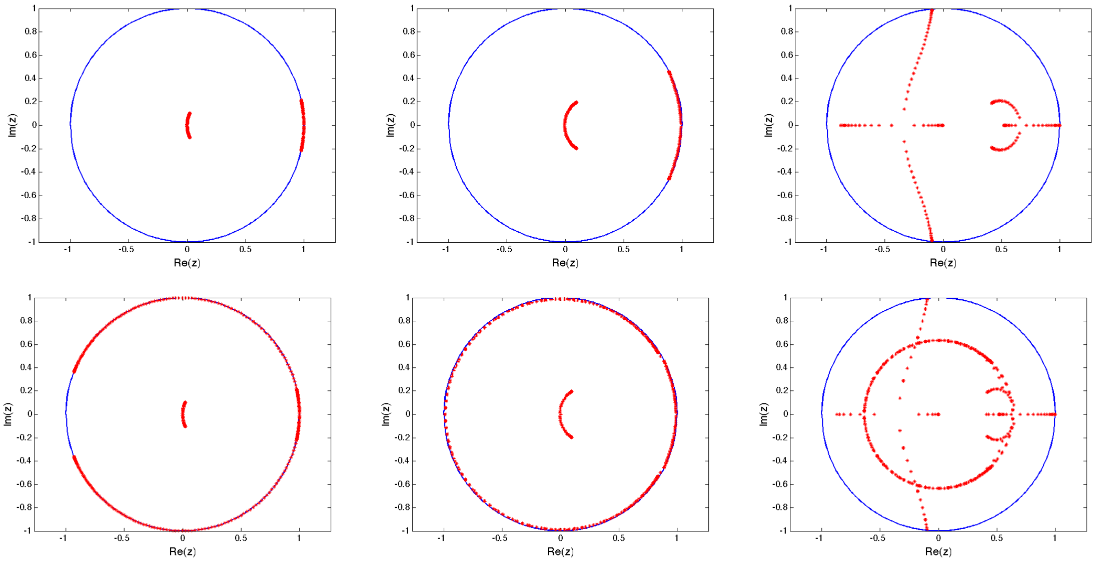


Figure 18: Eigenvalues of $\mathbf{C}_{AB2}$ (top) and $\mathbf{C}_{LexpAB2}$ (bottom) for $\sigma = 0.1$ (left), 1 (middle) and 20 (right) and $\Delta t = \Delta t_{\text{opt}}$.

In Fig. 18 we can see that even with a refinement (in this numerical example $p = 13$) Δt_{AB2} is small enough to ensure that our method for time integration is stable, i.e. all eigenvalues are still inside the unit circle. This is further confirmed by Fig. 19 where we observe, that the spectral radius $\rho(\mathbf{C}_{AB2})$ is below 1 for all $\Delta t \leq \Delta t_{\text{AB2}}$.

We repeat this approach for the case of continuous $\mathcal{P}^2$ FEM in combination with the third-order LexpAB3 scheme. Starting from the well-known AB3 method

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \left(\frac{23}{12}\mathbf{B}\mathbf{y}_n - \frac{16}{12}\mathbf{B}\mathbf{y}_{n-1} + \frac{5}{12}\mathbf{B}\mathbf{y}_{n-2} \right), \tag{11.3}$$

we rewrite (11.3) as a one-step method

$$\begin{bmatrix} \mathbf{y}_{n+1} \\ \mathbf{y}_n \\ \mathbf{y}_{n-1} \end{bmatrix} = \mathbf{C}_{AB3} \begin{bmatrix} \mathbf{y}_n \\ \mathbf{y}_{n-1} \\ \mathbf{y}_{n-2} \end{bmatrix}, \tag{11.4}$$

in order to determine the optimal Δt.

Here the matrix $\mathbf{C}_{AB3}$ is given by

$$\mathbf{C}_{AB3} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix}$$

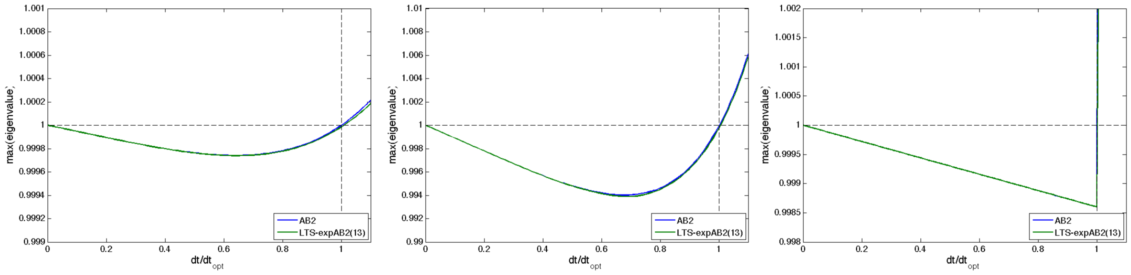


Figure 19: Maximal norm of the eigenvalues of $\mathbf{C}_{AB2}$ (top) and $\mathbf{C}_{LexpAB2}$ respectively with respect to Δt for $\sigma = 0.1$ (left), 1 (middle) and 20 (right)

with $\alpha_1 = \mathbf{I} + \Delta t \frac{23}{12}\mathbf{B}$, $\alpha_2 = -\Delta t \frac{16}{12}\mathbf{B}$ and $\alpha_3 = \Delta t \frac{5}{12}\mathbf{B}$.
In Table 8 we display Δt_{AB3} for different σ. This time we cannot observe the positive effect of σ on the stability behavior seen for AB2.

σ	0.1	1	20
Δt_{AB3}	0.0147	0.0146	0.0135

Table 8: Δt_{AB3} for quadratic FE combined with AB3 on equidistant coarse mesh for different σ.

Rewriting the 3-step exponential Adams method

$$\begin{aligned}\mathbf{y}_{n+1} = e^{\Delta t \mathbf{BP}}\mathbf{y}_n + \Delta t\big(&\gamma_0(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P})\mathbf{y}_n + \gamma_1(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P})(\mathbf{y}_n - \mathbf{y}_{n-1})\\ &+ \gamma_2(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P})(\mathbf{y}_n - 2\mathbf{y}_{n-1} + \mathbf{y}_{n-2})\big)\end{aligned} \tag{11.5}$$

into a one-step method leads to

$$\begin{bmatrix}\mathbf{y}_{n+1}\\ \mathbf{y}_n\\ \mathbf{y}_{n-1}\end{bmatrix} = \mathbf{C}_{LexpAB3}\begin{bmatrix}\mathbf{y}_n\\ \mathbf{y}_{n-1}\\ \mathbf{y}_{n-2}\end{bmatrix},$$

with

$$\mathbf{C}_{LexpAB3} = \begin{bmatrix}\tilde{\alpha}_1 & \tilde{\alpha}_2 & \tilde{\alpha}_3\\ \mathbf{I} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{I} & \mathbf{0}\end{bmatrix}$$

where

$$\begin{aligned}\tilde{\alpha}_1 &= e^{\Delta t\mathbf{BP}} + \Delta t\left(\gamma_0(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P}) + \gamma_1(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P}) + \gamma_2(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P})\right),\\ \tilde{\alpha}_2 &= -\Delta t\left(\gamma_1(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P}) + 2\gamma_2(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P})\right),\\ \tilde{\alpha}_3 &= \Delta t\left(\gamma_2(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P})\right).\end{aligned}$$

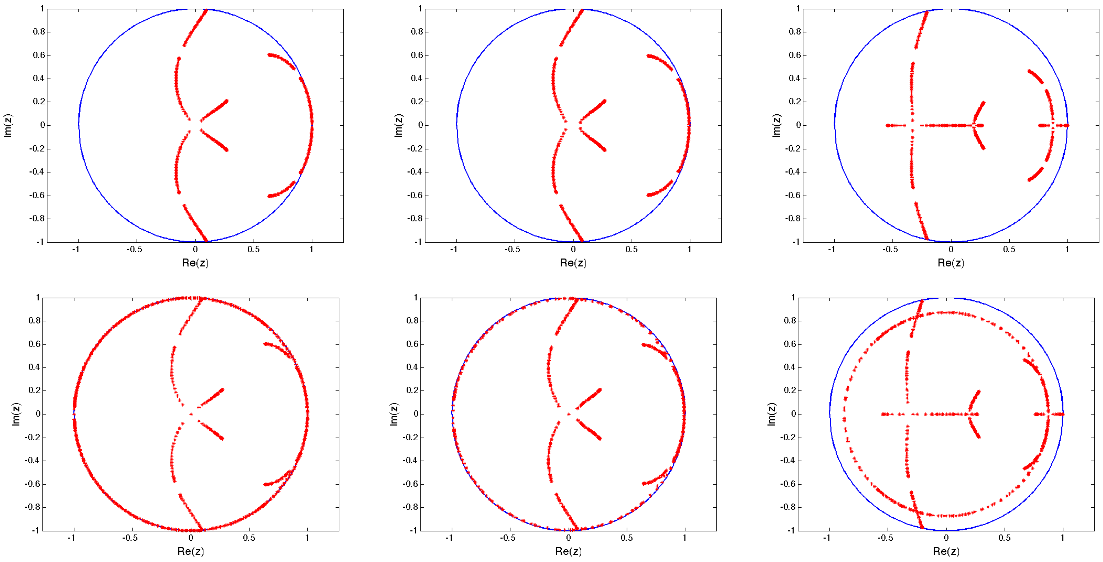


Figure 20: Eigenvalues of $\mathbf{C}_{AB3}$ (top) and $\mathbf{C}_{LexpAB3}$ (bottom) for $\sigma = 0.1$ (left), 1 (middle) and 20 (right) and $\Delta t = \Delta t_{\text{opt}}$.

A similar behavior as in the case of linear finite elements can be seen in Fig. 20 and 21. The choice of $\Delta t \leq \Delta t_{\text{AB3}}$ guarantees that the eigenvalues of both matrices have an absolute value strictly less than 1. For a larger damping factor σ, eigenvalues close to the unit circle seem to be moving inwards.

If we consider cubic finite elements the exponential Adams method, required to guarantee fourth order convergence, reduces to a 4-step Adams-Bashforth method in the case of an equidistant coarse mesh

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \left(\frac{55}{24}\mathbf{B}\mathbf{y}_n - \frac{59}{24}\mathbf{B}\mathbf{y}_{n-1} + \frac{37}{24}\mathbf{B}\mathbf{y}_{n-2} - \frac{3}{8}\mathbf{B}\mathbf{y}_{n-3} \right), \tag{11.6}$$

which can again be restated as a one-step method

$$\begin{bmatrix} \mathbf{y}_{n+1} \\ \mathbf{y}_n \\ \mathbf{y}_{n-1} \\ \mathbf{y}_{n-2} \end{bmatrix} = \mathbf{C}_{AB4} \begin{bmatrix} \mathbf{y}_n \\ \mathbf{y}_{n-1} \\ \mathbf{y}_{n-2} \\ \mathbf{y}_{n-3} \end{bmatrix}.$$

where

$$\mathbf{C}_{AB4} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix}$$

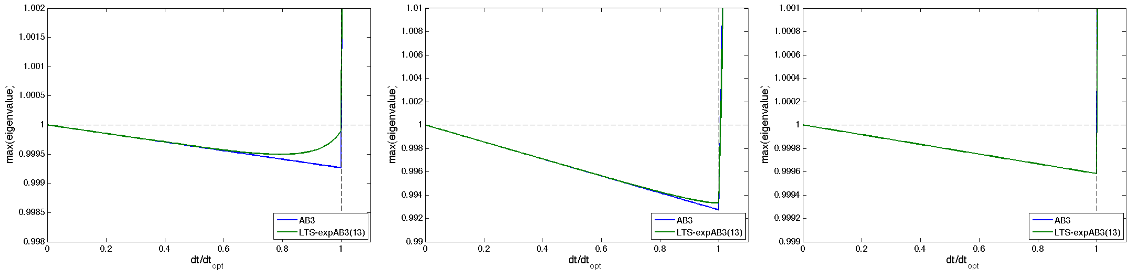


Figure 21: Maximal norm of the eigenvalues of $\mathbf{C}_{AB3}$ and $\mathbf{C}_{LexpAB3}$ respectively for $\sigma = 0.1$ (left), 1 (middle) and 20 (right)

Here $\alpha_1 = \mathbf{I} + \Delta t \frac{55}{24}\mathbf{B}$, $\alpha_2 = -\Delta t \frac{59}{24}\mathbf{B}$, $\alpha_3 = \Delta t \frac{37}{24}\mathbf{B}$ and $\alpha_4 = -\Delta t \frac{3}{8}\mathbf{B}$.
We display the optimal Δt for different σ in Table 9.

σ	0.1	1	20
Δt_{AB4}	0.00498	0.00495	0.00468

Table 9: Δt_{AB4} for cubic FE combined with AB4 on equidistant coarse mesh for different σ.

As done before for linear and quadratic elements we transform the corresponding exponential Adams method

$$\begin{aligned}\mathbf{y}_{n+1} = e^{\Delta t \mathbf{BP}}\mathbf{y}_n + \Delta t \Big(& \gamma_0(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P})\mathbf{y}_n + \gamma_1(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P})(\mathbf{y}_n - \mathbf{y}_{n-1}) \\ & + \gamma_2(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P})(\mathbf{y}_n - 2\mathbf{y}_{n-1} + \mathbf{y}_{n-2}) \\ & + \gamma_3(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P})(\mathbf{y}_n - 3\mathbf{y}_{n-1} + 3\mathbf{y}_{n-2} - \mathbf{y}_{n-3})\Big)\end{aligned}$$

into a one-step method

$$\begin{bmatrix}\mathbf{y}_{n+1}\\ \mathbf{y}_n\\ \mathbf{y}_{n-1}\\ \mathbf{y}y_{n-2}\end{bmatrix} = \mathbf{C}_{LexpAB4}\begin{bmatrix}\mathbf{y}_n\\ \mathbf{y}_{n-1}\\ \mathbf{y}_{n-2}\\ \mathbf{y}_{n-3}\end{bmatrix},$$

where

$$\mathbf{C}_{LexpAB4} = \begin{bmatrix}\tilde{\alpha}_1 & \tilde{\alpha}_2 & \tilde{\alpha}_3 & \tilde{\alpha_4}\\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0}\end{bmatrix}$$

with

$$\begin{aligned}
\tilde{\alpha}_1 &= e^{\Delta t\mathbf{BP}} + \Delta t\Big(\gamma_0(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P}) + \gamma_1(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P}) + \gamma_2(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P}) \\
&\qquad + \gamma_3(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P})\Big), \\
\tilde{\alpha}_2 &= -\Delta t\Big(\gamma_1(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P}) + 2\gamma_2(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P}) + 3\gamma_3(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P})\Big), \\
\tilde{\alpha}_3 &= \Delta t\Big(\gamma_2(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P}) + 3\gamma_3(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P})\Big), \\
\tilde{\alpha}_4 &= -\Delta t\gamma_3(\Delta t\mathbf{BP})\mathbf{B}(\mathbf{I}-\mathbf{P}).
\end{aligned}$$

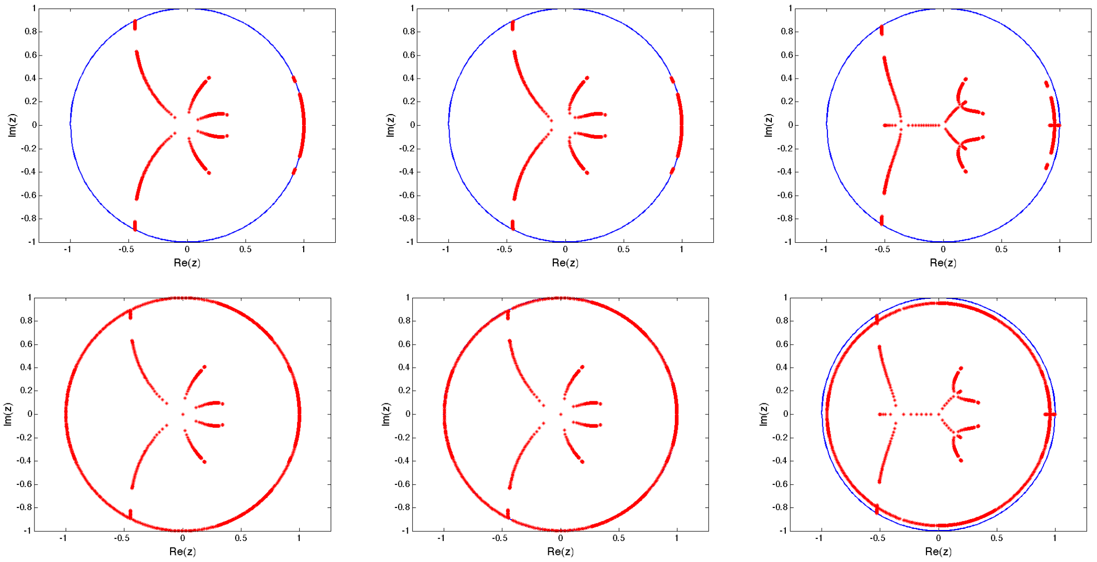


Figure 22: Eigenvalues of $\mathbf{C}_{AB4}$ (top) and $\mathbf{C}_{LexpAB4}$ (bottom) for $\sigma = 0.1$ (left), 1 (middle) and 20 (right) and $\Delta t = \Delta t_{\text{opt}}$.

Figure 22 shows us the eigenvalues of $\mathbf{C}_{AB4}$ and $\mathbf{C}_{\exp 4}$ for different σ's, $\Delta t = \Delta t_{\text{AB4}}$ and $p = 13$. They all have an absolute value less than 1. Thus the exponential 4-step Adams method is stable for Δt_{AB4}. This is confirmed by Fig. 23, where we can observe that the biggest eigenvalue in absolute value is below 1 for $\Delta t \leq \Delta t_{\text{AB4}}$.

11.3 TWO-DIMENSIONAL EXAMPLE

If the fine part of the mesh is small, LexpAB*k* schemes present a efficient alternative for solving the damped wave equation even in two space dimensions.

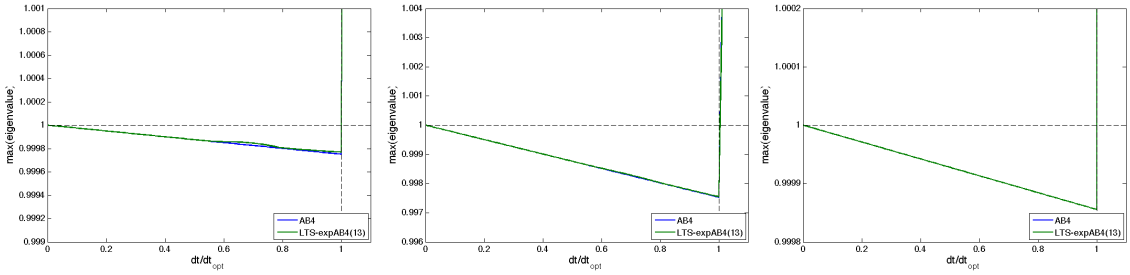


Figure 23: Maximal norm of the eigenvalues of $\mathbf{C}_{AB4}$ (top) and $\mathbf{C}_{LexpAB4}$ respectively for $\sigma = 0.1$ (left), 1 (middle) and 20 (right)

We consider here the model problem (3.1) on a domain Ω, rectangular of size $(0,2) \times (0,1)$ with two rectangular barriers separated by a narrow gap seen in Chapter 8. As shown in Fig. 24, the mesh resolves the small geometric features of the narrow gap. Notice that we choose the fine region here much smaller than in the case of LTS presented before. We want our mesh to have only a few fine elements in order for LexpAB to perform well.

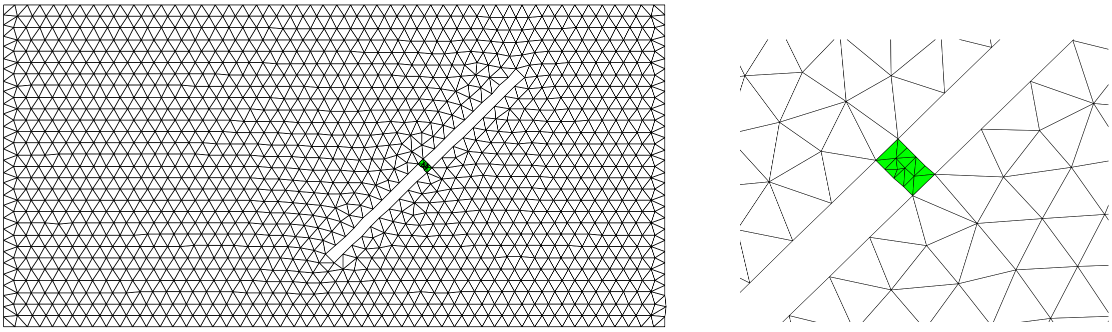


Figure 24: Initial triangular mesh with local refinement ratio $p = 6$, i.e. $h^{\text{fine}} \approx h^{\text{coarse}}/6$

At all boundaries we impose homogeneous Neumann boundary conditions; a vertical plane wave is excited through the initial conditions,

$$u(\mathbf{x}, 0) = \exp\left(-\frac{(x-0.8)^2}{0.0072^2}\right),$$

and $v(\mathbf{x}, 0) = u_t(\mathbf{x}, 0) = 0$. In space, we discretize using $\mathcal{P}^2$ continuous finite elements with mass lumping after having refined our initial mesh (see Fig. 24) three times, each time dividing every triangle into four.

As the initial mesh is unstructured, we have to decide which elements belong to the fine mesh, and which do not. Given h^{coarse} as the maximal radius of all triangles, the fine mesh consists of all triangles with $h < 0.7h^{\text{coarse}}$. For the time integration we choose the third-order scheme LexpAB3 with $\Delta t = 0.07h^{\text{coarse}}$ determined experimentally. Thus the numerical method is third order accurate in both space and time. We calculate the starting values by applying a standard fourth-order Runge-Kutta method. In Fig. 25, we follow the plane wave as it impinges on the obstacle and reflections occur, while the gap sheds circular waves into the medium.

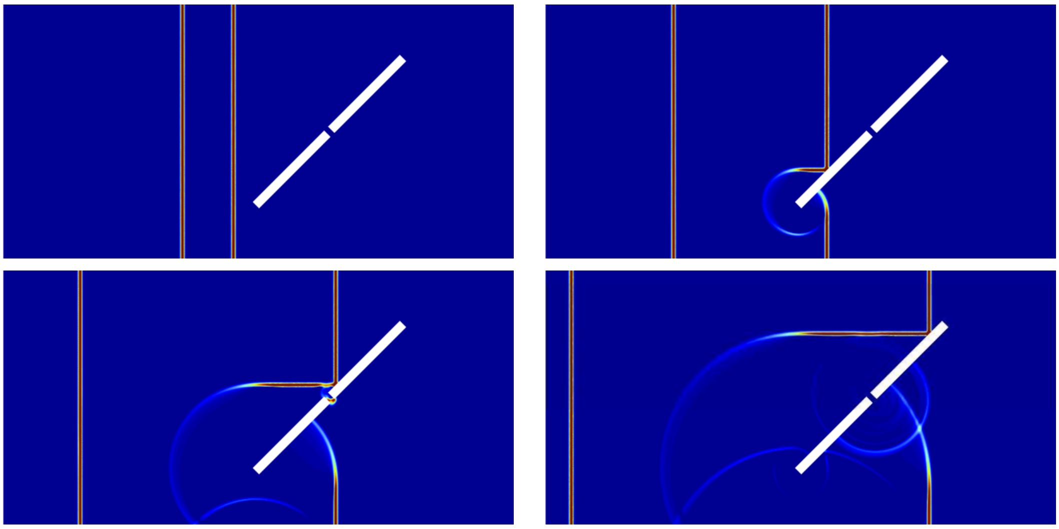


Figure 25: Numerical solution at times $t = 0.1$, 0.3, 0.5 and 0.7

11.4 COMPARISON

In this section we illustrate the performance of LexpAB methods compared to their underlying AB schemes in different geometrical situations. We want to show that for a fixed small number of fine elements, which are characterized by a very high ratio p between h^{coarse} and h^{fine}, LexpAB schemes outperform classical AB methods as the computational work only depends on the number of fine degrees of freedom, and not on p.

To do so we consider the damped wave equation (3.1) in one dimension for $\Omega = (0, 6)$, $c = 1$, $\sigma = 0.1$, $T = 5$ and $f(x) = 0$.

We start with an equidistant mesh with mesh size h^{coarse} and add $\mu = 1$ additional fine point at $h^{\text{fine}} = h^{\text{coarse}}/p$ – see Figure 26. We discretize (3.1) in space using continuous $\mathcal{P}^2$-FE with mass lumping.

We monitor the L^2-error at the final time $T = 5$ and the runtime in seconds for LexpAB3 and compare it to the results of AB3 on the same mesh with $\Delta t = \Delta t_{\text{AB3}}/p$ for different p.

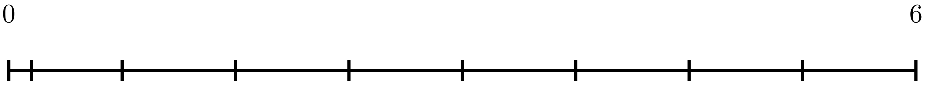


Figure 26: Ω with one fine element.

In a first test we set $h^{\text{coarse}} = 0.2$. The optimal Δt on the equidistant coarse mesh is here $\Delta t_{AB3} = 2.95e - 02$.

	LexpAB3		AB3	
p	L^2- Error	Runtime in s	L^2- Error	Runtime in s
5	$3.3845e-04$	0.476365	$3.8590e-04$	0.115519
11	$1.3254e-04$	0.477085	$3.8230e-04$	0.156068
53	$2.0152e-04$	0.473089	$3.7306e-04$	0.423811
100	$6.2434e-05$	0.481397	$3.7151e-04$	0.744742
1000	$1.4716e-04$	0.482522	$3.6978e-04$	6.659287
2000	$1.0888e-04$	0.479547	$3.6968e-04$	13.020158

Table 10: Runtime results for $\mu = 1$ and $h^{\text{coarse}} = 0.2$ for different p

In Table 10 we can observe the results for different p. Notice that the runtime for LexpAB3 is constant, i.e. only depends on the size of the matrix $\mathbf{B}_{\text{f}}$ and not on p, while the runtime for AB3 grows linearly with increasing p.

In a second experiment we reduce the mesh size to $h^{\text{coarse}} = 0.05$. Thus, the size of the first-order system (3.6) increases. The optimal Δt on the equidistant coarse mesh is now $\Delta t_{\text{AB3}} = 7.39e - 03$.

The results displayed in Table 11 confirm our results for the coarser mesh. The ratio between runtime of AB3 and runtime of corresponding LexpAB3 even increases for growing p.

We repeat the experiments above but increase the number of fine points at the beginning of the domain to $\mu = 3$ – see Figure 27 for the new grid.

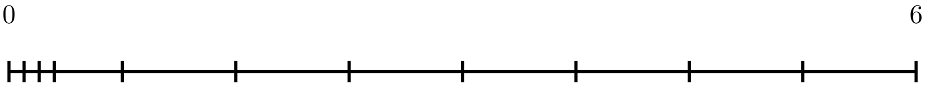


Figure 27: Ω with three fine element.

Again, we first consider the case with $h^{\text{coarse}} = 0.2$.

In Table 12, the results corroborate those in Table 10, i.e. the additional points do not change the outcome.

	LTS-expAB3		AB3	
p	L^2- Error	Runtime in s	L^2- Error	Runtime in s
5	$2.8130e-06$	1.009038	$1.4138e-06$	0.262177
11	$1.5893e-05$	1.010626	$1.3219e-06$	0.450336
53	$8.7525e-06$	1.005432	$1.4236e-06$	1.775844
100	$1.4042e-05$	1.020723	$1.4292e-06$	3.315734
1000	$1.8929e-05$	1.016705	$1.4272e-06$	31.811840
2000	$4.8814e-07$	1.017659	$1.4270e-06$	63.661842

Table 11: Runtime results for $\mu = 1$ and $h^{\text{coarse}} = 0.05$ for different p

Last we consider $h^{\text{coarse}} = 0.05$. The optimal Δt on the equidistant coarse mesh is again $\Delta t_{\text{AB3}} = 7.39e-03$.

Again the results in Table 13 confirm the results seen before. We note that p has to be high for LexpAB3 to actually beat AB3. When we further increase the number of fine points, the computation of the matrix functions gets more expensive and we thus have to choose a higher p to still beat AB3.

11.5 KRYLOV-METHODS

In Section 11.4 we presented experiments which suggest that in presence of a local mesh refinement of only a few, but very small elements LexpABk schemes appear to be an interesting alternative to LTS schemes. However, if the number of fine degrees of freedom increases, and thus the dimension of $\mathbf{B}_f$, directly evaluating the arising matrix functions becomes inefficient. Here we explain why standard Krylov method fail to be an alternative to direct methods when dealing with the damped wave equation.

We start by considering the damped wave equation (3.1) in one dimension for $\Omega = (0, L)$, $\sigma(x) \geq 0$ and $c(x)$ is piecewise smooth and strictly positive. We discretize (3.1) in space by using standard finite elements while leaving time continuous and thus get a system of ODEs of the form (3.6). For simplicity, we further assume that $F = 0$. Hence, (3.6) reduces to

$$\mathbf{y}'(t) = \mathbf{B}\mathbf{y}(t) \qquad \mathbf{y}(0) = \mathbf{y}_0. \tag{11.7}$$

	LexpAB3		AB3	
p	L^2- Error	Runtime in s	L^2- Error	Runtime in s
5	$3.8811e-04$	0.495881	$3.9625e-04$	0.113812
11	$1.5967e-04$	0.490194	$3.9759e-04$	0.157157
53	$1.9158e-04$	0.508014	$3.7946e-04$	0.420513
100	$2.0223e-04$	0.491436	$3.7518e-04$	0.747162
1000	$1.5863e-04$	0.496777	$3.7018e-04$	6.562576
2000	$1.8596e-04$	0.493427	$3.6988e-04$	12.929323

Table 12: Runtime results for $\mu = 3$ and $h^{\text{coarse}} = 0.2$ for different p

The exact solution of (11.7) can be written as

$$\mathbf{y}(t) = e^{t\mathbf{B}}\mathbf{y}_0. \tag{11.8}$$

If $\mathbf{B}$ is small, we can use the MATLAB function `expm.m`, which makes use of the Padé approximation for calculating the exponential of a matrix (see [58] for details). As soon as B gets too big, more efficient methods are desirable to evaluate (11.8).

The idea behind the use of Krylov subspace techniques is to project the exponential of the large matrix onto a small Krylov subspace [43]. In a first attempt we ignore the special structure of the matrix and just notice, that it is not symmetric. Hence we use the standard Arnoldi method to generate an orthonormal basis $\mathbf{V}_m = [\mathbf{v}_1, \dots, \mathbf{v}_m]$ of the Krylov space $\mathcal{K}_m = \operatorname{span}\{\mathbf{y}_0, \mathbf{B}\mathbf{y}_0, \dots, \mathbf{B}^{m-1}\mathbf{y}_0\}$ with $\mathbf{v}_1 = \dfrac{\mathbf{y}_0}{\|\mathbf{y}_0\|}$ and a Hessenberg matrix $\mathbf{H}_m$ of dimension m. We know that the Hessenberg matrix is of the form $\mathbf{H}_m = \mathbf{V}_m^\top \mathbf{B}\mathbf{V}_m$. Therefore $\mathbf{H}_m$ represents the projection of the linear transformation $\mathbf{B}$ onto the subspace $\mathcal{K}_m$ with respect to the basis $\mathbf{V}_m$. We can now approximate $\exp(t\mathbf{B})\mathbf{y}_0$

$$e^{t\mathbf{B}}\mathbf{y}_0 \approx \mathbf{V}_m e^{t\mathbf{H}_m}\mathbf{V}_m^\top \mathbf{y}_0 = \beta \mathbf{V}_m e^{t\mathbf{H}_m}\mathbf{e}_1, \tag{11.9}$$

where $\beta = \|\mathbf{y}_0\|$ and $\mathbf{e}_1$ is the first standard basis vector of $\mathbb{R}^N$. If m is small, we can use a direct method to evaluate the matrix exponential $e^{t\mathbf{H}_m}$, for example `expm.m`. Hochbruck and Lubich describe in [59] how to choose m for different classes of matrices. It turns out that superlinear convergence usually requires $m \gg \|t\mathbf{B}\|$. For symmetric negative definite matrices $\mathbf{B}$ we observe this behavior already for $m \geq \sqrt{\|t\mathbf{B}\|}$. As our matrix $\mathbf{B}$ in (11.7) is not symmetric, we only

	LTS-expAB3		AB3	
p	L^2- Error	Runtime in s	L^2- Error	Runtime in s
5	$9.4320e-06$	1.030653	$1.4286e-06$	0.263381
11	$8.7312e-06$	1.030607	$1.3415e-06$	0.454422
53	$1.8880e-05$	1.030758	$1.4312e-06$	1.789966
100	$1.0902e-05$	1.031568	$1.4336e-06$	3.356812
1000	$8.3640e-06$	1.026235	$1.4276e-06$	32.103770
2000	$8.9643e-06$	1.030787	$1.4273e-06$	63.582393

Table 13: Runtime results for $\mu = 3$ and $h^{\text{coarse}} = 0.05$ for different p

get convergence for $m < \|t\mathbf{B}\|$ in special cases, e.g. if the vectors, with which we approximate the matrix function, are particularly smooth (it turns out that our initial value $\mathbf{y}_0$ has this property).

This restriction on m may lead to difficulties in our case, where we deal with damped wave equations (3.1) in one space dimension. For simplicity we assume that $T = 3.5$ and that $c = 1$ and $\sigma = 0.1$ are constant. We discretize (3.1) in space using continuous $\mathcal{P}^2$-FE with mass-lumping on an equidistant mesh with mesh size $h = 0.1$ for $[0, L] = [0, 4]$. For this simple example $\|\mathbf{B}\| \approx 2.4 \cdot 10^3$ whereas the dimension of $\mathbf{B}$ is only $N = 158$, i.e. $m \gg N$ and we were hoping for a m that is a lot smaller than N. If we still want to approximate the exponential with Krylov methods and a reasonable m, we can reduce our t, which results in a time stepping scheme of the form

$$\mathbf{y}_k = \mathbf{V}_{m,k} e^{\Delta t \mathbf{H}_{m,k}} \mathbf{V}_{m,k}^\top \mathbf{y}_{k-1}, \qquad k = 1, \ldots, n. \tag{11.10}$$

Here $\Delta t = \frac{T}{n}$, where $\mathbf{V}_{m,k}$ and $\mathbf{H}_{m,k}$ belong to the Krylov projection associated with $\mathbf{B}$ and starting vector $\mathbf{y}_{k-1}$. For (11.10) to be stable we need to require that the eigenvalues of $e^{\Delta t \mathbf{H}_{m,k}}$ are in the unit circle. This gives us a restriction for Δt. So we have two different options to achieve superlinear convergence for the Arnoldi iteration. We can either increase the number of projections m or decrease the time-step Δt. As an increasing m leads to a bigger Hessenberg matrix $\mathbf{H}_{m,k}$ of which we then have to calculate the matrix exponential, we will keep $m = 10$ fixed and focus on the influence of Δt in the following experiments.

We first choose an example with initial conditions

$$\begin{aligned} u_0(x) &= e^{-\frac{(x-2)^2}{0.25^2}}, \\ v_0(x) &= 0. \end{aligned} \tag{11.11}$$

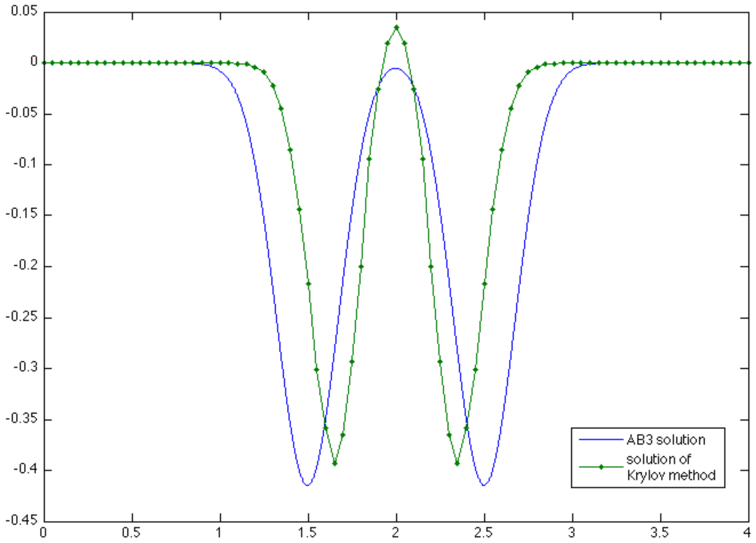


Figure 28: Reference solution (blue) and solution calculated using an Arnoldi method with $m = 10$ (green) at time $T = 3.5$

Obviously a small m won't guarantee convergence (see Figure 28). If we now use a time-stepping scheme but with a Δt that is too big, we even get problems with stability. While the eigenvalues of $\mathbf{H}_{m,0}$ approximate the eigenvalues of $\mathbf{B}$ rather well, the eigenvalues of $\mathbf{H}_{m,k}$ for $k \geq 1$ need not even be in the left half plane. Thus the solution of the time stepping scheme (11.10) might explode. To understand this behavior better, we will examine the eigenvalues of $\mathbf{H}_{m,k}$ carefully.

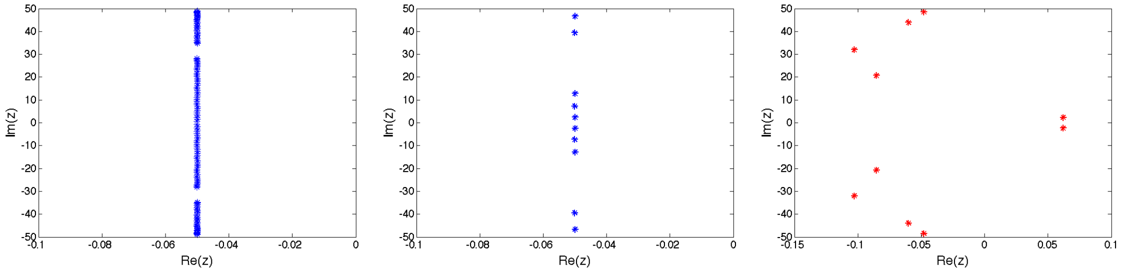


Figure 29: Eigenvalues of $\mathbf{B}$ (left), of $\mathbf{H}_{10,0}$ and of a Hessenberg matrix $\mathbf{H}$ generated with a random vector $\mathbf{v}$.

We can see in Figure 29 that the eigenvalues of $\mathbf{H}_{10,0}$ approximate the spectrum of $\mathbf{B}$ significantly better than the eigenvalues of the projection onto a Krylov subspace of the same dimension but generated by an arbitrary vector $\mathbf{v}$. Especially the two eigenvalues in the right half plane will cause problems. Similar behavior can be seen when doing more time steps. Thus the very good approximation of $\mathbf{B}$ that $\mathbf{H}_{10,0}$ seems to give us, is an exception probably related to the Krylov subspace $\mathcal{K}_m = \text{span}\{\mathbf{y}_0, \mathbf{B}\mathbf{y}_0, \dots, \mathbf{B}^{m-1}\mathbf{y}_0\}$ we project on.

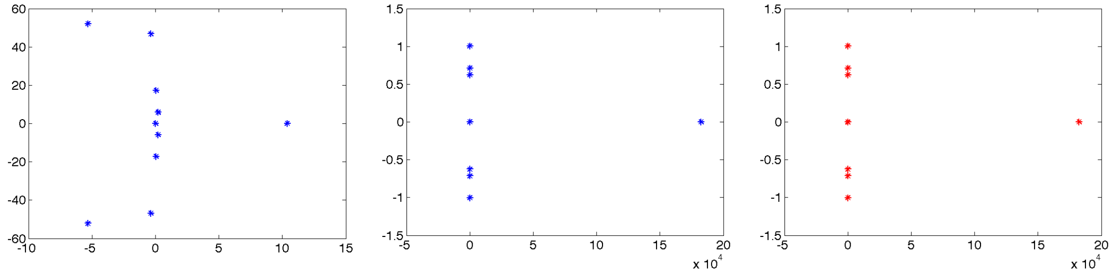


Figure 30: Eigenvalues of $\mathbf{H}_{10,1}$ (left), $e^{\frac{T}{3}\mathbf{H}_{10,1}}$ (middle) and $\mathbf{V}_{10,1}e^{\frac{T}{3}\mathbf{H}_{10,1}}\mathbf{V}_{10,1}^{\top}$ (right).

In Figure 30 we see the eigenvalues of the Hessenberg matrix after the first time step. It is important to notice that the eigenvalues of $e^{\frac{T}{3}\mathbf{H}_{10,1}}$ are no longer inside the unit circle. The eigenvalues of $\mathbf{H}_{10,1}$ located in the right half plane result in the instability of the time stepping method as the solution will grow. This can be prevented by choosing a smaller Δt, such that the eigenvalues of $e^{\Delta t \mathbf{H}_{10,1}}$ are again less than 1 in absolute value.

Our experiments show that it is usually more efficient to choose a small m and thus also a rather small Δt, as the effort of the Arnoldi method grows quadratically with m.

In fact, we can create special situations, where an exponential integrator with Krylov methods beats standard Adams-Bashforth methods. Due to the structure

of the matrix **B** in (11.7), i.e. especially the large norm of **B**, however, the exponential integrator can't compete with Adams-Bashforth methods in most of the examples. We illustrate this behavior now in the following.
We consider two different meshes for our tests. In both cases we start an equidistant mesh of size h on $[0,4]$. In case A we add two additional points at $2+h^{\text{fine}}$ and $2+2h^{\text{fine}}$ for $h^{\text{fine}} = 0.0001$, while in case B we refine the whole interval $[2, 2+h]$ with h^{fine}.
In the following we compare the performance of three different approaches. The first one calculates the solution (11.8) of (11.7) directly using the MATLAB function `expm.m`. As a second method we use the MATLAB function `expv.m` by [93]. This calculates an approximation of (11.8) with an Arnoldi method and adaptive time-stepping. In all experiments presented here we choose $m = 10$. We opt for a small m as the function `expv.m` calculates the exponential of the arising Hessenberg matrix in a direct manner. We compare both methods with the standard AB3 scheme with a small time-step dictated by the fine elements on the whole domain.

method	# of time-steps	L^2- Error	Runtime in s
direct	1	$1.3368e-03$	0.374
`expv`	1357	$1.3368e-03$	2.987
AB3	21605	$1.3368e-03$	1.345

Table 14: Case A with $h = 0.1$ and $h^{\text{fine}} = 0.001$.

For small problems the direct method is the best choice as its performance only depends on the size of the matrix and not on the refinement factor $p = h/h^{\text{fine}}$ – see Table 14 for details.

method	# of time-steps	L^2- Error	Runtime in s
direct	1	$1.3092e-03$	4.245
`expv`	3538	$1.3092e-03$	7.656
AB3	25000	$1.3092e-03$	2.778

Table 15: Case B with $h = 0.1$ and $h^{\text{fine}} = 0.001$.

If we, however, increase the number of fine unknowns and thus enlarge our system the direct method becomes less efficient than AB3 – see Table 15.

We now repeat both tests with a smaller h and h^{fine}.

method	# of time-steps	L^2- Error	Runtime in s
direct	1	$1.2683e-07$	60.561
`expv`	8215	$1.2602e-07$	21.148
AB3	214724	$1.2608e-07$	30.997

Table 16: Case A with $h = 0.01$ and $h^{\text{fine}} = 0.0001$.

In Table 16 we found an example, where the Krylov method actually presents the most efficient method.

method	# of time-steps	L^2- Error	Runtime in s
direct	1	$1.2610e-07$	109.140
`expv`	24035	$1.2522e-07$	67.776
AB3	258095	$1.2520e-07$	38.733

Table 17: Case B with $h = 0.01$ and $h^{\text{fine}} = 0.0001$.

If, however, we increase the number of fine elements, `expv` cannot compete with the classical AB3. Note further that the direct method fails in both cases as the system gets too large and thus the direct calculation of the matrix exponential too expensive.

Part IV

CONCLUSION AND FUTURE WORK

12

CONCLUSION AND FUTURE WORK

12.1 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 6.1, the LTS methods based on low-storage (LS) RK methods are given in Algorithm 6.3, and the LTS method based on the popular RK4 method is detailed in Algorithm 6.5. Like their standard RK counterparts, our LTS-RK methods preserve linear invariants.

Let Δt denote the time-step dictated by the CFL-condition in the coarser part of the mesh. Then, during every (global) time step Δt, 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 $\mathbf{P}$ 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 $\mathcal{P}^{k-1}$ FE spatial discretization, the numerical solution will converge to the true solution with optimal rate $\mathcal{O}(h^k, \Delta t^k)$, as $h, \Delta t \to 0$. Moreover, our numerical experiments suggest that the LTS-RK methods preserve the optimal CFL-condition imposed on Δt 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 [65]. Although the LTS-RK methods are particularly appropriate for computational wave propagation, where the use of explicit time integration prevails, they are likely to also prove useful for other partial differential equations.

As an alternative to LTS-RK schemes, we have presented exponential Adams methods for the time integration of damped wave equations as a way to overcome the severe stability constraints imposed by a locally refined mesh. In the last part of this thesis we have seen some numerical results which confirm that LexpAB schemes preserve the accuracy of the corresponding classical ABk method and we have studied the stability of our methods. We have observed that especially if the locally refined part is small in comparison to the entire mesh, LexpAB schemes present an interesting option for the time integration of wave equations.

12.2 FUTURE WORK

A central and important contribution of this thesis and of [47] is the proof in the ODE sense for LTS-RK schemes. We have shown in Section 7 that the LTS-RKs method preserves the accuracy of the underlying RK method. Further we have proved that for the LTS-RKs methods converge as $\Delta t \to 0$ to the semi-discrete solution $s = k = 2, 3, 4$. To study the fully discrete error analysis for LTS-RKs schemes when combined with a spatial discretization of suitable order seems a natural extension of this thesis and is of interest for our future work.

Until now we have distinguished between coarse and fine for our LTS approach. As common meshes normally have more levels of refinement, we are deriving multilevel LTS schemes which are able to handle hierarchical meshes. First results when combined with a Summation-By-Parts-Simultaneous Approximation Term (SBP-SAT) FD methods [76] seem very promising and a further investigation is work in progress.

To compare our LTS-RK schemes with local exponential integrators we are currently extending the ideas described in Part III to exponential integrators of RK type [61, 60]. Again, we hope that in cases where we have only a few tiny elements LexpRK methods present an exciting and new alternative to established LTS schemes.

Part V

APPENDIX

A

APPENDIX

Here we give the coefficients of the RK methods we use for the numerical experiments. For convenience they are presented in the well-known Butcher tableau – see [13] for details.

Table 18: Coefficients of the classical RKs methods.

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

(a) RK2 (order 2)

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ 1 & -1 & 2 & \\ \hline & \frac{1}{6} & \frac{4}{6} & \frac{1}{6} \end{array}$$

(b) RK3 (order 3)

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

(c) RK4 (order 4)

Table 19: Coefficients of the LSRK*s* methods.

$$\begin{array}{c|cc} 0 & 0 & \\ 1 & -1 & \\ \hline & 1 & \frac{1}{2} \end{array}$$

(a) LSRK2 (order 2)

$$\begin{array}{c|ccc} 0 & 0 & & \\ \frac{1}{3} & -\frac{5}{9} & & \\ -\frac{3}{4} & -\frac{153}{128} & & \\ \hline & \frac{1}{3} & \frac{15}{16} & \frac{8}{15} \end{array}$$

(b) LSRK3 (order 3)

i	A_i	B_i	C_i
1	0	$\frac{1432997174477}{9575080441755}$	0
2	$-\frac{567301805773}{1357537059087}$	$\frac{5161836677717}{13612068292357}$	$\frac{1432997174477}{9575080441755}$
3	$-\frac{2404267990393}{2016746695238}$	$\frac{1720146321549}{2090206949498}$	$\frac{2526269341429}{6820363962896}$
4	$-\frac{3550918686646}{2091501179385}$	$\frac{3134564353537}{4481467310338}$	$\frac{2006345519317}{3224310063776}$
5	$-\frac{1275806237668}{842570457699}$	$\frac{2277821191437}{14882151754819}$	$\frac{2802321613138}{2924317926251}$

(c)LSRK5 (order 4)

BIBLIOGRAPHY

[1] R. M. Alford, K. R. Kelly, and D. M. Boore, *Accuracy of finite-difference modeling of the acoustic wave equation*, Geophysics, 39 (1974), pp. 834–842.

[2] J. F. Andrus, *Stability of a multi-rate method for numerical integration of ODEs*, Comput. Math. Appl., 25 (1993), pp. 3–14.

[3] L. D. Angulo, J. Alvarez, F. L. Teixeira, M. F. Pantoja, and S. G. Garcia, *Causal-path local time-stepping in the discontinuous Galerkin method for Maxwell's equations*, J. Comput. Phys., 256 (2014), pp. 678–695.

[4] U. M. Ascher, S. J. Ruuth, and B. Wetton, *Implicit-explicit methods for time-dependent partial differential equations*, SIAM J. Numer. Anal., 32 (1995), pp. 797–823.

[5] A. Baeza and P. Mulet, *Adaptive mesh refinement techniques for high-order shock capturing schemes for multi-dimensional hydrodynamic simulations*, International Journal for Numerical Methods in Fluids, 52 (2006), pp. 455–471.

[6] J. Baglama, *Augmented block Householder Arnoldi method*, Linear Algebra Appl., 429 (2008), pp. 2315–2334.

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

[8] G. A. Baker and J. H. Bramble, *Semidiscrete and single step fully discrete approximations for second order hyperbolic equations*, RAIRO Anal. Numér., 13 (1979), pp. 75–100.

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

[10] M. J. Berger and P. Colella, *Local adaptive mesh refinement for shock hydrodynamics*, J. Comput. Phys., 82 (1989), pp. 64 – 84.

[11] M. J. Berger and R. J. LeVeque, *Adaptive Mesh Refinement Using Wave-Propagation Algorithms for Hyperbolic Systems*, SIAM J. Numer. Anal., 35 (1998), pp. 2298–2316.

[12] M. J. Berger and J. Oliger, *Adaptive mesh refinement for hyperbolic partial differential equations*, J. Comput. Phys., 53 (1984), pp. 484–512.

[13] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, John Wiley & Sons, Ltd., Chichester, second ed., 2008.

[14] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods*, Scientific Computation, Springer-Verlag, Berlin, 2006. Fundamentals in single domains.

[15] M. H. Carpenter and C. A. Kennedy, *Fourth-Order* 2*N-Storage Runge-Kutta Schemes*, NASA Report TM 109112, NASA Langley Research Center, 1994.

[16] J. Certaine, *The solution of ordinary differential equations with large time constants*, in Mathematical Methods for Digital Computers, Wiley, New York, 1960, pp. 128–132.

[17] M. J. S. Chin-Joe-Kong, W. A. Mulder, and M. Van Veldhuizen, *Higher-order triangular and tetrahedral finite elements with mass lumping for solving the wave equation*, J. Eng. Math., 35 (1999), pp. 405–426.

[18] B. Cockburn, G. E. Karniadakis, and C.-W. Shu, *The development of discontinuous Galerkin methods*, in Discontinuous Galerkin methods (Newport, RI, 1999), vol. 11 of Lect. Notes Comput. Sci. Eng., Springer, Berlin, 2000, pp. 3–50.

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

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

[21] G. C. Cohen, *Higher-Order Numerical Methods for Transient Wave Equations*, Scientific Computation, Springer-Verlag, Berlin, 2002.

[22] G. C. Cohen and P. Monk, *Mur-Nédélec finite element schemes for Maxwell's equations*, Comput. Methods Appl. Mech. Engrg., 169 (1999), pp. 197–217.

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

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

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

[26] E. Constantinescu and A. Sandu, *Multirate timestepping methods for hyperbolic conservation laws*, J. Sci. Comput., 33 (2007), pp. 239–278.

[27] ——, *Multirate Explicit Adams Methods for Time Integration of Conservation Laws*, J. Sci. Comput., 38 (2009), pp. 229–249.

[28] C. Dawson and R. Kirby, *High Resolution Schemes for Conservation Laws with Locally Varying Time Steps*, SIAM J. Sci. Comput., 22 (2001), pp. 2256–2281.

[29] A. Demirel, J. Niegemann, K. Busch, and M. Hochbruck, *Efficient Multiple Time-Stepping Algorithms of Higher Order*, J. Comput. Phys., (2015), p. in press.

[30] E. Deriaz, *Stability conditions for the numerical solution of convection-dominated problems with skew-symmetric discretizations*, SIAM J. Numer. Anal., 50 (2012), pp. 1058–1085.

[31] S. Descombes, C. Durochat, S. Lanteri, L. Moya, C. Scheid, and J. Viquerat, *Recent advances on a DGTD method for time-domain electromagnetics*, Photonics and Nanostructures - Fundamentals and Applications, 11 (2013), pp. 291 – 302.

[32] S. Descombes, S. Lanteri, and L. Moya, *Locally implicit time integration strategies in a discontinuous Galerkin method for Maxwell's equations*, J. Sci. Comp., 56 (2013), pp. 190–218.

[33] J. Diaz and M. J. Grote, *Energy conserving explicit local time-stepping for second-order wave equations*, SIAM J. Sci. Comput., 31 (2009), pp. 1985–2014.

[34] V. Dolean, H. Fahs, L. Fezoui, and S. Lanteri, *Locally implicit discontinuous Galerkin method for time domain electromagnetics*, J. Comput. Phys., 229 (2010), pp. 512–526.

[35] M. Dumbser, M. Käser, and E. Toro, *An arbitrary high-order Discontinuous Galerkin method for elastic waves on unstructured meshes - V. Local time stepping and p-adaptivity*, Geophys. J. Int., 171 (2007), pp. 695–717.

[36] M. Dumbser, O. Zanotti, A. Hidalgo, and D. S. Balsara, *ADER-WENO finite volume schemes with spacetime adaptive mesh refinement*, Journal of Computational Physics, 248 (2013), pp. 257 – 286.

[37] A. Elmkies and P. Joly, *Éléments finis d'arête et condensation de masse pour les équations de Maxwell: le cas 2d*, Comptes Rendus de l'Académie des Sciences - Series I - Mathematics, 324 (1997), pp. 1287 – 1293.

[38] ——, *Éléments finis d'arête et condensation de masse pour les équations de Maxwell: le cas de dimension 3*, Comptes Rendus de l'Académie des Sciences - Series I - Mathematics, 325 (1997), pp. 1217 – 1222.

[39] C. Engstler and C. Lubich, *Multirate extrapolation methods for differential equations with different time scales*, Computing, 58 (1997), pp. 173–185.

[40] L. C. Evans, *Partial Differential Equations*, vol. 19 of Graduate Studies in Mathematics, American Mathematical Society, Providence, RI, 1998.

[41] A. Ezziani and P. Joly, *Local time stepping and discontinuous Galerkin methods for symmetric first order hyperbolic systems*, J. Comput. Appl. Math., 234 (2010), pp. 1886–1895.

[42] J. E. Flaherty, R. M. Loy, M. S. Shephard, B. K. Szymanski, J. D. Teresco, and L. H. Ziantz, *Adaptive Local Refinement with Octree Load-Balancing for the Parallel Solution of Three-Dimensional Conservation Laws*, J. Parallel Distrib. Comput, 47 (1997), pp. 139–152.

[43] E. Gallopoulos and Y. Saad, *Efficient solution of parabolic equations by Krylov approximation methods*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 1236–1264.

[44] M. J. Gander and L. Halpern, *Techniques for Locally Adaptive Time Stepping Developed over the Last Two Decades*, in Domain Decomposition Methods in Science and Engineering XX, vol. 91 of Lecture Notes in Computational Science and Engineering, Springer Berlin Heidelberg, 2013, pp. 377–385.

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

[46] W. Gomm, *Stability analysis of explicit multirate methods*, Math. Comput. Simulation, 23 (1981), pp. 34–50.

[47] M. J. Grote, M. Mehlin, and T. Mitkova, *Runge-Kutta based explicit local time-stepping methods for wave propagation*, SIAM J. Sci. Comput., (2015), p. in press.

[48] M. J. Grote and T. Mitkova, *Explicit local time-stepping methods for Maxwell's equations*, J. Comput. Appl. Math., 234 (2010), pp. 3283–3302.

[49] ——, *High-order explicit local time-stepping methods for damped wave equations*, J. Comput. Appl. Math., 239 (2013), pp. 270–289.

[50] M. J. Grote, A. Schneebeli, and D. Schötzau, *Discontinuous Galerkin Finite Element Method for the Wave Equation*, SIAM J. Numer. Anal., 44 (2006), pp. 2408–2431.

[51] M. J. Grote, A. Schneebeli, and D. Schötzau, *Interior penalty discontinuous Galerkin method for Maxwell's equations: energy norm error estimates*, J. Comput. Appl. Math., 204 (2007), pp. 375–386.

[52] M. Günther, A. Kværnø, and P. Rentrop, *Multirate partitioned Runge-Kutta methods*, BIT, 41 (2001), pp. 504–514.

[53] M. Günther and P. Rentrop, *Multirate ROW methods and latency of electric circuits*, Applied Numerical Mathematics, 13 (1993), pp. 83 – 102.

[54] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration*, vol. 31 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2002. Structure-preserving algorithms for ordinary differential equations.

[55] E. Hairer, S. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer–Verlag, 2000.

[56] J. S. Hesthaven and T. Warburton, *Nodal High-Order Methods on Unstructured Grids: I. Time-Domain Solution of Maxwell's Equations*, J. Comput. Phys., 181 (2002), pp. 186 – 221.

[57] ——, *Nodal Discontinuous Galerkin Methods*, Springer, 2008.

[58] N. J. Higham, *The scaling and squaring method for the matrix exponential revisited*, SIAM Rev., 51 (2009), pp. 747–764.

[59] M. Hochbruck and C. Lubich, *On Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 34 (1997), pp. 1911–1925.

[60] M. Hochbruck and A. Ostermann, *Explicit Exponential Runge–Kutta Methods for Semilinear Parabolic Problems*, SIAM J. Numer. Anal., 43 (2005), pp. 1069–1090.

[61] M. Hochbruck and A. Ostermann, *Exponential RungeKutta methods for parabolic problems*, Appl. Numer. Math., 53 (2005), pp. 323 – 339.

[62] ——, *Exponential integrators*, Acta Numerica, 19 (2010), pp. 209–286.

[63] ——, *Exponential multistep methods of Adams-type*, BIT, 51 (2011), pp. 889–908.

[64] G. W. Howell, *Derivative Error Bounds for Langrange Interpolation: An Extention of Cauchy's Bound for the Error of Langrange Interpolation*, J. Approx. Theory, 67 (1991), pp. 164–173.

[65] F. Q. Hu, M. Y. Hussaini, and J. L. Manthey, *Low-dissipation and low-dispersion Runge-Kutta Schemes for computational acoustics*, J. Comput. Phys., 124 (1996), pp. 177–191.

[66] W. Hundsdorfer, A. Mozartova, and V. Savcenco, *Monotonicity conditions for multirate and partitioned explicit Runge-Kutta schemes*, in Recent developments in the numerics of nonlinear hyperbolic conservation laws, vol. 120 of Notes Numer. Fluid Mech. Multidiscip. Des., Springer, Heidelberg, 2013, pp. 177–195.

[67] W. Hundsdorfer and J. G. Verwer, *Numerical solution of time-dependent advection-diffusion-reaction equations*, vol. 33 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2003.

[68] A. Kanevsky, M. H. Carpenter, D. Gottlieb, and J. S. Hesthaven, *Application of implicit-explicit high order Runge-Kutta methods to discontinuous-Galerkin schemes*, J. Comput. Phys., 225 (2007), pp. 1753–1781.

[69] G. E. Karniadakis and S. J. Sherwin, *Spectral/hp element methods for computational fluid dynamics*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, second ed., 2005.

[70] C. A. Kennedy and M. H. Carpenter, *Additive Runge-Kutta schemes for convection-diffusion-reaction equations*, Appl. Numer. Math., 44 (2003), pp. 139–181.

[71] L. Krivodonova, *An efficient local time-stepping scheme for solution of nonlinear conservation laws*, J. Comput. Phys., 229 (2010), pp. 8537–8551.

[72] R. J. LeVeque, *Finite volume methods for hyperbolic problems*, Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge, 2002.

[73] J.-L. Lions and E. Magenes, *Non-homogeneous boundary value problems and applications. Vol. I*, Springer-Verlag, New York-Heidelberg, 1972.

[74] A. Logg, *Multi-adaptive Galerkin methods for ODEs. I*, SIAM J. Sci. Comput., 24 (2003), pp. 1879–1902.

[75] ——, *Multi-adaptive Galerkin methods for ODEs. II. Implementation and applications*, SIAM J. Sci. Comput., 25 (2003), pp. 1119–1141.

[76] K. Mattsson, F. Ham, and G. Iaccarino, *Stable and Accurate Wave-propagation in Discontinuous Media*, J. Comput. Phys., 227 (2008), pp. 8753–8767.

[77] B. V. Minchev and W. M. Wright, *A review of exponential integrators for first order semi-linear problems.*, Preprint Numerics No. 2/2005, NTNU Trondheim, 2005.

[78] P. Monk, *Finite element methods for Maxwell's equations*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2003.

[79] E. Montseny, S. Pernet, X. Ferrires, and G. Cohen, *Dissipative terms and local time-stepping improvements in a spatial high order Discontinuous Galerkin scheme for the time-domain maxwells equations*, Journal of Computational Physics, 227 (2008), pp. 6795 – 6820.

[80] W. A. Mulder, *Higher-order mass-lumped finite elements for the wave equation*, J. Comput. Accust., 9 (2001), pp. 671–680.

[81] J.-C. Nédélec, *Mixed finite elements in* $\mathbf{R}^3$, Numer. Math., 35 (1980), pp. 315–341.

[82] ——, *A new family of mixed finite elements in* $\mathbf{R}^3$, Numer. Math., 50 (1986), pp. 57–81.

[83] J. Niegemann, R. Diehl, and K. Busch, *Efficient low-storage Runge-Kutta schemes with optimized stability regions*, J. Comput. Phys., 231 (2012), pp. 364 – 372.

[84] S. P. Nørsett, *An A-stable modification of the Adams-Bashforth methods*, in Conf. on Numerical Solution of Differential Equations (Dundee, 1969), Springer, Berlin, 1969, pp. 214–219.

[85] S. Osher and R. Sanders, *Numerical approximations to nonlinear conservation laws with locally varying time and space grids*, Math. Comp., 41 (1983), pp. 321–336.

[86] A. Ostermann, M. Thalhammer, and W. M. Wright, *A class of explicit exponential general linear methods*, BIT, 46 (2006), pp. 409–431.

[87] S. Piperno, *Symplectic local time-stepping in non-dissipative DGTD methods applied to wave propagation problems*, M2AN Math. Model. Numer. Anal., 40 (2006), pp. 815–841.

[88] W. H. Reed and T. R. Hill, *Triangular mesh methods for the neutron transport equation*, Tech. Rep. LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[89] P. Rentrop, *Partitioned Runge-Kutta methods with stiffness detection and stepsize control*, Numerische Mathematik, 47 (1985), pp. 545–564.

[90] J. R. Rice, *Split Runge-Kutta method for simultaneous equations*, J. Res. Nat. Bur. Standards Sect. B, 64B (1960), pp. 151–170.

[91] B. Rivière and M. F. Wheeler, *Discontinuous finite element methods for acoustic and elastic wave problems*, in Current trends in scientific computing (Xi'an, 2002), vol. 329 of Contemp. Math., Amer. Math. Soc., Providence, RI, 2003, pp. 271–282.

[92] V. Savcenco, W. Hundsdorfer, and J. G. Verwer, *A multirate time stepping strategy for stiff ordinary differential equations*, BIT, 47 (2007), pp. 137–155.

[93] R. B. Sidje, Expokit. *A software package for computing matrix exponentials*, ACM Trans. Math. Softw., 24 (1998), pp. 130–156.

[94] S. Skelboe and P. Andersen, *Stability Properties of Backward Euler Multirate Formulas*, SIAM J. Sci. Stat. Comput., 10 (1989), pp. 1000–1009.

[95] A. Taflove and S. C. Hagness, *Computational Electrodynamics: The Finite-difference Time-domain Method*, Artech House antennas and propagation library, Artech House, Boston, Mass, third ed., 2005.

[96] H.-Z. Tang and G. Warnecke, *High resolution schemes for conservation laws and convection-diffusion equations with varying time and space grids*, J. Comput. Math., 24 (2006), pp. 121–140.

[97] A. Taube, M. Dumbser, C.-D. Munz, and R. Schneider, *A high-order discontinuous Galerkin method with time-accurate local time stepping for the Maxwell equations*, International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, 22 (2009), pp. 77–103.

[98] J. G. Verwer, *Component splitting for semi-discrete Maxwell equations*, BIT Numer. Math., 51 (2011), pp. 427–445.

[99] J. H. Williamson, *Low-storage Runge-Kutta schemes*, J. Comput. Phys., 35 (1980), pp. 48–56.

[100] K. S. Yee, *Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media*, IEEE Trans. Antennas Propag., 14 (1966), pp. 302–307.

B

CURRICULUM VITAE

Personal Information

Name	Michaela Mehlin
Date of Birth	3rd of October 1984
Place of Birth	Lörrach, Germany
Nationality	German

Education

2011–2015	**Doctoral studies in Mathematics**, University of Basel, Switzerland. Date of Defense: 21st of May 2015 (summa cum laude) Thesis: *Efficient Explicit Time Integration for the Simulation of Acoustic and Electromagnetic Waves* Advisor: Marcus J. Grote
2008–2011	**Master studies in Mathematics**, University of Basel, Switzerland. Master of Science received in April 2011 (Grade: 5.9/6) Thesis: *Exponential Adams methods with local time-stepping for damped wave equations* Advisor: Marcus J. Grote
2005–2008	**Bachelor studies in Mathematics**, University of Basel, Switzerland. Bachelor of Science received in October 2008 (Grade: 5.4/6)