DISSERTATION

Mapped Tent Pitching Schemes for Hyperbolic Systems

ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Doktors der technischen Wissenschaften unter der Leitung von

Prof. Dr. Joachim Schöberl
E101 – Institut für Analysis und Scientific Computing, TU Wien

eingereicht an der Technischen Universität Wien
Fakultät für Mathematik und Geoinformation

von

Dipl. Ing. Christoph Wintersteiger
Matrikelnummer: 01026218
Langobardenstraße 35/40
1220 Wien

Diese Dissertation haben begutachtet:

1. **Prof. Dr. Joachim Schöberl**
   Institut für Analysis und Scientific Computing, TU Wien
2. **Prof. Dr. Jay Gopalakrishnan**
   Fariborz Maseeh Department of Mathematics and Statistics, Portland State University
3. **Prof. Dr. Martin J. Gander**
   Section de Mathématiques, Université de Genève

Wien, am 20. Oktober 2020

Kurzfassung

Diese Arbeit behandelt eine Lösungsmethode für zeitabhängige hyperbolische Probleme, wie zum Beispiel die Maxwell- oder Eulergleichungen. Hyperbolische Probleme haben eine wohldefinierte Ausbreitungsgeschwindigkeit, welche im Weiteren dazu verwendet wird, das Raum-Zeit Gebiet in zeltförmige Elemente zu unterteilen. Diese zeltförmigen Raum-Zeit Elemente werden *tents* genannt und durch einen *Tent Pitching Algorithmus* erzeugt. Der *Tent Pitching Algorithmus* verwendet dabei immer die lokale Ausbreitungsgeschwindigkeit, daher passt sich der lokale Zeitschritt automatisch an diese und an die Elementgröße der räumlichen Zerlegung an. Um nun Raum und Zeit getrennt voneinander diskretisieren zu können, werden die *tents* auf Raum-Zeit Zylinder, ein Tensorprodukt von Raum und Zeit, transformiert. Auf den transformierten *tents* wird dann für die räumliche Diskretisierung eine Discontinuous Galerkin Methode verwendet. Das sich daraus ergebende System von gewöhnlichen Differentialgleichungen kann dann mittels expliziter oder impliziter Zeitschrittverfahren gelöst werden. Durch die Verwendung von impliziten Runge-Kutta Verfahren ergibt sich eine speicherintensive Methode, welche durch den großen Speicherbedarf nur begrenzt eingesetzt werden kann. Im Gegensatz dazu ergibt sich durch explizite Runge-Kutta Verfahren eine Methode mit sehr geringem Speicherbedarf. In Verbindung mit obiger Transformation führt dies allerdings, unabhängig von der Polynomordnung der räumlichen Diskretisierung, zu linearen Konvergenzraten. Um die zu erwartende höhere Konvergenzordnung wiederherzustellen, werden in dieser Arbeit spezielle explizite Zeitschrittverfahren entwickelt, welche die durch die Transformation auftretenden strukturellen Eigenschaften beachtet. Diese Konvergenzraten werden in weiterer Folge mit numerischen Beispielen belegt und es werden die Stabilitätseigenschaften der resultierenden Methode diskutiert. Abschließend wird die Anwendbarkeit dieser Methode anhand verschiedenster hyperbolischer Probleme demonstriert.

Abstract

This thesis introduces Mapped Tent Pitching (MTP) methods for hyperbolic systems. These hyperbolic system, like the Maxwell equations or the Euler equations, have a well-defined speed of propagation, which can be used to partition the spacetime domain using tent-shaped elements. These spacetime elements, denoted as tents, are generated with a tent pitching algorithm and mapped to spacetime cylinders, which allows to discretize space and time independently. Tent pitched meshes adapt to varying speeds of propagation and different sized spatial mesh leading to a naturally built in local time stepping.

The spatial discretization using a high order discontinuous Galerkin method leads to a system of ordinary differential equations, which can be solved by implicit or explicit time stepping methods. Although locally implicit MTP methods based on implicit Runge-Kutta schemes for the temporal discretization show high order convergence, the memory is a limiting factor for these methods.

Fully explicit methods have a low memory consumption, but they are limited to first order when using standard methods for the temporal discretization. To overcome this convergence order reduction, we construct suitable explicit time stepping schemes to propagate hyperbolic solutions within these tent-shaped spacetime elements. These structure aware time stepping schemes recover the high order convergence for linear and nonlinear problems.

To demonstrate the optimal convergence rates, we apply these MTP methods using structure aware times stepping schemes to various linear and nonlinear hyperbolic systems. Further we report the discrete stability properties of these methods applied to linear hyperbolic equations.

Acknowledgement

First of all I want to express my gratitude to my advisor Prof. Joachim Schöberl for his guidance and support over the last years, always coming up with new ideas when something did not work out as expected. He sparked my interest in numerical methods and their implementation and gave me the opportunity to join his group at TU Wien developing the software package Netgen/NGSolve.

Secondly, I would like to thank Prof. Jay Gopalakrishnan for all the input and feedback during our collaboration and for reviewing this thesis. Many thanks for the fruitful discussions and the good time we had during my visit in Portland.

Moreover, I wish to thank Prof. Martin J. Gander for reading and reviewing this thesis.

At this point I want to thank my colleagues in the work group for the great atmosphere and their support whenever needed. In particular, I would like to mention my (former) office mates Matthias Hochsteger, Lukas Kogler and Philip Lederer. Further I want to thank Francisco Orlandini, who joined our tent pitching club recently, and Paul Stocker for their input on the tent pitching algorithms.

I would like to acknowledge the support of the TU Wien and the Austrian Science Fund (FWF) through the research program "Taming complexity in partial differential systems" (F65) - project "Automated discretization in multiphysics" (P10).

Special thanks to family and friends for taking my mind of my studies and research to keep a good balance with other things in life.

Most importantly I want to thank my girlfriend Steffi, for her faith in me and all her support over the years. I am very grateful and happy to have her by my side.

Vienna, October 20, 2020 Christoph Wintersteiger

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Wien, am 20. Oktober 2020

Christoph Wintersteiger

Contents

1 Introduction

In many areas of continuum physics first order partial differential equations arise when formulating balance laws for quantities like mass, momentum or total energy for fluids or solid materials. Written in divergence form, these equations form a *hyperbolic system of conservation laws*. A famous nonlinear example is the Euler equations which arise when studying gas dynamics. There are many other well-known linear and nonlinear problems, which fit into the notion of hyperbolic systems. The Maxwell equations formulated as linear first order system can be used to describe electromagnetic wave propagation. These electromagnetic waves propagate with a finite speed, thus the electromagnetic field at a certain point in space and time depends only on field values within a dependency cone. These local cones are defined by the *causality condition*, which is used to delineate what is causally possible and impossible in spacetime. By subdividing spacetime into tent-shaped regions, the causality condition in naturally imposed when numerically solving hyperbolic systems. This allows to advance by different amounts in time at different spatial locations, which leads to a naturally built in local time stepping. For general nonlinear hyperbolic systems the speed of propagation is given by maximal characteristic speed, defined by a generalized eigenvalue problem, which makes this strategy applicable to general hyperbolic systems. Such spacetime meshing strategies were named tent pitching in [6, 33]. Methods using tent-pitched meshes can be traced back to [25, 29] and since then there has been active development. The dominant discretization technique is the spacetime discontinuous Galerkin (SDG) method, which can be found in works on numerical analysis [8, 12, 25] as well as in engineering applications [24, 27, 37].

These SDG methods formulate local variational problems within tents, for which linear systems are set up and solved. Although these systems are local, the matrix size can grow rapidly with the polynomial degree, especially in four-dimensional spacetime tents. The above-mentioned research into SDG methods has abundantly clarified the many advantages that tent pitched meshes offer. Perhaps the primary advantage they offer is a rational way to build high order methods (in space and time) that incorporates spatial adaptivity and locally varying time step size, even on complex structures. This local adaptivity in spacetime is utilized in [1, 2], where they apply the asynchronous SDG (aSDG) method to new engineering applications to track discontinuities is spacetime.

In this context it is natural to ask if one can develop explicit schemes (which usually perform well under low memory bandwidth) that take advantage of tents. Such a method was first introduced in [13], where the local problems within the tents are mapped to cylindrical domains. On these cylinders, space and time can be separated, so that standard spatial discretizations combined with time stepping can be used to solve the local problems within the tents. To obtain high order convergence, these Mappend Tent Pitching (MTP) methods require the use of structure aware time stepping methods within the mapped cylindrical domains, which were developed in [11, 14].

Another recent approach to reduce the complexity of the local problems within tents was presented in [28], where they discretize the tents using a spacetime Trefftz discontinuous Galerkin method. Since the Trefftz method just requires surface integrals, there is no need to build and discretize $N+1$-dimensional elements like in SDG methods.

Without tent meshes, many standard methods resort to ad hoc techniques (interpolation, extrapolation, projection, etc.) for locally adaptive time stepping [10] within inexpensive explicit strategies. If one is willing to pay the expense of solving global systems on space-time domains [22, 26, 34, 35], space and time can be treated equally and is it straight forward to obtain adaptivity in space and time. In between these options, there are interesting alternative methods, without using tents, able to perform explicit local time stepping while maintaining high order accuracy [5, 15, 16] by dividing the spatial mesh into fine and coarse regions. In contract to tent based methods, special techniques are needed at the interface of fine and coarse regions.

The main contribution of this dissertation is the construction of high order explicit MTP schemes, which have a low ratio of memory movements to flops, making them highly suitable for the newly emerging many-core processors. The presented construction of MTP schemes is based on the work published in [13]. For the derivation of the structure aware time stepping methods, the works [11, 14] serve as basis, which is extended in this thesis.

Structure of the thesis

The basis of tent pitching methods is the finite speed of propagation defined by the considered hyperbolic problem. We give a generic definition of hyperbolic systems in chapter 2 and derive the speed of propagation for some linear and nonlinear examples. Having the speed of propagation at hand, we discuss tent pitching algorithms and the novel mapping in chapter 3. Further we introduce a discontinuous Galerkin method for the spatial discretization after the mapping and apply explicit and implicit Runge-Kutta methods to propagate the solution in time within the tents. Chapter 4 is dedicated to the derivation of structure aware time stepping methods, which are essential to obtain high order methods. First we derive Structure Aware Taylor (SAT) time stepping schemes for linear problems and extent this idea to nonlinear problems by Structure Aware Runge-Kutta (SARK) time stepping methods. In chapter 5, we investigate discrete stability properties of these structure aware time stepping methods. This thesis is concluded with chapter 6, where we present several numerical examples illustrating the abilities of these explicit MTP methods.

2 Hyperbolic systems

In this chapter, we give the generic definition of the hyperbolic problems we want to consider in this thesis, which follows the definition in [4]. Let the integer $N \geq 1$ denote the dimension of our spatial domain $\Omega_0 \subset \mathbb{R}^N$. Further let the integer $L \geq 1$ denote the number of equations of our system, which are posed on the spacetime cylinder $\Omega = \Omega_0 \times (0, t_{\max})$ for a final time $t_{\max}$. For given functions $g : \Omega \times \mathbb{R}^L \to \mathbb{R}^L$ and $f : \Omega \times \mathbb{R}^L \to \mathbb{R}^{L \times N}$, the problem is to find a function $u : \Omega \to \mathbb{R}^L$ such that

$$\partial_t g(x, t, u(x, t)) + \operatorname{div}_x f(x, t, u(x, t)) = 0, \tag{2.1}$$

where $\partial_t = \partial/\partial t$ denotes the time derivative and $\operatorname{div}_x(\cdot)$ denotes the spatial divergence operator applied row-wise to matrix-valued functions. Using subscripts do denote the components (e.g. g_l for the lth component of g and f_{li} for the (l, i)th component of f), we can write (2.1) as

$$\partial_t g_l(x, t, u(x, t)) + \sum_{i=1}^{N} \partial_i \left(f_{li}(x, t, u(x, t)) \right) = 0, \tag{2.2}$$

for all components $l = 1 \dots L$. The differentiation along the ith direction in $\mathbb{R}^N$ is denoted by $\partial_i = \partial/\partial x_i$. In examples, we will supplement (2.2) by initial conditions u_0 on Ω_0 and boundary conditions on $\partial\Omega_0 \times (0, t_{\max})$.

To define hyperbolicity, we require that the first order derivatives of g and f_i, the ith column of f, with respect to u exit. These $L \times L$ derivative matrices are denoted by $\mathrm{D}_u g$ (whose (l, m)th entry is $\partial g_l / \partial u_m$) and $\mathrm{D}_u f_i$ for $i = 1, \dots, N$.

Definition 1. The system (2.1) is called *hyperbolic in t-direction* if, for any fixed $u \in \mathbb{R}^L$, $(x, t) \in \Omega$ and any direction $\nu \in \mathbb{S}^{N-1}$, $\mathrm{D}_u g$ is invertible and the eigenvalue problem

$$\left[\sum_{i=1}^{N} \nu_i \mathrm{D}_u f_i(x, t, u) - \lambda \mathrm{D}_u g(x, t, u) \right] v = 0 \tag{2.3}$$

has real eigenvalues $\lambda_1(\nu, x, t, u), \dots, \lambda_L(\nu, x, t, u)$ and L linearly independent eigenvectors $v_1(\nu, x, t, u), \dots, v_L(\nu, x, t, u)$. These eigenvalues $\lambda_i, i = 1, \dots, L$ are called *characteristic speeds*.

Let $c(x, t, u)$ denote the maximum of these characteristic speeds for all direction ν on the $N - 1$ dimensional unit sphere $\mathbb{S}^{N-1}$. For simplicity, we assume that $c(x, t, u)$ is given (even though it can often be computationally estimated), so that the meshing process in the next section can use it as input. The maximal characteristic speed is often called wave speed or speed of propagation.

2.1 Linear examples

For linear hyperbolic systems, we can use the linearity to write the hyperbolic system in a simpler form. Suppose that $A^{(i)} : \Omega \to \mathbb{R}^{L\times L}$, for $i = 1, \dots, N+1$, are symmetric matrix-valued functions and $A^{(t)} \equiv A^{(N+1)}$ is symmetric positive definite. By setting

$$f_{li}(x,t,u) = \sum_{m=1}^{L} A^{(i)}_{lm}(x,t)\, u_m \qquad \text{and} \qquad g_l(x,t,u) = \sum_{m=1}^{L} A^{(t)}_{lm}(x,t)\, u_m\,, \tag{2.4}$$

the system (2.1) can be written as

$$\partial_t\big(A^{(t)}u\big) + \sum_{i=1}^{N} \partial_i\big(A^{(i)}u\big) = 0. \tag{2.5}$$

The derivatives of g and f with respect to u, forming the eigenvalue problem (2.3), are given by $\mathrm{D}_u g = A^{(t)}$ and $\mathrm{D}_u f_i = A^{(i)}$. Next, we consider some linear examples to get a better understanding of the abstract definition of the characteristic speeds given by the eigenvalue problem (2.3).

Advection equation

The advection equation

$$\partial_t u + \operatorname{div}_x(\beta u) = 0 \tag{2.6}$$

describes the transport of a scalar density function $u : \Omega \to \mathbb{R}$ along a given divergence-free vector field $\beta : \Omega_0 \to \mathbb{R}^N$. This fits into the framework (2.5) with $L = 1$, $A^{(t)} = [1]$ and $A^{(i)} = [\beta_i(x)]$, for $i = 1, \dots, N$. The resulting eigenvalue problem is

$$\left[\sum_{i=1}^{N} \nu_i \beta_i(x) - \lambda\right] v = 0 \quad \Leftrightarrow \quad \lambda = \nu \cdot \beta\,.$$

Since $\nu \in \mathbb{S}^{N-1}$, we obtain $c(x,t,u) = \|\beta\|$. Thus the characteristic speeds are bounded by the Euclidean norm of the vector field $\beta(x)$, which is the bound we expected.

Wave equation

For a given symmetric and positive definite material coefficient $\alpha : \Omega_0 \to \mathbb{R}^{N\times N}$, the wave equation for the linearized pressure $\phi : \Omega \to \mathbb{R}$ is

$$\partial_{tt}\phi - \operatorname{div}_x(\alpha\, \nabla_x\, \phi) = 0 \qquad \text{in } \Omega. \tag{2.7}$$

To put (2.7) into the framework of (2.5), we set $L = N + 1$ and define

$$u = \begin{bmatrix} q \\ \mu \end{bmatrix} = \begin{bmatrix} -\alpha\, \nabla_x\, \phi \\ \partial_t \phi \end{bmatrix} \in \mathbb{R}^L. \tag{2.8a}$$

Then (2.7) yields

$$\alpha^{-1}\partial_t q + \nabla_x \mu = 0, \qquad \partial_t \mu + \operatorname{div}_x q = 0.$$

With the matrices

$$A^{(t)} = \begin{bmatrix} \alpha^{-1} & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad A^{(i)} = \begin{bmatrix} 0 & e_i \\ e_i^\intercal & 0 \end{bmatrix}, \tag{2.8b}$$

where $e_i \in \mathbb{R}^N$ denotes the ith unit vector, we obtain (2.7) in the form (2.5). This allows us to formulate the eigenvalue problem for the characteristic speed

$$\left[\sum_{i=1}^{N} \nu_i A^{(i)} - \lambda A^{(t)}\right] v = 0 \quad \Leftrightarrow \quad \begin{bmatrix} \alpha & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & \nu \\ \nu^\intercal & 0 \end{bmatrix} v = \lambda v\,.$$

This leads to the characteristic polynomial

$$p(\lambda) = \det \begin{bmatrix} -\lambda I & \alpha\nu \\ \nu^\intercal & -\lambda \end{bmatrix} = -\lambda^{N-1}\left(-\lambda^2 + \nu^\intercal \alpha \nu\right),$$

where $I \in \mathbb{R}^{N\times N}$ denotes the identity matrix. For an isotropic material holds $\alpha = c_s^2 I$, with the speed of sound c_s, and we obtain the characteristic speeds $\lambda_i \in \{0, \pm c_s\}$, for $i = 1, \dots, N$, as solutions of $p(\lambda) = 0$. For the maximum of these characteristic speeds holds

$$c(x,t,u) = \max_{1\le i\le N} |\lambda_i(x)| = c_s(x). \tag{2.9}$$

Maxwell equations

A more complex example is the Maxwell system

$$\partial_t (\varepsilon E) - \operatorname{curl} H = 0, \tag{2.10a}$$
$$\partial_t (\mu H) + \operatorname{curl} E = 0, \tag{2.10b}$$

for the electric field $E \in \mathbb{R}^3$ and magnetic field $H \in \mathbb{R}^3$, where permittivity ε and permeability μ are functions on Ω_0. With the notation

$$\operatorname{skew} E = \begin{bmatrix} 0 & E_z & -E_y \\ -E_z & 0 & E_x \\ E_y & -E_x & 0 \end{bmatrix},$$

we can rephrase the curl operator as

$$\operatorname{curl} E = \operatorname{div}_x \operatorname{skew} E\,.$$

With the definitions

$$u = \begin{bmatrix} E \\ H \end{bmatrix}, \quad g(u) = \begin{bmatrix} \varepsilon E \\ \mu H \end{bmatrix}, \quad \text{and} \quad f(u) = \begin{bmatrix} -\operatorname{skew} H \\ \operatorname{skew} E \end{bmatrix}, \tag{2.11a}$$

the Maxwell system (2.10) can be written as conservation law of the form (2.2). In the following considerations, we assume that the material parameters ε and μ are independent

of the propagation direction. Thus ε and μ are scalar functions and to fit into the linear framework (2.5), we define the matrices

$$A^{(t)} = \begin{bmatrix} \varepsilon I & 0 \\ 0 & \mu I \end{bmatrix} \quad \text{and} \quad A^{(i)} = \begin{bmatrix} 0 & [\epsilon^i] \\ [\epsilon^i]^\intercal & 0 \end{bmatrix}, \tag{2.11b}$$

where $I \in \mathbb{R}^{3\times 3}$ is the identity matrix and $\epsilon^i \in \mathbb{R}^{3\times 3}$ the matrix whose (j,k)th entry is the Levi-Civita alternator ϵ_{ijk}. The eigenvalues problem (2.3) for the characteristic speeds reads

$$\left[\sum_{i=1}^{N} \nu_i A^{(i)} - \lambda A^{(t)}\right] v = 0 \quad \Leftrightarrow \quad \begin{bmatrix} \varepsilon I & 0 \\ 0 & \mu I \end{bmatrix}^{-1} \begin{bmatrix} 0 & \operatorname{skew} \nu \\ -\operatorname{skew} \nu & 0 \end{bmatrix} v = \lambda v\,,$$

and leads to the characteristic polynomial

$$p(\lambda) = \det \begin{bmatrix} -\lambda I & \varepsilon^{-1} \operatorname{skew} \nu \\ -\mu^{-1} \operatorname{skew} \nu & -\lambda I \end{bmatrix} = \left(\frac{\lambda}{\varepsilon\mu}\right)^2 \left(\varepsilon\mu\lambda^2 - 1\right)^2.$$

The solutions of $p(\lambda) = 0$ define the characteristic speeds $\lambda_i \in \left\{0, \pm(\varepsilon\mu)^{-1/2}\right\}$, for $i = 1, \ldots, 6$, and we obtain

$$c(x,t,u) = \max_{1\le i\le 6} |\lambda_i(x)| = \frac{1}{\sqrt{\varepsilon(x)\mu(x)}}\,. \tag{2.12}$$

2.2 Nonlinear examples

When numerically solving nonlinear hyperbolic problem, one has to use stabilization techniques to handle nonsmooth artifacts, like shocks, of the solution. We will use and entropy based artificial viscosity, which is described in detail in §6.2.1.

Recall that a real function $\mathcal{E}(u)$ is called an entropy [32, Definition 3.4.1] of the hyperbolic system (2.1) if there exists an entropy flux $\mathcal{F}(u) \in \mathbb{R}^N$ such that every classical solution u of (2.1) satisfies $\partial_t \mathcal{E}(u) + \operatorname{div}_x \mathcal{F}(u) = 0$. Note that this equality does not need to hold for nonsmooth solutions u. The pair $(\mathcal{E}, \mathcal{F})$ is called the entropy pair. We say that this pair satisfies the *entropy admissibility condition* on $\Omega_0 \subset \mathbb{R}^N$ if

$$\partial_t \mathcal{E}(u(x,t)) + \operatorname{div}_x \mathcal{F}(u(x,t)) \le 0, \tag{2.13}$$

holds in the sense of distributions on Ω. The inequality is useful to study the violation of entropy conservation for nonsmooth solutions (like shocks). Nonlinear conservation laws often have multiple weak solutions and uniqueness is obtained by selecting a solution u satisfying the entropy admissibility condition. These theoretical considerations motivate the use of numerical analogues of (2.13) in designing schemes for conservation laws.

Next, we give two examples of nonlinear hyperbolic problems and their entropy pairs.

Burgers' equation

As first nonlinear model problem, we consider the Burgers' equation, which we obtain by setting $L = 1$, $N \in \{1, 2\}$,

$$g(x,t,u) = u \in \mathbb{R} \qquad \text{and} \qquad f(x,t,u) = \frac{1}{2}[u^2]^N \in \mathbb{R}^N. \tag{2.14}$$

There holds $\mathrm{D}_u g = 1$ and $\mathrm{D}_u f_i = u$, for $i = 1, \dots, N$, and the corresponding eigenvalue problem (2.3) reads

$$\left[\sum_{i=1}^{N} \nu_i u(x,t) - \lambda\right] v = 0 \quad \Leftrightarrow \quad \lambda = u \sum_{i=1}^{N} \nu_i \,.$$

For $\nu \in \mathbb{S}^{N-1}$ holds

$$c(x,t,u) = \max_{\nu \in \mathbb{S}^{N-1}} |\lambda| = |u(x,t)| \max_{\nu \in \mathbb{S}^{N-1}} \left|\sum_{i=1}^{N} \nu_i\right| = \sqrt{N}|u(x,t)|.$$

In contrast to the previously discussed linear problems, we now have a solution dependent characteristic speed.

An entropy pair $(\mathcal{E}, \mathcal{F})$ for the Burgers' equation, satisfying the condition (2.13), is given by the functions

$$\mathcal{E}(u) = u^2 \quad \text{and} \quad \mathcal{F}(u) = \frac{u^3}{3}. \tag{2.15}$$

Euler equations

Another well-known example is the Euler equations, described by the density $\rho : \Omega_0 \to \mathbb{R}$, the momentum $m : \Omega_0 \to \mathbb{R}^N$ and total energy $E : \Omega_0 \to \mathbb{R}$ of a perfect gas occupying $\Omega_0 \subset \mathbb{R}^N$. Set $L = N + 2$ and let

$$u = \begin{bmatrix} \rho \\ m \\ E \end{bmatrix}, \quad g(u) = u\,, \quad f(u) = \begin{bmatrix} m^\intercal \\ P I + m \otimes m/\rho \\ (E + P) m^\intercal / \rho \end{bmatrix}, \tag{2.16a}$$

where $m \otimes m = m\, m^\intercal \in \mathbb{R}^{N \times N}$ denotes the outer product of the momentum m. Here, the pressure P and temperature T are related to the state variables by

$$P = \frac{1}{2}\rho T \qquad \text{and} \qquad T = \frac{4}{d}\left(\frac{E}{\rho} - \frac{1}{2}\frac{\|m\|^2}{\rho^2}\right), \tag{2.16b}$$

where d, the degrees of freedom of the gas particles, is set to 5 for ideal gas. With these settings, the system of Euler equations is given by (2.1).

For the calculation of the wave speed we refer to [18] where they derive a methodology to obtain an upper bound of the wave speed without computing the full solution.

A well-known entropy pair $(\mathcal{E}, \mathcal{F})$ for the Euler equations, satisfying the condition (2.13), is given by the functions

$$\mathcal{E}(\rho, m, E) = \rho\big(\ln \rho - \tfrac{d}{2} \ln T\big) \quad \text{and} \quad \mathcal{F}(\rho, m, E) = \frac{m\mathcal{E}}{\rho}. \tag{2.17}$$

3 Tents

The Mapped Tent Pitching (MTP) schemes we present later in this chapter fall into the category of methods that use tent pitching for unstructured spacetime meshing. Accordingly, in this chapter, we first give a general description of tent meshing, clarifying the mathematical meaning of words we have already used colloquially such as *tent*, *tent pole*, *advancing front*, etc. in §3.1 and then give details of specific meshing algorithms that we have chosen to implement in §3.2. Based on such a tent mesh, we introduce a mapping in §3.3, which allows to discretize space and time separately. In §3.4 we discuss explicit and implicit temporal discretizations leading to fully explicit and locally implicit MTP methods. This chapter is ended by the examples of the inverse map in §3.5 which is required during the explicit time stepping.

3.1 Overview of a tent pitching scheme

We now describe how a tent pitching scheme advances the numerical solution in time. We mesh Ω_0 by a simplicial conforming shape regular finite element mesh $\mathcal{T} = \{T_i, i = 1, \dots, N_{\mathcal{T}}\}$. The mesh is unstructured to accommodate for any intricate features in the spatial geometry or in the evolving solution. Let $P_1(\mathcal{T})$ denote the set of continuous real-valued functions on Ω_0 which are linear on each element of $\mathcal{T}$. Clearly any function in $P_1(\mathcal{T})$ is completely determined by its values at the vertices $\mathcal{V} = \{\mathrm{v}_i, i = 1, \dots, N_{\mathcal{V}}\}$, of the mesh $\mathcal{T}$.

At the ith step of a tent pitching scheme, the numerical solution is available for all $x \in \Omega_0$ and all $0 < t < \tau_i(x)$. The function τ_i is in $P_1(\mathcal{T})$ and its graph, denoted by S_i, is called the *advancing front* (see Figure 3.1).

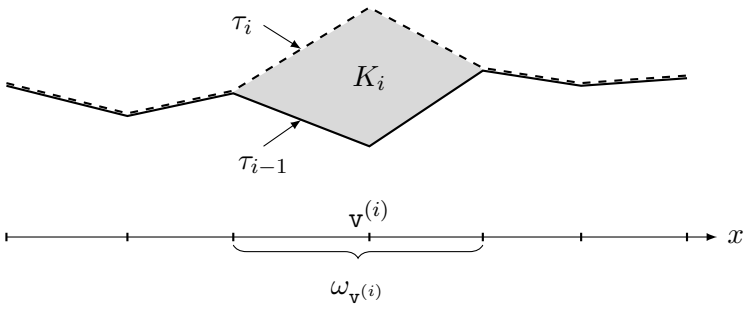


Figure 3.1: Advancing fronts τ_{i-1} and τ_i at the ith step and the tent K_i over the vertex patch $\omega_{\mathrm{v}^{(i)}}$ centered at the vertex $\mathrm{v}^{(i)}$ based on a one-dimensional spatial mesh.

We present a serial version of the algorithm first. A parallel generalization is straightforward as mentioned in Remark 1. A tent pitching scheme updates τ_i within the general outline of Algorithm 1.

Algorithm 1 Advancing front of tents and approximate solution

1. Initially, set $\tau_0 \equiv 0$. Then $S_0 = \Omega_0$. The solution on S_0 is determined by the initial data on Ω_0.
2. For $i = 1, 2, \ldots,$ do:
   a) Select a mesh vertex $\mathtt{v}^{(i)}$ and calculate the height (in time) k_i by which we can move the advancing front at $\mathtt{v}^{(i)}$. Detailed strategies are discussed in §3.2.
   b) Given the solution on the current advancing front S_{i-1}, pitch a *spacetime tent* K_i by erecting a *tent pole* of height k_i at the point $(\mathtt{v}^{(i)}, \tau_{i-1}(\mathtt{v}^{(i)}))$ on S_{i-1}. Let $\eta_i \in P_1(\mathcal{T})$ be the unique function that equals one at $\mathtt{v}^{(i)}$ and is zero at all other mesh vertices. Set
   $$\tau_i = \tau_{i-1} + k_i \eta_i \,. \tag{3.1}$$
   Define the *vertex patch* $\omega_{\mathtt{v}}$ of a mesh vertex $\mathtt{v}$ as the (spatial) open set in $\mathbb{R}^N$ that is the interior of the union of all simplices in $\mathcal{T}$ connected to $\mathtt{v}$. Then the tent K_i can be expressed as
   $$K_i = \{(x,t) : \; x \in \omega_{\mathtt{v}^{(i)}}, \; \tau_{i-1}(x) < t < \tau_i(x)\}, \tag{3.2}$$
   see Figure 3.1.
   c) Numerically solve (2.1) on K_i (e.g., by the methods proposed later in this thesis). The initial data is obtained from the given solution on S_{i-1}. If $\mathtt{v}^{(i)} \in \partial\Omega_0$, then the boundary conditions required to solve (2.1) on K_i are obtained from the given boundary conditions on the global boundary $\partial\Omega_0 \times (0, t_{\max})$.
   d) If $\tau(\mathtt{v}) \geq t_{\max}$ for all mesh vertices $\mathtt{v}$, then exit.

The height of the tent pole k_i in step 2a of Algorithm 1 should be determined using the causality constraint so that the hyperbolic problem (2.1) is solvable on the resulting tent K_i. The choice of the vertex $\mathtt{v}^{(i)}$ should be made considering the height of the neighboring vertices. Other authors have studied these issues [6, 33] and given appropriate advancing front meshing strategies. In the next section, we describe the strategies we have chosen to implement and discuss their advantages and disadvantages.

3.2 Tent pitching algorithms

We now describe strategies how to calculate the possible advance in time, e.g. the height of the tent pole k_i, such that the new advancing front satisfies the *causality constraint*. Let $\bar{c}(x)$ denote a given (or computed) approximation to the maximal characteristic speed

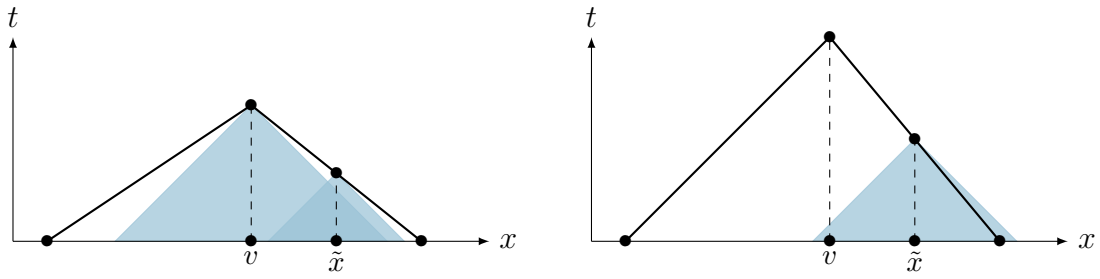


Figure 3.2: Domain of dependence (blue filled) of light cone with slope $\frac{1}{\bar{c}}$ in the context of tent pitching in one spatial dimensional.

at a point $(x, \tau_{i-1}(x))$ on the advancing front S_{i-1}, e.g., $\bar{c}(x) = c(x, \tau_{i-1}(x), u(x, \tau_{i-1}(x))$, where u is the computed numerical solution. We want to ensure that

$$\|\nabla_x \tau_i(x)\| \leq \frac{1}{\bar{c}(x)} \tag{3.3}$$

holds for all advancing fronts τ_i and all $x \in \Omega_0$. Here $\|\cdot\|$ denotes the Euclidean norm. This is our *causality condition*, which is imposed even before we have discretized the hyperbolic problem. Note that this has also been called the *cone constraint* in [6], where it is geometrically interpreted as every tent facets separates the domain of influence (light cone opening above) from the domain of dependence (light cone opening below). Thus the causality condition (3.3) is satisfied, if the domain of dependence lies below the new advancing front S_i for every point $(x, \tau_i(x))$. Figure 3.2 compares two tents with a lower (left) and a higher (right) advance in time at the central vertex $\mathtt{v}$. While the left tent satisfies the causality condition (3.3), the right one does not, because the domain of dependence exceeds the advancing front, as illustrated for the point $(\tilde{x}, \tau_i(\tilde{x}))$.

First, we present a strategy in §3.2.1, which has a low computational effort and applies verbatim in one, two and three space dimensions. The causality condition (3.3) is enforced on the edges of the spatial mesh. Thus, (3.3) is fulfilled up to a constant depending on the shape regularity of the spatial mesh. The second strategy in §3.2.2 obtains the maximal possible advance in time by solving the quadratic inequality arising from the causality condition (3.3). This approach guarantees the bound given by (3.3), but is computationally more expensive.

3.2.1 Edge-based tent pitching algorithm

For simplicity, we now assume that c is independent of time and instead of (3.3), we impose the more stringent condition

$$\|\nabla_x(\tau_i|_T)\| \leq \frac{1}{c_T}, \qquad \text{for all } T \in \mathcal{T}, \tag{3.4}$$

where $c_T = \max_{x \in T} \bar{c}(x)$. Since $\tau_i|_T$ is linear, its gradient is a constant vector that is determined by its tangential components along the edges of T. The tangential component on a mesh edge e of length $|e|$ is $(\tau_i(e_1) - \tau_i(e_2))/|e|$, where e_1 and e_2 denote the endpoints

of e. Due to our assumption that the initial spatial mesh is shape regular, we can guarantee that (3.4) holds by imposing

$$\frac{\tau_i(e_1) - \tau_i(e_2)}{|e|} \leq \frac{C_{\mathcal{T}}}{c_e}, \qquad \text{for all mesh edges } e, \tag{3.5}$$

where c_e is the maximum of c_T over all elements T which have e as an edge and $C_{\mathcal{T}}$ is a constant that depends only on the shape regularity of the mesh $\mathcal{T}$. In one space dimension, condition (3.5) with $C_{\mathcal{T}} = 1$ is equivalent to (3.4), since the edge corresponds to the volume element. Condition (3.5) is easier to work with in practice and is the same in one, two and three space dimensions. A practical strategy is to start with a guess for $C_{\mathcal{T}}$ like $1/3$, check if the values of $\nabla_x \tau_i$ at the integration nodes (which need to be computed anyway as will be clear later) satisfy (3.4), and revise if necessary.

To obtain an advancing front satisfying (3.5) at all stages i, we maintain a list of potential time advance $\tilde{k}_l^{(i)}$ that can be made at any vertex $\mathtt{v}_l \in \mathcal{V}$. Let $\mathcal{E}_l$ denote the set of all mesh edges connected to the vertex $\mathtt{v}_l$ and suppose edge endpoints are enumerated so that $e_1 = \mathtt{v}_l$ for all $e \in \mathcal{E}_l$. Given τ_i satisfying (3.5), while considering pitching a tent at $(\mathtt{v}_l, \tau_i(\mathtt{v}_l))$ so that (3.5) continues to hold, we want to ensure that

$$\frac{\left(\tau_i(\mathtt{v}_l) + \tilde{k}_l^{(i)}\right) - \tau_i(e_2)}{|e|} \leq \frac{C_{\mathcal{T}}}{c_e} \quad \text{and} \quad \frac{-\left(\tau_i(\mathtt{v}_l) + \tilde{k}_l^{(i)}\right) + \tau_i(e_2)}{|e|} \leq \frac{C_{\mathcal{T}}}{c_e}$$

hold for all $e \in \mathcal{E}_l$. The latter inequality is obvious from (3.5) since we are only interested in $\tilde{k}_l^{(i)} \geq 0$. The former inequality is ensured if we choose

$$\tilde{k}_l^{(i)} \leq \min_{e \in \mathcal{E}_l} \left(\tau_i(e_2) - \tau_i(\mathtt{v}_l) + |e| \frac{C_{\mathcal{T}}}{c_e} \right), \tag{3.6}$$

as done in the Algorithm 2 below. The algorithm also maintains a list of locations ready for pitching a tent. There exist various approaches how to select positions where to pitch next. In the following we want to discuss two approaches.

Relative progress criterion

For this approach, it needs the reference heights $r_l = \min_{e \in \mathcal{E}_l} |e| C_{\mathcal{T}} / c_e$ (the maximal tent pole heights on a flat advancing front) which can be precomputed. Set $\tilde{k}_l^{(0)} = r_l$. A vertex $\mathtt{v}_l$ is considered a location where "good" progress in time can be made if its index l is in the set

$$J_i = \left\{ l : \ \tilde{k}_l^{(i)} \geq \gamma r_l \right\}. \tag{3.7}$$

Here $0 < \gamma < 1$ is a parameter (usually set to $1/2$). While a lower value of γ identifies many vertices to progress in time moderately, a higher value of γ identifies fewer vertices where time can be advanced more aggressively.

Local minima criterion

In this approach tents are always pitched at one of the local minima of the advancing front. Then the neighboring vertices, which are all ahead in time of vertex v_l where we want to pitch, impose a less restrictive bound. This becomes clear when looking at (3.6), where $\tau_i(e_2) - \tau_i(\mathrm{v}_l)$ is guaranteed to be nonnegative for all edges $e \in \mathcal{E}_l$ connected to v_l. The list J_i of ready vertices at the ith step is given by

$$J_i = \{l : \ \tau_i(e_2) \geq \tau_i(\mathrm{v}_l) \ \forall e \in \mathcal{E}_l\} \,. \tag{3.8}$$

Algorithm 2 Updating potential pitch locations and time steps

Initially, $\tau_0 \equiv 0$, $\tilde{k}_l^{(0)} = r_l$ and $J_0 = \{1, 2, \ldots, N_\mathcal{V}\}$. For $i \geq 1$, given τ_{i-1}, $\{\tilde{k}_l^{(i-1)}\}$, and J_{i-1}, we choose the next tent pitching location $\mathrm{v}^{(i)}$ and the tent pole height k_i, and update as follows:

1. Pick any l_* in J_{i-1}.
2. Set $\mathrm{v}^{(i)} = \mathrm{v}_{l_*}$ and $k_i = \min\left(t_{\max} - \tau_{i-1}(\mathrm{v}_{l_*}), \tilde{k}_{l_*}^{(i-1)}\right)$.
3. Update τ_i by (3.1).
4. Update $\tilde{k}_l^{(i)}$ for all vertices v_l adjacent to $\mathrm{v}^{(i)}$ by

$$\tilde{k}_l^{(i)} = \min_{e \in \mathcal{E}_l} \left(\tau_i(e_2) - \tau_i(\mathrm{v}_l) + |e| \frac{C_\mathcal{T}}{c_e} \right) . \tag{3.9}$$

5. Use $\{\tilde{k}_l^{(i)}\}$ to set J_i using (3.7) or (3.8).

3.2.2 Volume-based tent pitching algorithm

Again, we assume that c is independent of time and impose the element-wise condition

$$\|\nabla_x(\tau_i|_T)\| \ \leq \ \frac{1}{c_T}, \qquad \text{for all } T \in \mathcal{T}, \tag{3.4}$$

where $c_T = \max_{x \in T} \bar{c}(x)$. The simplicial elements $T \in \mathcal{T}$ can be represented by the convex hull

$$T = \mathrm{conv}\{\mathrm{v}_T^j, j = 1, \ldots, N+1\}$$

of its $N+1$ vertices $\mathrm{v}_T^j \in \mathcal{V}$. Let $\lambda_T^j \in P_1(T)$, $j = 1, \ldots, N+1$, denote the barycentric coordinates, which satisfy

$$\lambda_T^j(\mathrm{v}_T^k) = \begin{cases} 1 & j = k \,, \\ 0 & j \neq k \,. \end{cases}$$

Since $\tau_i|_T$ is linear, we can decompose it with respect to the barycentric coordinates and obtain

$$(\tau_i|_T)\,(x) = \sum_{j=1}^{N+1} d_T^j \lambda_T^j(x) \,,$$

with the coefficients $d_T^j = \tau_i(\mathrm{v}_T^j)$ for $j = 1, \dots, N+1$. The gradient of $\tau_i|_T$ is then represented by the constant vector

$$\nabla_x(\tau_i|_T) = \sum_{j=1}^{N+1} d_T^j \, \nabla_x \lambda_T^j \,. \tag{3.10}$$

We now consider pitching a tent at the advancing front $(\mathrm{v}_l, \tau_i(\mathrm{v}_l))$ for any $\mathrm{v}_l \in \mathcal{V}$. Therefore we have to calculate the possible tent pole heights given by the neighboring elements $T \in \mathcal{T}_{\mathrm{v}_l}$ of v_l, where $\mathcal{T}_{\mathrm{v}_l} \subset \mathcal{T}$ denotes the collection of elements in the vertex patch ω_{v_l}. For all $T \in \mathcal{T}_{\mathrm{v}_l}$ we find a $k \in \{1, \dots, N+1\}$, such that $\mathrm{v}_T^k = \mathrm{v}_l$. Thus we want to find the maximal coefficient d_T^k, such that (3.4) holds true. Using (3.10), the quadratic norm of the gradient reads

$$\begin{aligned}
\|\nabla_x(\tau_i|_T)\|^2 &= \sum_{j=1}^{N+1} \sum_{l=1}^{N+1} d_T^j d_T^l \, \nabla_x \lambda_T^j \cdot \nabla_x \lambda_T^l \\
&= \| \nabla_x \lambda_T^k \|^2 \big(d_T^k\big)^2 + 2 \Big(\sum_{\substack{j=1 \\ j \neq k}}^{N+1} d_T^j \, \nabla_x \lambda_T^j \Big) \cdot \nabla_x \lambda_T^k \; d_T^k \\
&\quad + \sum_{\substack{j=1 \\ j \neq k}}^{N+1} \sum_{\substack{l=1 \\ l \neq k}}^{N+1} d_T^j d_T^l \, \nabla_x \lambda_T^j \cdot \nabla_x \lambda_T^l \,.
\end{aligned}$$

With the constants $\alpha, \beta, \gamma \in \mathbb{R}$ defined by

$$\alpha = \| \nabla_x \lambda_T^k \|^2 \,, \qquad \beta = 2 \Big(\sum_{\substack{j=1 \\ j \neq k}}^{N+1} d_T^j \, \nabla_x \lambda_T^j \Big) \cdot \nabla_x \lambda_T^k \,,$$

$$\gamma = \sum_{\substack{j=1 \\ j \neq k}}^{N+1} \sum_{\substack{l=1 \\ l \neq k}}^{N+1} d_T^j d_T^l \, \nabla_x \lambda_T^j \cdot \nabla_x \lambda_T^l \,,$$

we obtain the inequality

$$\|\nabla_x(\tau_i|_T)\|^2 = \alpha \big(d_T^k\big)^2 + \beta d_T^k + \gamma < \frac{1}{c_T^2} \,, \tag{3.11}$$

which is equivalent to the element-wise causality condition (3.4). To obtain the allowed advance at the vertex v_T^k, we have to find the maximal d_T^k such that the quadratic inequality (3.11) holds. By solving

$$\|\nabla_x(\tau_i|_T)\|^2 - \frac{1}{c_T^2} = \alpha \big(d_T^k\big)^2 + \beta d_T^k + \gamma - \frac{1}{c_T^2} = 0 \,,$$

we obtain two candidates

$$\big(d_T^k\big)_{1,2} = \frac{-\beta \pm \sqrt{\beta^2 - 4\alpha\tilde{\gamma}}}{2\alpha} \,, \tag{3.12}$$

where $\tilde{\gamma} = \gamma - \frac{1}{c_T^2}$. Due to the symmetry of the causality condition, one solution lies in the "past" and one in the "future". Thus we are interested in the maximum of these two and we ensure the causality condition if we choose

$$\tilde{k}_l^{(i)} \leq \min_{T \in \mathcal{T}_{\mathtt{v}_l}} \left(\max \left\{ \left(d_T^k\right)_1, \left(d_T^k\right)_2 \right\} - \tau_i(\mathtt{v}_l) \right).$$

Algorithm 2 can be easily modified by replacing (3.9) with the above derived bound

$$\tilde{k}_l^{(i)} = \min_{T \in \mathcal{T}_{\mathtt{v}_l}} \left(\max \left\{ \left(d_T^k\right)_1, \left(d_T^k\right)_2 \right\} - \tau_i(\mathtt{v}_l) \right).$$

3.2.3 Discussion of the presented algorithms

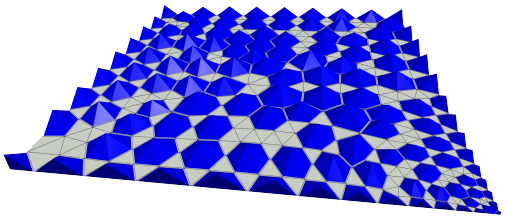
To generate a tent pitched spacetime mesh, one can follow the general framework described in Algorithm 1. Based on this, one has to specify a selection criterion for the next location to pitch, where we gave two possibilities in (3.7) and (3.8). After selecting a vertex to pitch at, one needs an estimate for the allowed advance in time. Here we described two procedures, one based on the gradients along the edges of the tent in §3.2.1 and one calculating the exact gradients of the tent facets in §3.2.2. Next, we discuss the properties of the resulting tent pitching algorithms when combining the mentioned approaches.

The criterion (3.8) to pitch tents only at local minima is very restrictive and leads to a lower number of tents in the resulting time slab, compared to criterion (3.7) selecting vertices where "good" progress can be made. Thinking about parallelization as described in Remark 1, pitching at local minima results in a larger number of layers in the time slab and hence less tents per layer. This does not favor the parallelism which performs best with as many tents per layer as possible. How restrictive the criterion (3.8) is, becomes clear when using it in combination with the volume-based tent pitching algorithm in §3.2.2, where locks can occur – see Remark 2. Having all this in mind, we decided to use the "good" progress criterion (3.7) in the remainder of this thesis.

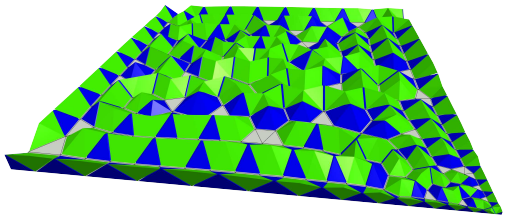
After the selection of a vertex, we can use the volume-based or the edge-based tent pitching algorithm to create the tent. The volume-based algorithm given in §3.2.2 pitches more aggressively, which can cause locks as discussed in Remark 2. In comparison, the edge-based algorithm given in §3.2.1 controls the tent height by one-dimensional gradients along the edges and a shape regularity constant $C_{\mathcal{T}}$. These one-dimensional problems along the edges always allow an advance in time, since for a given vertex $\mathtt{v}_l$, the estimate $\tilde{k}_l^{(i)}$ in (3.6) will always be strictly positive if $\tau_i(e_2) > \tau_i(\mathtt{v}_l)$ for all $e \in \mathcal{E}_l$. With a global shape regularity constant $C_{\mathcal{T}}$, the edge-based algorithm leads to around 30% more tents in the resulting time slab based on a two-dimensional spatial mesh than the volume-based one. Using a local constant $C_{\mathcal{T}}$ for each vertex $\mathtt{v} \in \mathcal{V}$, representing the shape regularity of the vertex patch $\omega_{\mathtt{v}}$, both algorithms generate a comparable number of tents in the resulting time slab.

Using the volume-based algorithm we rarely saw locks occurring with the "good" progress criterion as well. Thus an artificial constant would be needed to control how "aggressive" the volume-based algorithm pitches the tents to guarantee completion.

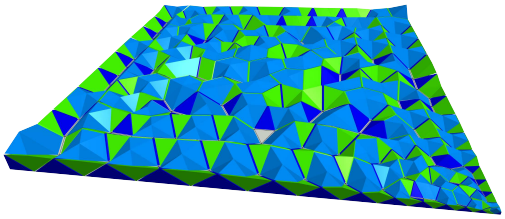
Overall we favor the edge-based algorithm because of its simplicity and the fact that we can ensure completion, due to the reduction to the one-dimensional gradients in (3.5).



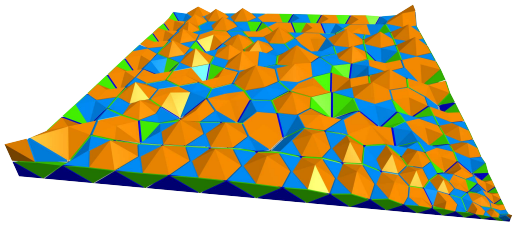
(a) Initial tents forming layer 1



(b) Tents forming layers 1 and 2



(c) Tents forming layers 1, 2 and 3



(d) Tents forming layers 1, 2, 3 and 4

Figure 3.3: Parallel tents within different layers in two spatial dimensions.

Remark 1 (Parallel tent pitching). To pitch multiple tents in parallel, at the ith step, instead of picking l_* arbitrarily as in Algorithm 2, we choose $l_* \in J_{i-1}$ with the property that $\omega_{\mathbf{v}_{l_*}} = \omega_{\mathbf{v}^{(i)}}$ does not intersect $\omega_{\mathbf{v}^{(j)}}$ for all $j < i$. As we step through i, we continue to pick such l_* until we reach an index $i = i_1$ where no such l_* exists. All the tents made until this point, say $K_1, K_2, \ldots, K_{i_1}$ form the *layer* L_1. (An example of tents within such layers are shown in Figure 3.3 – in this example one of the corners of the domain has a singularity.) We then repeat this process to find greater indices $i_2 < i_3 < \cdots$ and layers $L_k = \{K_{i_{k-1}}, K_{i_{k-1}+1}, \ldots, K_{i_k}\}$ with the property that $\omega_{\mathbf{v}^{(j)}}$ does not intersect $\omega_{\mathbf{v}^{(i)}}$ for any distinct i and j in the range $i_{k-1} \leq i, j \leq i_k$. Computations on tents within each layer can proceed in parallel.

Remark 2 (Lock in two spatial dimensions). To show that the criterion (3.8) is to restrictive, we consider a single triangle and set the advancing front so that the element-wise causality condition (3.4) does not allow any advance in time at the position of the local minimum. This example just applies to the volume-based algorithm since the edge-based algorithm relaxes the element-wise constraint to one-dimensional constraints along the edges.

For a general triangle $T = \operatorname{conv}\{(a_1, a_2), (b_1, b_2), (c_1, c_2)\}$ with the coordinates $a_i, b_i, c_i \in \mathbb{R}$, $i = 1, 2$, the advancing front $\tau \in P_1(T)$ is determined by its values $\tau_A = \tau(a_1, a_2)$, $\tau_B = \tau(b_1, b_2)$ and $\tau_C = \tau(c_1, c_2)$ at the vertices A, B and C. Using the affine mapping $\Psi : \hat{T} \to T$, where $\hat{T} = \operatorname{conv}\{(0, 0), (1, 0), (0, 1)\}$ denotes the reference triangle, we obtain

$$\nabla_x \tau = \frac{1}{\det \Psi'} \begin{bmatrix} -c_2 + b_2 & c_2 - a_2 & -b_2 + a_2 \\ c_1 - b_1 & -c_1 + a_1 & b_1 - a_1 \end{bmatrix} \begin{bmatrix} \tau_A \\ \tau_B \\ \tau_C \end{bmatrix}, \tag{3.13}$$

with $\det \Psi' = (b_1 - a_1)(c_2 - a_2) - (b_2 - a_2)(c_1 - a_1)$. For $\varepsilon > 0$ we get an obtuse triangle

$T = \operatorname{conv}\{(0,0),(1,0),(-\varepsilon,1)\}$ and (3.13) yields

$$\nabla_x \tau = \begin{bmatrix} -1 & 1 & 0 \\ -(1+\varepsilon) & \varepsilon & 1 \end{bmatrix} \begin{bmatrix} \tau_A \\ \tau_B \\ \tau_C \end{bmatrix} = \begin{bmatrix} \tau_B - \tau_A \\ \varepsilon(\tau_B - \tau_A) + \tau_C - \tau_A \end{bmatrix}, \tag{3.14}$$

where we used $\det \Psi' = 1$. Assuming a local wave speed $c_T = 1$, we substitute $\tau_A = \delta > 0$, $\tau_B = 0$ into (3.14) and solve $\| \nabla_x \tau \| = 1$ for the highest possible τ_C. For $\delta < 1$ we obtain the solution $\tau_C = \sqrt{1-\delta^2} + \delta(1+\varepsilon)$ leading to a local minimum $\tau_B = 0$ at the vertex B. The criterion (3.8) would now select the vertex B to pitch the next tent. As function of τ_B the gradient of the advancing front reads

$$\nabla_x \tau = \begin{bmatrix} \tau_B - \delta \\ \varepsilon\tau_B + \sqrt{1-\delta^2} \end{bmatrix}.$$

We now want to find $\tau_B > 0$, such that $\| \nabla_x \tau \|$ decreases. Therefore we consider

$$\begin{aligned} \| \nabla_x \tau \|^2 &= (\tau_B - \delta)^2 + \big(\varepsilon\tau_B + \sqrt{1-\delta^2}\big)^2 \\ &= (1+\varepsilon^2)\tau_B^2 + 2\big(\varepsilon\sqrt{1-\delta^2} - \delta\big)\tau_B + 1 \end{aligned}$$

and its derivative

$$\frac{d}{d\tau_B} \| \nabla_x \tau \|^2 = 2(1+\varepsilon^2)\tau_B + 2\big(\varepsilon\sqrt{1-\delta^2} - \delta\big).$$

For $\delta < \frac{\varepsilon}{\sqrt{1+\varepsilon^2}}$ holds $\delta < \varepsilon\sqrt{1-\delta^2}$ and for any $\tau_B > 0$ follows

$$\frac{d}{d\tau_B} \| \nabla_x \tau \|^2 > 0.$$

Thus increasing τ_B to a value larger than 0 would lead to $\| \nabla_x \tau \| > 1$, violating the causality condition. A similar calculation for τ_A shows that increasing τ_A leads to a decrease of $\| \nabla_x \tau \|$. Hence the lock occurs for the local minima criterion (3.8) while the "good" progress criterion (3.7) could select vertex A to proceed in time.

3.3 Mapped tent pitching

In this section we discuss a mapping technique that allows us to separate space and time discretizations within tents. Domains like $\Omega_0 \times (0,T)$ formed by a tensor product of a spatial domain with a time interval are referred to as spacetime cylinders. This tensor product structure is used in many numerical methods since space and time discretizations neatly separate. However, the tent

$$K_i = \{(x,t) : \ x \in \omega_{\mathbf{v}^{(i)}}, \ \tau_{i-1}(x) < t < \tau_i(x)\}, \tag{3.2}$$

is not of this form. Therefore, we now introduce a mapping that transforms K_i one-to-one onto the spacetime cylinder $\hat{K}_i = \omega_{\mathbf{v}^{(i)}} \times (0,1)$.

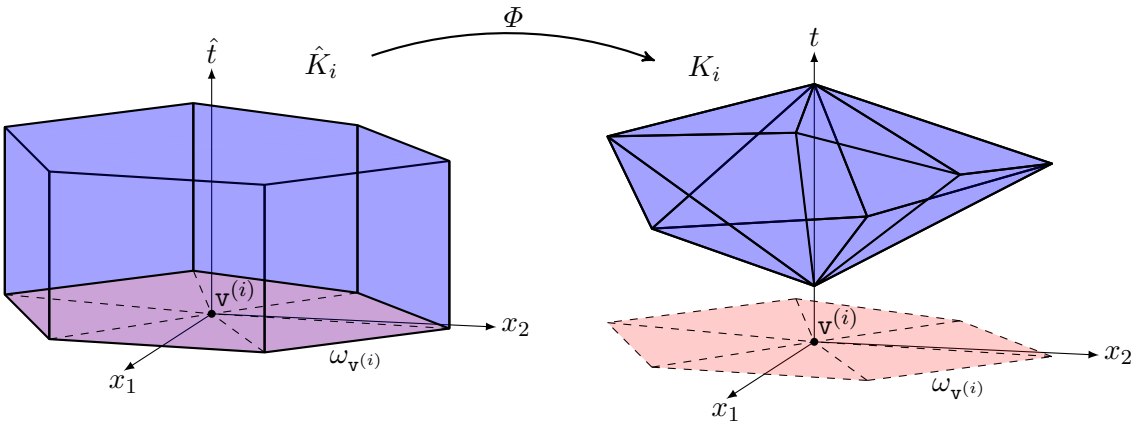


Figure 3.4: Tent K_i in two spatial dimensions mapped from a tensor product domain $\hat{K}_i$.

Define the mapping $\Phi : \hat{K}_i \to K_i$ (see Figure 3.4) by $\Phi(x, \hat{t}) = (x, \varphi(x, \hat{t}))$, where

$$\varphi(x, \hat{t}) = (1 - \hat{t})\tau_{i-1}(x) + \hat{t}\tau_i(x), \tag{3.15}$$

for all $(x, \hat{t})$ in $\hat{K}_i$. Note that the $(N + 1) \times (N + 1)$ Jacobian matrix of derivatives of Φ takes the form

$$\Phi' = \begin{bmatrix} I & 0 \\ (\nabla_x \varphi)^\intercal & \delta \end{bmatrix}, \tag{3.16}$$

where $\delta(x) = \tau_i(x) - \tau_{i-1}(x)$ is a linear function in space describing the tent height and I denotes the $N \times N$ identity matrix. In analogy to the abbreviated notation $\partial_i = \partial/\partial x_i$ for spatial derivatives, we use the abbreviation $\partial_{\hat{t}} = \partial/\partial\hat{t}$ for the pseudo-temporal derivative on $\hat{K}_i$.

Theorem 1. *The function $u : K_i \to \mathbb{R}^L$ satisfies (2.1) if and only if $\hat{u} = u \circ \Phi : \hat{K}_i \to \mathbb{R}^L$ satisfies*

$$\partial_{\hat{t}}\big(g(x, \hat{t}, \hat{u}(x, \hat{t})) - f(x, \hat{t}, \hat{u}(x, \hat{t}))\, \nabla_x \varphi(x, \hat{t})\big) + \operatorname{div}_x \big(\delta(x) f(x, \hat{t}, \hat{u}(x, \hat{t}))\big) = 0 \tag{3.17}$$

for all $(x, \hat{t})$ in $\hat{K}_i$. In component form, (3.17) reads

$$\partial_{\hat{t}}\Big(g_l(x, \hat{t}, \hat{u}(x, \hat{t})) - \sum_{i=1}^{N} f_{li}(x, \hat{t}, \hat{u}(x, \hat{t}))\partial_i\varphi(x, \hat{t})\Big) + \sum_{i=1}^{N} \partial_i\big(\delta(x) f_{li}(x, \hat{t}, \hat{u}(x, \hat{t}))\big) = 0 \tag{3.18}$$

for all $(x, \hat{t})$ in $\hat{K}_i$ and all $l = 1, \ldots, L$.

Proof. The proof proceeds by calculating the component-wise pull back of the system (2.2) from K_i to $\hat{K}_i$ using the map Φ. Using the given u, define $F_l : K_i \to \mathbb{R}^{N+1}$ by

$$F_l(x, t) = \begin{bmatrix} f_{l1}(x, t, u(x, t)) \\ \vdots \\ f_{lN}(x, t, u(x, t)) \\ g_l(x, t, u(x, t)) \end{bmatrix},$$

and the pullback on $\hat{K}_i$ by

$$\hat{F}_l = (\det \Phi')\,(\Phi')^{-1}(F_l \circ \Phi).$$

By the well-known properties of the Piola map,

$$\operatorname{div}_{(x,\hat{t})} \hat{F}_l = (\det \Phi')\,(\operatorname{div}_{(x,t)} F_l) \circ \Phi, \tag{3.19}$$

where the divergence on either side is now taken in spacetime ($\mathbb{R}^{N+1}$). Note that $\det \Phi' = \delta$ is never zero at any point of (the open set) $\hat{K}_i$. Writing equation (2.2) in these new notations, we obtain $(\operatorname{div}_{(x,t)} F_l)(x,t) = 0$ for all $(x,t) \in K_i$, or equivalently,

$$(\operatorname{div}_{(x,t)} F_l)(\Phi(x,\hat{t})) = 0$$

for all $(x,\hat{t}) \in \hat{K}_i$. Multiplying through by $\det \Phi'$ and using (3.19), this becomes

$$\operatorname{div}_{(x,\hat{t})} \hat{F}_l(x,\hat{t}) = 0, \qquad \text{on } \hat{K}_i. \tag{3.20}$$

To finish the proof, we simplify this equation. Inverting the block triangular matrix Φ' displayed in (3.16) and using it in the definition for $\hat{F}_l$, we obtain

$$\hat{F}_l = (\det \Phi')\,\delta^{-1} \begin{bmatrix} \delta I & 0 \\ -(\nabla_x \varphi)^\intercal & 1 \end{bmatrix} (F_l \circ \Phi) = \begin{bmatrix} \delta \hat{f}_l \\ \hat{g}_l - \nabla_x \varphi \cdot \hat{f}_l \end{bmatrix},$$

where $\hat{f}_l$ is the vector whose ith component is $f_{li}(x,\hat{t},\hat{u}(x,\hat{t}))$ and $\hat{g}_l$ denotes the lth component of $g(x,\hat{t},\hat{u}(x,\hat{t}))$. Substituting these into (3.20) and expanding, we obtain (3.18). □

In the following, we consider a general tent

$$K = \{(x,t) : x \in \omega_{\mathrm{v}}, \varphi_b(x) \le t \le \varphi_t(x)\} \tag{3.21}$$

over any given vertex patch ω_{v}. The functions φ_b and φ_t denote the bottom and top advancing fronts restricted to the vertex patch ω_{v}. Theorem 1 maps the hyperbolic system to the cylinder $\hat{K} = \omega_{\mathrm{v}} \times (0,1)$, which opens up the possibility to construct tensor product discretizations – rather than spacetime discretizations – within each tent K. For readability, we omit the spatial variable x and pseudo-time $\hat{t}$ from the arguments of functions in (3.17) and simply write

$$\partial_{\hat{t}}\,(g(\hat{u}) - f(\hat{u})\,\nabla_x \varphi) + \operatorname{div}_x\,(\delta f(\hat{u})) = 0, \tag{3.22}$$

which describes the evolution of $\hat{u}$ along pseudo-time from $\hat{t} = 0$ to $\hat{t} = 1$. Since

$$\varphi(x,\hat{t}) = (1-\hat{t})\varphi_b(x) + \hat{t}\varphi_t(x) = \varphi_b(x) + \hat{t}\delta(x),$$

we may split $g(\hat{u}) - f(\hat{u})\,\nabla_x \varphi$ into parts with and without explicit dependence on pseudo-time, allowing us to rewrite (3.22) as

$$\partial_{\hat{t}}\left((g(\hat{u}) - f(\hat{u})\,\nabla_x \varphi_b) - \hat{t} f(\hat{u})\,\nabla_x \delta\right) + \operatorname{div}_x\,(\delta f(\hat{u})) = 0. \tag{3.23}$$

Further we define

$$M_0(w) = g(w) - f(w)\,\nabla_x\,\varphi_b, \tag{3.24a}$$
$$M_1(w) = f(w)\,\nabla_x\,\delta, \tag{3.24b}$$
$$M(\hat{t}, w) = M_0(w) - \hat{t}M_1(w). \tag{3.24c}$$

Equation (3.22) is the starting point for our spatial discretization. Let $P_p(T)$ denote the space of polynomials of degree at most p in x, restricted to the N-simplex T. We use a discontinuous Galerkin method based on

$$V_h = \{v \in [L_2(\Omega_0)]^L : v|_T \in [P_p(T)]^L \;\forall\, T \in \mathcal{T}\}.$$

When restricted to the vertex patch $\omega_{\mathtt{v}}$ we obtain

$$V_h^{\mathtt{v}} = \{v|_{\omega_{\mathtt{v}}} : v \in V_h\} = \{v \in [L_2(\Omega_0)]^L : v|_T \in [P_p(T)]^L \;\forall\, T \in \mathcal{T}_{\mathtt{v}}\}, \tag{3.25}$$

where $\mathcal{T}_{\mathtt{v}} \subset \mathcal{T}$ denotes the spatial mesh of the vertex patch $\omega_{\mathtt{v}}$. Multiplying (3.22) by a test function $v_h \in V_h^{\mathtt{v}}$ and integrating by parts over the patch $\omega_{\mathtt{v}}$, we obtain

$$\int_{\omega_{\mathtt{v}}} \partial_{\hat{t}} M(\hat{t}, \hat{u}) \cdot v_h = \sum_{T \in \mathcal{T}_{\mathtt{v}}} \int_T \delta f(\hat{u}) : \nabla v_h - \sum_{F \in \mathcal{F}_{\mathtt{v}}} \int_F \delta f_n(\hat{u}^+, \hat{u}^-) \cdot [\![v_h]\!], \tag{3.26}$$

for all $v_h \in V_h^{\mathtt{v}}$ and all $\hat{t} \in [0,1]$. The set of facets F, i.e., $(N-1)$-subsimplices, of the simplicial mesh $\mathcal{T}_{\mathtt{v}}$ of the vertex patch $\omega_{\mathtt{v}}$ is denoted by $\mathcal{F}_{\mathtt{v}}$. Here and throughout, every facet F is assigned a unit normal, simply denoted by n, whose direction is arbitrarily fixed, except when $F \subset \partial\Omega$, in which case it points outward. The traces $\hat{u}^+$ and $\hat{u}^-$ of $\hat{u}$ from either side are defined by

$$\hat{u}^+ = \lim_{s \to 0^+} \hat{u}(x + sn) \quad \text{and} \quad \hat{u}^- = \lim_{s \to 0^+} \hat{u}(x - sn).$$

In (3.26), we also used a numerical flux f_n on each facet F (that takes values in $\mathbb{R}^L$ depending on values $\hat{u}^+, \hat{u}^-$ from either side) and the jump $[\![\hat{v}_h]\!] = \hat{v}_h^+ - \hat{v}_h^-$. Further we define the mean value on the facet F by $\{\hat{u}\} = \frac{1}{2}(\hat{u}^+ + \hat{u}^-)$. In these definitions, whenever $\hat{u}^+$ falls outside Ω, it is prescribed using some given boundary conditions.

Let $m = \dim V_h^{\mathtt{v}}$ and let ψ_i, $i = 1, \dots, m$ denote any standard local basis for $V_h^{\mathtt{v}}$. Introducing $\mathtt{u} : [0,1] \to \mathbb{R}^m$, we obtain the basis expansion

$$\hat{u}_h(x, \hat{t}) = \sum_{i=1}^{m} \mathtt{u}_i(\hat{t})\psi_i(x). \tag{3.27}$$

Further we define the maps $\mathtt{M}_0, \mathtt{M}_1$, and $\mathtt{A}$ on $\mathbb{R}^m$ by

$$[\mathtt{M}_0(\mathtt{w})]_j = \int_{\omega_{\mathtt{v}}} M_0\Big(\sum_{i=1}^{m} \mathtt{w}_i\psi_i(x)\Big)\psi_j(x)\,, \tag{3.28a}$$
$$[\mathtt{M}_1(\mathtt{w})]_j = \int_{\omega_{\mathtt{v}}} M_1\Big(\sum_{i=1}^{m} \mathtt{w}_i\psi_i(x)\Big)\psi_j(x)\,, \tag{3.28b}$$
$$[\mathtt{A}(\mathtt{w})]_j = A\Big(\sum_{i=1}^{m} \mathtt{w}_i\psi_i, \psi_j\Big), \tag{3.28c}$$

with

$$A(w, v_h) := \sum_{T\in\mathcal{T}_v} \int_T \delta f(w) : \nabla v_h - \sum_{F\in\mathcal{F}_v} \int_F \delta f_n(w^+, \hat{w}^-) \cdot [\![v_h]\!] \,. \tag{3.29}$$

With these notations, (3.26) becomes the semi-discrete problem of finding $\mathtt{u} : [0,1] \to \mathbb{R}^m$ satisfying

$$\frac{d}{d\hat{t}} \mathtt{M}(\hat{t}, \mathtt{u}(\hat{t})) = \mathtt{A}(\mathtt{u}(\hat{t})), \tag{3.30}$$

given some $\mathtt{u}(0) = \mathtt{u}_0 \in \mathbb{R}^m$. Here $\mathtt{M} : [0,1] \times \mathbb{R}^m \to \mathbb{R}^m$ is defined by

$$\mathtt{M}(\hat{t}, \mathtt{w}) := \mathtt{M}_0(\mathtt{w}) - \hat{t}\,\mathtt{M}_1(\mathtt{w}), \qquad 0 \le \hat{t} \le 1, \quad \mathtt{w} \in \mathbb{R}^m.$$

The nonstandard feature of (3.30) is that $\mathtt{M}$ is an affine-linear function of the pseudo-time $\hat{t}$, since our mapping enters through $\nabla_x \varphi$ in (3.24).

3.4 Time stepping within tents

To get a better understanding of the nonstandard feature of the semi-discrete ODE (3.30), we considering a general linear problem, as described in (2.5). The functions g and f can be expressed by symmetric matrix-valued functions $A^{(i)} : \Omega \to \mathbb{R}^{L\times L}$, for $i = 1, \dots, N+1$, where $A^{(t)} \equiv A^{(N+1)}$ is symmetric positive definite – see (2.4). The linear hyperbolic problem takes the form

$$\partial_t\big(A^{(t)}u\big) + \sum_{i=1}^{N} \partial_i\big(A^{(i)}u\big) = 0 \tag{2.5}$$

on the tent K. By Theorem 1, we obtain the mapped problem

$$\partial_{\hat{t}}\Big[\Big(A^{(t)} - \sum_{i=1}^{N} A^{(i)}\partial_i\varphi\Big)\hat{u}\Big] + \sum_{i=1}^{N} \partial_i\big(\delta A^{(i)}\hat{u}\big) = 0 \tag{3.31}$$

on the cylinder $\hat{K}$. In this linear setting $M(\hat{t}, \hat{u})$ defined in (3.24) reads

$$M(\hat{t}, \hat{u}) = \Big(A^{(t)} - \sum_{i=1}^{N} A^{(i)}\partial_i\varphi\Big)\hat{u}. \tag{3.32}$$

Thus we obtain $\mathtt{M}(\hat{t}, \mathtt{u}(\hat{t})) = \mathtt{M}(\hat{t})\mathtt{u}(\hat{t})$, where $\mathtt{M}(\hat{t}) \in \mathbb{R}^{m\times m}$ corresponds to a mass matrix whose entries are

$$\mathtt{M}_{kl}(\hat{t}) = \int_{\omega_v} M(\hat{t}, \psi_k(x))\psi_l(x) = \int_{\omega_v} \Big(A^{(t)} - \sum_{i=1}^{N} A^{(i)}\partial_i\varphi\Big)(x, \hat{t})\psi_k(x)\psi_l(x). \tag{3.33}$$

Then, (3.30) yields

$$\frac{d}{d\hat{t}}\big(\mathtt{M}(\hat{t})\mathtt{u}(\hat{t})\big) = \mathtt{A}(\mathtt{u}(\hat{t})). \tag{3.34}$$

This allows the usage of either explicit of implicit Runge-Kutta methods for the temporal discretization. In §3.4.1, we apply explicit Runge-Kutta methods to the system (3.34),

which do not give the expected orders of convergence. This was first reported in [11], where we introduced a Structure Aware Taylor (SAT) time stepping to overcome this issue for linear problems. The idea of structure aware time stepping methods was then extended to nonlinear problems by Structure Aware Runge Kutta (SARK) methods in [14]. These structure aware time stepping methods will be discussed in chapter 4. In §3.4.2, we apply implicit Runge-Kutta methods to the system (3.34). These methods do not show the reduced convergence order, but they are memory intensive due to the rather large local problems.

3.4.1 Explicit time stepping

With the local discontinuous Galerkin space

$$V_h^{\mathtt{v}} = \{\psi \in [L^2(\omega_{\mathtt{v}})]^L : \psi|_T \in [P_p(T)]^L \ \forall T \in \mathcal{T}_{\mathtt{v}}\},$$

we obtain the semi-discrete ODE (3.34). Introducing the new variable $\mathtt{Y}(\hat{t}) := \mathtt{M}(\hat{t})\mathtt{u}(\hat{t})$, we can rephrase (3.34) as

$$\frac{d}{d\hat{t}}\mathtt{Y}(\hat{t}) = \mathtt{A}\big(\mathtt{M}(\hat{t})^{-1}\mathtt{Y}(\hat{t})\big). \tag{3.35}$$

This form allows the direct application of standard ODE solvers, such as explicit Runge-Kutta methods. As model problem, we consider the two-dimensional wave equation

$$\partial_{tt}\phi - \operatorname{div}_x(\alpha\, \nabla_x\, \phi) = 0 \tag{3.36a}$$

on the spacetime cube $\Omega = [0,\pi]^2 \times [0, t_{\max}]$, with $t_{\max} = \sqrt{2}\pi$. The material parameter α is set to the identity matrix, such that the speed of propagation is $c_s = 1$. It is easy to see that the classical standing wave

$$\phi(x,t) = \cos(x_1)\cos(x_2)\sin(t\sqrt{2})/\sqrt{2}, \tag{3.36b}$$

is a solution of (3.36a). Written as hyperbolic system – see (2.8) – we obtain

$$u(x,t) = \begin{bmatrix} q(x,t) \\ \mu(x,t) \end{bmatrix} = \begin{bmatrix} -(\nabla_x\,\phi)(x,t) \\ (\partial_t \phi)(x,t) \end{bmatrix} = \begin{bmatrix} \sin(x_1)\cos(x_2)\sin(t\sqrt{2})/\sqrt{2} \\ \cos(x_1)\sin(x_2)\sin(t\sqrt{2})/\sqrt{2} \\ \cos(x_1)\cos(x_2)\cos(t\sqrt{2}) \end{bmatrix}. \tag{3.36c}$$

The initial data q_0 and μ_0 is set to

$$q_0 = 0, \qquad \mu_0 = \cos(x_1)\cos(x_2), \tag{3.36d}$$

for $(x_1, x_2) \in \Omega_0$. Further we set the boundary conditions $q \cdot n = 0$ on $\partial\Omega_0 \times (0, t_{\max})$, where n denotes the spatial component of the outward unit normal and we set the numerical flux f_n in (3.29) to the upwind flux

$$f_n(\hat{u}^+, \hat{u}^-) = \begin{bmatrix} 0 & n \\ n^\intercal & 0 \end{bmatrix} \{\hat{u}\} + \frac{1}{2}\begin{bmatrix} n n^\intercal & 0 \\ 0 & 1 \end{bmatrix} [\![\hat{u}]\!]. \tag{3.37}$$

Before we apply the time stepping, we subdivide the interval $[0,1]$ into r subintervals

$$[\hat{t}_k, \hat{t}_{k+1}], \qquad k = 0, 1, \dots r-1, \qquad \text{where } \hat{t}_k = \frac{k}{r}.$$

Thus we obtain subintervals of the size $\tau^{[k]} = \hat{t}_{k+1} - \hat{t}_k$. Then we set the initial data $\mathtt{Y}^{[0]} = \mathtt{M}(0)\mathtt{u}(0)$ and perform the time stepping for $k = 0, 1, \dots, r-1$:

$$\mathtt{k}_i^{[k]} = \mathtt{A}\Big(\mathtt{Y}^{[k]} + \tau^{[k]} \sum_{j=1}^{i-1} a_{ij} \mathtt{Ak}_j^{[k]}\Big), \qquad 1 \le i \le s,$$

$$\mathtt{Y}^{[k+1]} = \mathtt{Y}^{[k]} + \tau^{[k]} \sum_{i=1}^{s} b_i \mathtt{Ak}_i^{[k]}.$$

After the final step, we obtain $\mathtt{Y}^{[r]}$ as approximation to $\mathtt{Y}(1)$ at the tent top. The explicit Runge-Kutta method is determined by the coefficient matrices

$$b = (b_1, \dots, b_s) \in \mathbb{R}^{s\times 1}, \quad \text{and} \quad \mathcal{A} = \begin{pmatrix} 0 & & & \\ a_{21} & 0 & & \\ \vdots & \ddots & 0 & \\ a_{s1} & \dots & a_{s,s-1} & 0 \end{pmatrix} \in \mathbb{R}^{s\times s},$$

usually expressed by the standard Butcher tableau $\begin{array}{c|c} c & \mathcal{A} \\ \hline & b \end{array}$. The remaining coefficients $c \in \mathbb{R}^s$ are set by the consistency condition

$$c_i = \sum_{j=1}^{i-1} a_{ij}.$$

Based on a spatial mesh with mesh size h, we generate a tent pitched mesh using the edge-based algorithm in §3.2.1 with $c_e = 2$ and $C_\tau = \frac{1}{2}$. The maximal slope $\| \nabla_x \varphi\|$ is bounded by 0.494 and we apply a discontinuous Galerkin method in space using polynomials of degree p, with $1 \le p \le 4$. On each cylinder we perform the classical fourth order Runge-Kutta (RK4) method, given by the coefficients in Table 3.1, with $r = 2p$ subintervals. Letting $q_h(x)$ and $\mu_h(x)$ denote the computed solution at the final time $t_{\max}$, we measure the spatial L_2-error e_h of all field components by

$$e_h^2 := \|q(x, t_{\max}) - q_h(x)\|^2_{L_2(\Omega_0)} + \|\mu(x, t_{\max}) - \mu_h(x)\|^2_{L_2(\Omega_0)}. \tag{3.38}$$

The errors are reported in Figure 3.5, where we observe that the convergence rates drop to first order after showing slightly higher rates for the first refinement steps. For $2 \le p \le 4$, the error goes to zero at the rate of $\mathcal{O}(h)$, where the rate drops earlier for higher polynomial degree p. In the case $p = 1$, we see a reduction of the convergence rate to 1.48 in the last refinement step. We observe higher rates for larger mesh sizes because the error of the spatial discretization dominates, while the error arising from the temporal discretization comes into play when h tends to 0, causing the drop in the convergence rate.

0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
1	0	0	1	0
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Table 3.1: Butcher tableau of the classical forth order Runge-Kutta (RK4) method.

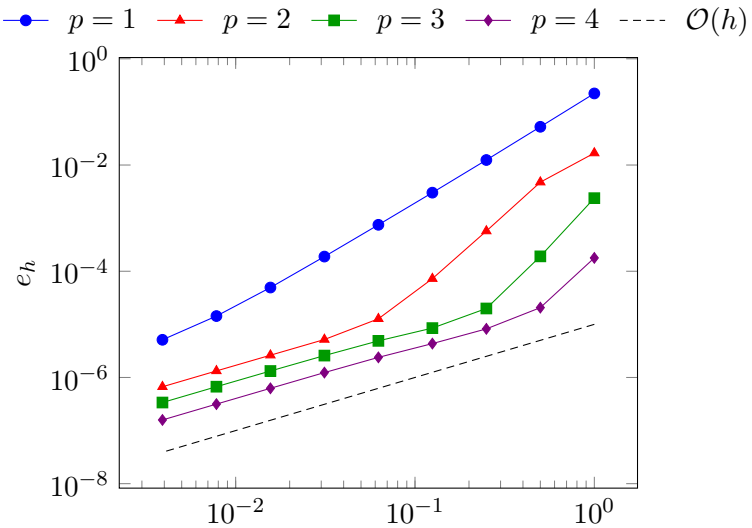


Figure 3.5: Spatial L_2-error e_h of all field components for a two-dimensional standing wave over the mesh size h when using the RK4 method.

3.4.2 Implicit time stepping

For linear problems, implicit Runge-Kutta methods can be used to solve the local problem on each tent. In the following, we consider the two-dimensional wave equation as described in (3.36) and discuss the construction of locally implicit MTP methods. Here we use the Brezzi-Douglas-Marini (BDM) mixed method and set

$$V_h^{\mathtt{v}} := \{\psi \equiv (r,\eta) \in H(\operatorname{div},\omega_{\mathtt{v}}) \times L^2(\omega_{\mathtt{v}}) : r|_T \in [P_p(T)]^N, \eta|_T \in P_p(T)\ \forall\, T \in \mathcal{T}_{\mathtt{v}}\}.$$

Using the definitions (2.8), the system (3.31) reads

$$\partial_{\hat{t}} \begin{bmatrix} \alpha^{-1} & -\nabla_x \varphi \\ -\nabla_x \varphi^{\intercal} & 1 \end{bmatrix} \begin{bmatrix} \hat{q} \\ \hat{\mu} \end{bmatrix} + \begin{bmatrix} \nabla_x(\delta\hat{\mu}) \\ \operatorname{div}_x(\delta\hat{q}) \end{bmatrix} = 0. \tag{3.39}$$

Multiplying (3.39) by $\psi \equiv (r,\eta)$ and integrating the first equation by parts, we obtain

$$\frac{d}{d\hat{t}} \int_{\omega_{\mathtt{v}}} \left(\begin{bmatrix} \alpha^{-1} & -\nabla_x \varphi \\ -\nabla_x \varphi^{\intercal} & 1 \end{bmatrix} \begin{bmatrix} \hat{q} \\ \hat{\mu} \end{bmatrix} \right) \cdot \begin{bmatrix} r \\ \eta \end{bmatrix} = \int_{\omega_{\mathtt{v}}} \begin{bmatrix} \delta\hat{\mu} \\ -\operatorname{div}_x(\delta\hat{q}) \end{bmatrix} \cdot \begin{bmatrix} \operatorname{div}_x r \\ \eta \end{bmatrix}$$

for all $(r,\eta) \in V_h^{\mathtt{v}}$. Using a basis $\psi_i \equiv (r_i,\eta_i)$, $i = 1,\dots,m$, of $V_h^{\mathtt{v}}$, we can define the matrices

$$\mathtt{M}_{ij}(\hat{t}) = \int_{\omega_{\mathtt{v}}} \left(\begin{bmatrix} \alpha^{-1} & -\nabla_x \varphi \\ -\nabla_x \varphi^{\intercal} & 1 \end{bmatrix} \begin{bmatrix} r_j \\ \eta_j \end{bmatrix} \right) \cdot \begin{bmatrix} r_i \\ \eta_i \end{bmatrix},$$

$$\mathtt{B}_{ij} = \int_{\omega_{\mathtt{v}}} \begin{bmatrix} \delta\eta_j \\ -\operatorname{div}_x(\delta r_j) \end{bmatrix} \cdot \begin{bmatrix} \operatorname{div}_x r_i \\ \eta_i \end{bmatrix},$$

to obtain the ODE system

$$\frac{d}{d\hat{t}} \left(\mathtt{M}(\hat{t})\mathtt{u}(\hat{t}) \right) = \mathtt{B}\mathtt{u}(\hat{t}) \qquad 0 \le \hat{t} \le 1, \tag{3.40}$$

for the coefficient vector $\mathtt{u}(\hat{t}) \in \mathbb{R}^m$ of the basis expansion $\hat{u}_h(x,\hat{t}) = \sum_i \mathtt{u}_i(\hat{t})\psi_i(x)$. The ODE in (3.40) has the same structure as (3.34), which we would have obtained using a space $V_h^{\mathtt{v}} \subset [L_2(\omega_{\mathtt{v}})]^L$ for the spatial discretization. The advantages of the mixed method are that it does not require a numerical flux and the sparsity pattern allows a more efficient inversion of the arising matrices in the implicit Runge-Kutta scheme.

For the temporal discretization of (3.40), we use an implicit high order s-stage Runge-Kutta method of Radau IIA type [20, Chapter IV.5]. Note that due to the implicit nature of the scheme, there is no CFL constraint on the number of stages (within the mapped tent), irrespective of the polynomial degree p of the spatial discretization. These Runge-Kutta methods are characterized by coefficients a_{lm} and c_l for $l,m = 1,\dots,s$ (forming entries of a Butcher tableau) with the property that $c_s = 1$. The remaining c_l are determined by the roots of appropriate Jacobi polynomials. With $\mathtt{M}^{[l]} = \mathtt{M}(\hat{t}_l)$ and the approximation $\mathtt{u}^{[l]}$ to $\mathtt{u}(\hat{t}_l)$, we obtain the linear system

$$\mathtt{M}^{[l]}\mathtt{u}^{[l]} = \mathtt{M}^{[0]}\mathtt{u}^{[0]} + \sum_{m=1}^{s} a_{lm}\mathtt{B}\mathtt{u}^{[m]}, \qquad l = 1,\dots,s, \tag{3.41}$$

which can be easily solved for the final stage solution $\mathfrak{u}^{[s]}$, given $\mathfrak{u}^{[0]}$.

We report the results obtained for (3.39) applied to the standing wave described in (3.36). The final time $t_{\max} = \sqrt{2}\pi$ and the spatial domain $\Omega_0 = [0,\pi]^2$ is meshed by simplices using a mesh size h. The parameters to be varied in each experiment are the spatial mesh size $h = 2^{-l}, l = 0,\dots,7$ and the polynomial degree $1 \le p \le 4$ of the space discretization. The tent meshing algorithm in §3.2.1 is driven by an input wave speed $c_e = 1$ and $C_\tau = \frac{1}{2}$ (leading to maximal slope $\| \nabla_x \varphi\| \approx 0.722$) to mesh a time slab of size $t_{\max} \cdot 2^{-l}/8$. This time slab is stacked in time to mesh the entire spacetime region of simulation $\Omega_0 \times (0, t_{\max})$. Letting $q_h(x)$ and $\mu_h(x)$ denote the computed solutions at time $t = t_{\max}$, we measure the error norm e_h defined by

$$e_h^2 = \|q(\cdot, t_{\max}) - q_h\|^2_{L^2(\Omega_0)} + \|\mu(\cdot, t_{\max}) - \mu_h\|^2_{L^2(\Omega_0)}.$$

Figure 3.6a shows the convergence history for a fixed polynomial degree $p = 3$. We observe that the rate is limited by the number of stages s of the Radau IIA method and the polynomial degree p. Thus we set the number stages $s = p$ for further observations. These are compiled in Figure 3.6b, where the values of e_h as a function of degree p and h are plotted. We observe that e_h appears to go to 0 at a rate of $O(h^p)$.

Next, we consider the case of three spatial dimensions, where $\Omega_0 = [0,\pi]^3$ is spatially meshed with tetrahedral elements of size $h = 2^{-l+1}$ for $l = 0,\dots,l_{\max}$, with

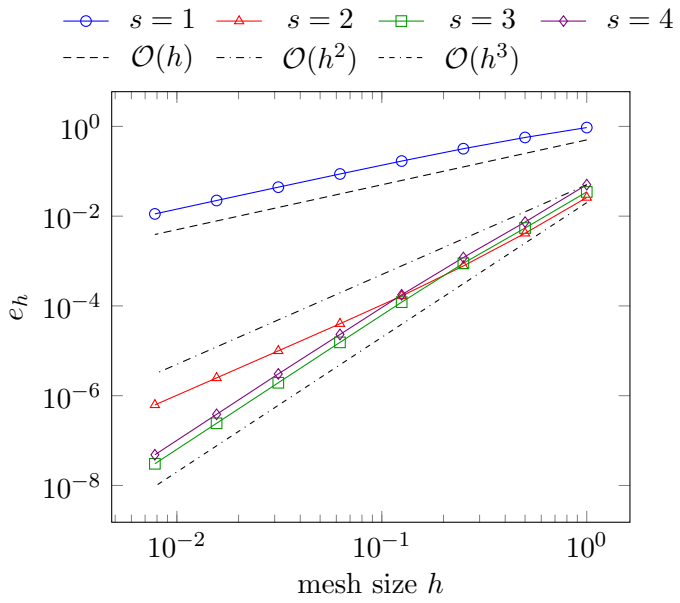
$$l_{\max} = \begin{cases} 5, & p = 1, \\ 4, & p = 2, \\ 3, & p = 3. \end{cases}$$

Here, the number of refinement levels $l_{\max}$ is limit by the available 320GB shared memory. Again, we use the algorithm in §3.2.1 with wave speed $c_e = 1$ and $C_\tau = \frac{1}{3}$ to generate the tents in the time slab of size $t_{\max} \cdot 2^{-l}/8$, leading to maximal slope $\| \nabla_x \varphi\| \approx 0.979$. The final time $t_{\max} = \sqrt{3}\pi$ and the exact solution is $\phi(x,t) = \cos(x_1)\cos(x_2)\cos(x_3)\sin(t\sqrt{3})/\sqrt{3}$. This leads to the system

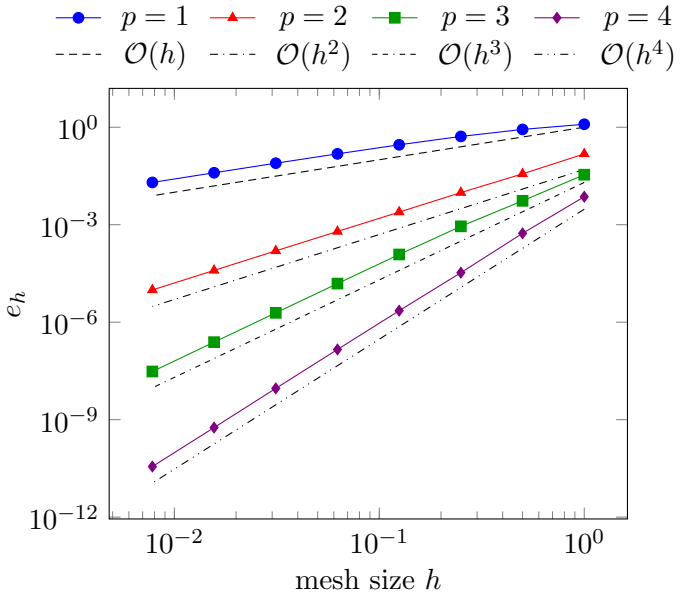
$$u(x,t) = \begin{bmatrix} q(x,t) \\ \mu(x,t) \end{bmatrix} = \begin{bmatrix} \sin(x_1)\cos(x_2)\cos(x_3)\sin(t\sqrt{3})/\sqrt{3} \\ \cos(x_1)\sin(x_2)\cos(x_3)\sin(t\sqrt{3})/\sqrt{3} \\ \cos(x_1)\cos(x_2)\sin(x_3)\sin(t\sqrt{3})/\sqrt{3} \\ \cos(x_1)\cos(x_2)\cos(x_3)\cos(t\sqrt{3}) \end{bmatrix}, \tag{3.42}$$

which is used to set the initial values $q_0 = q(\cdot, 0)$ and $\mu_0 = \mu(\cdot, 0)$. The convergence history plotted in Figure 3.7 shows that e_h, just as in the previous case, goes to zero at a rate of $O(h^p)$. Note that the spacetime mesh of tents is now formed by *four-dimensional* simplices. They are represented by the (three-dimensional) spatial mesh and the time coordinates of the tent pole endpoints. Hence the storage requirements are those of a three-dimensional mesh and not of an general four-dimensional simplicial mesh.

Although seeing high order rates $O(h^p)$ for the spatial error e_h at the final time for these locally implicit MTP methods, we observe that the available memory is a limiting factor.



(a) Rates for spatial polynomial degree $p = 3$ and an implicit Radau IIA method with $s = 1, \ldots, 4$ stages.



(b) Rate for spatial polynomial degrees $p = 1, \ldots, 4$ and an implicit Radau IIA method with $s = p$.

Figure 3.6: Convergence rates for a standing wave in two spatial dimension using implicit Radau IIA methods for the time stepping.

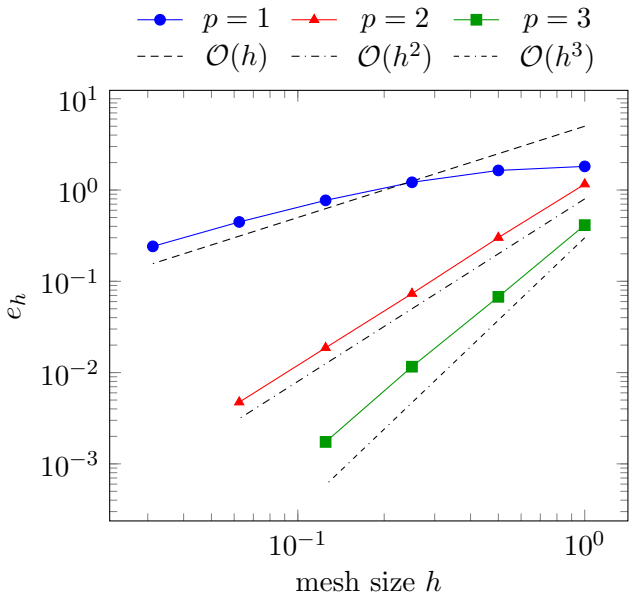


Figure 3.7: Convergence rates for a standing wave in three spatial dimension using implicit Radau IIA methods for the time stepping.

3.5 Inverse map

An important detail of the fully explicit MTP schemes, we did not discuss so far, is the inversion of $\mathtt{M}$. This could be done by solving $\mathtt{Y}(\hat{t}) = \mathtt{M}(\hat{t}, \mathtt{u}(\hat{t}))$ for $\mathtt{u}(\hat{t})$, but in many cases it is possible to calculate an explicit inverse. Therefore we introduce the function $\hat{y} = M(\hat{t}, \hat{u})$ and its basis expansion

$$\hat{y}_h(x, \hat{t}) = \sum_{i=1}^{m} \mathtt{y}_i(\hat{t}) \psi_i(x).$$

With the mass matrix

$$\mathtt{G}_{ij} := \int_{\omega_\mathtt{v}} \psi_i(x) \psi_j(x), \qquad 1 \le i, j \le m,$$

we obtain the relation $\mathtt{G}\mathtt{y}(\hat{t}) = \mathtt{Y}(\hat{t})$ for the vectors $\mathtt{y}(\hat{t})$ and $\mathtt{Y}(\hat{t})$. At any pseudo-time $\hat{t}$, given a $\hat{y}_h \in V_h^\mathtt{v}$ whose coefficient vector in the basis expansion is $\mathtt{y}(\hat{t}) = \mathtt{G}^{-1}\mathtt{Y}(\hat{t})$, we have to solve $\mathtt{Y}(\hat{t}) = \mathtt{M}(\hat{t}, \mathtt{u}(\hat{t}))$ for $\mathtt{u}(\hat{t})$. This equation, in variational form, is

$$\int_{\omega_\mathtt{v}} M(\hat{t}, \hat{u}_h) \cdot v_h = \int_{\omega_\mathtt{v}} \hat{y}_h \cdot v_h, \qquad \text{for all } v_h \in V_h^\mathtt{v}. \tag{3.43}$$

This variational form is used in §3.5.1 to prove the existence of a solution $\hat{u}_h \in V_h^\mathtt{v}$ for linear problems on a tent K satisfying the strict causality condition

$$\|\nabla_x \varphi\| < \frac{1}{c}, \tag{3.44}$$

with the maximal characteristic speed c. The causality condition clearly holds for the function φ, defined as convex combination of two advancing fronts in (3.15). The strict bound in (3.44) can be guaranteed by choosing $\bar{c}$ in (3.3) so that $\bar{c} > c$ when pitching the tent K.

Whenever the inverse $M^{-1}(\hat{t}, \cdot)$ of $M(\hat{t}, \cdot)$ is at hand in closed form or some other computationally convenient form, we use this form to obtain the solution $\hat{u} = M^{-1}(\hat{t}, \hat{y})$. We then perform a projection into $V_h^{\mathfrak{v}}$ to obtain the coefficients $\mathfrak{u}(\hat{t})$. For uncurved elements, this just involves the inversion of a diagonal mass matrix. For the small number of curved elements, we use a highly optimized algorithm which uses an approximation instead of the exact inverse mass matrix. These inverse maps for some important nonlinear problems are discussed in §3.5.2.

3.5.1 Linear examples

Before we show the existence of a solution of the variational form (3.43), we recall the notation for linear hyperbolic problems introduced in §2.1, where we used symmetric matrix-valued functions $A^{(i)} : \Omega \to \mathbb{R}^{L\times L}$, $i = 1, \dots, N+1$, to describe the hyperbolic problem. Further the matrix $A^{(t)} \equiv A^{(N+1)}$ is assumed to be symmetric positive definite. The corresponding eigenvalue problem (2.3) reads

$$\left[\sum_{i=1}^{N} \nu_i A^{(i)} - \lambda A^{(t)}\right] v = 0, \qquad v \in \mathbb{R}^L,$$

and we obtain the real eigenvalues $\lambda_1, \dots, \lambda_L$, which still depend on the direction $\nu \in \mathbb{S}^{N-1}$ on the $N-1$ dimensional unit sphere. The accompanying L linearly independent eigenvectors $e_j \in \mathbb{R}^L$ satisfy

$$\Big(\sum_{i=1}^{N} \nu_i A^{(i)}\Big) e_j = \lambda_j A^{(t)} e_j. \tag{3.45}$$

Since the matrices $A^{(i)}$, $i = 1, \dots, N+1$, are symmetric, it is easy to see that the eigenvectors e_i must be orthogonal in the $\langle x, y\rangle_{A^{(t)}} = y^\intercal A^{(t)} x$ inner product.

Using this setting, we show that the strict causality condition (3.44) implies the existence of a solution of (3.43) in Theorem 2, which implies that the inverse map $M^{-1}(\hat{t}, \cdot)$ is well defined.

Lemma 1. *For a linear hyperbolic system given by symmetric matrix-valued functions $A^{(i)} : \Omega \to \mathbb{R}^{L\times L}$, $i = 1, \dots, N+1$, where $A^{(t)} \equiv A^{(N+1)}$ is assumed to be symmetric positive definite, holds*

$$v^\intercal \Big(\sum_{i=1}^{N} \nu_i A^{(i)}\Big) v \le c\; v^\intercal A^{(t)} v, \qquad \forall v \in \mathbb{R}^L, \tag{3.46}$$

where c denotes the maximal characteristic speed of the system and $\nu \in \mathbb{S}^{N-1}$. Further there exists a constant $C_1 > 0$, so that

$$v^\intercal \Big(A^{(t)} - \sum_{i=1}^{N} \partial_i \varphi A^{(i)}\Big) u \le C_1\; \|u\|\|v\|, \qquad \forall u, v \in \mathbb{R}^L. \tag{3.47}$$

Proof. Expanding any vector $v \in \mathbb{R}^L$ in the eigenbasis e_j as follows,

$$v = \sum_{j=1}^{L} v_j e_j \qquad \text{with } v_j = \langle v, e_j \rangle_{A^{(t)}},$$

and by (3.45) we see that

$$v^\intercal \Big(\sum_{i=1}^{N} \nu_i A^{(i)} \Big) v = v^\intercal \sum_{j=1}^{L} v_j \lambda_j A^{(t)} e_j = \sum_{j=1}^{L} \lambda_j \, |v_j|^2.$$

The maximum of the eigenvalues λ_j, $j = 1, \dots L$, over all possible directions ν defines our maximal characteristic speed c. Hence we obtain

$$v^\intercal \Big(\sum_{i=1}^{N} \nu_i A^{(i)} \Big) v \leq c \sum_{j=1}^{L} |v_j|^2 = c \, \langle v, v \rangle_{A^{(t)}} = c \; v^\intercal A^{(t)} v$$

for all $v \in \mathbb{R}^L$.

To show the second inequality, we expand any vectors $u, v \in \mathbb{R}^L$ in the eigenbasis

$$u = \sum_{j=1}^{L} u_j e_j \qquad \text{and} \qquad v = \sum_{j=1}^{L} v_j e_j,$$

with $u_j = \langle u, e_j \rangle_{A^{(t)}}$ and $v_j = \langle v, e_j \rangle_{A^{(t)}}$. Using the expansion of u and (3.45), there holds

$$\begin{aligned} v^\intercal \Big(A^{(t)} - \sum_{i=1}^{N} \partial_i \varphi A^{(i)} \Big) u &= v^\intercal A^{(t)} u - v^\intercal \sum_{j=1}^{L} u_j \Big(\sum_{i=1}^{N} \partial_i \varphi A^{(i)} \Big) e_j \\ &= v^\intercal \sum_{j=1}^{L} u_j (1 - \| \nabla_x \varphi \| \lambda_j) A^{(t)} e_j. \end{aligned}$$

Since $|\lambda_j| \leq c$, for $j = 1, \dots, L$, the causality condition implies $0 \leq (1 - \| \nabla_x \varphi \| \lambda_j) \leq 2$. Together with the expansion of v, we obtain

$$v^\intercal \Big(A^{(t)} - \sum_{i=1}^{N} \partial_i \varphi A^{(i)} \Big) u = \sum_{j=1}^{L} (1 - \| \nabla_x \varphi \| \lambda_j) u_j v_j \leq 2 \sum_{j=1}^{L} |u_j| |v_j|.$$

Using the Cauchy-Schwarz inequality, we get

$$\sum_{j=1}^{L} |u_j| |v_j| \leq \sqrt{\sum_{j=1}^{L} |u_j|^2} \sqrt{\sum_{j=1}^{L} |v_j|^2} = \sqrt{u^\intercal A^{(t)} u} \sqrt{v^\intercal A^{(t)} v}.$$

Since $A^{(t)}$ is symmetric positive definite, there exits $\mu > 0$, so that $w^\intercal A^{(t)} w \leq \mu w^\intercal w$ for all $w \in \mathbb{R}^L$. Combining these inequalities, we obtain

$$v^\intercal \Big(A^{(t)} - \sum_{i=1}^{N} \partial_i \varphi A^{(i)} \Big) u \leq 2 \sqrt{u^\intercal A^{(t)} u} \sqrt{v^\intercal A^{(t)} v} \leq 2\mu \sqrt{u^\intercal u} \sqrt{v^\intercal v},$$

and setting $C_1 = 2\mu$ concludes the proof. □

Theorem 2. *Assuming the strict causality condition (3.44) on a tent K, there exists an unique solution $\hat{u}$ of the mapped equation $\hat{y} = M(\hat{t}, \hat{u})$.*

Proof. To prove the solvability of $\hat{y} = M(\hat{t}, \hat{u})$, we use its variational form (3.43) and show that

$$a(\hat{u}, \hat{v}) = \int_{\omega_v} M(\hat{t}, \hat{u}) \cdot \hat{v}$$

defines a coercive, continuous bilinear form on $[L_2(\omega_v)]^L$ for any $\hat{t} \in [0, 1]$.

Inserting $M(\hat{t}, \hat{u})$, given by (3.32), into $a(\hat{u}, \hat{u})$ yields

$$\begin{aligned} a(\hat{u}, \hat{u}) &= \int_{\omega_v} \left[\left(A^{(t)} - \sum_{i=1}^{N} A^{(i)} \partial_i \varphi \right) \hat{u} \right] \cdot \hat{u} \\ &\geq \int_{\omega_v} \left[\left(A^{(t)} - c\, A^{(t)} \| \nabla_x \varphi \| \right) \hat{u} \right] \cdot \hat{u} \\ &= \int_{\omega_v} (1 - c \| \nabla_x \varphi \|) \left(A^{(t)} \hat{u} \right) \cdot \hat{u}, \end{aligned}$$

where we used (3.46) with $\nu = \frac{\nabla_x \varphi}{\| \nabla_x \varphi \|}$. The causality condition (3.44) for φ implies $(1 - c \| \nabla_x \varphi \|) > 0$ and together with the positive definiteness of the matrix $A^{(t)}$, we get that there exists constant $C > 0$, such that

$$a(\hat{u}, \hat{u}) \geq \int_{\omega_v} (1 - c \| \nabla_x \varphi \|) \left(A^{(t)} \hat{u} \right) \cdot \hat{u} \geq C \| \hat{u} \|_{L_2(\omega_v)}, \qquad \forall \hat{u} \in [L_2(\omega_v)]^L. \tag{3.48}$$

To show continuity, we apply (3.47) to $a(\hat{u}, \hat{v})$ and obtain

$$a(\hat{u}, \hat{v}) = \int_{\omega_v} \left[\left(A^{(t)} - \sum_{i=1}^{N} A^{(i)} \partial_i \varphi \right) \hat{u} \right] \cdot \hat{v} \leq C_1 \int_{\omega_v} \sqrt{\hat{u} \cdot \hat{u}} \sqrt{\hat{v} \cdot \hat{v}} \leq C_1 \| \hat{u} \|_{L_2(\omega_v)} \| \hat{v} \|_{L_2(\omega_v)},$$

where we used the Cauchy-Schwarz inequality to conclude the proof. □

As consequence of the proof of Theorem 2, we obtain the positive definiteness of the matrix

$$\mathtt{M}_{ij}(\hat{t}) = a(\psi_i, \psi_j), \qquad 1 \leq i, j \leq m, \tag{3.49}$$

with $\psi_i \in V_h^{\mathrm{v}}$, as defined in (3.33).

In the following we give explicit inverse maps for some linear hyperbolic systems, which are then used in our numerical examples later in this thesis.

Advection equation

The advection equation is described by a divergence-free vector field $\beta : \Omega_0 \to \mathbb{R}^N$, which leads to the functions $A^{(t)} = [1]$ and $A^{(i)} = [\beta_i]$, for $i = 1, \ldots, N$. For a given $\hat{y} = M(\hat{t}, \hat{u})$, (3.32) reads

$$\hat{y} = (1 - \beta \cdot \nabla_x \varphi)\, \hat{u}.$$

The causality condition $\|\beta\|\|\nabla_x \varphi\| < 1$ implies $(1 - \beta \cdot \nabla_x \varphi) > 0$ and we obtain the solution

$$\hat{u} = \frac{\hat{y}}{1 - \beta \cdot \nabla_x \varphi}. \tag{3.50}$$

Wave equation

We recall the definitions in (2.8) and restrict to an isotropic material, e.g. $\alpha = c_s^2 I$, for the following inverse map. Then, (3.32) yields

$$\begin{pmatrix} \hat{y}_q \\ \hat{y}_\mu \end{pmatrix} = \begin{pmatrix} c_s^{-2}\hat{q} - \hat{\mu}\,\nabla_x \varphi \\ -\nabla_x \varphi \cdot \hat{q} + \hat{\mu} \end{pmatrix}, \tag{3.51}$$

for a given $\hat{y} = [\hat{y}_q, \hat{y}_\mu] \in \mathbb{R}^N \times \mathbb{R}$. We now have to solve (3.51) for $\hat{q} \in \mathbb{R}^N, \hat{\mu} \in \mathbb{R}$. The inner product of $\hat{y}_q$ and $\nabla_x \varphi$ reads

$$\hat{y}_q \cdot \nabla_x \varphi = c_s^{-2}(\hat{q} \cdot \nabla_x \varphi) - \hat{\mu}\|\nabla_x \varphi\|^2 = c_s^{-2}\,(\hat{\mu} - \hat{y}_\mu) - \hat{\mu}\|\nabla_x \varphi\|^2, \tag{3.52}$$

where we used the second component of (3.51) to substitute $\nabla_x \varphi \cdot \hat{q}$. Rephrasing (3.52) leads to

$$\hat{\mu} = \frac{\hat{y}_\mu + c_s^2(\hat{y}_q \cdot \nabla_x \varphi)}{1 - c_s^2\|\nabla_x \varphi\|^2}, \tag{3.53a}$$

which is well-defined due to the causality condition $c_s\|\nabla_x \varphi\| < 1$. Further we obtain

$$\hat{q} = c_s^2\,(\hat{y}_q + \hat{\mu}\,\nabla_x \varphi)\,. \tag{3.53b}$$

Maxwell equations

For the derivation of the inverse map for the Maxwell equations, we recall the definitions in (2.11). We split the given function $\hat{y} = M(\hat{t}, \hat{u}) \in \mathbb{R}^6$ into $\hat{y}_E, \hat{y}_H \in \mathbb{R}^3$, with $\hat{y} = [\hat{y}_E, \hat{y}_H]^\intercal$. Thus we have to find $\hat{E}, \hat{H} \in \mathbb{R}^3$, such that

$$\begin{pmatrix} \hat{y}_E \\ \hat{y}_H \end{pmatrix} = \begin{pmatrix} \varepsilon\hat{E} - \hat{H} \times \nabla_x \varphi \\ \mu\hat{H} + \hat{E} \times \nabla_x \varphi \end{pmatrix}. \tag{3.54}$$

The outer product of $\hat{y}_E$ and $\nabla_x \varphi$ reads

$$\begin{aligned} \hat{y}_E \times \nabla_x \varphi &= \varepsilon\hat{E} \times \nabla_x \varphi - (\hat{H} \times \nabla_x \varphi) \times \nabla_x \varphi \\ &= \varepsilon\hat{E} \times \nabla_x \varphi - \big((\hat{H} \cdot \nabla_x \varphi)\,\nabla_x \varphi - (\nabla_x \varphi \cdot \nabla_x \varphi)\hat{H}\big) \\ &= \varepsilon(\hat{y}_H - \mu\hat{H}) - \big(\nabla_x \varphi(\nabla_x \varphi)^\intercal \hat{H} - \|\nabla_x \varphi\|^2 \hat{H}\big), \end{aligned} \tag{3.55}$$

where we used the identity $(a \times b) \times c = (a \cdot c)\,b - (b \cdot c)\,a$, for any $a, b, c \in \mathbb{R}^3$, and expressed $\hat{E} \times \nabla_x \varphi$ by known quantities and $\hat{H}$ using (3.54). Rephrasing (3.55) leads to

$$\big[(\varepsilon\mu - \|\nabla_x \varphi\|^2)I + \nabla_x \varphi(\nabla_x \varphi)^\intercal\big]\hat{H} = (\varepsilon\hat{y}_H - \hat{y}_E \times \nabla_x \varphi),$$

with the identity matrix $I \in \mathbb{R}^{3\times 3}$. This allows us to express the solution $\hat{H}$ by

$$\hat{H} = \frac{1}{\varepsilon\mu - \|\nabla_x \varphi\|^2}\Big[I - \frac{1}{\varepsilon\mu}\nabla_x \varphi(\nabla_x \varphi)^\intercal\Big](\varepsilon\hat{y}_H - \hat{y}_E \times \nabla_x \varphi). \tag{3.56a}$$

Repeating this derivation for $\hat{E}$, we obtain

$$\hat{E} = \frac{1}{\varepsilon\mu - \| \nabla_x \varphi\|^2} \Big[I - \frac{1}{\varepsilon\mu} \nabla_x \varphi (\nabla_x \varphi)^\intercal \Big] (\mu \hat{y}_E - \hat{y}_H \times \nabla_x \varphi). \tag{3.56b}$$

The solution $(\hat{E}, \hat{H})$ of (3.54) is well defined for $\| \nabla_x \varphi\| < \sqrt{\varepsilon\mu}$ – our causality condition.

3.5.2 Nonlinear examples

In this section, we present example of inverse maps for nonlinear problem, which we will discuss later in this thesis. For a general nonlinear hyperbolic problem (2.1) defined by the functions g and f, we have to find $\hat{u}$ for a given $\hat{y}$, such that

$$\hat{y} = M(\hat{t}, \hat{u}) = g(\hat{u}) - f(\hat{u}) \nabla_x \varphi \,. \tag{3.57}$$

Burgers' equations

Using the functions g and f in (2.14), defining the Burgers' equation for $N \in \{1, 2\}$, the nonlinear equation (3.57) reads

$$\hat{y} = \hat{u} - \sum_{i=1}^{N} \frac{1}{2} \hat{u}^2 \partial_i \varphi \,.$$

With the constant $b = \sum_{i=1}^{N} \partial_i \varphi$, we are left to solve the quadratic equation

$$\frac{b}{2} \hat{u}^2 - \hat{u} + \hat{y} = 0, \tag{3.58}$$

which has the solutions

$$\hat{u} = \frac{1 \pm \sqrt{1 - 2b\hat{y}}}{b} = \frac{2\hat{y}}{1 \mp \sqrt{1 - 2b\hat{y}}}. \tag{3.59}$$

We will now show that the causality condition (3.3) implies that these roots are real and only one of the two roots is valid. Recall that the maximal characteristic speed is $\bar{c} = \sqrt{N}|u|$ and the causality condition reads

$$\| \nabla_x \varphi\| < \frac{1}{\bar{c}} = \frac{1}{\sqrt{N}|u|}.$$

For the constant b holds $|b| \leq \sqrt{N} \| \nabla_x \varphi\|$, which implies

$$|b\hat{u}| < 1. \tag{3.60}$$

Rewriting (3.58) as $2b\hat{y} = b\hat{u}(2 - b\hat{u})$, we obtain

$$1 - 2b\hat{y} = 1 - b\hat{u}(2 - b\hat{u}) = (1 - b\hat{u})^2 \geq 0$$

and hence the existence of the roots in (3.59). To choose the correct sign in (3.59), we rewrite the same as

$$b\hat{u} - 1 = \pm\sqrt{1 - 2b\hat{y}}. \tag{3.61}$$

From (3.60), we conclude $b\hat{u} - 1 \leq |b\hat{u}| - 1 < 0$, i.e., we must choose the negative sign in the $\pm$ on the right hand side of (3.61). Thus we obtain the correct root

$$\hat{u} = M^{-1}(\hat{t}, \hat{y}) = \frac{2\hat{y}}{1 + \sqrt{1 - 2b\hat{y}}}. \tag{3.62}$$

Euler equations

For the Euler equations, using the definitions in (2.16), we want to find an explicit solution $\hat{u} = [\hat{\rho}, \hat{m}, \hat{E}]^\intercal$ of (3.57) for a given $\hat{y} = [\hat{y}_\rho, \hat{y}_m, \hat{y}_E]^\intercal$. Thus we have to solve

$$\begin{bmatrix} \hat{y}_\rho \\ \hat{y}_m \\ \hat{y}_E \end{bmatrix} = \begin{bmatrix} \hat{\rho} \\ \hat{m} \\ \hat{E} \end{bmatrix} - \begin{bmatrix} \hat{m} \cdot \nabla_x \varphi \\ \hat{P} \nabla_x \varphi + \frac{1}{\hat{\rho}} (\hat{m} \cdot \nabla_x \varphi) \hat{m} \\ (\hat{E} + \hat{P}) \frac{1}{\hat{\rho}} (\hat{m} \cdot \nabla_x \varphi) \end{bmatrix} \tag{3.63}$$

for $\hat{\rho} \in \mathbb{R}, \hat{m} \in \mathbb{R}^N$ and $\hat{E} \in \mathbb{R}$. The system (3.63) is equivalent to

$$\frac{\hat{y}_\rho}{\hat{\rho}} = 1 - \frac{\hat{m} \cdot \nabla_x \varphi}{\hat{\rho}}, \tag{3.64}$$

$$\hat{y}_m = \left(1 - \frac{\hat{m} \cdot \nabla_x \varphi}{\hat{\rho}}\right) \hat{m} - \hat{P} \nabla_x \varphi = \frac{\hat{y}_\rho}{\hat{\rho}} \hat{m} - \hat{P} \nabla_x \varphi, \tag{3.65}$$

$$\hat{y}_E = \left(1 - \frac{\hat{m} \cdot \nabla_x \varphi}{\hat{\rho}}\right) \hat{E} - \hat{P} \frac{\hat{m} \cdot \nabla_x \varphi}{\hat{\rho}} = \frac{\hat{y}_\rho}{\hat{\rho}} \hat{E} - \hat{P} \frac{\hat{m} \cdot \nabla_x \varphi}{\hat{\rho}}. \tag{3.66}$$

Further we recall the definition (2.16b) of the pressure

$$\hat{P} = \frac{2}{d} \left(\frac{\hat{E}}{\hat{\rho}} - \frac{1}{2} \frac{\|\hat{m}\|^2}{\hat{\rho}^2} \right), \tag{3.67}$$

which we use to express the total energy

$$\hat{E} = \frac{d}{2} \hat{P} \hat{\rho} + \frac{1}{2} \frac{\|\hat{m}\|^2}{\hat{\rho}}. \tag{3.68}$$

Substituting (3.68) into (3.66) yields

$$\hat{y}_E = \frac{d}{2} \hat{P} \hat{y}_\rho + \frac{1}{2} \frac{\|\hat{m}\|^2 \hat{y}_\rho}{\hat{\rho}^2} - \hat{P} \frac{\hat{m} \cdot \nabla_x \varphi}{\hat{\rho}}. \tag{3.69}$$

Next, we want to eliminate $\|\hat{m}\|^2$, $\hat{m} \cdot \nabla_x \varphi$ and $\hat{\rho}$ in (3.69). Thus, we rewrite (3.69) as

$$\frac{\|m\|^2 \hat{y}_\rho^2}{\hat{\rho}^2} = \|\hat{y}_m\|^2 + 2\hat{P}(\hat{y}_m \cdot \nabla_x \varphi) + \hat{P}^2 \|\nabla_x \varphi\|. \tag{3.70}$$

The inner product of $\hat{y}_m$ and $\nabla_x \varphi$ reads

$$\hat{y}_m \cdot \nabla_x \varphi = \frac{\hat{y}_\rho}{\hat{\rho}} \hat{m} \cdot \nabla_x \varphi - \hat{P} \|\nabla_x \varphi\|^2,$$

and we obtain

$$\frac{\hat{m} \cdot \nabla_x \varphi}{\hat{\rho}} = \frac{1}{\hat{y}_\rho} \left(\hat{y}_m \cdot \nabla_x \varphi + \hat{P} \|\nabla_x \varphi\|^2 \right). \tag{3.71}$$

Using (3.70) and (3.71), we can rephrase (3.69) to

$$\hat{y}_\rho \hat{y}_E = \frac{d}{2} \hat{P} \hat{y}_\rho^2 + \frac{1}{2} \left(\|\hat{y}_m\|^2 + 2\hat{P}(\hat{y}_m \cdot \nabla_x \varphi) + \hat{P}^2 \|\nabla_x \varphi\| \right) - \hat{P} \left(\hat{y}_m \cdot \nabla_x \varphi + \hat{P} \|\nabla_x \varphi\|^2 \right),$$

which solely depends on $\hat{P}$ and known quantities. This leads to a quadratic equation in $\hat{P}$, which reads

$$0 = \frac{1}{2}\| \nabla_x \varphi\|^2 \hat{P}^2 - \frac{d}{2}\hat{y}_\rho^2 \hat{P} + \hat{y}_\rho \hat{y}_E - \frac{1}{2}\|\hat{y}_m\|^2. \tag{3.72}$$

The roots of (3.72) are given by

$$\hat{P} = \frac{a_1 \pm \sqrt{a_1^2 - \| \nabla_x \varphi\|^2 a_2}}{\| \nabla_x \varphi\|^2} = \frac{a_2}{a_1 \mp \sqrt{a_1^2 - \| \nabla_x \varphi\|^2 a_2}}, \tag{3.73}$$

where

$$a_1 = \frac{d}{2}\hat{y}_\rho^2, \qquad a_2 = 2\hat{y}_E \hat{y}_\rho - \|\hat{y}_m\|^2.$$

To choose the correct sign in (3.73), we consider the limit $\nabla_x \varphi \to 0$, e.g. flat advancing front, for which holds

$$\begin{bmatrix} \hat{y}_\rho \\ \hat{y}_m \\ \hat{y}_E \end{bmatrix} \xrightarrow{\nabla_x \varphi \to 0} \begin{bmatrix} \hat{\rho} \\ \hat{m} \\ \hat{E} \end{bmatrix}.$$

Further we obtain

$$a_1 = \frac{d}{2}\hat{y}_\rho^2 \xrightarrow{\nabla_x \varphi \to 0} \frac{d}{2}\hat{\rho}^2,$$
$$a_2 = 2\hat{y}_E \hat{y}_\rho - \|\hat{y}_m\|^2 \xrightarrow{\nabla_x \varphi \to 0} 2\hat{E}\hat{\rho} - \|\hat{m}\|^2,$$

and therefore

$$\hat{P} = \frac{a_2}{a_1 \mp \sqrt{a_1^2 - \| \nabla_x \varphi\|^2 a_2}} \xrightarrow{\nabla_x \varphi \to 0} \frac{2\hat{E}\hat{\rho} - \|\hat{m}\|^2}{\frac{d}{2}\hat{\rho}^2 \mp \frac{d}{2}\hat{\rho}^2}.$$

To obtain a well defined $\hat{P}$ agreeing with the definition (3.67), we must choose the positive sign in the $\mp$ on the right hand side of (3.73) and the correct root is

$$\hat{P} = \frac{a_2}{a_1 + \sqrt{a_1^2 - \| \nabla_x \varphi\|^2 a_2}}. \tag{3.74a}$$

Substituting (3.71) into (3.64) yields

$$\frac{\hat{y}_\rho}{\hat{\rho}} = 1 - \frac{1}{\hat{y}_\rho}\left(\hat{y}_m \cdot \nabla_x \varphi + \hat{P}\| \nabla_x \varphi\|^2\right),$$

and we get

$$\hat{\rho} = \frac{\hat{y}_\rho^2}{\hat{y}_\rho - \left(\hat{y}_m \cdot \nabla_x \varphi + \hat{P}\| \nabla_x \varphi\|\right)}. \tag{3.74b}$$

The solutions $\hat{m}$ and $\hat{E}$ can be expressed using (3.65) and (3.66), respectively, and we obtain

$$\hat{m} = \frac{\hat{\rho}}{\hat{y}_\rho}\left(\hat{y}_m \cdot \nabla_x \varphi + \hat{P}\| \nabla_x \varphi\|\right), \tag{3.74c}$$

$$\hat{E} = \frac{\hat{\rho}}{\hat{y}_\rho}\left(\hat{y}_E - \frac{\hat{P}}{\hat{\rho}}(\hat{m} \cdot \nabla_x \varphi)\right). \tag{3.74d}$$

4 Structure aware time stepping schemes

In the previous chapter we observed high order, but still suboptimal, convergence rates $\mathcal{O}(h^p)$ for locally implicit MTP methods when using polynomials of degree at most p for the spatial discretization. Another limiting factor of these locally implicit methods is the available memory, while fully explicit schemes have, using a matrix-free implementation, a very low memory consumption. Thus our focus lies on fully explicit MTP methods and we construct suitable time stepping schemes to recover the high order convergence.

Due to the mapping introduced in §3.3, the temporal discretization has a direct influence on the spatial convergence. This becomes clear with the following lemma.

Lemma 2. *The function $\hat{u} = u \circ \Phi$ satisfies*

$$\frac{\partial^k \hat{u}}{\partial \hat{t}^k}(x, \hat{t}) = \delta(x)^k \frac{\partial^k u}{\partial t^k}(x, t)$$

at almost every point $(x, t) = \Phi(x, \hat{t})$ in the tent K.

Proof. Let e denote the spacetime unit vector in the time direction i.e., all its components are zero except for the last (time) component which is 1. Then, at some fixed spacetime point $\hat{P} = (x, \hat{t})$ and $P \equiv (x, t) = \Phi(\hat{P})$, we have

$$\frac{\partial^k \hat{u}}{\partial \hat{t}^k}(\hat{P}) = \hat{u}^{(k)}(\hat{P})(e, e, \dots, e).$$

Here, in the argument list of the multilinear form representing the Frechet derivative $\hat{u}^{(k)}$, the vector e is repeated k times. By standard arguments [3] for affine maps,

$$\begin{aligned}\frac{\partial^k \hat{u}}{\partial \hat{t}^k}(\hat{P}) &= (u \circ \Phi)^{(k)}(\hat{P})(e, e, \dots, e) \\ &= u^{(k)}(P)(\Phi' e, \Phi' e, \dots, \Phi' e) \\ &= u^{(k)}(P)(\delta e, \delta e, \dots, \delta e)\end{aligned}$$

where we have used (3.16) in the last step. Since the last term above equals the product of δ^k and the derivative $\partial^k u / \partial t^k$ at P, the proof is complete. □

The causality condition (3.3) implies that the tent height $\delta \lesssim h$. Since we perform the time stepping for $\hat{u} = u \circ \Phi$ in $\hat{t}$, it is crucial that the time stepping scheme approximates the Taylor expansion of $\hat{u}$ properly. Otherwise the remainder term would introduce a scaling with the mesh size h leading to reduced convergence rates, as we observed in §3.4.1.

With this knowledge, we want to design time stepping schemes that are aware of this structure. In §4.1, we construct the Structure Aware Taylor (SAT) time stepping scheme

for linear problems and extend this idea to nonlinear problem in §4.2, where we derive Structure Aware Runge-Kutta (SARK) methods. Both methods are applied to numerical examples in §4.3 showing the recovered high order convergence rates.

The starting point for the temporal discretization is the semi-discrete ODE (3.30) reconsidered as differential-algebraic system

$$\frac{d}{d\hat{t}}\mathtt{Y}(\hat{t}) = \mathtt{A}(\mathtt{u}(\hat{t})), \qquad \mathtt{Y}(\hat{t}) = \mathtt{M}(\hat{t}, \mathtt{u}(\hat{t})). \tag{4.1}$$

Recall that $\mathtt{A}$ is independent of pseudo-time $\hat{t}$, and $\mathtt{M}$ is an affine-linear function of $\hat{t}$, i.e.,

$$\mathtt{M}(\hat{t}, \cdot) = \mathtt{M}_0(\cdot) - \hat{t}\,\mathtt{M}_1(\cdot), \qquad 0 \le \hat{t} \le 1. \tag{4.2}$$

4.1 Structure aware Taylor time stepping

In this section, we develop a structure aware time stepping for linear problems, using the notation introduced in §2.1. For any $\mathtt{w} \in \mathbb{R}^m$, the maps defined in (3.28) can be expressed by the matrices $\mathtt{A}, \mathtt{M}_0$ and $\mathtt{M}_1$ in $\mathbb{R}^{m\times m}$ and the holds

$$\mathtt{A}(\mathtt{w}) = \mathtt{A}\,\mathtt{w}, \qquad \mathtt{M}_0(\mathtt{w}) = \mathtt{M}_0\,\mathtt{w}, \qquad \mathtt{M}_1(\mathtt{w}) = \mathtt{M}_1\,\mathtt{w}, \qquad \text{and} \quad \mathtt{M}(\hat{t}) = \mathtt{M}_0 - \hat{t}\,\mathtt{M}_1.$$

With this notation, (4.1) simplifies to

$$\mathtt{Y}'(\hat{t}) = \mathtt{A}\,\mathtt{u}(\hat{t}), \tag{4.3a}$$

$$\mathtt{Y}(\hat{t}) = \mathtt{M}(\hat{t})\,\mathtt{u}(\hat{t}). \tag{4.3b}$$

Here and throughout we use primes ($'$) to abbreviate $d/d\hat{t}$. Further we subdivide the interval $[0,1]$ into r subintervals

$$[\hat{t}_k, \hat{t}_{k+1}], \qquad k = 0, 1, \dots r-1, \qquad \text{where } \hat{t}_k = \frac{k}{r}.$$

This subdivision of the interval is done to obtain stability when using higher polynomial degrees p for the spatial discretization, which we discuss in detail in chapter 5. Using the subintervals, the matrix $\mathtt{M}(\hat{t})$ can be expressed by

$$\mathtt{M}(\hat{t}) = \mathtt{M}_0^{[k]} - (\hat{t} - \hat{t}_k)\mathtt{M}_1, \qquad \hat{t} \in [\hat{t}_k, \hat{t}_{k+1}] \tag{4.4}$$

where $\mathtt{M}_0^{[k]} = \mathtt{M}(\hat{t}_k)$.

Remark 3. Subdividing the interval $[0,1]$ into r subintervals corresponds to splitting the tent K into r "subtents", as illustrated in Figure 4.1.

Consider the approximations to $\mathtt{Y}, \mathtt{u}$ on $[\hat{t}_k, \hat{t}_{k+1}]$ in the form of Taylor polynomials $\mathtt{Y}^{[k+1]}, \mathtt{u}^{[k+1]}$ of degree s and $s-1$, respectively, defined for $\hat{t} \in [\hat{t}_k, \hat{t}_{k+1}]$ by

$$\mathtt{Y}^{[k+1]}(\hat{t}) = \sum_{n=0}^{s} \frac{(\hat{t} - \hat{t}_k)^n}{n!}\mathtt{Y}^{[k,n]}, \tag{4.5a}$$

$$\mathtt{u}^{[k+1]}(\hat{t}) = \sum_{n=0}^{s-1} \frac{(\hat{t} - \hat{t}_k)^n}{n!}\mathtt{u}^{[k,n]}, \tag{4.5b}$$

where $\mathtt{Y}^{[k,n]} = \big(\mathtt{Y}^{[k+1]}\big)^{(n)}(\hat{t}_k)$ and $\mathtt{u}^{[k,n]} = \big(\mathtt{u}^{[k+1]}\big)^{(n)}(\hat{t}_k)$. To find these derivatives, we differentiate both equations of (4.3) n times to get

$$\begin{aligned} \mathtt{Y}^{(n+1)}(\hat{t}) &= \mathtt{A}\,\mathtt{u}^{(n)}(\hat{t})\,, && n \geq 0\,, \\ \mathtt{Y}^{(n)}(\hat{t}) &= \mathtt{M}(\hat{t})\,\mathtt{u}^{(n)}(\hat{t}) - n\,\mathtt{M}_1\,\mathtt{u}^{(n-1)}(\hat{t})\,, && n \geq 1\,. \end{aligned}$$

For the second equation we used Leibniz' formula $(fg)^{(n)} = \sum_{i=0}^{n} \binom{n}{i} f^{(i)} g^{(n-i)}$, and the fact that $\mathtt{M}(\hat{t}) = \mathtt{M}_0 - \hat{t}\,\mathtt{M}_1$ is affine-linear. Evaluating these equations for the Taylor polynomials $\mathtt{Y}^{[k+1]}, \mathtt{u}^{[k+1]}$ at $\hat{t} = \hat{t}_k$, we obtain a recursive formula for $\mathtt{Y}^{[k,n]}$ and $\mathtt{u}^{[k,n]}$ in terms of $\mathtt{u}^{[k,n-1]}$, namely

$$\begin{aligned} \mathtt{Y}^{[k,n]} &= \mathtt{A}\,\mathtt{u}^{[k,n-1]}\,, && 1 \leq n \leq s\,, \\ \mathtt{M}_0^{[k]}\,\mathtt{u}^{[k,n]} &= \mathtt{Y}^{[k,n]} + n\,\mathtt{M}_1\,\mathtt{u}^{[k,n-1]}\,, && 1 \leq n \leq s-1\,, \end{aligned} \tag{4.5c}$$

for all $0 \leq k \leq r-1$. Given $\mathtt{Y}^{[0,0]} = \mathtt{Y}(\hat{t}_0)$, $\mathtt{M}_0\,\mathtt{u}^{[0,0]} = \mathtt{Y}^{[0,0]}$, applying (4.5c) with $k = 0$ gives the approximate functions $\mathtt{Y}^{[1]}(\hat{t}), \mathtt{u}^{[1]}(\hat{t})$ in the first subinterval $[\hat{t}_0, \hat{t}_1]$. The recursive formulas are initiated for later subintervals at $n = 0$ by

$$\mathtt{Y}^{[k,0]} = \mathtt{Y}^{[k]}(\hat{t}_k), \qquad \mathtt{M}_0^{[k]}\,\mathtt{u}^{[k,0]} = \mathtt{Y}^{[k,0]}\,, \qquad 1 \leq k \leq r-1\,. \tag{4.5d}$$

After the final subinterval, we get $\mathtt{Y}^{[r]}(\hat{t}_r)$, our approximation to $\mathtt{Y}(1)$. We shall refer to the new time-stepping scheme generated by (4.5) as the s-stage *SAT (structure aware Taylor) time stepping*.

Remark 4. Note that $\mathtt{Y}^{[r]}(\hat{t}_r)$ is our approximation to $\mathtt{Y} = \mathtt{M}\,\mathtt{u}$ at the top of the tent. This value is then passed to the next tent in time. The time dependence of $\mathtt{M}$ arises from the time dependence of $\nabla\varphi$. This gradient is continuous along spacetime lines of constant spatial coordinates. Therefore, when passing from one element of a tent to the same element within the next tent in time, $\mathtt{Y}$ is continuous (since the solution $\mathtt{u}$ is continuous). Of course, on flat fronts $\nabla\varphi = \nabla\tau = 0$, so there $\mathtt{M}$ is just a diagonal matrix containing the material parameters.

4.1.1 Propagation operator of SAT methods

For the later use, we define the linear propagation operator on the tent $\mathtt{T}_{r,s} : \mathbb{R}^m \to \mathbb{R}^m$, which relates input $\mathtt{Y}(\hat{t}_0)$ and output $\mathtt{Y}^{[r]}(\hat{t}_r)$ of the scheme described in (4.5) by

$$\mathtt{Y}^{[r]}(\hat{t}_r) = \mathtt{T}_{r,s}\,\mathtt{Y}(\hat{t}_0). \tag{4.6}$$

Further we introduce a partial propagation operator $\mathtt{T}_{r,s}^{[k+1]} : \mathbb{R}^m \to \mathbb{R}^m$, relating the intermediate quantities

$$\mathtt{Y}^{[k+1]}(\hat{t}_{k+1}) = \mathtt{T}_{r,s}^{[k+1]}\,\mathtt{Y}^{[k,0]} \tag{4.7}$$

at the kth step of the scheme and there holds

$$\mathtt{T}_{r,s} = \prod_{i=0}^{r-1} \mathtt{T}_{r,s}^{[r-i]}.$$

Given $\mathtt{Y}^{[k,0]}$ and $\mathtt{M}_0^{[k]}\,\mathtt{u}^{[k,0]} = \mathtt{Y}^{[k,0]}$, solving (4.5c) leads to the coefficients

$$\mathtt{Y}^{[k,n]} = \tilde{\mathtt{A}}^{[k]} \prod_{i=1}^{n-1} \Big(\tilde{\mathtt{A}}^{[k]} + (n-i)\tilde{\mathtt{M}}_1^{[k]}\Big)\mathtt{Y}^{[k,0]}, \qquad 1 \le n \le s,$$

with $\tilde{\mathtt{A}}^{[k]} = \mathtt{A}\big(\mathtt{M}_0^{[k]}\big)^{-1}$ and $\tilde{\mathtt{M}}_1^{[k]} = \mathtt{M}_1\big(\mathtt{M}_0^{[k]}\big)^{-1}$. The resulting Taylor polynomial (4.5a) at $\hat{t}_{k+1}$ for a s-stage SAT method reads

$$\mathtt{Y}^{[k+1]}(\hat{t}_{k+1}) = \Big(\mathtt{I} + \sum_{n=1}^{s} \frac{(\tau^{[k]})^n}{n!}\tilde{\mathtt{A}}^{[k]} \prod_{i=1}^{n-1}\Big(\tilde{\mathtt{A}}^{[k]} + (n-i)\tilde{\mathtt{M}}_1^{[k]}\Big)\Big)\mathtt{Y}^{[k,0]},$$

where $\mathtt{I} \in \mathbb{R}^{m\times m}$ denotes the identity matrix and $\tau^{[k]} = \hat{t}_{k+1} - \hat{t}_k = \frac{1}{r}$. This leads to the partial propagation operator

$$\mathtt{T}_{r,s}^{[k+1]} = \mathtt{I} + \sum_{n=1}^{s} \frac{(\tau^{[k]})^n}{n!}\tilde{\mathtt{A}}^{[k]} \prod_{i=1}^{n-1}\Big(\tilde{\mathtt{A}}^{[k]} + (n-i)\tilde{\mathtt{M}}_1^{[k]}\Big). \tag{4.8}$$

Concluding this section, we give examples of $\mathtt{T}_{r,s}^{[k+1]}$ for $s = 2, 3, 4$, for which we will discuss discrete stability properties in chapter 5:

$$\mathtt{T}_{r,2}^{[k+1]} = \mathtt{I} + \tau^{[k]}\tilde{\mathtt{A}}^{[k]} + \frac{1}{2}(\tau^{[k]})^2\tilde{\mathtt{A}}^{[k]}\big(\tilde{\mathtt{A}}^{[k]} + \tilde{\mathtt{M}}_1^{[k]}\big), \tag{4.9}$$

$$\begin{aligned}\mathtt{T}_{r,3}^{[k+1]} = \mathtt{I} &+ \tau^{[k]}\tilde{\mathtt{A}}^{[k]} + \frac{1}{2}(\tau^{[k]})^2\tilde{\mathtt{A}}^{[k]}\big(\tilde{\mathtt{A}}^{[k]} + \tilde{\mathtt{M}}_1^{[k]}\big)\\ &+ \frac{1}{6}(\tau^{[k]})^3\tilde{\mathtt{A}}^{[k]}\big(\tilde{\mathtt{A}}^{[k]} + 2\tilde{\mathtt{M}}_1^{[k]}\big)\big(\tilde{\mathtt{A}}^{[k]} + \tilde{\mathtt{M}}_1^{[k]}\big),\end{aligned} \tag{4.10}$$

$$\begin{aligned}\mathtt{T}_{r,4}^{[k+1]} = \mathtt{I} &+ \tau^{[k]}\tilde{\mathtt{A}}^{[k]} + \frac{1}{2}(\tau^{[k]})^2\tilde{\mathtt{A}}^{[k]}\big(\tilde{\mathtt{A}}^{[k]} + \tilde{\mathtt{M}}_1^{[k]}\big)\\ &+ \frac{1}{6}(\tau^{[k]})^3\tilde{\mathtt{A}}^{[k]}\big(\tilde{\mathtt{A}}^{[k]} + 2\tilde{\mathtt{M}}_1^{[k]}\big)\big(\tilde{\mathtt{A}}^{[k]} + \tilde{\mathtt{M}}_1^{[k]}\big)\\ &+ \frac{1}{24}(\tau^{[k]})^4\tilde{\mathtt{A}}^{[k]}\big(\tilde{\mathtt{A}}^{[k]} + 3\tilde{\mathtt{M}}_1^{[k]}\big)\big(\tilde{\mathtt{A}}^{[k]} + 2\tilde{\mathtt{M}}_1^{[k]}\big)\big(\tilde{\mathtt{A}}^{[k]} + \tilde{\mathtt{M}}_1^{[k]}\big).\end{aligned} \tag{4.11}$$

4.2 Structure aware Runge-Kutta methods

In this section, we develop specialized Runge-Kutta type schemes that do not show the above mentioned convergence order loss of classical Runge-Kutta schemes when being used in explicit MTP methods.

We motivate the definition of the new scheme by reformulating (4.1) in terms of two variables $\mathtt{Z}(\hat{t})$ and $\mathtt{Y}(\hat{t})$, defined by

$$\mathtt{Z}(\hat{t}) = \mathtt{M}_0(\mathtt{u}(\hat{t})), \qquad \mathtt{Y}(\hat{t}) = \mathtt{M}(\hat{t}, \mathtt{u}(\hat{t})) = \mathtt{Z}(\hat{t}) - \hat{t}\,\mathtt{M}_1(\mathtt{u}(\hat{t})).$$

Then (4.1) implies

$$\mathtt{Y}' = \mathtt{A}(\mathtt{u}(\hat{t})), \qquad \mathtt{Z}' = \mathtt{A}(\mathtt{u}(\hat{t})) + (\hat{t}\,\mathtt{M}_1(\mathtt{u}(\hat{t})))', \tag{4.12}$$

together with the initial conditions $\mathtt{Y}(0) = \mathtt{Z}(0) = \mathtt{M}_0(\mathtt{u}_0)$. The key idea is to avoid the inversion of the time-dependent $\mathtt{M}$ at all $\hat{t}$, limiting the inversion to just that of $\mathtt{M}_0$. Assuming we can compute the time-independent inverse $\mathtt{M}_0^{-1}$, we define

$$\tilde{\mathtt{A}} = \mathtt{A} \circ \mathtt{M}_0^{-1}, \qquad \tilde{\mathtt{M}}_1 = \mathtt{M}_1 \circ \mathtt{M}_0^{-1}.$$

Then, (4.12) yields the following ODE system for $\mathtt{Y}$ and $\mathtt{Z}$ on $0 < \hat{t} < 1$:

$$\mathtt{Z}' = \tilde{\mathtt{A}}(\mathtt{Z}(\hat{t})) + (\hat{t}\,\tilde{\mathtt{M}}_1(\mathtt{Z}(\hat{t})))', \qquad \mathtt{Z}(0) = \mathtt{Y}_0, \tag{4.13a}$$

$$\mathtt{Y}' = \tilde{\mathtt{A}}(\mathtt{Z}(\hat{t})), \qquad \mathtt{Y}(0) = \mathtt{Y}_0, \tag{4.13b}$$

where $\mathtt{Y}_0 = \mathtt{M}_0(\mathtt{u}_0)$. The Integrating the equations of (4.13) from 0 to τ, we obtain

$$\mathtt{Z}(\tau) = \mathtt{Z}(0) + \tau\,\tilde{\mathtt{M}}_1(\mathtt{Z}(\tau)) + \int_0^\tau \tilde{\mathtt{A}}(\mathtt{Z}(s))\,\mathrm{d}s, \tag{4.14a}$$

$$\mathtt{Y}(\tau) = \mathtt{Y}(0) + \int_0^\tau \tilde{\mathtt{A}}(\mathtt{Z}(s))\,\mathrm{d}s. \tag{4.14b}$$

The new scheme, defined below, may be thought of as motivated by quadrature approximations to the integrals above. Note that we are only interested in such quadratures that result in explicit schemes. Moreover, we must also approximate $\tau\,\tilde{\mathtt{M}}_1(\mathtt{Z}(\tau))$ by an extrapolation formula that uses prior values of $\mathtt{Z}$, in order to keep the scheme explicit.

Definition 2. Given an initial condition $\mathtt{Y}_0$, an *s-stage structure aware Runge-Kutta (SARK) scheme* for (4.13) computes

$$\mathtt{Z}_i = \mathtt{Y}_0 + \tau \sum_{j=1}^{i-1} d_{ij}\tilde{\mathtt{M}}_1(\mathtt{Z}_j) + \tau \sum_{j=1}^{i-1} a_{ij}\tilde{\mathtt{A}}(\mathtt{Z}_j), \qquad 1 \le i \le s, \tag{4.15a}$$

$$\mathtt{Y}_\tau = \mathtt{Y}_0 + \tau \sum_{i=1}^{s} b_i\tilde{\mathtt{A}}(\mathtt{Z}_i). \tag{4.15b}$$

This explicit method is determined by the coefficient matrices $b \in \mathbb{R}^{s\times 1}$, $\mathcal{A} \in \mathbb{R}^{s\times s}$, and $\mathcal{D} \in \mathbb{R}^{s\times s}$, taking the form

$$b = (b_1, \ldots, b_s), \quad \mathcal{A} = \begin{pmatrix} 0 & & & \\ a_{21} & 0 & & \\ \vdots & \ddots & 0 & \\ a_{s1} & \ldots & a_{s,s-1} & 0 \end{pmatrix}, \quad \mathcal{D} = \begin{pmatrix} 0 & & & \\ d_{21} & 0 & & \\ \vdots & \ddots & 0 & \\ d_{s1} & \ldots & d_{s,s-1} & 0 \end{pmatrix}.$$

Hence we use $\begin{array}{c|c|c} c & \mathcal{A} & \mathcal{D} \\ \hline & b & \end{array}$ instead of the standard Butcher tableau $\begin{array}{c|c} c & \mathcal{A} \\ \hline & b \end{array}$ to express our scheme. Here we restrict ourselves to schemes where $c \in \mathbb{R}^s$ is set by the consistency condition

$$c_i = \sum_{j=1}^{i-1} a_{ij}. \tag{4.16}$$

In §4.2.1, we shall develop a theory to choose appropriate values of a_{ij}, d_{ij}, and b_i. Specific examples of SARK scheme tableaus can be found in §4.2.2.

Recall that we need to solve the ODE system (4.13) for $0 < \hat{t} < 1$ within each mapped tent. Since the $\hat{t}$ interval is not small, we subdivide it into r subintervals and use the previously described s-stage SARK scheme within each subinterval, as described next. This is done to enforce stability when using higher polynomial degrees p for the spatial discretization, similar to the p-dependence of the CFL-condition for standard time stepping schemes.

Application of multiple steps within a tent

As for the SAT time stepping, we subdivide the unit interval $[0, 1]$ into r subintervals

$$[\hat{t}_k, \hat{t}_{k+1}], \qquad k = 0, 1, \ldots, r-1, \qquad \text{where } \hat{t}_k = \frac{k}{r},$$

and apply (4.15) within each subinterval as described next.

First observe that the above splitting of the unit $\hat{t}$-interval corresponds to subdividing the original tent K, as given by (3.21), into r "subtents" (see Figure 4.1) of the form

$$K_k = \{(x, t) : \ x \in \omega_{\mathtt{v}}, \ \varphi^{[k]} \le t \le \varphi^{[k+1]}\}, \tag{4.17}$$

where $\varphi^{[k]} = \varphi(\hat{t}_k)$. Clearly $\varphi^{[0]} = \varphi_b$ and $\varphi^{[r]} = \varphi_t$.

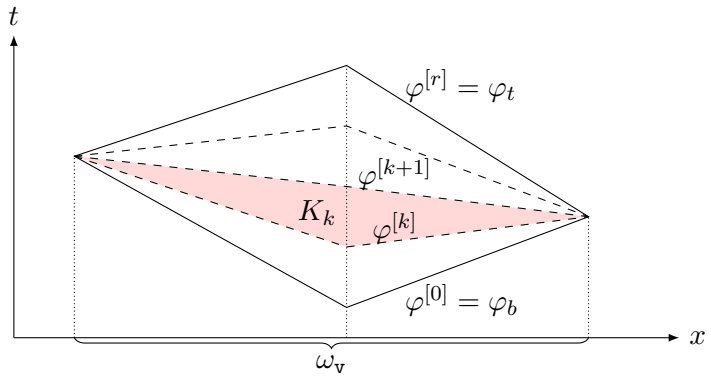


Figure 4.1: Illustration of the subtent K_k (shaded) defined in (4.17). It is the image under Φ of the tensor product domain $\hat{K}_k = \omega_{\mathtt{v}} \times (\hat{t}_k, \hat{t}_{k+1})$.

We then apply (4.15) to each of these subtents. Accordingly, let $\mathtt{M}_0^{[k]}$ be defined by (3.24) after replacing φ_b by $\varphi^{[k]}$. Keeping the same definition of $\mathtt{A}$ and $\mathtt{M}_1$, let $\tilde{\mathtt{M}}_1^{[k]} = \mathtt{M}_1 \circ \big(\mathtt{M}_0^{[k]}\big)^{-1}$, $\tilde{\mathtt{A}}^{[k]} = \mathtt{A} \circ \big(\mathtt{M}_0^{[k]}\big)^{-1}$, and $\tau^{[k]} = \hat{t}_{k+1} - \hat{t}_k$. Then the application of (4.15) on each interval $[\hat{t}_k, \hat{t}_{k+1}]$ results in Algorithm 3.

We conclude this section by defining the propagation operators of the above algorithm, which we shall use later. At step k, we define the (generally nonlinear) partial propagation operator $\mathtt{T}_{r,s}^{[k+1]} : \mathbb{R}^m \to \mathbb{R}^m$, using the intermediate quantities in the algorithm so that

$$\mathtt{T}_{r,s}^{[k+1]}(\mathtt{Y}^{[k]}) = \mathtt{Y}^{[k+1]}. \tag{4.18a}$$

Algorithm 3 SARK time stepping with subtents

1. If the input is $\mathtt{Y}_0$, an approximation to $\mathtt{Y}(0)$ at the tent bottom, then set $\mathtt{Y}^{[0]} = \mathtt{Y}_0$. If the input is $\mathtt{u}_0$, an approximation to $\hat{u}(0)$ at the tent bottom, then set $\mathtt{Y}^{[0]} = \mathtt{Y}_0 = \mathtt{M}_0(\mathtt{u}_0)$.

2. For $k = 0, 1, \ldots, r-1$ do:

   a) For $i = 1, 2, \ldots, s$, compute

$$\mathtt{Z}_i^{[k]} = \mathtt{Y}^{[k]} + \tau^{[k]} \sum_{j=1}^{i-1} d_{ij} \tilde{\mathtt{M}}_1^{[k]}(\mathtt{Z}_j^{[k]}) + \tau^{[k]} \sum_{j=1}^{i-1} a_{ij} \tilde{\mathtt{A}}^{[k]}(\mathtt{Z}_j^{[k]}).$$

   b) Compute

$$\mathtt{Y}^{[k+1]} = \mathtt{Y}^{[k]} + \tau^{[k]} \sum_{i=1}^{s} b_i \tilde{\mathtt{A}}^{[k]}(\mathtt{Z}_i^{[k]}).$$

3. Set

$$\mathtt{Y}_1^{r,s} = \mathtt{Y}^{[r]}.$$

   Output this as the approximation to $\mathtt{Y}(1)$ at the tent top.

Let the total propagation operator on the tent $\mathtt{T}_{r,s} : \mathbb{R}^m \to \mathbb{R}^m$ be defined by

$$\mathtt{T}_{r,s} = \mathtt{T}_{r,s}^{[r]} \circ \ldots \circ \mathtt{T}_{r,s}^{[2]} \circ \mathtt{T}_{r,s}^{[1]}. \tag{4.18b}$$

Clearly, the input and output of the algorithm are related to $\mathtt{T}_{r,s}$ by

$$\mathtt{Y}_1^{r,s} = \mathtt{T}_{r,s}(\mathtt{Y}_0). \tag{4.19}$$

4.2.1 Order conditions for the scheme

Appropriate values of a_{ij}, d_{ij}, and b_i can be found by order conditions obtained by matching terms in the Taylor expansions of the exact solution $\mathtt{Y}(\tau)$ and the discrete solution $\mathtt{Y}_\tau$. To derive these order conditions we follow the general methodology laid out in [19]. For this, we need to first compute the derivatives of the exact flow, then the derivatives of the discrete flow, followed by the formulation of resulting order conditions.

Derivatives of the exact solution

Continuing to use primes ($'$) for total derivatives with respect to a single variable like $d/d\tau$, to ease the tedious calculations below, we shall also employ the nth order Frechet derivative of a function $g : D \subset \mathbb{R}^m \to V$, for some vector space V. It is denoted by $g^{(n)}(\mathtt{w}) : \mathbb{R}^m \times \cdots \times \mathbb{R}^m \to V$ and defined by the symmetric multilinear form

$$g^{(n)}(\mathtt{w})(\mathtt{v}_1, \ldots, \mathtt{v}_n) = \sum_{i_1, i_2, \ldots, i_n = 1}^{m} \frac{\partial^n g(\mathtt{w})}{\partial x_{i_1} \cdots \partial x_{i_n}} [\mathtt{v}_1]_{i_1} \cdots [\mathtt{v}_n]_{i_n}$$

for any $\mathtt{v}_1, \dots, \mathtt{v}_n \in \mathbb{R}^m$. Whenever g and $\mathtt{w} : (0,1) \to \mathbb{R}^m$ are sufficiently smooth for the derivatives below to exist continuously, we have the following formulae:

$$\frac{d}{d\tau} g(\mathtt{w}(\tau)) = g^{(1)}(\mathtt{w}(\tau))(\mathtt{w}'(\tau)), \tag{4.20a}$$

$$\frac{d^2}{d\tau^2} g(\mathtt{w}(\tau)) = g^{(2)}(\mathtt{w}(\tau))(\mathtt{w}'(\tau), \mathtt{w}'(\tau)) + g^{(1)}(\mathtt{w}(\tau))(\mathtt{w}''(\tau)), \tag{4.20b}$$

$$\begin{aligned}\frac{d^3}{d\tau^3} g(\mathtt{w}(\tau)) &= g^{(3)}(\mathtt{w}(\tau))(\mathtt{w}'(\tau), \mathtt{w}'(\tau), \mathtt{w}'(\tau)) \\ &\quad + 3g^{(2)}(\mathtt{w}(\tau))(\mathtt{w}'(\tau), \mathtt{w}''(\tau)) + g^{(1)}(\mathtt{w}(\tau))(\mathtt{w}'''(\tau)),\end{aligned} \tag{4.20c}$$

$$\begin{aligned}\frac{d^4}{d\tau^4} g(\mathtt{w}(\tau)) &= g^{(4)}(\mathtt{w}(\tau))(\mathtt{w}'(\tau), \mathtt{w}'(\tau), \mathtt{w}'(\tau), \mathtt{w}'(\tau)) \\ &\quad + 6g^{(3)}(\mathtt{w}(\tau))(\mathtt{w}'(\tau), \mathtt{w}'(\tau), \mathtt{w}''(\tau)) + 4g^{(2)}(\mathtt{w}(\tau))(\mathtt{w}'(\tau), \mathtt{w}'''(\tau)) \\ &\quad + 3g^{(2)}(\mathtt{w}(\tau))(\mathtt{w}''(\tau), \mathtt{w}''(\tau)) + g^{(1)}(\mathtt{w}(\tau))(\mathtt{w}''''(\tau)).\end{aligned} \tag{4.20d}$$

These formulae can be derived by repeated application of the chain rule (or by applying the Faà di Bruno formula). We will also need to use

$$\frac{d^k}{d\tau^k}\big(\tau g(\mathtt{w}(\tau))\big) = \tau \frac{d^k}{d\tau^k} g(\mathtt{w}(\tau)) + k \frac{d^{k-1}}{d\tau^{k-1}} g(\mathtt{w}(\tau)), \tag{4.21}$$

which is a simple consequence of the Leibniz rule.

We start by computing the derivatives of $\mathtt{Z}(\tau)$ at $\tau = 0$. To express such derivatives concisely, we introduce the notation

$$\begin{aligned}\alpha &= \tilde{\mathtt{A}}(\mathtt{Z}(0)), & \alpha^{(n)}(\mathtt{v}_1, \dots, \mathtt{v}_n) &= \tilde{\mathtt{A}}^{(n)}(\mathtt{Z}(0))(\mathtt{v}_1, \dots, \mathtt{v}_n), \\ \mu &= \tilde{\mathtt{M}}_1(\mathtt{Z}(0)), & \mu^{(n)}(\mathtt{v}_1, \dots, \mathtt{v}_n) &= \tilde{\mathtt{M}}^{(n)}(\mathtt{Z}(0))(\mathtt{v}_1, \dots, \mathtt{v}_n).\end{aligned}$$

From (4.13a), it is immediate that $\mathtt{Z}'(0) = \tilde{\mathtt{A}}(\mathtt{Z}(0)) + \tilde{\mathtt{M}}(\mathtt{Z}(0))$. Thus,

$$\mathtt{Z}'(0) = \alpha + \mu. \tag{4.22a}$$

For the next derivative, we differentiate (4.13a) and with (4.21) we get

$$\mathtt{Z}''(\tau) = \tilde{\mathtt{A}}(\mathtt{Z}(\tau))' + (\tau \tilde{\mathtt{M}}_1(\mathtt{Z}(\tau)))'' = \tilde{\mathtt{A}}(\mathtt{Z}(\tau))' + \tau \tilde{\mathtt{M}}_1(\mathtt{Z}(\tau))'' + 2\,\tilde{\mathtt{M}}_1(\mathtt{Z}(\tau))'.$$

Simplifying $\mathtt{Z}''(\tau)$ using (4.20) yields

$$\begin{aligned}\mathtt{Z}''(\tau) &= \tilde{\mathtt{A}}^{(1)}(\mathtt{Z}(\tau))(\mathtt{Z}'(\tau)) + 2\,\tilde{\mathtt{M}}_1^{(1)}(\mathtt{Z}(\tau))(\mathtt{Z}'(\tau)) \\ &\quad + \tau\big(\tilde{\mathtt{M}}_1^{(2)}(\mathtt{Z}(\tau))(\mathtt{Z}'(\tau), \mathtt{Z}'(\tau)) + \tilde{\mathtt{M}}_1^{(1)}(\mathtt{Z}(\tau))(\mathtt{Z}''(\tau))\big)\end{aligned}$$

and by evaluating at $\tau = 0$, we obtain

$$\mathtt{Z}''(0) = \big(\alpha^{(1)} + 2\mu^{(1)}\big)(\alpha + \mu). \tag{4.22b}$$

By the same procedure, starting with $\mathtt{Z}'''(\tau) = \tilde{\mathtt{A}}(\mathtt{Z}(\tau))'' + (\tau\tilde{\mathtt{M}}_1(\mathtt{Z}(\tau)))'''$ and using (4.21) and (4.20), we obtain

$$\begin{aligned}\mathtt{Z}'''(\tau) &= \tilde{\mathtt{A}}(\mathtt{Z}(\tau))'' + \tau\tilde{\mathtt{M}}_1(\mathtt{Z}(\tau))''' + 3\,\tilde{\mathtt{M}}_1(\mathtt{Z}(\tau))'' \\ &= \tilde{\mathtt{A}}^{(2)}(\mathtt{Z}(\tau))(\mathtt{Z}'(\tau),\mathtt{Z}'(\tau)) + \tilde{\mathtt{A}}^{(1)}(\mathtt{Z}(\tau))(\mathtt{Z}''(\tau)) + \tau\tilde{\mathtt{M}}_1(\mathtt{Z}(\tau))''' \\ &\quad + 3\big(\tilde{\mathtt{M}}_1^{(2)}(\mathtt{Z}(\tau))(\mathtt{Z}'(\tau),\mathtt{Z}'(\tau)) + \tilde{\mathtt{M}}_1^{(1)}(\mathtt{Z}(\tau))(\mathtt{Z}''(\tau))\big).\end{aligned}$$

Evaluating at $\tau = 0$ leads to

$$\begin{aligned}\mathtt{Z}'''(0) &= \big(\alpha^{(2)} + 3\mu^{(2)}\big)(\alpha+\mu,\alpha+\mu) \\ &\quad + \big(\alpha^{(1)} + 3\mu^{(1)}\big)\big(\big(\alpha^{(1)} + 2\mu^{(1)}\big)(\alpha+\mu)\big).\end{aligned} \tag{4.22c}$$

Armed with (4.22), we proceed to compute the derivatives of $\mathtt{Y}$. The first derivative $\mathtt{Y}'(\tau) = \tilde{\mathtt{A}}(Z(\tau))$ is given by (4.13b). Using (4.20), we obtain the higher derivatives

$$\begin{aligned}\mathtt{Y}''(\tau) &= \tilde{\mathtt{A}}^{(1)}(\mathtt{Z}(\tau))(\mathtt{Z}'(\tau)), \\ \mathtt{Y}'''(\tau) &= \tilde{\mathtt{A}}^{(2)}(\mathtt{Z}(\tau))(\mathtt{Z}'(\tau),\mathtt{Z}'(\tau)) + \tilde{\mathtt{A}}^{(1)}(\mathtt{Z}(\tau))(\mathtt{Z}''(\tau)), \\ \mathtt{Y}''''(\tau) &= \tilde{\mathtt{A}}^{(3)}(\mathtt{Z}(\tau))(\mathtt{Z}'(\tau),\mathtt{Z}'(\tau),\mathtt{Z}'(\tau)) \\ &\quad + 3\,\tilde{\mathtt{A}}^{(2)}(\mathtt{Z}(\tau))(\mathtt{Z}'(\tau),\mathtt{Z}''(\tau)) + \tilde{\mathtt{A}}^{(1)}(\mathtt{Z}(\tau))(\mathtt{Z}'''(\tau)).\end{aligned}$$

Evaluating these derivatives at $\tau = 0$ using the previously computed derivatives of $\mathtt{Z}$ in (4.22), we get

$$\mathtt{Y}'(0) = \tilde{\mathtt{A}}(Z(0)) = \alpha, \tag{4.23a}$$

$$\mathtt{Y}''(0) = \alpha^{(1)}(\alpha+\mu), \tag{4.23b}$$

$$\mathtt{Y}'''(0) = \alpha^{(2)}(\alpha+\mu,\alpha+\mu) + \alpha^{(1)}\big(\big(\alpha^{(1)} + 2\mu^{(1)}\big)(\alpha+\mu)\big), \tag{4.23c}$$

$$\begin{aligned}\mathtt{Y}''''(0) &= \alpha^{(3)}(\alpha+\mu,\alpha+\mu,\alpha+\mu) \\ &\quad + 3\alpha^{(2)}(\alpha+\mu,\big(\alpha^{(1)} + 2\mu^{(1)}\big)(\alpha+\mu)) \\ &\quad + \alpha^{(1)}\big(\big(\alpha^{(2)} + 3\mu^{(2)}\big)(\alpha+\mu,\alpha+\mu)\big) \\ &\quad + \alpha^{(1)}\big(\big(\alpha^{(1)} + 3\mu^{(1)}\big)\big(\big(\alpha^{(1)} + 2\mu^{(1)}\big)(\alpha+\mu)\big)\big).\end{aligned} \tag{4.23d}$$

Derivatives of the discrete flow

The next task is to compute the coefficients of the Taylor expansion of the function $\mathtt{Y}_\tau$ defined in (4.15b). The arguments $\mathtt{Z}_i$ in (4.15b) are also functions of τ, as given by (4.15a). Therefore, in what follows, we first differentiate $\mathtt{Z}_i \equiv \mathtt{Z}_i(\tau)$ and then $\mathtt{Y}_\tau$.

Obviously, $\mathtt{Z}_i(0)$ and $\mathtt{Z}(0)$ coincide, so we will focus on the first and higher derivatives of

$\mathtt{Z}_i$ at $\tau = 0$. To this end, we differentiate (4.15a) k times, using (4.21), to get

$$\begin{aligned}\frac{d^k \mathtt{Z}_i}{d\tau^k} &= \sum_{j<i} \left[d_{ij} \frac{d^k}{d\tau^k}(\tau \tilde{\mathtt{M}}_1(\mathtt{Z}_j(\tau))) + a_{ij} \frac{d^k}{d\tau^k}(\tau \tilde{\mathtt{A}}(\mathtt{Z}_j(\tau))) \right] \\ &= \sum_{j<i} \left[\tau \frac{d^k}{d\tau^k} \Big(d_{ij} \tilde{\mathtt{M}}_1(\mathtt{Z}_j(\tau)) + a_{ij} \tilde{\mathtt{A}}(\mathtt{Z}_j(\tau)) \Big) \right. \\ &\qquad \left. + k \frac{d^{k-1}}{d\tau^{k-1}} \Big(d_{ij} \tilde{\mathtt{M}}_1(\mathtt{Z}_j(\tau)) + a_{ij} \tilde{\mathtt{A}}(\mathtt{Z}_j(\tau)) \Big) \right].\end{aligned}$$

The kth derivatives vanish when evaluating at $\tau = 0$, thus we introduce

$$\theta_k(\tau) := \frac{d^k}{d\tau^k} \Big(d_{ij} \tilde{\mathtt{M}}_1(\mathtt{Z}_j(\tau)) + a_{ij} \tilde{\mathtt{A}}(\mathtt{Z}_j(\tau)) \Big),$$

containing the vanishing terms. The kth derivative of $\mathtt{Z}_i$ reads

$$\frac{d^k \mathtt{Z}_i}{d\tau^k} = \sum_{j<i} \left[k \frac{d^{k-1}}{d\tau^{k-1}} \Big(d_{ij} \tilde{\mathtt{M}}_1(\mathtt{Z}_j(\tau)) + a_{ij} \tilde{\mathtt{A}}(\mathtt{Z}_j(\tau)) \Big) + \tau \theta_k(\tau) \right]. \tag{4.24}$$

The first derivative

$$\mathtt{Z}_i'(\tau) = \sum_{j<i} \left[d_{ij} \tilde{\mathtt{M}}_1(\mathtt{Z}_j(\tau)) + a_{ij} \tilde{\mathtt{A}}(\mathtt{Z}_j(\tau)) + \tau \theta_1(\tau) \right] \tag{4.25a}$$

is directly given by (4.24) for $k = 1$. Using (4.20), we obtain the higher derivatives

$$\begin{aligned}\mathtt{Z}_i''(\tau) &= \sum_{j<i} \left[2 \Big(d_{ij} \tilde{\mathtt{M}}_1(\mathtt{Z}_j(\tau))' + a_{ij} \tilde{\mathtt{A}}(\mathtt{Z}_j(\tau))' \Big) + \tau \theta_2(\tau) \right] \qquad (4.25b) \\ &= \sum_{j<i} \left[2 \Big(d_{ij} \tilde{\mathtt{M}}_1^{(1)}(\mathtt{Z}_j(\tau))(\mathtt{Z}_j'(\tau)) + a_{ij} \tilde{\mathtt{A}}^{(1)}(\mathtt{Z}_j(\tau))(\mathtt{Z}_j'(\tau)) \Big) + \tau \theta_2(\tau) \right]. \\ \mathtt{Z}_i'''(\tau) &= \sum_{j<i} \left[3 \Big(d_{ij} \tilde{\mathtt{M}}_1(\mathtt{Z}_j(\tau))'' + a_{ij} \tilde{\mathtt{A}}(\mathtt{Z}_j(\tau))'' \Big) + \tau \theta_3(\tau) \right] \\ &= \sum_{j<i} \left[3\, d_{ij} \Big(\tilde{\mathtt{M}}_1^{(2)}(\mathtt{Z}_j(\tau))(\mathtt{Z}_j'(\tau), \mathtt{Z}_j'(\tau)) + \tilde{\mathtt{M}}_1^{(1)}(\mathtt{Z}_j(\tau))(\mathtt{Z}_j''(\tau)) \Big) \right. \qquad (4.25c) \\ &\qquad \left. + 3\, a_{ij} \Big(\tilde{\mathtt{A}}^{(2)}(\mathtt{Z}_j(\tau))(\mathtt{Z}_j'(\tau), \mathtt{Z}_j'(\tau)) + \tilde{\mathtt{A}}^{(1)}(\mathtt{Z}_j(\tau))(\mathtt{Z}_j''(\tau)) \Big) + \tau \theta_3(\tau) \right].\end{aligned}$$

Evaluating (4.25) at $\tau = 0$, we get

$$\mathtt{Z}_i'(0) = \sum_{j<i} d_{ij}\mu + a_{ij}\alpha, \tag{4.26a}$$

$$\mathtt{Z}_i''(0) = 2\sum_{j<i}\sum_{k<j} \big(d_{ij}\mu^{(1)} + a_{ij}\alpha^{(1)}\big)(d_{jk}\mu + a_{jk}\alpha), \tag{4.26b}$$

$$\begin{aligned}\mathtt{Z}_i'''(0) = {} & 3\sum_{j<i}\sum_{k<j}\sum_{l<j} \big(d_{ij}\mu^{(2)} + a_{ij}\alpha^{(2)}\big)(d_{jk}\mu + a_{jk}\alpha, d_{jl}\mu + a_{jl}\alpha), \\ & + 6\sum_{j<i}\sum_{k<j}\sum_{l<k} \big(d_{ij}\mu^{(1)} + a_{ij}\alpha^{(1)}\big)\big(\big(d_{jk}\mu^{(1)} + a_{jk}\alpha^{(1)}\big)(d_{kl}\mu + a_{lk}\alpha)\big).\end{aligned} \tag{4.26c}$$

Next, we focus on $\mathtt{Y}_\tau$. By (4.15b), using (4.21), we obtain

$$\frac{d^k \mathtt{Y}_\tau}{d\tau^k} = \sum_{i=1}^{s} b_i \frac{d^k}{d\tau^k}(\tau\tilde{\mathtt{A}}(\mathtt{Z}_i(\tau))) = \sum_{i=1}^{s} b_i \Big[k\frac{d^{k-1}}{d\tau^{k-1}}(\tilde{\mathtt{A}}(\mathtt{Z}_i(\tau))) + \tau\frac{d^k}{d\tau^k}(\tilde{\mathtt{A}}(\mathtt{Z}_i(\tau)))\Big]. \tag{4.27}$$

Using (4.27) for $k = 1, 2, 3, 4$ and (4.20), the derivatives of $\mathtt{Y}_\tau$ read

$$\begin{aligned}
\mathtt{Y}_\tau'(\tau) &= \sum_{i=1}^{s} b_i \Big[\tilde{\mathtt{A}}(\mathtt{Z}_i(\tau)) + \tau\tilde{\mathtt{A}}(\mathtt{Z}_i(\tau))'\Big], \\
\mathtt{Y}_\tau''(\tau) &= \sum_{i=1}^{s} b_i \Big[2\,\tilde{\mathtt{A}}^{(1)}(\mathtt{Z}_i(\tau))(\mathtt{Z}_i'(\tau)) + \tau\tilde{\mathtt{A}}(\mathtt{Z}_i(\tau))''\Big], \\
\mathtt{Y}_\tau'''(\tau) &= \sum_{i=1}^{s} b_i \Big[3\Big(\tilde{\mathtt{A}}^{(2)}(\mathtt{Z}_i(\tau))(\mathtt{Z}_i'(\tau), \mathtt{Z}_i'(\tau)) + \tilde{\mathtt{A}}^{(1)}(\mathtt{Z}_i(\tau))(\mathtt{Z}_i''(\tau))\Big) + \tau\tilde{\mathtt{A}}(\mathtt{Z}_i(\tau))'''\Big], \\
\mathtt{Y}_\tau''''(\tau) &= \sum_{i=1}^{s} b_i \Big[4\Big(\tilde{\mathtt{A}}^{(3)}(\mathtt{Z}_i(\tau))(\mathtt{Z}_i'(\tau), \mathtt{Z}_i'(\tau), \mathtt{Z}_i'(\tau)) + 3\,\tilde{\mathtt{A}}^{(2)}(\mathtt{Z}_i(\tau))(\mathtt{Z}_i'(\tau), \mathtt{Z}_i''(\tau)) \\
&\qquad + \tilde{\mathtt{A}}^{(1)}(\mathtt{Z}_i(\tau))(\mathtt{Z}_i'''(\tau))\Big) + \tau\tilde{\mathtt{A}}(\mathtt{Z}_i(\tau))''''\Big].
\end{aligned}$$

Evaluating the resulting terms at $\tau = 0$ by means of (4.26), we obtain

$$\mathtt{Y}'_\tau(0) = \sum_{i=1}^{s} b_i \alpha, \tag{4.28a}$$

$$\mathtt{Y}''_\tau(0) = 2\sum_{i=1}^{s}\sum_{j<i} b_i \alpha^{(1)}(d_{ij}\mu + a_{ij}\alpha), \tag{4.28b}$$

$$\begin{aligned}\mathtt{Y}'''_\tau(0) = {} & 3\sum_{i=1}^{s}\sum_{j<i}\sum_{k<i} b_i \alpha^{(2)}(d_{ij}\mu + a_{ij}\alpha, d_{ik}\mu + a_{ik}\alpha) \\ & + 6\sum_{i=1}^{s}\sum_{j<i}\sum_{k<j} b_i \alpha^{(1)}\big((d_{ij}\mu^{(1)} + a_{ij}\alpha^{(1)})(d_{jk}\mu + a_{jk}\alpha)\big),\end{aligned} \tag{4.28c}$$

$$\begin{aligned}\mathtt{Y}''''_\tau(0) = {} & 4\sum_{i=1}^{s}\sum_{j<i}\sum_{k<i}\sum_{l<i} b_i \alpha^{(3)}(d_{ij}\mu + a_{ij}\alpha, d_{ik}\mu + a_{ik}\alpha, d_{il}\mu + a_{il}\alpha) \\ & + 24\sum_{i=1}^{s}\sum_{j<i}\sum_{k<i}\sum_{l<k} b_i \alpha^{(2)}(d_{ij}\mu + a_{ij}\alpha, (d_{ik}\mu^{(1)} + a_{ik}\alpha^{(1)})(d_{kl}\mu + a_{kl}\alpha)) \\ & + 12\sum_{i=1}^{s}\sum_{j<i}\sum_{k<j}\sum_{l<j} b_i \alpha^{(1)}\big((d_{ij}\mu^{(2)} + a_{ij}\alpha^{(2)})(d_{jk}\mu + a_{jk}\alpha, d_{jl}\mu + a_{jl}\alpha)\big) \\ & + 24\sum_{i=1}^{s}\sum_{j<i}\sum_{k<j}\sum_{l<k} b_i \alpha^{(1)}\big((d_{ij}\mu^{(1)} + a_{ij}\alpha^{(1)})\big((d_{jk}\mu^{(1)} + a_{jk}\alpha^{(1)})(d_{kl}\mu + a_{lk}\alpha)\big)\big).\end{aligned} \tag{4.28d}$$

Formulation of order conditions

To obtain a specific method, we find values for a_{ij}, d_{ij} and b_i by matching the coefficients in the Taylor expansions of $\mathtt{Y}(\tau)$ and $\mathtt{Y}_\tau$. Note that $\mathtt{Y}_\tau(0) = \mathtt{Y}_0 = \mathtt{Y}(0)$, so the 0th order coefficients match.

The next terms in the Taylor expansions will match if $\mathtt{Y}'(0) = \mathtt{Y}'_\tau(0)$. For this it is sufficient that

$$\sum_{i=1}^{s} b_i = 1, \tag{4.29}$$

because of (4.23a) and (4.28a). To match the third terms in the Taylor expansions, equating (4.23b) and (4.28b),

$$\alpha^{(1)}(\alpha) + \alpha^{(1)}(\mu) = \sum_{i=1}^{s}\sum_{j<i} 2b_i d_{ij}\alpha^{(1)}(\mu) + 2b_i a_{ij}\alpha^{(1)}(\alpha).$$

Equating the coefficients of $\alpha^{(1)}(\alpha)$ and $\alpha^{(1)}(\mu)$, we conclude that $\mathtt{Y}''(0) = \mathtt{Y}''_\tau(0)$ if

$$2\sum_{i=1}^{s}\sum_{j<i} b_i d_{ij} = 1 \quad \text{and} \quad 2\sum_{i=1}^{s}\sum_{j<i} b_i a_{ij} = 1. \tag{4.30}$$

To match the next higher order terms, $\mathtt{Y}_\tau'''(0) = \mathtt{Y}'''(0)$, the expressions in (4.23c) and (4.28c) must be equated, i.e.,

$$\begin{aligned}
&\alpha^{(2)}(\alpha,\alpha) + 2\alpha^{(2)}(\alpha,\mu) + \alpha^{(2)}(\mu,\mu) \\
&\quad + \alpha^{(1)}(\alpha^{(1)}(\alpha)) + \alpha^{(1)}(\alpha^{(1)}(\mu)) + 2\alpha^{(1)}(\mu^{(1)}(\alpha)) + 2\alpha^{(1)}(\mu^{(1)}(\mu)) \\
&\quad = \sum_{i=1}^{s}\sum_{j<i}\sum_{k<i}\Big[3b_i d_{ij} d_{ik}\alpha^{(2)}(\mu,\mu) + 6b_i d_{ij} a_{ik}\alpha^{(2)}(\mu,\alpha) + 3b_i a_{ij} a_{ik}\alpha^{(2)}(\alpha,\alpha)\Big] \\
&\quad + 6\sum_{i=1}^{s}\sum_{j<i}\sum_{k<j}\Big[b_i d_{ij} d_{jk}\alpha^{(1)}(\mu^{(1)}(\mu)) + b_i d_{ij} a_{jk}\alpha^{(1)}(\mu^{(1)}(\alpha)) \\
&\qquad\qquad + b_i a_{ij} d_{jk}\alpha^{(1)}(\alpha^{(1)}(\mu)) + b_i a_{ij} a_{jk}\alpha^{(1)}(\alpha^{(1)}(\alpha))\Big].
\end{aligned}$$

For this equality to hold, the following seven conditions are sufficient as can be seen by equating the coefficients of $\alpha^{(2)}(\alpha,\alpha)$, $\alpha^{(2)}(\mu,\mu)$, $\alpha^{(2)}(\alpha,\mu)$, $\alpha^{(1)}(\alpha^{(1)}(\alpha))$, $\alpha^{(1)}(\alpha^{(1)}(\mu))$, $\alpha^{(1)}(\mu^{(1)}(\alpha))$, and $\alpha^{(1)}(\mu^{(1)}(\mu))$, respectively:

$$3\sum_{i=1}^{s} b_i\Big(\sum_{j<i} a_{ij}\Big)^2 = 1, \tag{4.31a}$$

$$3\sum_{i=1}^{s} b_i\Big(\sum_{j<i} d_{ij}\Big)^2 = 1, \tag{4.31b}$$

$$3\sum_{i=1}^{s} b_i\Big(\sum_{j<i} a_{ij}\Big)\Big(\sum_{j<i} d_{ij}\Big) = 1, \tag{4.31c}$$

$$6\sum_{i=1}^{s}\sum_{j<i}\sum_{k<j} b_i a_{ij} a_{jk} = 1, \tag{4.31d}$$

$$6\sum_{i=1}^{s}\sum_{j<i}\sum_{k<j} b_i a_{ij} d_{jk} = 1, \tag{4.31e}$$

$$3\sum_{i=1}^{s}\sum_{j<i}\sum_{k<j} b_i d_{ij} a_{jk} = 1, \tag{4.31f}$$

$$3\sum_{i=1}^{s}\sum_{j<i}\sum_{k<j} b_i d_{ij} d_{jk} = 1. \tag{4.31g}$$

If one desires to further match the next higher order terms, $\mathtt{Y}_\tau''''(0) = \mathtt{Y}''''(0)$, the expressions in (4.23d) and (4.28d) must be equated. By equating the coefficients of $\alpha^{(3)}(\alpha,\alpha,\alpha)$,

$\alpha^{(3)}(\alpha,\alpha,\mu)$, $\alpha^{(3)}(\alpha,\mu,\mu)$, and $\alpha^{(3)}(\mu,\mu,\mu)$, we get the conditions

$$4\sum_{i=1}^{s} b_i\Big(\sum_{j<i} a_{ij}\Big)^3 = 1, \tag{4.32a}$$

$$4\sum_{i=1}^{s} b_i\Big(\sum_{j<i} a_{ij}\Big)^2\Big(\sum_{j<i} d_{ij}\Big) = 1, \tag{4.32b}$$

$$4\sum_{i=1}^{s} b_i\Big(\sum_{j<i} a_{ij}\Big)\Big(\sum_{j<i} d_{ij}\Big)^2 = 1, \tag{4.32c}$$

$$4\sum_{i=1}^{s} b_i\Big(\sum_{j<i} d_{ij}\Big)^3 = 1. \tag{4.32d}$$

The next eight conditions are obtained by equating the coefficients of $\alpha^{(2)}(\alpha,\alpha^{(1)}(\alpha))$, $\alpha^{(2)}(\alpha,\alpha^{(1)}(\mu))$, $\alpha^{(2)}(\alpha,\mu^{(1)}(\alpha))$, $\alpha^{(2)}(\alpha,\mu^{(1)}(\mu))$, $\mu^{(2)}(\alpha,\alpha^{(1)}(\alpha))$, $\mu^{(2)}(\alpha,\alpha^{(1)}(\mu))$, $\mu^{(2)}(\alpha,\mu^{(1)}(\alpha))$, and $\mu^{(2)}(\alpha,\mu^{(1)}(\mu))$:

$$8\sum_{i=1}^{s} b_i\Big(\sum_{j<i} a_{ij}\Big)\Big(\sum_{j<i}\sum_{k<j} a_{ij}a_{jk}\Big) = 1, \tag{4.32e}$$

$$8\sum_{i=1}^{s} b_i\Big(\sum_{j<i} a_{ij}\Big)\Big(\sum_{j<i}\sum_{k<j} a_{ij}d_{jk}\Big) = 1, \tag{4.32f}$$

$$8\sum_{i=1}^{s} b_i\Big(\sum_{j<i} a_{ij}\Big)\Big(\sum_{j<i}\sum_{k<j} d_{ij}a_{jk}\Big) = 1, \tag{4.32g}$$

$$8\sum_{i=1}^{s} b_i\Big(\sum_{j<i} a_{ij}\Big)\Big(\sum_{j<i}\sum_{k<j} d_{ij}a_{jk}\Big) = 1, \tag{4.32h}$$

$$4\sum_{i=1}^{s} b_i\Big(\sum_{j<i} d_{ij}\Big)\Big(\sum_{j<i}\sum_{k<j} a_{ij}a_{jk}\Big) = 1, \tag{4.32i}$$

$$4\sum_{i=1}^{s} b_i\Big(\sum_{j<i} d_{ij}\Big)\Big(\sum_{j<i}\sum_{k<j} a_{ij}d_{jk}\Big) = 1, \tag{4.32j}$$

$$4\sum_{i=1}^{s} b_i\Big(\sum_{j<i} d_{ij}\Big)\Big(\sum_{j<i}\sum_{k<j} d_{ij}a_{jk}\Big) = 1, \tag{4.32k}$$

$$4\sum_{i=1}^{s} b_i\Big(\sum_{j<i} d_{ij}\Big)\Big(\sum_{j<i}\sum_{k<j} d_{ij}a_{jk}\Big) = 1. \tag{4.32l}$$

Equating the coefficients of $\alpha^{(1)}(\alpha^{(2)}(\alpha,\alpha))$, $\alpha^{(1)}(\alpha^{(2)}(\alpha,\mu))$, $\alpha^{(1)}(\alpha^{(2)}(\mu,\mu))$,

$\alpha^{(1)}(\mu^{(2)}(\alpha,\alpha))$, $\alpha^{(1)}(\mu^{(2)}(\alpha,\mu))$, and $\alpha^{(1)}(\mu^{(2)}(\mu,\mu))$ leads further six conditions

$$12\sum_{i=1}^{s}\sum_{j<i} b_i a_{ij}\Big(\sum_{k<j} a_{jk}\Big)^2 = 1, \tag{4.32m}$$

$$12\sum_{i=1}^{s}\sum_{j<i} b_i a_{ij}\Big(\sum_{k<j} a_{jk}\Big)\Big(\sum_{k<j} d_{jk}\Big) = 1, \tag{4.32n}$$

$$12\sum_{i=1}^{s}\sum_{j<i} b_i a_{ij}\Big(\sum_{k<j} d_{jk}\Big)^2 = 1, \tag{4.32o}$$

$$4\sum_{i=1}^{s}\sum_{j<i} b_i d_{ij}\Big(\sum_{k<j} a_{jk}\Big)^2 = 1, \tag{4.32p}$$

$$4\sum_{i=1}^{s}\sum_{j<i} b_i d_{ij}\Big(\sum_{k<j} a_{jk}\Big)\Big(\sum_{k<j} d_{jk}\Big) = 1, \tag{4.32q}$$

$$4\sum_{i=1}^{s}\sum_{j<i} b_i d_{ij}\Big(\sum_{k<j} d_{jk}\Big)^2 = 1. \tag{4.32r}$$

The final eight conditions are obtained by equating the coefficients of $\alpha^{(1)}(\alpha^{(1)}(\alpha^{(1)}(\alpha)))$, $\alpha^{(1)}(\alpha^{(1)}(\alpha^{(1)}(\mu)))$, $\alpha^{(1)}(\alpha^{(1)}(\mu^{(1)}(\alpha)))$, $\alpha^{(1)}(\alpha^{(1)}(\mu^{(1)}(\mu)))$, $\alpha^{(1)}(\mu^{(1)}(\alpha^{(1)}(\alpha)))$, $\alpha^{(1)}(\mu^{(1)}(\alpha^{(1)}(\mu)))$, $\alpha^{(1)}(\mu^{(1)}(\mu^{(1)}(\alpha)))$, and $\alpha^{(1)}(\mu^{(1)}(\mu^{(1)}(\mu)))$:

$$24\sum_{i=1}^{s}\sum_{j<i}\sum_{k<j}\sum_{l<k} b_i a_{ij} a_{jk} a_{kl} = 1, \tag{4.32s}$$

$$24\sum_{i=1}^{s}\sum_{j<i}\sum_{k<j}\sum_{l<k} b_i a_{ij} a_{jk} d_{kl} = 1, \tag{4.32t}$$

$$12\sum_{i=1}^{s}\sum_{j<i}\sum_{k<j}\sum_{l<k} b_i a_{ij} d_{jk} a_{kl} = 1, \tag{4.32u}$$

$$12\sum_{i=1}^{s}\sum_{j<i}\sum_{k<j}\sum_{l<k} b_i a_{ij} d_{jk} d_{kl} = 1, \tag{4.32v}$$

$$8\sum_{i=1}^{s}\sum_{j<i}\sum_{k<j}\sum_{l<k} b_i d_{ij} a_{jk} a_{kl} = 1, \tag{4.32w}$$

$$8\sum_{i=1}^{s}\sum_{j<i}\sum_{k<j}\sum_{l<k} b_i d_{ij} a_{jk} d_{kl} = 1, \tag{4.32x}$$

$$4\sum_{i=1}^{s}\sum_{j<i}\sum_{k<j}\sum_{l<k} b_i d_{ij} d_{jk} a_{kl} = 1, \tag{4.32y}$$

$$4\sum_{i=1}^{s}\sum_{j<i}\sum_{k<j}\sum_{l<k} b_i d_{ij} d_{jk} d_{kl} = 1. \tag{4.32z}$$

Matching the coefficients of the fourth order derivatives of $\mathtt{Y}''''$ and $\mathtt{Y}_\tau''''$ at $\tau = 0$ leads to a total number of 26 conditions.

Thus, we have proved the following result, which summarizes our discussions on order conditions.

Theorem 3. *Whenever* $\tilde{\mathtt{A}}$ *and* $\tilde{\mathtt{M}}$ *are smooth enough for the derivatives below to exist continuously,*

1. *the condition* (4.29) *implies* $\mathtt{Y}'(0) = \mathtt{Y}_\tau'(0)$,
2. *the conditions of* (4.30) *imply* $\mathtt{Y}''(0) = \mathtt{Y}_\tau''(0)$,
3. *the conditions of* (4.31) *imply* $\mathtt{Y}'''(0) = \mathtt{Y}_\tau'''(0)$, *and*
4. *the conditions of* (4.32) *imply* $\mathtt{Y}''''(0) = \mathtt{Y}_\tau''''(0)$.

4.2.2 Examples of methods up to third order

Observe that the standard order conditions of Runge-Kutta methods are a subset of the order conditions (4.29) - (4.31). Thus we base our SARK methods on existing Runge-Kutta methods. Below, we shall refer to an s-stage SARK method based on an existing Runge-Kutta method called "RKname" as "SARK(s, RKname)".

A second order two-stage SARK method can be derived from a second order Runge-Kutta method once we find d_{ij} satisfying the additional condition

$$2\sum_{i=1}^{2}\sum_{j<i} b_i d_{ij} = 1 \quad \Leftrightarrow \quad b_2 d_{21} = \frac{1}{2}, \tag{4.33}$$

which was introduced in (4.30). For example, one may start with the standard explicit midpoint rule and select $d_{21} = 1/2$ to satisfy (4.33), thus arriving at the "SARK(2, midpoint)" method, listed first in Table 4.1. The table continues on to display further such methods obtained from other well-known second order Runge-Kutta schemes.

The third order SARK methods in Table 4.2 are based on known third order Runge-Kutta methods with three stages. The additional coefficients d_{ij} are chosen, such that (4.30)-(4.31) are satisfied.

0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0
	0	1		

(a) SARK(2, midpoint), based on the explicit midpoint rule

0	0	0	0	0
$\frac{2}{3}$	$\frac{2}{3}$	0	$\frac{2}{3}$	0
	$\frac{1}{4}$	$\frac{3}{4}$		

(b) SARK(2, Ralston), based on Ralston's second order method

0	0	0	0	0
1	1	0	1	0
	$\frac{1}{2}$	$\frac{1}{2}$		

(c) SARK(2, Heun), based on Heun's second order method

Table 4.1: Examples of two-stage SARK methods.

0	0	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	0	0
1	-1	2	0	-3	4	0
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$			

(a) SARK(3, Kutta) method, based on Kutta's third order method

0	0	0	0	0	0	0
$\frac{1}{3}$	$\frac{1}{3}$	0	0	$\frac{1}{3}$	0	0
$\frac{2}{3}$	0	$\frac{2}{3}$	0	$-\frac{2}{3}$	$\frac{4}{3}$	0
	$\frac{1}{4}$	0	$\frac{3}{4}$			

(b) SARK(3, Heun) method, based on Heun's third order method

Table 4.2: Examples of three-stage SARK methods.

4.2.3 Fourth order methods

In the previous section we constructed s-stage SARK methods of order s. As we will see in this section, such methods do not exist for $s = 4$. Finally we discuss the construction of 5-stage SARK method of fourth order.

Nonexistence of 4-stage, fourth order SARK methods

Since the standard Runge-Kutta order conditions are a subset of the SARK order conditions, we start with a set of coefficients $a_{ij},\ i,j = 1,\dots,4$ and $b_i, c_i,\ i = 1,\dots,4$ forming an explicit fourth order Runge-Kutta method with $s = 4$ stages. This class of methods is derived in [19, Chapter II.1] and has to satisfy the conditions $c_1 = 0$ and $c_4 = 1$. Further considerations show, that all possible sets of Runge-Kutta coefficients can be represented by one of the following four cases.

RK4.I $c_2 \neq c_3$ with $0 < c_i < 1, i \in \{2,3\}$ and

$$b_1 = \frac{1 - 2(c_2 + c_3) + 6c_2c_3}{12c_2c_3}, \qquad b_2 = \frac{2c_3 - 1}{12c_2(c_2 - 1)(c_3 - c_2)},$$
$$b_3 = \frac{1 - 2c_2}{12c_3(c_3 - 1)(c_2 - c_3)}, \qquad b_4 = \frac{3 - 4(c_2 + c_3) + 6c_2c_3}{12(1 - c_2)(1 - c_3)},$$

with c_2, c_3 such that $b_3 \neq 0$ and $b_4 \neq 0$.

RK4.II $c_2 = \frac{1}{2},\ c_3 = 0,\ b_1 = \frac{1}{6} - b_3,\ b_2 = \frac{4}{6},\ b_3 \neq 0,\ b_4 = \frac{1}{6}$

RK4.III $c_2 = c_3 = \frac{1}{2},\ b_1 = \frac{1}{6},\ b_2 = \frac{4}{6} - b_3,\ b_3 \neq 0,\ b_4 = \frac{1}{6}$

RK4.IV $c_2 = 1,\ c_3 = \frac{1}{2},\ b_1 = \frac{1}{6},\ b_2 = \frac{1}{6} - b_4,\ b_3 = \frac{4}{6},\ b_4 \neq 0$

Note that there holds $b_3 \neq 0$ and $b_4 \neq 0$ for all cases. For $s = 4$, the conditions (4.32s) and (4.32t) yield

$$24b_4a_{43}a_{32}a_{21} = 1 \quad \text{and} \quad 24b_4a_{43}a_{32}d_{21} = 1,$$

which implies, using the consistency condition (4.16), that there holds

$$d_{21} = a_{21} = c_2. \tag{4.34a}$$

The conditions (4.31d) and (4.31e) read

$$\begin{aligned} 6\big[b_3 a_{32} a_{21} + b_4 a_{42} a_{21} + b_4 a_{43} \overbrace{(a_{32} + a_{31})}^{=c_3}\big] &= 1, \\ 6\big[b_3 a_{32} d_{21} + b_4 a_{42} d_{21} + b_4 a_{43}(d_{32} + d_{31})\big] &= 1, \end{aligned}$$

where we again used the consistency condition (4.16). Subtracting these two equations leads to

$$6 b_4 a_{43}\big(c_3 - (d_{32} + d_{31})\big) = 0,$$

and we obtain

$$d_{32} + d_{31} = c_3, \tag{4.34b}$$

since b_4 and a_{43} have to be nonzero to satisfy the order conditions of the underlying Runge-Kutta method. Proceeding with the conditions (4.30), we obtain

$$\begin{aligned} 2\big[b_2 c_2 + b_3 c_3 + b_4 c_4\big] &= 1, \\ 2\big[b_2 \underbrace{d_{21}}_{=c_2} + b_3 \underbrace{(d_{32} + d_{31})}_{=c_3} + b_4(d_{43} + d_{42} + d_{41})\big] &= 1, \end{aligned}$$

which leads to the condition

$$d_{43} + d_{42} + d_{41} = c_4 = 1. \tag{4.34c}$$

Using the consistency condition (4.16) and the corresponding condition (4.34) for the additional coefficients $d_{ij},\ i, j = 1, \dots, 4$, the order conditions for second order SARK methods reduce to the standard Runge-Kutta conditions

$$\sum_{i=1}^{4} b_i = 1, \tag{4.35a}$$

$$\sum_{i=1}^{4} b_i c_i = \frac{1}{2}. \tag{4.35b}$$

For the extension to third order, they have to fulfill the standard Runge-Kutta conditions

$$\sum_{i=1}^{4} b_i c_i^2 = \frac{1}{3}, \tag{4.35c}$$

$$\sum_{i=1}^{4} \sum_{j<i} b_i a_{ij} c_j = \frac{1}{6}, \tag{4.35d}$$

and one additional condition

$$\sum_{i=1}^{4} \sum_{j<i} b_i d_{ij} c_j = \frac{1}{3}. \tag{4.35e}$$

Going to fourth order, the conditions separate into the standard Runge-Kutta conditions

$$\sum_{i=1}^{4} b_i c_i^3 = \frac{1}{4}, \tag{4.35f}$$

$$\sum_{i=1}^{4}\sum_{j<i} b_i c_i a_{ij} c_j = \frac{1}{8}, \tag{4.35g}$$

$$\sum_{i=1}^{4}\sum_{j<i} b_i a_{ij} c_j^2 = \frac{1}{12}, \tag{4.35h}$$

$$\sum_{i=1}^{4}\sum_{j<i}\sum_{k<j} b_i a_{ij} a_{jk} c_k = \frac{1}{24}, \tag{4.35i}$$

and five additional SARK conditions

$$\sum_{i=1}^{4}\sum_{j<i} b_i c_i d_{ij} c_j = \frac{1}{4}, \tag{4.35j}$$

$$\sum_{i=1}^{4}\sum_{j<i} b_i d_{ij} c_j^2 = \frac{1}{4}, \tag{4.35k}$$

$$\sum_{i=1}^{4}\sum_{j<i}\sum_{k<j} b_i d_{ij} d_{jk} c_k = \frac{1}{4}, \tag{4.35l}$$

$$\sum_{i=1}^{4}\sum_{j<i}\sum_{k<j} b_i d_{ij} a_{jk} c_k = \frac{1}{8}, \tag{4.35m}$$

$$\sum_{i=1}^{4}\sum_{j<i}\sum_{k<j} b_i a_{ij} d_{jk} c_k = \frac{1}{12}. \tag{4.35n}$$

Based on a 4-stage, fourth order Runge-Kutta method, given by RK4.I–RK4.IV, we now have to find a set of coefficients $d_{ij},\ i,j = 1,\dots,4$ satisfying

$$b_3 d_{32} c_2 + b_4(d_{42} c_2 + d_{43} c_3) = \frac{1}{3}, \tag{4.35e}$$

$$b_3 c_3 d_{32} c_2 + b_4 c_4(d_{42} c_2 + d_{43} c_3) = \frac{1}{4}, \tag{4.35j}$$

$$b_3 d_{32} c_2^2 + b_4(d_{42} c_2^2 + d_{43} c_3^2) = \frac{1}{4}, \tag{4.35k}$$

$$b_4 d_{43} d_{32} c_2 = \frac{1}{4}, \tag{4.35l}$$

$$b_4 d_{43} a_{32} c_2 = \frac{1}{4}, \tag{4.35m}$$

$$b_4 a_{43} d_{32} c_2 = \frac{1}{4}, \tag{4.35n}$$

and the consistency condition (4.34) to obtain a 4-stage, fourth order SARK method. The conditions (4.35e), (4.35j) and (4.35k) lead to the linear system

$$\begin{pmatrix} b_3c_2 & b_4c_2 & b_4c_3 \\ b_3c_3c_2 & b_4c_2 & b_4c_3 \\ b_3c_2^2 & b_4c_2^2 & b_4c_3^2 \end{pmatrix} \begin{pmatrix} d_{32} \\ d_{42} \\ d_{43} \end{pmatrix} = \frac{1}{12} \begin{pmatrix} 4 \\ 3 \\ 3 \end{pmatrix}, \tag{4.36}$$

where we substituted the condition $c_4 = 1$. The determinant of the linear system (4.36) yields

$$\det \begin{pmatrix} b_3c_2 & b_4c_2 & b_4c_3 \\ b_3c_3c_2 & b_4c_2 & b_4c_3 \\ b_3c_2^2 & b_4c_2^2 & b_4c_3^2 \end{pmatrix} = b_3b_4^2c_2^2c_3(c_3 - c_2)(1 - c_3). \tag{4.37}$$

For the cases RK4.II and RK4.III, with $c_3 = 0$ and $c_2 = c_3$ respectively, the determinant of the linear system is zero, which implies the nonexistence of a solution for these cases. The determinant (4.37) is nonzero for the other cases, since there holds $b_3 \neq 0$ and $b_4 \neq 0$. Based on the coefficient in RK4.I, the solution of (4.36) is

$$d_{32} = \frac{-1}{12b_3c_2(c_3 - 1)}, \tag{4.38a}$$

$$d_{42} = \frac{-2c_3(2c_3 - 3) + c_2 - 3}{12b_4c_2(c_2 - c_3)(c_3 - 1)}, \tag{4.38b}$$

$$d_{43} = \frac{4c_3 - 3}{12b_4c_3(c_2 - c_3)}. \tag{4.38c}$$

Substituting (4.38a) and (4.38c) into the order condition (4.35l), using the definition of b_3 given in RK4.I, we obtain

$$\frac{1}{4} = b_4d_{43}d_{32}c_2 = \frac{4c_2 - 3}{12(2c_2 - 1)} \quad \Leftrightarrow \quad c_2 = 0,$$

contradictory to our assumption $c_2 > 0$. For the final case RK4.IV, we get the coefficients

$$d_{32} = \frac{1}{4}, \qquad d_{42} = 0, \qquad \text{and} \qquad d_{43} = \frac{1}{3b_4},$$

as solution of (4.36). Again, these coefficients lead to a contradiction with the order condition (4.35l), which reads

$$b_4d_{43}d_{32}c_2 = \frac{1}{12} \neq \frac{1}{4}.$$

Thus we can conclude, that there exists no fourth order SARK method with $s = 4$ stages.

SARK methods of fourth order

As discussion previously in this section, there exists no solution of the order conditions (4.29)–(4.32) for fourth order SARK methods with four stages. Thus we set $s = 5$ and assume

$$c_i = \sum_{j=1}^{i-1} d_{ij}, \tag{4.39}$$

similar to the consistency condition of Runge-Kutta methods (4.16). For these consistent methods the conditions (4.29)–(4.32) reduce to 14 conditions (4.40). Given c_i, $i = 1, \dots, 5$, we want to solve (4.40) for b_i, $i = 1, \dots, 5$, a_{ij}, $i, j = 1, \dots, 5$ and d_{ij}, $i, j = 1, \dots, 5$.

The conditions (4.29) and (4.30) for second order reduce to the standard Runge-Kutta conditions

$$\sum_{i=1}^{5} b_i = 1, \tag{4.40a}$$

$$\sum_{i=1}^{5} b_i c_i = \frac{1}{2}. \tag{4.40b}$$

The third order conditions (4.31) split into two standard Runge-Kutta conditions

$$\sum_{i=1}^{5} b_i c_i^2 = \frac{1}{3}, \tag{4.40c}$$

$$\sum_{i=1}^{5} \sum_{j<i} b_i a_{ij} c_j = \frac{1}{6}, \tag{4.40d}$$

and one additional condition

$$\sum_{i=1}^{5} \sum_{j<i} b_i d_{ij} c_j = \frac{1}{3}. \tag{4.40e}$$

Finally, the fourth order conditions (4.32) separate into the standard Runge-Kutta conditions

$$\sum_{i=1}^{5} b_i c_i^3 = \frac{1}{4}, \tag{4.40f}$$

$$\sum_{i=1}^{5} \sum_{j<i} b_i c_i a_{ij} c_j = \frac{1}{8}, \tag{4.40g}$$

$$\sum_{i=1}^{5} \sum_{j<i} b_i a_{ij} c_j^2 = \frac{1}{12}, \tag{4.40h}$$

$$\sum_{i=1}^{5} \sum_{j<i} \sum_{k<j} b_i a_{ij} a_{jk} c_k = \frac{1}{24}, \tag{4.40i}$$

and five additional SARK conditions

$$\sum_{i=1}^{5}\sum_{j<i} b_i c_i d_{ij} c_j = \frac{1}{4}, \tag{4.40j}$$

$$\sum_{i=1}^{5}\sum_{j<i} b_i d_{ij} c_j^2 = \frac{1}{4}, \tag{4.40k}$$

$$\sum_{i=1}^{5}\sum_{j<i}\sum_{k<j} b_i d_{ij} d_{jk} c_k = \frac{1}{4}, \tag{4.40l}$$

$$\sum_{i=1}^{5}\sum_{j<i}\sum_{k<j} b_i d_{ij} a_{jk} c_k = \frac{1}{8}, \tag{4.40m}$$

$$\sum_{i=1}^{5}\sum_{j<i}\sum_{k<j} b_i a_{ij} d_{jk} c_k = \frac{1}{12}. \tag{4.40n}$$

To obtain an explicit method, we set $c_1 = 0$ and solve the standard Runge-Kutta conditions (4.40a)–(4.40d) and (4.40f)–(4.40i) using a computer algebra software. There is set of well defined solution for distinct coefficients $0 \neq c_i$, $i = 2, \dots, 5$. In the following we give two examples of fourth order SARK methods based on two of these solutions.

For the first solution the coefficient c_2 has to satisfy $c_2 \neq \frac{1}{2}$. Further we have the freedom to choose $a_{43} \in \mathbb{R}$ and $b_5 \neq 0$. Substituting these Runge-Kutta coefficients into the remaining conditions (4.40e) and (4.40j)-(4.40n), there exists a solution for d_{ij}, $i, j = 1, \dots, 5$ when $c_5 \neq 1$. Thus we choose

$$c_2 = \frac{1}{3}, \qquad c_3 = \frac{2}{3}, \qquad c_4 = 1, \qquad c_5 = \frac{1}{2},$$

and set $b_5 = \frac{1}{2}$ to obtain positive weights b_i, $i = 1, \dots, 5$. Furthermore we set $a_{43} = d_{43} = 0$ to define the SARK(5) method of fourth order presented in Table 4.3.

The second solution of the Runge-Kutta conditions is well-defined for any $0 \neq c_i$, $i = 2, \dots, 5$ and we have the freedom to choose $a_{32}, a_{42} \in \mathbb{R}$ and $b_5 \neq 0$. Therefore we set

$$c_2 = \frac{1}{4}, \qquad c_3 = \frac{1}{2}, \qquad c_4 = \frac{3}{4}, \qquad c_5 = 1,$$

and $b_5 = \frac{1}{8}$, again leading to positive weights b_i, $i = 1, \dots, 5$. Setting $a_{32} = \frac{1}{4}$ and $a_{42} = 0$, we obtain a solution for all d_{ij}, $i, j = 1, \dots, 5$, except of d_{42}, which we set to 0. The coefficients of the second fourth order SARK(5) method are presented in Table 4.4.

4.2.4 Propagation operator of SARK methods

After the derivation of the order conditions in §4.2.1, we can specify the partial propagation operators of s-stage SARK methods defined in (4.18a) for linear problems.

0	0	0	0	0	0	0	0	0	0	0
$\frac{1}{3}$	$\frac{1}{3}$	0	0	0	0	$\frac{1}{3}$	0	0	0	0
$\frac{2}{3}$	$-\frac{1}{3}$	1	0	0	0	$-\frac{4}{3}$	2	0	0	0
1	$-\frac{2}{5}$	$\frac{7}{5}$	0	0	0	$-\frac{9}{5}$	$\frac{14}{5}$	0	0	0
$\frac{1}{2}$	$\frac{3}{8}$	$-\frac{1}{8}$	$\frac{1}{4}$	0	0	$\frac{11}{16}$	$-\frac{13}{16}$	$\frac{5}{16}$	$\frac{5}{16}$	0
	$\frac{5}{32}$	$\frac{3}{32}$	$\frac{3}{32}$	$\frac{5}{32}$	$\frac{1}{2}$					

Table 4.3: First example of a five-stage SARK method of fourth order.

0	0	0	0	0	0	0	0	0	0	0
$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	0	0	0	0
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0	$\frac{1}{2}$	0	0	0
$\frac{3}{4}$	$-\frac{5}{8}$	0	$\frac{11}{8}$	0	0	-2	0	$\frac{11}{4}$	0	0
1	$\frac{2}{3}$	$\frac{1}{6}$	$-\frac{1}{6}$	$\frac{1}{3}$	0	$\frac{17}{3}$	$-\frac{31}{3}$	5	$\frac{2}{3}$	0
	$\frac{1}{8}$	$\frac{1}{6}$	$\frac{5}{12}$	$\frac{1}{6}$	$\frac{1}{8}$					

Table 4.4: Second example of a five-stage SARK method of fourth order.

Propagation operator of second order SARK methods

For an arbitrary two-stage SARK method the only nonzero coefficients are b_1, b_2, a_{21}, d_{21}. In the kth step of Algorithm 3, given $\mathtt{Y}^{[k]} = \mathtt{M}_0^{[k]}\mathtt{u}^{[k]}$, we obtain

$$\begin{aligned}
\mathtt{Z}_1^{[k]} &= \mathtt{Y}^{[k]},\\
\mathtt{Z}_2^{[k]} &= \mathtt{Y}^{[k]} + \tau^{[k]} d_{21} \tilde{\mathtt{M}}_1^{[k]} \mathtt{Z}_1^{[k]} + \tau^{[k]} a_{21} \tilde{\mathtt{A}}^{[k]} \mathtt{Z}_1^{[k]}\\
&= \left(\mathtt{I} + \tau^{[k]} \left(d_{21}\tilde{\mathtt{M}}_1^{[k]} + a_{21}\tilde{\mathtt{A}}^{[k]}\right)\right) \mathtt{Y}^{[k]},
\end{aligned}$$

with the identity matrix $\mathtt{I} \in \mathbb{R}^{m\times m}$. The propagation from $\hat{t}_k$ to $\hat{t}_{k+1}$ reads

$$\begin{aligned}
\mathtt{Y}^{[k+1]} &= \mathtt{Y}^{[k]} + \tau^{[k]} \left(b_1 \tilde{\mathtt{A}}^{[k]} \mathtt{Z}_1^{[k]} + b_2 \tilde{\mathtt{A}}^{[k]} \mathtt{Z}_2^{[k]}\right)\\
&= \left(\mathtt{I} + \tau^{[k]}(b_1+b_2)\tilde{\mathtt{A}}^{[k]} + (\tau^{[k]})^2 \tilde{\mathtt{A}}^{[k]} \left(b_2 d_{21} \tilde{\mathtt{M}}_1^{[k]} + b_2 a_{21} \tilde{\mathtt{A}}^{[k]}\right)\right) \mathtt{Y}^{[k]}\\
&= \left(\mathtt{I} + \tau^{[k]} \tilde{\mathtt{A}}^{[k]} + \tfrac{1}{2}(\tau^{[k]})^2 \tilde{\mathtt{A}}^{[k]} \left(\tilde{\mathtt{M}}_1^{[k]} + \tilde{\mathtt{A}}^{[k]}\right)\right) \mathtt{Y}^{[k]},
\end{aligned}$$

where we used the order conditions (4.29) and (4.30) for second order methods. This results in the partial propagation matrix

$$\mathtt{T}_{r,2}^{[k+1]} = \mathtt{I} + \tau^{[k]} \tilde{\mathtt{A}}^{[k]} + \tfrac{1}{2}(\tau^{[k]})^2 \tilde{\mathtt{A}}^{[k]} \left(\tilde{\mathtt{M}}_1^{[k]} + \tilde{\mathtt{A}}^{[k]}\right), \tag{4.41}$$

such that $\mathtt{Y}^{[k+1]} = \mathtt{T}_{r,2}^{[k+1]} \mathtt{Y}^{[k]}$.

Propagation operator of third order SARK methods

A similar calculation for three-stage SARK methods, using the order conditions (4.29)-(4.31), leads to the partial propagation matrix

$$\begin{aligned}\mathtt{T}_{r,3}^{[k+1]} = \mathtt{I} + \tau^{[k]}\tilde{\mathtt{A}}^{[k]} + \tfrac{1}{2}(\tau^{[k]})^2\tilde{\mathtt{A}}^{[k]}\left(\tilde{\mathtt{M}}_1^{[k]} + \tilde{\mathtt{A}}^{[k]}\right) \\ + \tfrac{1}{6}(\tau^{[k]})^3\tilde{\mathtt{A}}^{[k]}\left(2\tilde{\mathtt{M}}_1^{[k]} + \tilde{\mathtt{A}}^{[k]}\right)\left(\tilde{\mathtt{M}}_1^{[k]} + \tilde{\mathtt{A}}^{[k]}\right).\end{aligned} \tag{4.42}$$

Propagation operator of fourth order SARK methods

Extending this to fourth order using the conditions (4.29)-(4.32), the partial propagation matrix yields

$$\begin{aligned}\mathtt{T}_{r,4}^{[k+1]} = \mathtt{I} + \tau^{[k]}\tilde{\mathtt{A}}^{[k]} + \tfrac{1}{2}(\tau^{[k]})^2\tilde{\mathtt{A}}^{[k]}\left(\tilde{\mathtt{M}}_1^{[k]} + \tilde{\mathtt{A}}^{[k]}\right) \\ + \tfrac{1}{6}(\tau^{[k]})^3\tilde{\mathtt{A}}^{[k]}\left(2\tilde{\mathtt{M}}_1^{[k]} + \tilde{\mathtt{A}}^{[k]}\right)\left(\tilde{\mathtt{M}}_1^{[k]} + \tilde{\mathtt{A}}^{[k]}\right) \\ + \tfrac{1}{24}(\tau^{[k]})^4\tilde{\mathtt{A}}^{[k]}\left(3\tilde{\mathtt{M}}_1^{[k]} + \tilde{\mathtt{A}}^{[k]}\right)\left(2\tilde{\mathtt{M}}_1^{[k]} + \tilde{\mathtt{A}}^{[k]}\right)\left(\tilde{\mathtt{M}}_1^{[k]} + \tilde{\mathtt{A}}^{[k]}\right).\end{aligned} \tag{4.43}$$

4.3 Numerical examples - convergence rates

In this section we apply these structure aware time stepping methods to linear and nonlinear problems to investigate to convergence rates. First, we reconsider the example of the wave equation discussed in §3.4.1 and §3.4.2 and apply an appropriate SAT time stepping schemes. Then, we construct a nonlinear model problem and apply SARK time stepping methods to observe the high order rates also for nonlinear problems.

4.3.1 SAT time stepping

Recall the example of the two-dimensional standing wave (3.36), where we observed a reduced convergence order in §3.4.1. We use the same tent mesh, generated by the edge-based algorithm in §3.2.1 with $c_e = 2$ and $C_\tau = \frac{1}{2}$, resulting in a maximal slope $\| \nabla_x \varphi\| \approx 0.494$. For the spatial discretization, we apply the same discontinuous Galerkin method using polynomials of degree p, with $1 \leq p \leq 4$. On each cylinder we perform a $(p+1)$-stage SAT time stepping, described in §4.1, with $r = 2p$ subintervals. The spatial errors at the final time are measured in the norm e_h defined in (3.38) and they are reported in Figure 4.2. We observe that the error goes to zero at the optimal rate of $\mathcal{O}(h^{p+1})$ until we are close to machine precision.

Compared to the implicit time stepping in §3.4.2, where we observed the suboptimal rate $\mathcal{O}(h^p)$, the explicit $(p+1)$-stage SAT time stepping yields to the optimal rate $\mathcal{O}(h^{p+1})$. Furthermore, the explicit time stepping uses less than 9GB of memory at the refinement step $l = 8$, while the implicit method would exceed the available 320GB.

Next, we consider the three-dimensional standing wave, with the exact solution given by (3.42). The domain $\Omega_0 = [0, \pi]^3$ is spatially meshed with tetrahedral elements of size $h = 2^{-l+1}$ for $l = 0, \ldots, 5$ and the final time $t_{\max}$ is set to $t_{\max} = \sqrt{3}\pi$. The tents in the

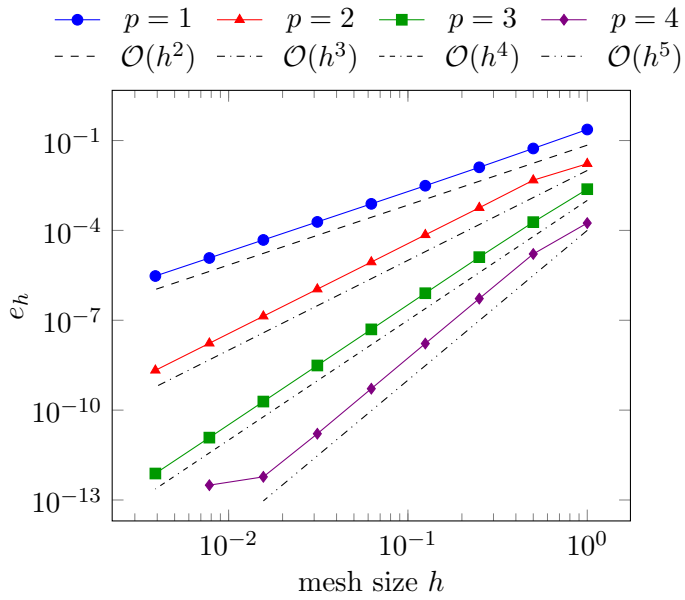


Figure 4.2: Spatial L_2-error e_h of all field components for a two-dimensional standing wave over the mesh size h using a $(p+1)$-stage SAT time stepping.

time slab of size $t_{\max} \cdot 2^{-l}/8$ are generated using the algorithm in §3.2.1 with wave speed $c_e = 2$ and $C_\tau = \frac{1}{3}$, leading to maximal slope $\| \nabla_x \varphi \| \approx 0.53$. Again, we observe that the spatial error e_h at $t_{\max}$ goes to zero at the optimal rate of $\mathcal{O}(h^{p+1})$, as reported in Figure 4.3.

Using the implicit time stepping in §3.4.2, we were limited to refinement level $l = 3$ for spatial polynomial degree $p = 3$. A comparable 4-stage SAT time stepping with $p = 3$ uses less than 1GB memory for the refinement level $l = 4$, while the implicit method would exceed the available 320GB.

4.3.2 SARK time stepping

We consider the example of the one-dimensional Burgers' equation

$$\partial_t u(x,t) + \partial_x u(x,t)^2 = 0, \quad \forall (x,t) \in [0,1] \times (0, t_{\max}], \tag{4.44a}$$

with initial values set by

$$u(x,0) = \exp\big(-50\big(x - \tfrac{1}{2}\big)^2\big), \quad \forall x \in [0,1], \tag{4.44b}$$

an inflow boundary condition an $x = 0$, and an outflow boundary condition at $x = 1$. The semi-discretized ODE system (3.30) is obtained using a discontinuous Galerkin method with polynomial degree p in (3.28). Further we use the upwind flux

$$f_n(u^+, u^-) = \frac{1}{2} \begin{cases} (u^-)^2, & \{u\} \cdot n \geq 0, \\ (u^+)^2, & \{u\} \cdot n < 0, \end{cases}$$

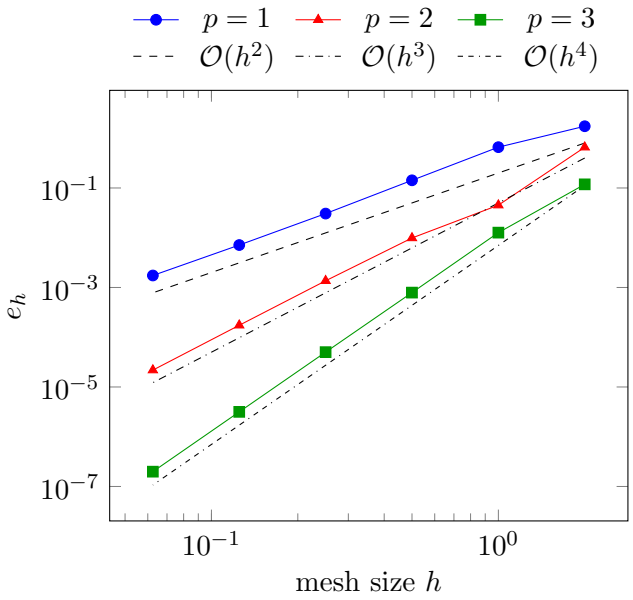


Figure 4.3: Spatial L_2-error e_h of all field components for a three-dimensional standing wave over the mesh size h using a $(p+1)$-stage SAT time stepping.

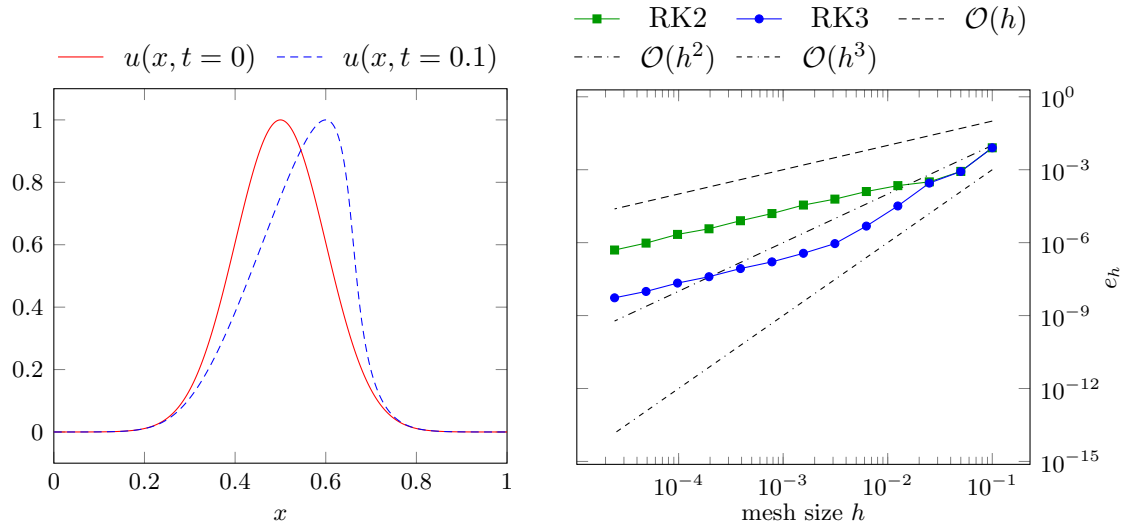
where $\{u\} = \frac{1}{2}(u^- + u^+)$. Let $u_h(x)$ be the numerical solution of (4.44) at $t = t_{\max}$. The final time $t_{\max} = 0.1$ is chosen such that the exact solution is still a smooth function (before the onset of shock). Therefore no regularization or limiting is expected to be essential to witness high order convergence. The exact solution at $t_{\max}$ shown in Figure 4.4a is obtained by the method of characteristics together with a Newton method. Thus one would expect the error

$$e_h := \|u(\cdot, 0.1) - u_h\|_{L^2([0,1])} \tag{4.44c}$$

to go to zero at a rate of $\mathcal{O}(h^{p+1})$.

Again, the standard reformulation to the variable $\mathtt{Y}(\hat{t}) = \mathtt{M}(\hat{t}, \mathtt{u}(\hat{t}))$, as discussed in §3.4, combined with standard time stepping methods leads to first order convergence. Figure 4.4b reports the rates we observed when two standard time stepping schemes were used to solve the reformulated ODE obtained with a polynomial degree $p = 2$, namely the two-stage Ralston (RK2) and the three-stage Heun (RK3) time stepping schemes. The Butcher tableaus can be found in Tables 4.1b and 4.2b. The tents were built such that (3.3) is satisfied with $\bar{c} = 2$ and based on spatial meshes with mesh size $h = 2^{-i}/10$ for $i = 0 \dots 12$. Although we see third order convergence for the first few refinement steps, the rate eventually drops to first order for both methods.

Now we show that SARK methods do not suffer from the previously described convergence order reduction. The tents in the slab of size 0.1 are generated with the same wave speed $\bar{c} = 2$ as before. We apply Algorithm 3 with SARK schemes of $s = 2$ and $s = 3$ stages using the same spatial mesh size $h = 2^{-i}/10$ for $i = 0 \dots 12$ and $r = 4$ and $r = 10$ substeps within each tent for $s = 2$ and $s = 3$ respectively. The values of e_h are plotted in Figure 4.5



(a) Initial values (4.44b) (solid) and exact solution of (4.44) at $t_{\max} = 0.1$ (dashed).

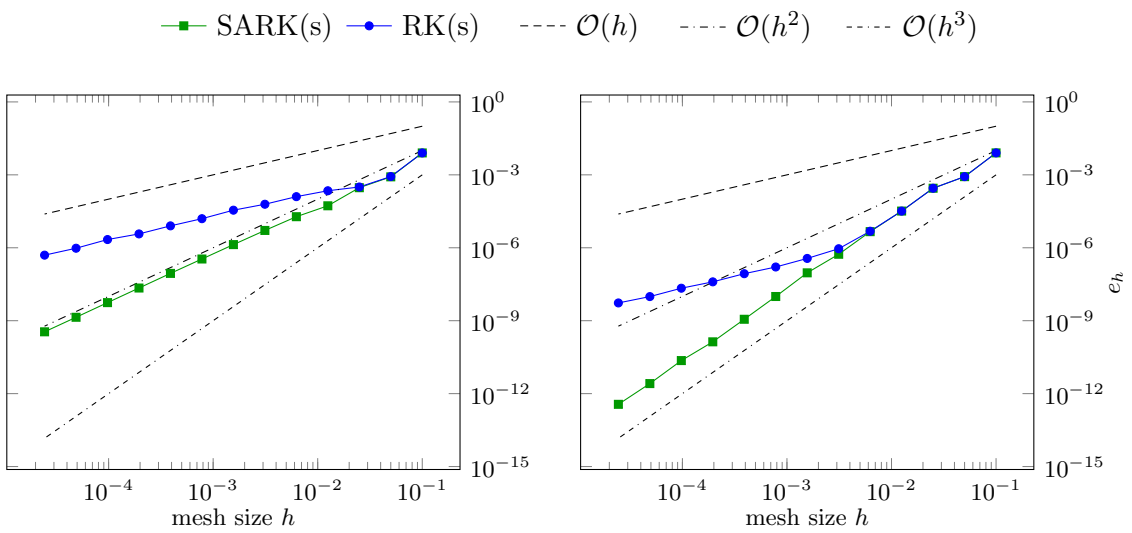
(b) Convergence rates of the spatial error at $t_{\max}$ for Ralston's method (RK2) and Heun's method (RK3).

Figure 4.4: Exact solution u and convergence rates of the error e_h, defined in (4.44c), for the example of the Burgers' equation described in (4.44).

for both methods. As h decreases, in Figure 4.5a we eventually see quadratic convergence for the two-stage SARK method (although the convergence rate seems to be slightly higher in a preasymptotic regime), while the rate of the underlying standard Runge-Kutta method drops to first order. The three-stage SARK method in Figure 4.5b shows cubic convergence while the rate of the underlying standard Runge-Kutta method drops to first order again.

To observe fourth order convergence, we have to use polynomials of degree $p = 3$ for the spatial discretization and a SARK(5) method given by the coefficients in Table 4.3 or Table 4.3, with $r = 12$ substeps. The tents were built with $\bar{c} = 4$, based on a spatial mesh with mesh $h = 2^{-i}/10$ for $i = 0 \dots 10$. The values of e_h obtained by the SARK(5) and the underlying RK(5) method are shown in Figure 4.6. As expected, we see the convergence order reduction for the Runge-Kutta method, while the both SARK methods converges at fourth order until we are close to machine precision.

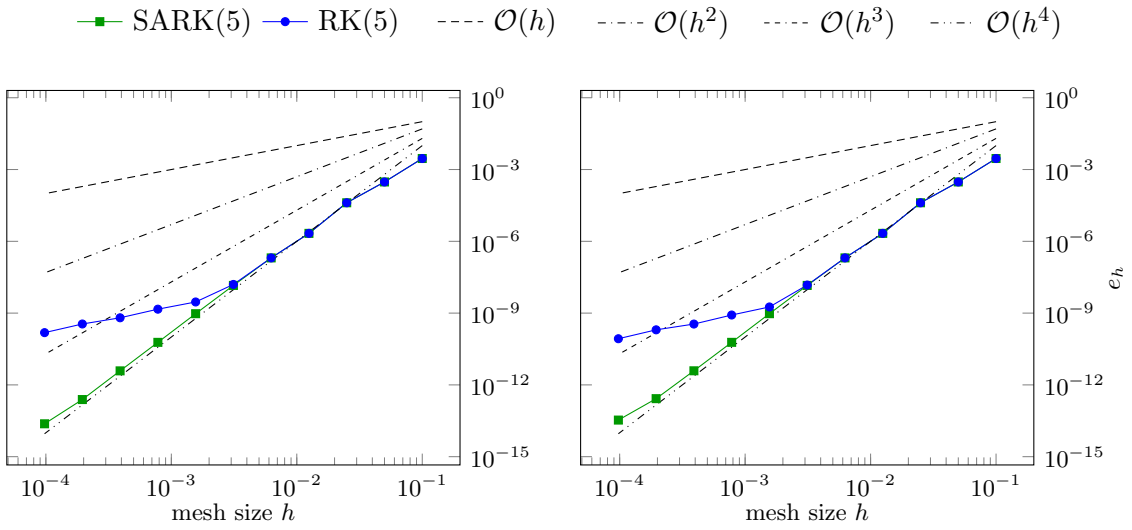
These plots clearly show the benefit of SARK schemes over the corresponding standard Runge-Kutta schemes.



(a) Convergence rates obtained from SARK(2, Ralston) method (see Table 4.1b) and the standard Ralston method.

(b) Convergence rates obtained from SARK(3, Heun) method (see Table 4.2b) and the standard Heun scheme.

Figure 4.5: Plots of the error e_h defined in (4.44c) for SARK and Runge-Kutta (RK) methods applied to the Burgers' example described in (4.44).



(a) SARK(5) method given in Table 4.3.

(b) SARK(5) method given in Table 4.4.

Figure 4.6: Plots of the error e_h defined in (4.44c) for SARK(5) methods and the underlying RK(5) methods applied to the Burgers' example described in (4.44).

5 Investigation of discrete stability

This chapter is devoted to remarks on the stability of the new structure aware time stepping schemes. While it is common to study stability of ODE solvers by applying them to a simple scalar ODE, keeping our application of spatially varying hyperbolic solutions in mind, we consider changes in an energy-like measure on the solution $\mathtt{u}(\hat{t})$. Recall that $\mathtt{u}(\hat{t}) \in \mathbb{R}^m$ is the coefficient vector of the basis expansion of the mapped finite element solution $\hat{u}(x,\hat{t}) \in V_h^{\mathtt{v}}$, as defined by (3.27). We limit ourselves to the case where the energy-like quantity

$$\|\hat{u}\|_{M(\hat{t})}^2 := \int_{\omega_{\mathtt{v}}} M(\hat{t},\hat{u}(x,\hat{t})) \cdot \hat{u}(x,\hat{t}) = \int_{\omega_{\mathtt{v}}} \big(M_0(\hat{u}) - \hat{t}M_1(\hat{u})\big) \cdot \hat{u} \tag{5.1}$$

is a *norm* and (the generally nonlinear operators) M, M_0 and M_1 defined in (3.24) are linear, so that we may rewrite $M(\hat{t},\hat{u}) = M(\hat{t})\hat{u}$ using the linear operator $M(\hat{t}) = M_0 - \hat{t}M_1 : V_h^{\mathtt{v}} \to V_h^{\mathtt{v}}$. For linear hyperbolic systems, the causality condition (3.3) implies that $M(\hat{t})$ is identifiable with a symmetric positive definite matrix $\mathtt{M}(\hat{t})$ – see (3.49) – so that (5.1) indeed defines a norm. For the coefficients $\mathtt{u}(\hat{t})$ of the basis expansion of $\hat{u}(\cdot,\hat{t})$ holds

$$\|\hat{u}\|_{M(\hat{t})}^2 = \|\mathtt{u}(\hat{t})\|_{\mathtt{M}(\hat{t})}^2 = \mathtt{u}(\hat{t})^{\intercal}\mathtt{M}(\hat{t})\mathtt{u}(\hat{t}). \tag{5.2}$$

In the special case of $g(v) = v$, e.g. that on flat advancing fronts, where $\varphi(x,\hat{t})$ is independent of x for some fixed $\hat{t}$, (5.1) reduces to

$$\|\hat{u}\|_{M(\hat{t})}^2 = \int_{\omega_{\mathtt{v}}} \hat{u} \cdot \hat{u},$$

so $\|\hat{u}\|_{M(\hat{t})}$ becomes the familiar spatial L^2 norm of $\hat{u}(\cdot,\hat{t})$.

5.1 Our procedure to study linear stability

Stability of the scheme within a tent can be understood by studying the discrete analogue of the ratio $\|\hat{u}(\cdot,\hat{t})\|_{M(1)}/\|\hat{u}(\cdot,\hat{t})\|_{M(0)}$ for all possible initial data $\hat{u}(0)$. This amounts to studying the norm of the discrete propagation operator for $\mathtt{u}$, which we proceed to formulate. First, recall the connection between $\mathtt{u}$ and $\mathtt{Y}$, namely $\mathtt{Y}(\hat{t}) = \mathtt{M}(\hat{t})\mathtt{u}(\hat{t})$ – see (3.33). Algorithm 3, takes as input an approximation $\mathtt{u}_0$ to $\hat{u}(0)$ at the tent bottom and outputs $\mathtt{Y}_1^{r,s}$, an approximation to $\mathtt{Y}(1)$ at the tent top. Hence the associated approximation to $\hat{u}(1)$ is

$$\mathtt{u}_1^{r,s} := \mathtt{M}(1)^{-1}\mathtt{Y}_1^{r,s}.$$

Next, recall the discrete propagation operator defined by (4.18). It is now a linear operator that maps $\mathtt{Y}_0 = \mathtt{M}(0)\mathtt{u}_0$ to $\mathtt{Y}_1^{r,s}$ according to (4.19). Define the *tent propagation matrix* $\mathtt{S}_{r,s} : \mathbb{R}^m \to \mathbb{R}^m$ by

$$\mathtt{S}_{r,s} = \mathtt{M}(1)^{-1}\mathtt{T}_{r,s}\,\mathtt{M}(0). \tag{5.3}$$

Clearly, (4.19) implies that

$$\mathtt{u}_1^{r,s} = \mathtt{S}_{r,s}\,\mathtt{u}_0. \tag{5.4}$$

The discrete analogue of $\|\hat{u}(\cdot,\hat{t})\|_{M(1)}/\|\hat{u}(\cdot,0)\|_{M(0)}$ is $\|\mathtt{u}_1^{r,s}\|_{\mathtt{M}(1)}/\|\mathtt{u}_0\|_{\mathtt{M}(0)}$ which can be bounded using the following norm of $\mathtt{S}_{r,s}$:

$$\|\mathtt{S}_{r,s}\|_{L(\mathtt{M}(0),\mathtt{M}(1))} = \sup_{0\neq\mathtt{w}\in\mathbb{R}^m} \frac{\|\mathtt{S}_{r,s}\mathtt{w}\|_{\mathtt{M}(1)}}{\|\mathtt{w}\|_{\mathtt{M}(0)}}. \tag{5.5}$$

It is immediate from (5.4) that $\|\mathtt{u}_1^{r,s}\|_{\mathtt{M}(1)} \le \|\mathtt{S}_{r,s}\|_{L(\mathtt{M}(0),\mathtt{M}(1))}\|\mathtt{u}_0\|_{\mathtt{M}(0)}$. Thus the study of stability of these structure aware schemes is reduced to computing estimates for the norm of $\mathtt{S}_{r,s}$. When being applied to linear problems, the partial propagation matrices $\mathtt{T}_{r,s}^{[k+1]}$ of the structure aware time stepping methods are identical. This becomes clear when comparing $\mathtt{T}_{r,2}^{[k+1]}$ in (4.9) and (4.41) for $s=2$, $\mathtt{T}_{r,3}^{[k+1]}$ in (4.10) and (4.42) for $s=3$ and $\mathtt{T}_{r,4}^{[k+1]}$ in (4.11) and (4.43) for $s=4$. In §5.2, we prove an upper bound for the discrete propagation matrix $\mathtt{S}_{r,2}$ of two-stage structure aware schemes in the norm defined in (5.5).

Next, we describe how we compute the norm of $\mathtt{S}_{r,s}$ for some examples in §5.3. With the $m\times m$ symmetric positive definite matrix $\mathtt{M}(\hat{t})$ holds

$$\begin{aligned}
\|\mathtt{S}_{r,s}\|^2_{L(\mathtt{M}(0),\mathtt{M}(1))} &= \sup_{0\neq\mathtt{w}\in\mathbb{R}^m} \frac{(\mathtt{S}_{r,s}\mathtt{w})^{\mathsf{T}}\mathtt{M}(1)(\mathtt{S}_{r,s}\mathtt{w})}{\mathtt{w}^{\mathsf{T}}\mathtt{M}(0)\mathtt{w}} \\
&= \sup_{0\neq\mathtt{w}\in\mathbb{R}^m} \frac{\mathtt{w}^{\mathsf{T}}(\mathtt{S}_{r,s}^{\mathsf{T}}\mathtt{M}(1)\mathtt{S}_{r,s})\mathtt{w}}{\mathtt{w}^{\mathsf{T}}\mathtt{M}(0)\mathtt{w}} \\
&= \sup\{|\lambda| :\ \exists 0\neq\mathtt{X}\in\mathbb{R}^m \text{ satisfying } (\mathtt{S}_{r,s}^{\mathsf{T}}\mathtt{M}(1)\mathtt{S}_{r,s})\mathtt{X} = \lambda\mathtt{M}(0)\mathtt{X}\}.
\end{aligned}$$

Thus, to investigate the stability of a scheme, we compute $\mathtt{T}_{r,s}^{[k+1]}$ by (4.18b), then $\mathtt{T}_{r,s}$ by (4.18b), followed by $\mathtt{S}_{r,s}$ per (5.3), and finally, the square root of the spectral radius of $\mathtt{M}(0)^{-1}(\mathtt{S}_{r,s}^{\mathsf{T}}\mathtt{M}(1)\mathtt{S}_{r,s})$, which equals $\|\mathtt{S}_{r,s}\|^2_{L(\mathtt{M}(0),\mathtt{M}(1))}$ as shown above. Numerical estimates for $\|\mathtt{S}_{r,s}\|_{L(\mathtt{M}(0),\mathtt{M}(1))}$ are reported for an example at the end of this chapter.

5.2 Discrete stability of two-stage structure aware time stepping schemes

Suppose that $A^{(i)}:\Omega_0\to\mathbb{R}^{L\times L}$, $i=1,\dots,N+1$, are symmetric matrix-valued functions, where $A^{(t)}\equiv A^{(N+1)}$ is symmetric positive definite. Following the notation introduced in §2.1, a general hyperbolic system of L equations in L unknowns $u\in\mathbb{R}^L$ takes the form

$$\partial_t\big(A^{(t)}u\big) + \sum_{i=1}^{N}\partial_i\big(A^{(i)}u\big) = 0. \tag{2.5}$$

We impose boundary conditions of the following form studied by Friedrichs [9],

$$(\mathcal{D}^n - \mathcal{B})u = 0 \qquad \text{on } \partial\Omega_0\times(0,t_{\max}), \tag{5.6}$$

where

$$\mathcal{D}^n = \sum_{j=1}^{N} n_j A^{(j)}. \tag{5.7}$$

Here n denotes the unit outward normal on $\partial\Omega_0$, and later it will be used to generically denote the outward unit normal of other domain boundaries. The operator $\mathcal{B} : \partial\Omega_0 \to \mathbb{R}^{L\times L}$ must satisfy certain well-studied conditions for the boundary condition to be admissible. For our purposes, we recall one of these conditions,

$$\mathcal{B} + \mathcal{B}^\intercal \geq 0 \qquad \text{on } \partial\Omega_0 \times (0, t_{\max}), \tag{5.8}$$

which we shall use. Inequality (5.8) means that at every point $x \in \partial\Omega_0$, the matrix $\mathcal{B}(x)$ satisfies $(\mathcal{B} + \mathcal{B}^\intercal)y \cdot y = 2\mathcal{B}(x)y \cdot y \geq 0$ for all vectors $y \in \mathbb{R}^L$. Of course, the system must also be supplemented with an initial condition,

$$u(x, 0) = u_0(x) \qquad \forall\ x \in \Omega_0, \tag{5.9}$$

at time $t = 0$ for some given initial data u_0. Thus we have restricted the model problem to the simple case of a linear hyperbolic system driven solely by the nonzero initial data u_0. To avoid some technicalities, we assume

$$\sum_{j=1}^{N} \partial_j A^{(j)} = 0 \tag{5.10}$$

in the sense of distributions, i.e., jumps in $A^{(j)}(x)$ are allowed so long as (5.10) holds.

The discontinuous Galerkin discretization of the mapped equation for $\hat{u} = u \circ \Phi$,

$$\partial_{\hat{t}}\Big[\Big(A^{(t)} - \sum_{i=1}^{N} A^{(i)}\partial_i\varphi\Big)\hat{u}\Big] + \sum_{i=1}^{N} \partial_i\big(\delta A^{(i)}\hat{u}\big) = 0, \tag{5.11}$$

discussed in §3.3, can be seen as simplified version of the framework presented in [7], which we discuss next. Recall, that $\mathcal{T}_\mathtt{v}$ denotes the collection of elements in the vertex patch $\omega_\mathtt{v}$ of a mesh vertex $\mathtt{v}$. Let $V_h^\mathtt{v}$ denote the restriction of the spatial discontinuous Galerkin space on $\omega_\mathtt{v}$ and let $\psi_j(x)$ denote a basis for $V_h^\mathtt{v}$. The semi-discrete approximation of $\hat{u}$ is of the form

$$\hat{u}_h(x, \hat{t}) = \sum_j \mathtt{u}_j(\hat{t})\psi_j(x).$$

We consider a discontinuous Galerkin semi-discretization of (5.11) of the following form:

$$\int_{\omega_\mathtt{v}} \partial_{\hat{t}}\Big[\Big(A^{(t)} - \sum_{i=1}^{N} A^{(i)}\partial_i\varphi\Big)\hat{u}_h\Big] \cdot v_h = \sum_{T\in\mathcal{T}_\mathtt{v}} \int_T \sum_{i=1}^{N} \big(\delta A^{(i)}\hat{u}_h\big) \cdot \partial_i v_h - \int_{\partial T} \delta f_n^{\hat{u}_h} \cdot v_h, \tag{5.12}$$

for all $v_h \in V_h^\mathtt{v}$, where the numerical flux $f_n^{\hat{u}_h}$ on an element boundary ∂T is defined using the values of $\hat{u}_h$ from the element T as well as from the neighbouring element T_o, as follows. For any $w \in V_h^\mathtt{v}$, at a point $x \in \partial T \cap \partial T_o$, letting $w_o = w|_{T_o}$, define

$$\{w\}(x) = \frac{1}{2}(w + w_o), \qquad [\![w]\!](x) = w - w_o.$$

Then, f_n^w is assumed to take the form

$$f_n^w = \begin{cases} \mathcal{D}^n\{w\} + \mathcal{S}[\![w]\!] & \text{on } \partial T \setminus \partial\Omega_0, \\ \frac{1}{2}(\mathcal{D}^n + \mathcal{B})w & \text{on } \partial T \cap \partial\Omega_0. \end{cases} \tag{5.13}$$

Here, $\mathcal{D}^n$ is defined by (5.7) with the n now denoting the outward unit normal on ∂T, and

$$\mathcal{S} : \bigcup_{T \in \mathcal{T}_\mathrm{v}} \partial T \to \mathbb{R}^{L \times L}$$

is a uniformly bounded stabilization matrix of the particular discontinuous Galerkin discretization under consideration. We assume that S is single-valued on facets shared by two elements and that

$$\mathcal{S} + \mathcal{S}^\intercal \geq 0. \tag{5.14}$$

The following examples show a variety of equations and their well-known discretizations that conform to this framework.

Advection equation

The advection equation $\partial_t u + \operatorname{div}_x(\beta u) = 0$ with some vector field $\beta : \Omega_0 \to \mathbb{R}^N$ fits the above setting with $L = 1$ (and arbitrary spatial dimension N), $A^{(j)} = [\beta_j] \in \mathbb{R}^{1\times 1}$, $A^{(t)} = [1]$ and $\mathcal{D}^n = \beta \cdot n$. A boundary condition $u = 0$ on the inflow boundary $\partial_\mathrm{in}\Omega_0 = \{x \in \partial\Omega_0 : n \cdot \beta(x) < 0\}$ can be represented in the form (5.6) using $\mathcal{B} = |\beta \cdot n|$. With this choice, (5.6) trivially holds on the outflow boundary, while at the inflow boundary points, it takes the form $(\mathcal{D}^n - \mathcal{B})u = 2(\beta \cdot n)u = 0$. Finally, if we choose

$$\mathcal{S} = \frac{1}{2}|\beta \cdot n|$$

in this example, the scheme (5.12) at once reduces to the well-known upwind discontinuous Galerkin scheme. Obviously, the condition (5.14) holds for this choice of $\mathcal{S}$.

Wave equation

The wave equation $\partial_{tt}\phi - \Delta\phi = 0$ in N space dimensions can be written as a first order hyperbolic system for $L = N + 1$ variables comprised of the components of the flux $q = -\nabla_x \phi$ and $\mu = \partial_t \phi$. To match the prior abstract setting, we set $A^{(t)}$ to identity, and

$$A^{(j)} = \begin{bmatrix} 0 & e_j \\ e_j^\intercal & 0 \end{bmatrix},$$

using the standard unit basis vectors e_j of $\mathbb{R}^N$. Thus we get

$$\mathcal{D}^n = \sum_{j=1}^{N} n_j A^{(j)} = \begin{bmatrix} 0 & n \\ n^\intercal & 0 \end{bmatrix}.$$

Dirichlet boundary conditions can be imposed using

$$\mathcal{B} = \begin{bmatrix} 0 & -n \\ n^\intercal & 2 \end{bmatrix}.$$

For this choice of $\mathcal{B}$, we clearly have (5.8). Moreover,

$$(\mathcal{D}^n - \mathcal{B})u = 2 \begin{bmatrix} n \\ -1 \end{bmatrix} \mu$$

on the boundary, thus allowing us to impose Dirichlet conditions on the variable $u_{N+1} = \mu = \partial_t \phi$. We set $\mathcal{S}$ by

$$\mathcal{S} = \frac{1}{2} \begin{bmatrix} n\, n^\intercal & 0 \\ 0 & 1 \end{bmatrix},$$

for which the condition (5.14) obviously holds. The resulting scheme is the discontinuous Galerkin scheme with upwind fluxes for the wave equation. Clearly, the definition of $\mathcal{S}$ does not depend on the choice of the orientation of the normal vector n, i.e., it is single-valued on element interfaces.

Maxwell equations

The Maxwell system for the electric field $E(x,t)$ and magnetic field $H(x,t)$ consists of

$$\begin{aligned} \partial_t(\varepsilon E) - \operatorname{curl} H &= 0, \\ \partial_t(\mu H) + \operatorname{curl} E &= 0, \end{aligned}$$

with the permittivity $\varepsilon(x)$ and permeability $\mu(x)$. To fit this into the prior setting, we choose $N = 3$ and $L = 6$ and write the unknowns in the block form $u = [E, H]^\intercal$. As discussed in §2.1, we set

$$A^{(j)} = \begin{bmatrix} 0 & [\epsilon^j] \\ [\epsilon^j]^\intercal & 0 \end{bmatrix} \quad \text{and} \quad A^{(t)} = \begin{bmatrix} \varepsilon I & 0 \\ 0 & \mu I \end{bmatrix},$$

where ϵ^j is the 3×3 matrix whose (l,m)th entry equals the value of the Levi-Civita alternator ϵ_{jlm}. With these settings and $\mathcal{N} = \sum_{j=1}^3 n_j \epsilon^j \in \mathbb{R}^{3\times 3}$, we get

$$\mathcal{D}^n = \sum_{j=1}^N n_j A^{(j)} = \begin{bmatrix} 0 & \mathcal{N} \\ \mathcal{N}^\intercal & 0 \end{bmatrix}. \tag{5.15}$$

It is easy to see that $\mathcal{N} H = H \times n$ and there holds

$$\mathcal{D}^n \begin{bmatrix} E \\ H \end{bmatrix} = \begin{bmatrix} H \times n \\ -E \times n \end{bmatrix}.$$

Next, we set

$$\mathcal{B} \begin{bmatrix} E \\ H \end{bmatrix} = \begin{bmatrix} 2\mathcal{N}^\intercal \mathcal{N} & \mathcal{N} \\ \mathcal{N} & 0 \end{bmatrix} \begin{bmatrix} E \\ H \end{bmatrix} = \begin{bmatrix} 2E_t + H \times n \\ E \times n \end{bmatrix}, \tag{5.16}$$

with the tangential component $E_t = n \times (E \times n)$ and there holds

$$(\mathcal{D}^n - \mathcal{B}) \begin{bmatrix} E \\ H \end{bmatrix} = \begin{bmatrix} 2E_t \\ -2(E \times n) \end{bmatrix}.$$

Therefore equation (5.6) imposes the Dirichlet boundary condition corresponding to electrical isolation by a perfect electric conductor. Also note that since $\mathcal{N}^\intercal = -\mathcal{N}$, we have

$$\mathcal{B}^\intercal + \mathcal{B} = \begin{bmatrix} 4\mathcal{N}^\intercal\mathcal{N} & 0 \\ 0 & 0 \end{bmatrix},$$

so (5.8) holds. For the discontinuous Galerkin discretization, we set

$$\mathcal{S} = \frac{1}{2} \begin{bmatrix} \mathcal{N}^\intercal\mathcal{N} & 0 \\ 0 & \mathcal{N}^\intercal\mathcal{N} \end{bmatrix}.$$

Clearly, $\mathcal{S} + \mathcal{S}^\intercal = 2\mathcal{S}$. Furthermore, $E^\intercal\mathcal{N}^\intercal\mathcal{N}E = \|\mathcal{N}E\|_2^2 \geq 0$, which implies $\mathcal{S} + \mathcal{S}^\intercal \geq 0$. Note that the sign of the normal vector n is immaterial while computing the entries of $\mathcal{S}$, i.e., $\mathcal{S}$ is single-valued on element interfaces, as required by our general setting. This choice of $\mathcal{S}$ leads to the upwind flux used in [11, 21].

Turning back to general linear hyperbolic system, we define $A : V_h^{\mathrm{v}} \times V_h^{\mathrm{v}} \to \mathbb{R}$, $M_0 : V_h^{\mathrm{v}} \to V_h^{\mathrm{v}}$, and $M_1 : V_h^{\mathrm{v}} \to V_h^{\mathrm{v}}$ by

$$A(w, v) = \sum_{T \in \mathcal{T}_{\mathrm{v}}} \int_T \sum_{i=1}^N \left(\delta A^{(i)} w\right) \cdot \partial_i v - \int_{\partial T} \delta f_n^w \cdot v, \tag{5.17}$$

$$\int_{\omega_{\mathrm{v}}} M_0 w \cdot v = \int_{\omega_{\mathrm{v}}} \Big(A^{(t)} - \sum_{i=1}^N A^{(i)} \partial_i \varphi_b\Big) w \cdot v, \tag{5.18}$$

$$\int_{\omega_{\mathrm{v}}} M_1 w \cdot v = \int_{\omega_{\mathrm{v}}} \Big(\sum_{i=1}^N A^{(i)} \partial_i \delta\Big) w \cdot v, \tag{5.19}$$

for all $w, v \in V_h^{\mathrm{v}}$. Using these definitions, the semi-discrete problem (5.12) reads

$$\int_{\omega_{\mathrm{v}}} \partial_{\hat{t}} \Big[\Big(M_0 - \hat{t} M_1\Big) \hat{u}_h\Big] \cdot v_h = A(\hat{u}_h, v_h) \tag{5.20}$$

for all $v_h \in V_h^{\mathrm{v}}$. Now, let

$$d(w, v) = -\left[2\, A(w, v) + \int_{\omega_{\mathrm{v}}} M_1 w \cdot v\right] \tag{5.21}$$

for $w, v \in V_h^{\mathrm{v}}$. Recall that $\mathcal{F}_{\mathrm{v}}$ denotes the set of facets F, i.e., $(N-1)$-subsimplices, of the simplicial mesh $\mathcal{T}_{\mathrm{v}}$ of the vertex patch ω_{v}. This set is partitioned into facets on the boundary $\partial\omega_{\mathrm{v}}$ of the vertex patch denoted by ${\mathcal{F}_{\mathrm{v}}}^b$, and the remaining facets are collected into ${\mathcal{F}_{\mathrm{v}}}^i$, the set of interior facets of $\mathcal{T}_{\mathrm{v}}$. We assume that each facet F of the entire spatial mesh $\mathcal{T}$ is endowed with a unit normal n whose orientation is arbitrarily fixed, unless if F is contained in the global boundary $\partial\Omega_0$, in which case it points outward. Then, for any $x \in F$, set $[\![u]\!](x) = \lim_{\varepsilon \to 0} u(x + \varepsilon n) - u(x - \varepsilon n)$.

Lemma 3. *For all* $v, w \in V_h^{\mathtt{v}}$,

$$d(w,w) = \sum_{F \in \mathcal{F}_{\mathtt{v}}{}^i} 2 \int_F \delta\, \mathcal{S}[\![w]\!] \cdot [\![w]\!] + \sum_{F \in \mathcal{F}_{\mathtt{v}}{}^b} \int_F \delta\, \mathcal{B} w \cdot w\,. \tag{5.22}$$

Proof. Integration by parts on an element $T \in \mathcal{T}_{\mathtt{v}}$,

$$\sum_{j=1}^N \int_T \big(\delta A^{(j)} w\big) \cdot \partial_j w = \sum_{j=1}^N \int_{\partial T} \big(\delta A^{(j)} w\big) \cdot w - \sum_{j=1}^N \int_T \partial_j \big(\delta A^{(j)} w\big) \cdot w.$$

Applying the product rule to expand the derivative in the last term and using (5.10), we obtain

$$\sum_{j=1}^N \int_T \big(\delta A^{(j)} w\big) \cdot \partial_j w = \sum_{j=1}^N \int_{\partial T} \big(\delta A^{(j)} w\big) \cdot w - \sum_{j=1}^N \int_T \big((\partial_j \delta) A^{(j)} w + \delta A^{(j)} \partial_j w\big) \cdot w$$

and the symmetry of $A^{(j)}$ implies

$$2 \sum_{j=1}^N \int_T \big(\delta A^{(j)} w\big) \cdot \partial_j w = \sum_{j=1}^N \int_{\partial T} \big(\delta A^{(j)} w\big) \cdot w - \sum_{j=1}^N \int_T (\partial_j \delta) A^{(j)} w \cdot w. \tag{5.23}$$

Substituting the definitions of $A(w,w)$ and M_1 in (5.17) and (5.19), respectively, into (5.21), we get

$$\begin{aligned} d(w,w) &= \sum_{T \in \mathcal{T}_{\mathtt{v}}} \Bigg(- \int_T \sum_{j=1}^N \big(2\delta A^{(j)} w\big) \cdot \partial_j w + \int_{\partial T} 2\delta f_n^w \cdot w - \int_T \Big(\sum_{j=1}^N A^{(j)} \partial_j \delta \Big) w \cdot w \Bigg) \\ &= \sum_{T \in \mathcal{T}_{\mathtt{v}}} \Bigg(\int_{\partial T} 2\delta f_n^w \cdot w - \int_{\partial T} \delta \mathcal{D}^n w \cdot w \Bigg), \end{aligned} \tag{5.24}$$

using (5.23) and the definition of $\mathcal{D}^n$ in (5.7). With the numerical flux f_n^w defined in (5.13), we have

$$\begin{aligned} d(w,w) = \sum_{T \in \mathcal{T}_{\mathtt{v}}} \Big(&\int_{\partial T \setminus \partial\Omega_0} 2\delta (\mathcal{D}^n \{w\} + \mathcal{S}[\![w]\!]) \cdot w - \int_{\partial T \setminus \partial\Omega_0} \delta \mathcal{D}^n w \cdot w \Big) \\ &+ \Big(\int_{\partial T \cap \partial\Omega_0} \delta (\mathcal{D}^n + \mathcal{B}) w \cdot w - \int_{\partial T \cap \partial\Omega_0} \delta \mathcal{D}^n w \cdot w \Big). \end{aligned}$$

Rearranging to sums over interior and boundary facets completes the proof and (5.22) holds. □

Corollary 1. *Let* $m = \dim V_h^{\mathtt{v}}$ *and let* ψ_k, $k = 1, \dots, m$, *denote any standard local basis for* $V_h^{\mathtt{v}}$. *For the matrices* $\mathtt{A}_{kl} = A(\psi_k, \psi_l)$ *and* $(\mathtt{M}_1)_{kl} = \int_{\omega_{\mathtt{v}}} M_1 \psi_k \cdot \psi_l$, *for* $k, l = 1, \dots, m$, *holds*

$$\mathtt{w}^{\mathsf{T}} (2\mathtt{A} + \mathtt{M}_1)\, \mathtt{w} \le 0 \qquad \forall\, \mathtt{w} \in \mathbb{R}^m. \tag{5.25}$$

Proof. Recalling the definition of $d(w, w)$ in (5.22), we obtain

$$d(\psi_k, \psi_l) = -(2\mathtt{A}_{kl} + (\mathtt{M}_1)_{kl})$$

for $\psi_k, \psi_l \in V_h^{\mathrm{v}}$ and the result follows by (5.22). □

Theorem 4. *For the propagation matrix* $\mathtt{S}_{r,2}$*, defined in (5.3), of a two-stage structure aware scheme holds*

$$\|\mathtt{S}_{r,2}\|_{L(\mathtt{M}(0),\mathtt{M}(1))} = 1 + \mathcal{O}(r^{-2}), \tag{5.26}$$

where $r \in \mathbb{N}$ *denotes the number of substeps used to obtain* $\mathtt{u}_1^{r,2} = \mathtt{S}_{r,2}\,\mathtt{u}_0$*.*

Proof. Using the notation introduced in §4.2.4 for SARK methods, the partial propagation of $\mathtt{Y}^{[k]}$ reads

$$\mathtt{Y}^{[k+1]} = \mathtt{T}_{r,2}^{[k+1]}\mathtt{Y}^{[k]},$$

and with the relation $\mathtt{Y}^{[k]} = \mathtt{M}_0^{[k]}\mathtt{u}^{[k]}$, we obtain

$$\mathtt{M}_0^{[k+1]}\mathtt{u}^{[k+1]} = \mathtt{T}_{r,2}^{[k+1]}\mathtt{M}_0^{[k]}\mathtt{u}^{[k]}.$$

This allows us to express the partial propagation of $\mathtt{u}^{[k]}$ by

$$\mathtt{u}^{[k+1]} = \mathtt{S}_{r,2}^{[k+1]}\mathtt{u}^{[k]},$$

with $\mathtt{S}_{r,2}^{[k+1]} = (\mathtt{M}_0^{[k+1]})^{-1}\mathtt{T}_{r,2}^{[k+1]}\mathtt{M}_0^{[k]}$. For a given input $\mathtt{u}^{[0]} = \mathtt{u}_0$, the propagation to the solution $\mathtt{u}_1^{r,2} = \mathtt{u}^{[r]}$ at $\hat{t}_r = 1$ yields

$$\mathtt{u}_1^{r,2} = \mathtt{u}^{[r]} = \mathtt{S}_{r,2}\,\mathtt{u}^{[0]},$$

where $\mathtt{S}_{r,2} = \mathtt{S}_{r,2}^{[r]} \circ \ldots \circ \mathtt{S}_{r,2}^{[2]} \circ \mathtt{S}_{r,2}^{[1]}$ denotes the propagation matrix.

Using (4.41), we express the partial propagation from $\mathtt{u}^{[k]}$ to $\mathtt{u}^{[k+1]}$ by

$$\begin{aligned}\mathtt{u}^{[k+1]} &= \mathtt{S}_{r,2}^{[k+1]}\mathtt{u}^{[k]} = (\mathtt{M}_0^{[k+1]})^{-1}\mathtt{T}_{r,2}^{[k+1]}\mathtt{M}_0^{[k]}\mathtt{u}^{[k]} \\ &= (\mathtt{M}_0^{[k+1]})^{-1}\left(\mathtt{M}_0^{[k]} + \tau^{[k]}\mathtt{A} + \tfrac{1}{2}(\tau^{[k]})^2\mathtt{A}\left(\mathtt{M}_0^{[k]}\right)^{-1}(\mathtt{M}_1 + \mathtt{A})\right)\mathtt{u}^{[k]}.\end{aligned} \tag{5.27}$$

Further, utilizing the definition (4.4) of $\mathtt{M}(\hat{t})$, we rewrite $\mathtt{M}_0^{[k+1]}$ in terms of $\mathtt{M}_0^{[k]}$ and obtain

$$\mathtt{M}_0^{[k+1]} = \mathtt{M}(\hat{t}_{k+1}) = \mathtt{M}_0^{[k]} - \tau^{[k]}\mathtt{M}_1 = \left(\mathtt{I} - \tau^{[k]}\mathtt{M}_1\left(\mathtt{M}_0^{[k]}\right)^{-1}\right)\mathtt{M}_0^{[k]}.$$

For its inverse holds

$$\begin{aligned}\left(\mathtt{M}_0^{[k+1]}\right)^{-1} &= \left(\mathtt{M}_0^{[k]}\right)^{-1}\left(\mathtt{I} - \tau^{[k]}\mathtt{M}_1\left(\mathtt{M}_0^{[k]}\right)^{-1}\right)^{-1} \\ &= \left(\mathtt{M}_0^{[k]}\right)^{-1} + \tau^{[k]}\left(\mathtt{M}_0^{[k]}\right)^{-1}\mathtt{M}_1\left(\mathtt{M}_0^{[k]}\right)^{-1} \\ &\quad + (\tau^{[k]})^2\left(\mathtt{M}_0^{[k]}\right)^{-1}\mathtt{M}_1\left(\mathtt{M}_0^{[k]}\right)^{-1}\mathtt{M}_1\left(\mathtt{M}_0^{[k]}\right)^{-1} + \mathcal{O}((\tau^{[k]})^3).\end{aligned} \tag{5.28}$$

Next, we relate the $\|\cdot\|_{\mathtt{M}(\hat{t})}$ norms of the intermediate solutions $\mathtt{u}^{[k]}$ at $\hat{t}=\hat{t}_k$ and $\mathtt{u}^{[k+1]}$ at $\hat{t}=\hat{t}_{k+1}$. Using the representation (5.27) of $\mathtt{u}^{[k+1]}$ and the symmetry of $\mathtt{M}_0^{[k]}$, we get

$$\begin{aligned}\|\mathtt{u}^{[k+1]}\|^2_{\mathtt{M}(\hat{t}_{k+1})} &= \big(\mathtt{u}^{[k+1]}\big)^\intercal \mathtt{M}_0^{[k+1]}\mathtt{u}^{[k+1]}\\ &= \big(\mathtt{u}^{[k]}\big)^\intercal \Big(\mathtt{M}_0^{[k]} + \tau^{[k]}\mathtt{A}^\intercal + \tfrac{1}{2}(\tau^{[k]})^2\,(\mathtt{M}_1+\mathtt{A})^\intercal \big(\mathtt{M}_0^{[k]}\big)^{-1}\mathtt{A}^\intercal\Big)\\ &\quad \big(\mathtt{M}_0^{[k+1]}\big)^{-1}\Big(\mathtt{M}_0^{[k]} + \tau^{[k]}\mathtt{A} + \tfrac{1}{2}(\tau^{[k]})^2\mathtt{A}\big(\mathtt{M}_0^{[k]}\big)^{-1}(\mathtt{M}_1+\mathtt{A})\Big)\,\mathtt{u}^{[k]}.\end{aligned} \tag{5.29}$$

Substituting (5.28) into (5.29) and gathering the powers of $\tau^{[k]}$ yields

$$\begin{aligned}\|\mathtt{u}^{[k+1]}\|^2_{\mathtt{M}(\hat{t}_{k+1})} &= \big(\mathtt{u}^{[k]}\big)^\intercal \mathtt{M}_0^{[k]}\mathtt{u}^{[k]} + \tau^{[k]}\big(\mathtt{u}^{[k]}\big)^\intercal\big(\mathtt{A}^\intercal+\mathtt{A}+\mathtt{M}_1\big)\mathtt{u}^{[k]}\\ &\quad + \frac{(\tau^{[k]})^2}{2}\big(\mathtt{u}^{[k]}\big)^\intercal \mathtt{C}\mathtt{u}^{[k]} + \mathcal{O}((\tau^{[k]})^3),\end{aligned} \tag{5.30}$$

with the matrix

$$\begin{aligned}\mathtt{C} &= (\mathtt{M}_1+\mathtt{A})^\intercal\big(\mathtt{M}_0^{[k]}\big)^{-1}\mathtt{A}^\intercal + 2\mathtt{A}^\intercal\big(\mathtt{M}_0^{[k]}\big)^{-1}\mathtt{M}_1 + 2\mathtt{M}_1^\intercal\big(\mathtt{M}_0^{[k]}\big)^{-1}\mathtt{A}\\ &\quad + 2\mathtt{A}^\intercal\big(\mathtt{M}_0^{[k]}\big)^{-1}\mathtt{A} + 2\mathtt{M}_1^\intercal\big(\mathtt{M}_0^{[k]}\big)^{-1}\mathtt{M}_1 + \mathtt{A}\big(\mathtt{M}_0^{[k]}\big)^{-1}(\mathtt{M}_1+\mathtt{A})\\ &= (\mathtt{M}_1+\mathtt{A})^\intercal\big(\mathtt{M}_0^{[k]}\big)^{-1}\mathtt{A}^\intercal + 2\,(\mathtt{M}_1+\mathtt{A})^\intercal\big(\mathtt{M}_0^{[k]}\big)^{-1}(\mathtt{M}_1+\mathtt{A}) + \mathtt{A}\big(\mathtt{M}_0^{[k]}\big)^{-1}(\mathtt{M}_1+\mathtt{A})\\ &= (\mathtt{M}_1+\mathtt{A})^\intercal\big(\mathtt{M}_0^{[k]}\big)^{-1}(\mathtt{A}^\intercal+\mathtt{M}_1+\mathtt{A}) + (\mathtt{M}_1^\intercal+\mathtt{A}^\intercal+\mathtt{A})\big(\mathtt{M}_0^{[k]}\big)^{-1}(\mathtt{M}_1+\mathtt{A}).\end{aligned}$$

Thus, (5.30) simplifies to

$$\begin{aligned}\|\mathtt{u}^{[k+1]}\|^2_{\mathtt{M}(\hat{t}_{k+1})} &= \big(\mathtt{u}^{[k]}\big)^\intercal \mathtt{M}_0^{[k]}\mathtt{u}^{[k]} + \tau^{[k]}\big(\mathtt{u}^{[k]}\big)^\intercal\big(\mathtt{A}^\intercal+\mathtt{A}+\mathtt{M}_1\big)\mathtt{u}^{[k]}\\ &\quad + \frac{(\tau^{[k]})^2}{2}\big(\mathtt{u}^{[k]}\big)^\intercal(\mathtt{M}_1+\mathtt{A})^\intercal\big(\mathtt{M}_0^{[k]}\big)^{-1}(\mathtt{A}^\intercal+\mathtt{M}_1+\mathtt{A})\,\mathtt{u}^{[k]}\\ &\quad + \frac{(\tau^{[k]})^2}{2}\big(\mathtt{u}^{[k]}\big)^\intercal(\mathtt{M}_1+\mathtt{A}^\intercal+\mathtt{A})\big(\mathtt{M}_0^{[k]}\big)^{-1}(\mathtt{M}_1+\mathtt{A})\,\mathtt{u}^{[k]} + \mathcal{O}((\tau^{[k]})^3),\end{aligned} \tag{5.31}$$

where we used the symmetry of $\mathtt{M}_1$. To obtain a complete square, we add and subtract

$$\frac{(\tau^{[k]})^3}{4}(\mathtt{M}_1+\mathtt{A})^\intercal\big(\mathtt{M}_0^{[k]}\big)^{-1}\big(\mathtt{A}^\intercal+\mathtt{A}+\mathtt{M}_1\big)\big(\mathtt{M}_0^{[k]}\big)^{-1}(\mathtt{M}_1+\mathtt{A})$$

in (5.31) an get

$$\|\mathtt{u}^{[k+1]}\|^2_{\mathtt{M}(\hat{t}_{k+1})} = \big(\mathtt{u}^{[k]}\big)^\intercal \mathtt{M}_0^{[k]}\mathtt{u}^{[k]} + \tau^{[k]}\big(\mathtt{u}^{[k]}\big)^\intercal\mathtt{D}\mathtt{u}^{[k]} + \mathcal{O}((\tau^{[k]})^3),$$

where $\mathtt{D} = \big(\mathtt{I} + \frac{\tau^{[k]}}{2}(\mathtt{M}_1+\mathtt{A})^\intercal\big(\mathtt{M}_0^{[k]}\big)^{-1}\big)\big(\mathtt{A}^\intercal+\mathtt{A}+\mathtt{M}_1\big)\big(\mathtt{I} + \frac{\tau^{[k]}}{2}\big(\mathtt{M}_0^{[k]}\big)^{-1}(\mathtt{M}_1+\mathtt{A})\big)$. Corollary 1 implies $\mathtt{w}^\intercal\big(\mathtt{A}^\intercal+\mathtt{A}+\mathtt{M}_1\big)\mathtt{w} \le 0$, for any $\mathtt{w}\in\mathbb{R}^m$, and there holds $\big(\mathtt{u}^{[k]}\big)^\intercal\mathtt{D}\mathtt{u}^{[k]} \le 0$. For the intermediate solutions $\mathtt{u}^{[k]}$ at the bottom and $\mathtt{u}^{[k+1]}$ at the top of the subtent, we obtain the relation

$$\|\mathtt{u}^{[k+1]}\|^2_{\mathtt{M}(\hat{t}_{k+1})} \le \big(\mathtt{u}^{[k]}\big)^\intercal \mathtt{M}_0^{[k]}\mathtt{u}^{[k]} + \mathcal{O}((\tau^{[k]})^3) = \|\mathtt{u}^{[k]}\|^2_{\mathtt{M}(\hat{t}_k)} + \mathcal{O}((\tau^{[k]})^3). \tag{5.32}$$

Thus we can bound $\|\mathtt{S}_{r,2}\|_{L(\mathtt{M}(0),\mathtt{M}(1))}$ using (5.32) multiple times to obtain

$$\|\mathtt{S}_{r,2}\|_{L(\mathtt{M}(0),\mathtt{M}(1))} = \frac{\|\mathtt{u}^{[r]}\|_{\mathtt{M}(\hat{t}_r)}}{\|\mathtt{u}^{[0]}\|_{\mathtt{M}(\hat{t}_0)}} = \prod_{k=0}^{r-1} \frac{\|\mathtt{u}^{[k+1]}\|_{\mathtt{M}(\hat{t}_{k+1})}}{\|\mathtt{u}^{[k]}\|_{\mathtt{M}(\hat{t}_k)}} \tag{5.33}$$

$$\leq \left(1 + \mathcal{O}((\tau^{[k]})^3)\right)^r = 1 + \mathcal{O}(r^{-2}), \tag{5.34}$$

where we used $\tau^{[k]} = \frac{1}{r}$. □

5.3 Discrete stability measure for a model problem

We report the practically observed values of the previously described stability measure (namely the norm $\|\mathtt{S}_{r,s}\|_{L(\mathtt{M}(0),\mathtt{M}(1))}$) for some structure aware schemes applied to the two-dimensional convection equation

$$\partial_t u(x,t) + \operatorname{div}_x (b\, u(x,t)) = 0, \qquad \forall (x,t) \in \Omega_0 \times (0, t_{\max}],$$

with $\Omega_0 = [0,1]^2$, $t_{\max} = 0.05$, the flux field $b = (1,1)^\top$ and periodic boundary conditions. The time slab $\Omega = \Omega_0 \times (0, t_{\max})$ is filled with tents. Within each such tent K_i, let C_i denote the norm $\|\mathtt{S}_{r,s}\|_{L(\mathtt{M}(0),\mathtt{M}(1))}$ computed with $\mathtt{S}_{r,s}, \mathtt{M}(0)$, and $\mathtt{M}(1)$ specific to that tent. We expect C_i to be close to one for a stable method. Let

$$\bar{C} := \max_i \{C_i - 1\}, \tag{5.35}$$

where the maximum is taken over all tents in the time slab. To gain an understanding of practical stability, we examine the values of $\bar{C}$ as a function of the number of stages (s), polynomial degree (p), and more importantly, the number of substeps per tent (r).

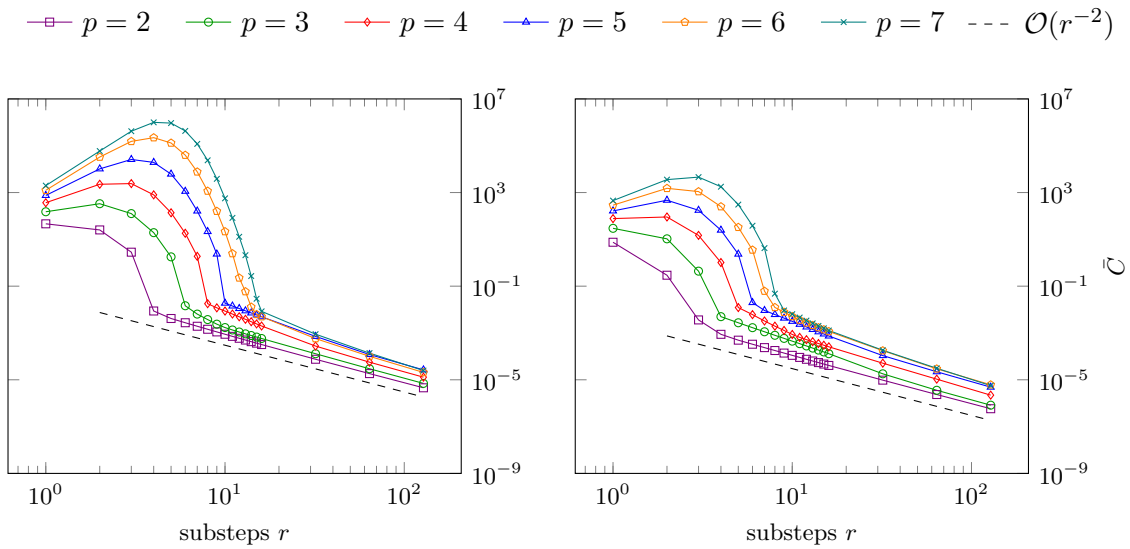
In all our numerical experiments, we observed that on each tent, for a fixed s, the norm $\|\mathtt{S}_{r,s}\|_{L(\mathtt{M}(0),\mathtt{M}(1))}$ tends to 1 with increasing number of substeps r, and moreover, we discovered a dependence of the following form

$$\|\mathtt{S}_{r,s}\|_{L(\mathtt{M}(0),\mathtt{M}(1))} = 1 + \mathcal{O}\left(r^{-s}\right)$$

on each tent K_i. Therefore, we organize our report on numerical stability observations into plots of values of $\bar{C}$ as a function of r. We do so for s-stage methods with $s = 2$ in Figure 5.1, $s = 3$ in Figure 5.2 and $s = 4$ in Figure 5.3. After a prominent preasymptotic region, we observe that $\bar{C}$, as a function of r, exhibits a rate of at least $\mathcal{O}\left(r^{-s}\right)$.

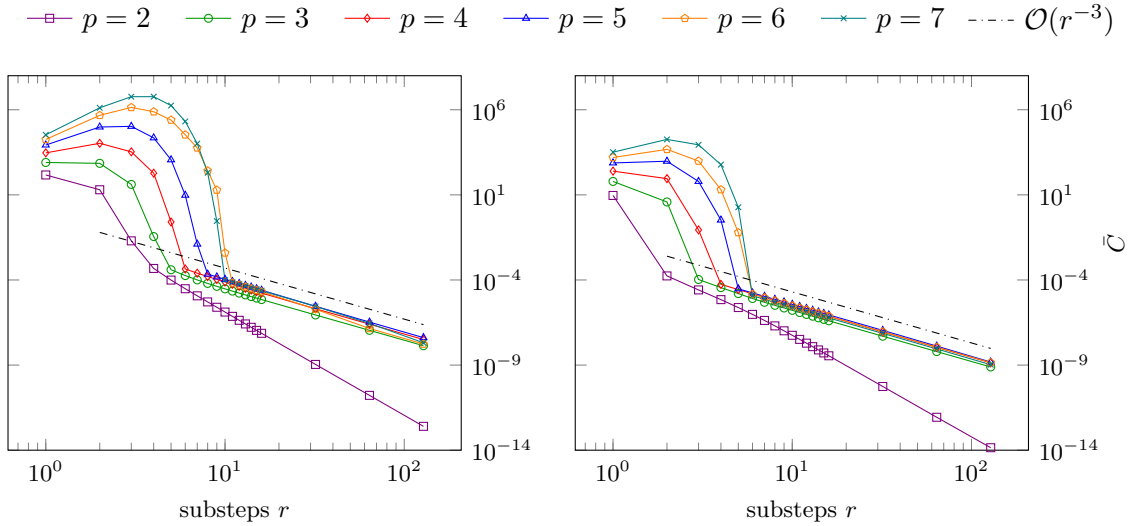
The preasymptotic region also depends on the wave speed $\bar{c}$ used to build the tents. When comparing the values of $\bar{C}$ for a two-stage method using a wave speed $\bar{c} = 4$ in Figure 5.1a and $\bar{c} = 8$ in Figure 5.1b, we observe that the curves are shifted to the bottom left. Thus a higher wave speed favors the stability properties due to smaller tents and the resulting smaller subtents. This behavior can also be observed for three- and four-stage methods in Figures 5.2 and 5.3.

Note that all curves plotted in Figures 5.1–5.3 shift to the top and right as p increases, i.e., the number of substeps r required to keep the same stability measure $\bar{C}$ increases with p. This behavior is akin to the p-dependence of the CFL-conditions of standard time stepping schemes.



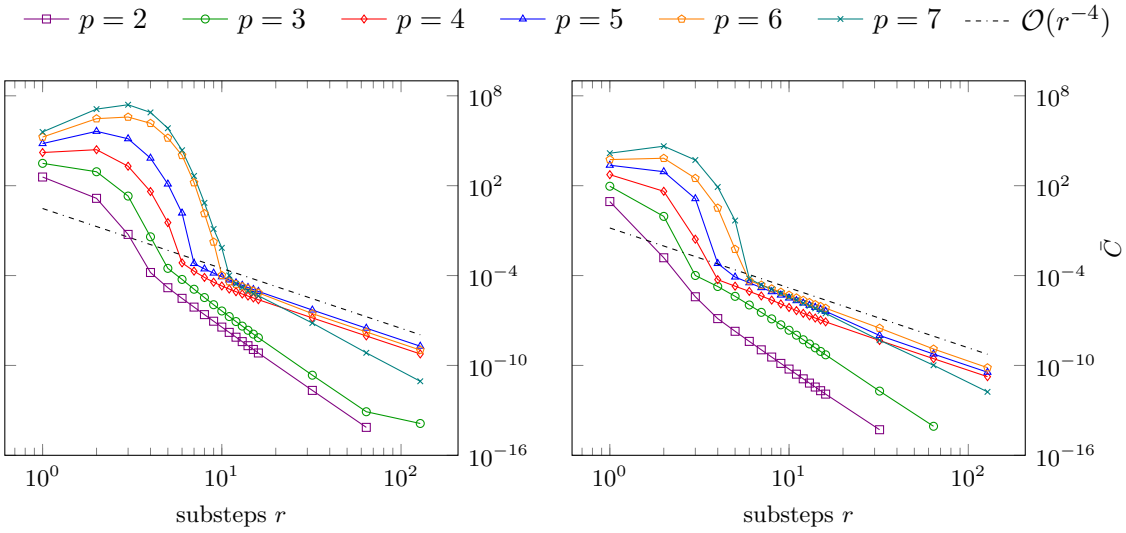
(a) Wave speed $\bar{c} = 4$ (b) Wave speed $\bar{c} = 8$

Figure 5.1: Observed dependence of $\bar{C}$ on r for a two-stage structure aware method with $2 \leq p \leq 7$.



(a) Wave speed $\bar{c} = 4$ (b) Wave speed $\bar{c} = 8$

Figure 5.2: Observed dependence of $\bar{C}$ on r for a three-stage structure aware method with $2 \leq p \leq 7$.



(a) Wave speed $\bar{c} = 4$

(b) Wave speed $\bar{c} = 8$

Figure 5.3: Observed dependence of $\bar{C}$ on r for a four-stage structure aware method with $2 \leq p \leq 7$.

6 Numerical examples

In this chapter we present numerical examples for linear and nonlinear hyperbolic systems to illustrate the applicability of MTP methods. These methods were implemented within the finite element library Netgen/NGSolve – see [30, 31] and www.ngsolve.org.

6.1 Linear examples

The first example in §6.1.1 shows how MTP methods adapt to varying speeds of propagation leading to a naturally built in local time stepping. Local time stepping also plays an import role for locally refined meshes as in §6.1.2, where we solve a large three-dimensional problem and discuss the speed up compared to SDG methods.

6.1.1 Wave equation with heterogeneous material

In the following example we show how tent pitching methods handle varying material parameters. We consider an example where a wave is partially reflected at an interface of two different materials, which was also performed in [28].

We consider the wave equation (2.7) on the spatial domain $\Omega_0 = [0,2]^2$ and the final time is set to $t_{\max} = 0.4$. The speed of propagation c_s is given by the piecewise constant function

$$c_s(x) = \begin{cases} 1, & x_1 \leq 1.2, \\ 3, & x_1 > 1.2, \end{cases}$$

which defines the material parameter $\alpha = c_s^2 I$, where $I \in \mathbb{R}^{2\times 2}$ denotes the identity matrix. The initial condition q_0, μ_0 is given by

$$u_0 = \begin{bmatrix} q_0 \\ \mu_0 \end{bmatrix} = \begin{bmatrix} -\alpha(x)(\nabla_x \phi)(x,0) \\ 0 \end{bmatrix}, \tag{6.1}$$

where ϕ is defined as Gaussian peak

$$\phi(x,t) = \exp\left(-\varepsilon^{-2}\|x - x_0\|^2\right),$$

at $x_0 = (1,1)$, with the constant $\varepsilon = 0.01$.

Based on the spatial mesh with simplicial elements of size $h = 0.05$, we pitch tents in the time slab of height 0.1 using the wave speed $\bar{c} = c_s$ and the constant $C_{\mathcal{T}} = \frac{1}{3}$. The resulting time slab is shown in Figure 6.1. We observe that the tents height, e.g. the local time step, adapts to the local speed of propagation resulting in larger tents in the subdomain with smaller c_s and smaller tents in the subdomain with larger c_s. When using standard discontinuous Galerkin methods combined explicit time stepping, one would have

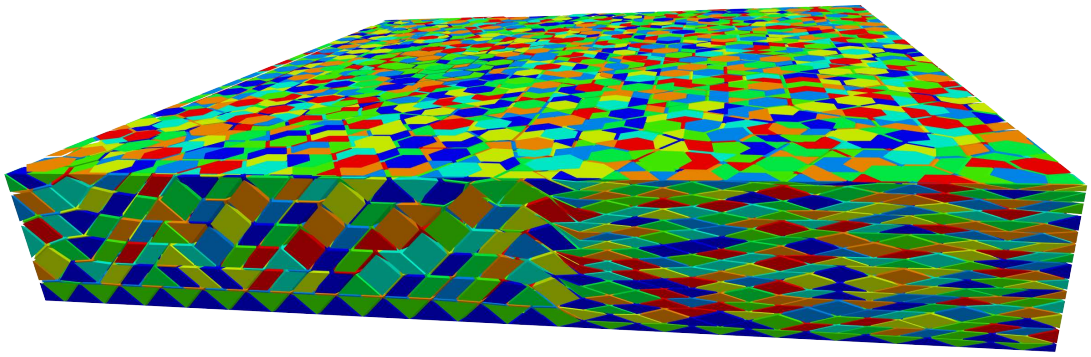


Figure 6.1: Time slab showing tents pitched on a spatial mesh with varying wave speed.

to enforce the more stringent condition globally. For the spatial discretization we choose fourth order polynomial and set the local spaces V_h^{v} to

$$V_h^{\mathrm{v}} = \{\psi \in [L^2(\omega_{\mathrm{v}})]^3 : \psi|_T \in [P_4(T)]^3 \ \forall\, T \in \mathcal{T}_{\mathrm{v}}\},$$

and use the upwind flux f_n in (3.37). The solution of the scalar component μ_h at the final time $t_{\max} = 0.4$ is shown in Figure 6.2. We see that after reaching the interface, the transmitted part of the wave travels at higher speed than the superposition of the initial and the reflected wave.

6.1.2 Maxwell equations

To illustrate the abilities of MTP schemes, we present a large scale problem in three space dimensions, as discussed in [11]. We solve the Maxwell equations in conservation form

$$\partial_t \begin{bmatrix} \varepsilon E \\ \mu H \end{bmatrix} + \operatorname{div}_x \begin{bmatrix} -\operatorname{skew} H \\ \operatorname{skew} E \end{bmatrix} = 0\,,$$

as defined in (2.11), on a domain $\Omega_0 \subset [0, 1196.8] \times [-133, 133] \times [-133, 133]$ similar to the resonator shown in [22]. The geometry is given as body of revolution of smooth B-spline curves with the first inner rounding located at $x = 217.6$ and a distance of 108.8 to the next inner rounding. The mesh consisting of is 489k curved elements is shown in Figure 6.3, which has a ratio of largest to smallest element of approximately $5 : 1$.

The initial data is set to

$$u_0 = \begin{bmatrix} E_0 \\ H_0 \end{bmatrix}, \quad E_0 = \begin{bmatrix} 0 \\ 0 \\ E_{0,z} \end{bmatrix}, \quad H_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

where the third component of the electrical field $E_{0,z}(x) = \exp(-10^{-3}((x_1 - \bar{x}_1)^2 + x_2^2 + x_3^2))$ is set to a Gaussian peak located at the axis of revolution and the position $\bar{x}_1 = 625.8$ of the fifth inner rounding – see Figure 6.4. Further we set the material parameters $\varepsilon = \mu = 1$, which leads to the constant characteristic speed $\bar{c} = 1$, as derived in (2.12).

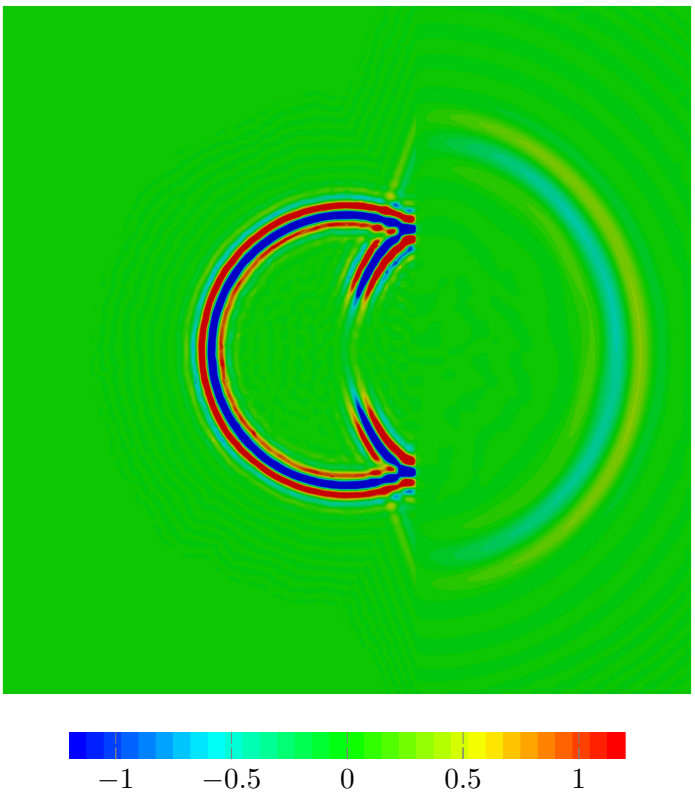


Figure 6.2: Scalar component μ_h of a wave traveling through heterogeneous media at the final time $t_{\max} = 0.4$.

To obtain the ODE system (3.30), we have to specify

$$V_h^{\mathrm{v}} = \{\psi \in [L^2(\omega_{\mathrm{v}})]^6 : \psi|_T \in [P_p(T)]^6 \ \forall\, T \in \mathcal{T}_{\mathrm{v}}\},$$

and the numerical flux f_n in (3.26). On interior facets $F \in \mathcal{F}_{\mathrm{v}}{}^i$, we used the upwind flux [21, p. 434]

$$f_n(\hat{u}^+, \hat{u}^-) = \begin{bmatrix} \{\hat{H}\} \times n + [\![\hat{E}_t]\!] \\ -\{\hat{E}\} \times n + [\![\hat{H}_t]\!] \end{bmatrix},$$

with the tangential components $\hat{E}_t = n \times (\hat{E} \times n)$, $\hat{H}_t = n \times (\hat{H} \times n)$ and the mean values $\{\hat{E}\} = \frac{1}{2}(\hat{E}^+ + \hat{E}^-)$, $\{\hat{H}\} = \frac{1}{2}(\hat{H}^+ + \hat{H}^-)$ of $\hat{E} = E \circ \Phi$ and $\hat{H} = H \circ \Phi$. This numerical flux coincides with the general definition in (5.13), which is we also used on the boundary facets $F \in \mathcal{F}_{\mathrm{v}}{}^b$, with $\mathcal{D}^n$ and $\mathcal{B}$ given in (5.15) and (5.16).

The H_y component of the solution at $t_{\max} = 260$ shown in Figure 6.5 was computed with polynomials of degree $p = 3$ in space and time slabs of height 1, each slab composed of $N_{\text{tents}} = 149072$ tents. On each tent we used a $(p+1)$-stage SAT time-stepping with $r = 2p$ intervals. With the spatial degrees of freedom $N_{\text{dof},i}$ of the ith tent and the number of stages $q = p + 1$, we obtain the total spacetime degrees of freedom per time slab

$$\sum_{i=1}^{N_{\text{tents}}} N_{\text{dof},i}\, m\, q = \left(\sum_{i=1}^{N_{\text{tents}}} N_{\text{dof},i} \right) 2p(p+1)\ .$$

The corresponding numbers of degrees of freedom and the simulation times are shown in Table 6.1. In [22] a similar problem is solved using a discontinuous Galerkin method with quadratic elements, combined with a polynomial Krylov subspace method in time. Using 96 cores it took them 7:10 hours to reach the final time. Our simulation with polynomial order $p = 3$, which has a comparable number of unknowns, took 3:33 hours on 64 cores. This significant speed up is an illustration of the capability of the new method.

	$p = 2$	$p = 3$
number of spatial dof	2.938×10^7	5.875×10^7
number of spacetime dof per slab	1.908×10^9	7.632×10^9
simulation time per slab	4.6 s	49.2 s
total simulation time	20 min	3 h 33 min

Table 6.1: Number of degrees of freedom and simulation times for spatial polynomial degrees $p = 2, 3$. This data was generated using a shared memory server with 4 E7-8867 CPUs with 16 cores each.

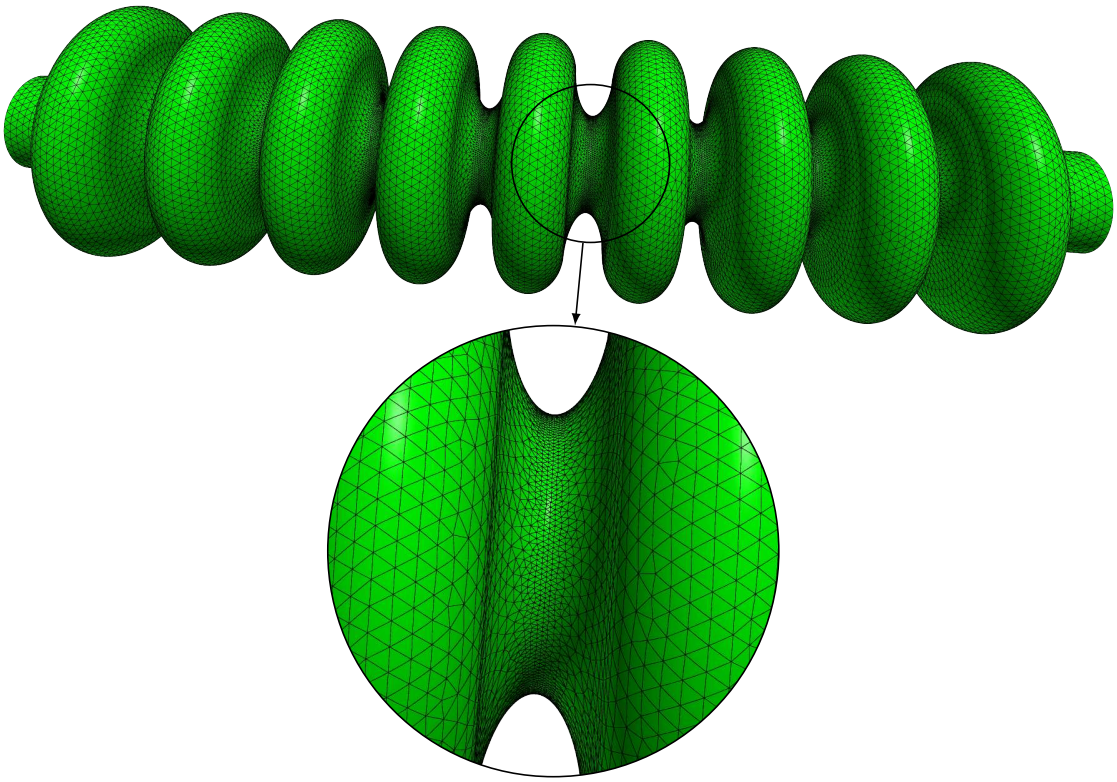


Figure 6.3: Tetrahedral mesh with 489k curved elements, ratio of the largest to the smallest element of approximately 5:1.

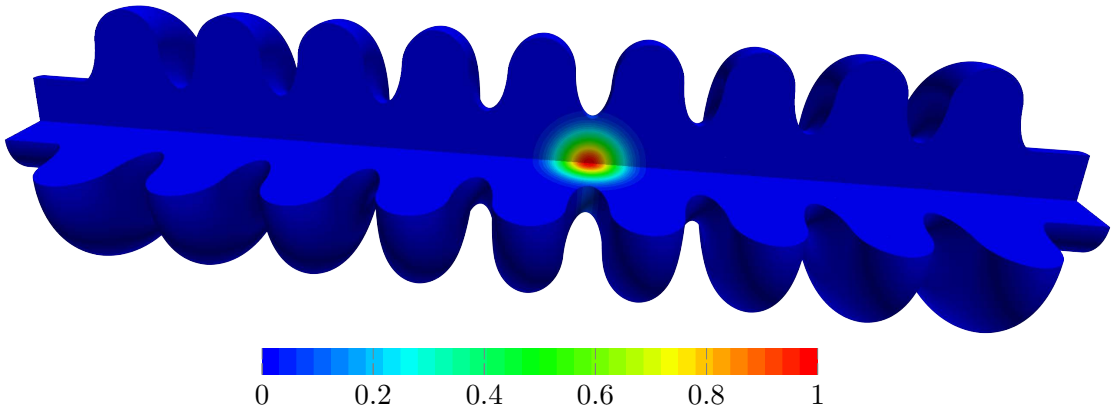


Figure 6.4: Third component of the electrical field $E_{0,z}(x)$ set as initial data.

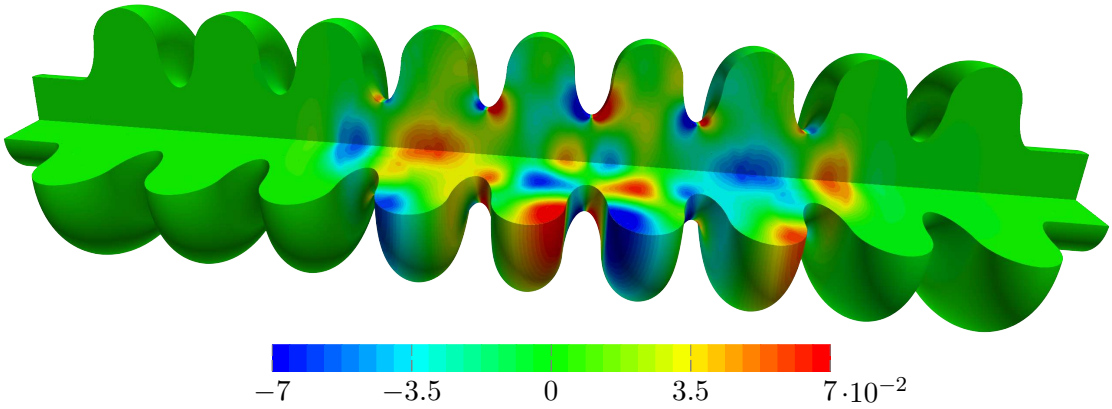


Figure 6.5: Second component $H_y(x)$ of the magnetic field at t = 260 solved with spatial polynomial degree $p = 3$.

6.2 Nonlinear examples

In this section we construct an explicit MTP scheme including an entropy viscosity regularization – described in §6.2.1 – to handle occurring shocks and discontinuities while solving nonlinear hyperbolic problems. This is then applied to the Euler equations in §6.2.2, where we consider the well-known benchmark of the Mach 3 wind tunnel.

Suppose that on a tent K, we are given a solution $u(x,t)$ of (2.1) and an entropy pair $(\mathcal{E},\mathcal{F})$. The mapped solution, as before, is $\hat{u} = u \circ \Phi$. Define

$$\hat{\mathcal{E}}(w) = \mathcal{E}(w) - \mathcal{F}(w)\,\nabla_x \varphi, \tag{6.2a}$$

$$\hat{\mathcal{F}}(w) = \delta \mathcal{F}(w). \tag{6.2b}$$

Theorem 5. *Suppose u solves (2.1) on K and $\hat{u} = u \circ \Phi$ solves the mapped equation (3.17). Then, whenever $(\mathcal{E},\mathcal{F})$ is an entropy pair for (2.1), $(\hat{\mathcal{E}},\hat{\mathcal{F}})$ is an entropy pair for (3.17). Moreover, if $\mathcal{E}(u)$ and $\mathcal{F}(u)$ satisfies the entropy admissibility condition (2.13) on K, then $\hat{\mathcal{E}}(\hat{u})$ and $\hat{\mathcal{F}}(\hat{u})$ satisfies the entropy admissibility condition on $\hat{K}$.*

Proof. Repeating the calculations in the proof of Theorem 1, with $g = \mathcal{E}$ and $f = \mathcal{F}$, we obtain

$$(\partial_t \mathcal{E}(u) + \operatorname{div}_x \mathcal{F}(u)) \circ \Phi = \frac{1}{\delta}\left(\partial_{\hat{t}} \hat{\mathcal{E}}(\hat{u}) + \operatorname{div}_x \hat{\mathcal{F}}(\hat{u})\right),$$

from which the statements of the theorem follow. □

6.2.1 Entropy viscosity regularization

The addition of "artificial viscosity" (a diffusion term) to the right hand side of nonlinear conservation laws makes their solutions dissipative. When the limit of such solutions, as the diffusion term goes to zero, exist in some sense, it is referred to as a vanishing viscosity solution. It is known [4, Theorem 4.6.1] that the vanishing viscosity solution satisfies the entropy admissibility condition for entropy pairs satisfying certain conditions. Motivated

by this property, the entropy viscosity regularization method of [17], suggests modifying the numerical scheme by selectively adding small amounts of artificial viscosity, to avoid spurious oscillations near discontinuities of the solutions. We borrow this technique and incorporate it into the MTP schemes as follows.

Consider the problem on the tent K mapped to $\hat{K}$. We set the spatial discretization space to $V_h^{\mathrm{v}} = \{u \in [L^2(\omega_{\mathrm{v}})]^L : u|_T \in P_p(T)\ \forall T \in \mathcal{T}_{\mathrm{v}}\}$ and consider a discontinuous Galerkin discretization

$$\int_{\omega_{\mathrm{v}}} \partial_{\hat{t}} M(\hat{t}, \hat{u}_h) \cdot v_h = \sum_{T\in\mathcal{T}_{\mathrm{v}}} \int_T \delta f(\hat{u}_h) : \nabla v_h - \sum_{F\in\mathcal{F}_{\mathrm{v}}} \int_F \delta f_n(\hat{u}_h^+, \hat{u}_h^-) \cdot [\![v_h]\!], \tag{3.26}$$

of the mapped equation (3.22). Let $(\cdot,\cdot)_h$ denote integral over the vertex patch ω_{v}, such that

$$(w_h, v_h)_h = \int_{\omega_{\mathrm{v}}} w_h v_h = \sum_{T\subset\omega_{\mathrm{v}}} \int_T w_h v_h \qquad \forall w_h, v_h \in V_h^{\mathrm{v}}.$$

Recalling the definition (3.29) of $A(\cdot,\cdot)$ and using the variable $\hat{y}_h = M(\hat{t}, \hat{u}_h)$, as discussed in §3.5, the semi-discretization takes the form

$$(\partial_{\hat{t}}\, \hat{y}_h, v_h)_h = A(M^{-1}(\hat{y}_h), v_h) \tag{6.3}$$

for all $v_h \in V_h^{\mathrm{v}}$.

Suppose that an entropy pair $(\mathcal{E}, \mathcal{F})$ is given for (2.1). On the mapped tent $\hat{K}$, let $(\hat{\mathcal{E}}, \hat{\mathcal{F}})$ be defined by (6.2). Suppose a numerical approximation $\hat{y}_h(x, \hat{t}_1)$ has been computed at some time $0 \le \hat{t}_1 < 1$ and we want to compute a numerical approximation at the next time stage, say at $\hat{t} = \hat{t}_1 + \Delta\hat{t} \le 1$. The *entropy residual* of the approximation $\hat{u}_h = M^{-1}(\hat{y}_h)$ to $\hat{u}$ is a weak form of the quantity $\partial_{\hat{t}}\hat{\mathcal{E}}(\hat{u}_h) + \mathrm{div}_x\, \hat{\mathcal{F}}(\hat{u}_h)$, which by Theorem 5, is nonpositive. The discrete entropy residual at time $\hat{t}_1$ is $R_h = \min(r_h, 0)$ where $r_h \in V_h^{\mathrm{v}}$ is defined by

$$\begin{aligned}(\delta r_h, v_h)_h &= \sum_{T\in\mathcal{T}_{\mathrm{v}}} \int_T \partial_{\hat{t}}\hat{\mathcal{E}}(\hat{u}_h) v_h - \sum_{T\in\mathcal{T}_{\mathrm{v}}} \int_T \hat{\mathcal{F}}(\hat{u}_h)\, \nabla_x\, v_h + \sum_{F\in\mathcal{F}_{\mathrm{v}}} \int_F \delta\mathcal{F}_n(\hat{u}_h) v_h \\ &= \sum_{T\in\mathcal{T}_{\mathrm{v}}} \int_T \frac{\partial(\hat{\mathcal{E}}\circ M^{-1})}{\partial\hat{y}} \partial_{\hat{t}}\hat{y}_h v_h - \sum_{T\in\mathcal{T}_{\mathrm{v}}} \int_T \hat{\mathcal{F}}(\hat{u}_h)\, \nabla_x\, v_h + \sum_{F\in\mathcal{F}_{\mathrm{v}}} \int_F \delta\mathcal{F}_n(\hat{u}_h) v_h\end{aligned}$$

for all $v_h \in V_h^{\mathrm{v}}$, where $\hat{u}_h = M^{-1}(\hat{y}_h)$. Here $\mathcal{F}_n$ is a numerical flux prescribed by a discontinuous Galerkin approximation to the entropy conservation equation. The term $\partial_{\hat{t}}\, \hat{y}_h$ can be replaced by its approximation available from (6.3) while computing r_h.

Next, following [17], we quantify the amount of viscosity to be added to (6.3). Define the *entropy viscosity coefficient* on one spatial element $T \in \mathcal{T}_{\mathrm{v}}$ by

$$\nu_{e,T} = c_X^2 \frac{\|R_h\|_{L^\infty(T)}}{|\bar{\mathcal{E}}|}, \tag{6.4}$$

where $\bar{\mathcal{E}}$ is the mean value of $\hat{\mathcal{E}}(M^{-1}(\hat{y}_h))$ on T and c_X is an effective local grid size of V_h^{v}, typically chosen as $c_X = \kappa_1\, \mathrm{diam}(T)/p$ for some fixed number κ_1 and the spatial polynomial degree p. To limit the viscosity added based on the local wave speed, define

$$\nu_{*,T} = \kappa_2\, \mathrm{diam}(T) \|D_u f(x, \hat{t}_1, \hat{u}_h(x, \hat{t}_1))\|_{L^\infty(T)}, \tag{6.5}$$

where κ_2 is another fixed number and set

$$\nu = \max_{T \in \mathcal{T}_\mathrm{v}} \min(\nu_{*,T}, \nu_{e,T}). \tag{6.6}$$

This artificial viscosity coefficient proposed in [17] leads to generous viscosity at discontinuities (where the entropy residual is high) and little viscosity in smooth regions. To apply this artificial viscosity, we first solve (6.3) in $\hat{t}_1 \le \hat{t} \le \hat{t}_1 + \Delta\hat{t} = \hat{t}_2$ to obtain $\hat{y}_h(x, \hat{t}_2)$ and $\hat{u}_h(x, \hat{t}_2) = M^{-1}(\hat{y}_h(x, \hat{t}_2))$ which allows us to calculate viscosity coefficient ν. Then we proceed by solving

$$\partial_{\hat{t}} \hat{w}_h - \delta \operatorname{div}_x(\nu \nabla_x \hat{w}_h) = 0 \tag{6.7}$$

in $\omega_\mathrm{v} \times (\hat{t}_1, \hat{t}_2)$, with initial data and boundary conditions set to

$$\begin{aligned} \hat{w}_h(x, \hat{t}_1) &= \hat{u}_h(x, \hat{t}_2), && \forall x \in \omega_\mathrm{v}, \\ \hat{w}_h(x, \hat{t}) &= \hat{u}_h(x, \hat{t}_2), && \forall (x, \hat{t}) \in \partial\omega_\mathrm{v} \times (\hat{t}_1, \hat{t}_2). \end{aligned}$$

Note, that the viscous term $-\operatorname{div}_x(\nu \nabla_x \hat{w}_h)$ is weighted with the tent height δ. The semi-discretization takes the form

$$(\delta^{-1} \partial_{\hat{t}} \hat{u}_h, v_h)_h + \nu A_{\mathrm{IP}}(\hat{u}_h, v_h) = 0, \tag{6.8}$$

for all $v_h \in V_h^\mathrm{v}$, where $A_{\mathrm{IP}}(\cdot,\cdot)$ is the standard interior penalty discontinuous Galerkin approximation of the viscous term $-\operatorname{div}_x(\nu \nabla_x \hat{w})$ defined below. On an interface F with the unit normal n, set

$$[\![w]\!] = w^+ - w^- \qquad \text{and} \qquad \{w\} = \frac{1}{2}\left(w^+ + w^-\right),$$

with the understanding that $w(x,t)$ is set to the boundary condition if x is outside the vertex patch ω_v. Then

$$\begin{aligned} A_{\mathrm{IP}}(w, v) = &\sum_{T \in \mathcal{T}_\mathrm{v}} \int_T \nabla_x w \, \nabla_x v - \sum_{F \in \mathcal{F}_\mathrm{v}{}^b} \int_F \left(\nabla_x w \cdot n \; v + [\![w]\!] \; \nabla_x v \cdot n - \frac{\alpha}{h} [\![w]\!] \; v \right) \\ &- \sum_{F \in \mathcal{F}_\mathrm{v}{}^i} \int_F \left(\{\nabla_x w\} \cdot n \; [\![v]\!] + [\![w]\!] \, \{\nabla_x v\} \cdot n - \frac{\alpha}{h} [\![w]\!] [\![v]\!] \right). \end{aligned}$$

Here, as usual, the penalization parameter α must be chosen large enough to obtain coercivity. Again, the set $\mathcal{F}_\mathrm{v}{}^b$ denotes the facets F on the boundary $\partial\omega_\mathrm{v}$ of the vertex patch ω_v and $\mathcal{F}_\mathrm{v}{}^i$ consists of the remaining inner facets in $\mathcal{T}_\mathrm{v}$.

To obtain the approximation $\hat{w}_h(x, \hat{t}_2)$ to $\hat{u}$ at $\hat{t}_2$, we solve (6.8) using a standard time stepping method. Finally, we apply the mapping M to obtain the regularized approximation $\hat{y}_h(x, \hat{t}_2) = M(\hat{t}_2, \hat{w}_h(x, \hat{t}_2))$, which is then used as input for the next time step.

Remark 5. The weight δ in (6.7) leads to a consistent partition of the viscous term over the whole time slab. Consider a constant viscosity coefficient ν for all tents in the time slab of height Δt illustrated in Figure 6.6.

Considering the spatial domain Ω_0, the discontinuous Galerkin semi-discretization of $\partial_t u - \operatorname{div}_x(\nu \nabla_x u) = 0$ takes the form

$$\frac{d}{dt}(\mathtt{G}\mathtt{u}) = -\mathtt{A}\mathtt{u},$$

where we used a basis $\psi_l \in V_h(\Omega_0) \subset [L_2(\Omega_0)]^L$, $l = 1, \dots m' = \dim V_h(\Omega_0)$. The matrix $\mathtt{G}$ represent the mass matrix

$$\mathtt{G}_{kl} = \sum_{T \in \Omega_0} \int_T \psi_k \psi_l, \qquad 1 \le k, l \le m',$$

and the matrix $\mathtt{A}$ the interior penalty discretization of the viscous term. The solution for the coefficient vector $\mathtt{u} \in \mathbb{R}^{m'}$ of the basis expansion at Δt reads

$$\mathtt{u}(\Delta t) = \exp(-(\Delta t)\, \mathtt{G}^{-1}\mathtt{A})\mathtt{u}(0). \tag{6.9}$$

Now we consider a time slab with final time Δt with tents K_i pitched at the vertex $\mathtt{v}^{(i)}$ for $i = 1, \dots, N_{\text{tents}}$ – see Figure 6.6. For the mapped variable $\hat{u}(x, \hat{t}) = u \circ \Phi(x, \hat{t})$ on the tent K_i, we spatially discretize (6.7) reformulated to

$$\frac{1}{\delta_i}\partial_{\hat{t}}\hat{u} - \operatorname{div}_x(\nu \nabla_x \hat{u}) = 0$$

on $\omega_{\mathtt{v}^{(i)}}$. Using the basis $\psi_l \in V_h^{\mathtt{v}^{(i)}} \subset V_h(\Omega_0)$, $l = 1, \dots m = \dim V_h^{\mathtt{v}^{(i)}}$, we obtain the semi-discrete ODE

$$\frac{d}{d\hat{t}}(\mathtt{G}_i\hat{\mathtt{u}}) = -\mathtt{A}_i\hat{\mathtt{u}}$$

for the local coefficient vector $\hat{\mathtt{u}} \in \mathbb{R}^m$ of the basis expansion on K_i. The matrix

$$[\mathtt{G}_i]_{kl} = \sum_{T \in \mathcal{T}_{\mathtt{v}^{(i)}}} \int_T \frac{1}{\delta_i}\psi_k \psi_l, \qquad 1 \le k, l \le m,$$

denotes the weighted mass matrix on the tent K_i and $\mathtt{A}_i$ the viscous term on the vertex patch $\omega_{\mathtt{v}^{(i)}}$. Then the solution at the top of the tent, e.g. at $\hat{t} = 1$, is given by

$$\hat{\mathtt{u}}(\hat{t} = 1) = \exp(-\mathtt{G}_i^{-1}\mathtt{A}_i)\hat{\mathtt{u}}(0), \tag{6.10}$$

for the local coefficient vector $\hat{\mathtt{u}} \in \mathbb{R}^m$. By defining the matrix

$$[\bar{\mathtt{G}}_i]_{kl} = \sum_{T \in \Omega_0} \int_T \delta_i \psi_k \psi_l, \qquad 1 \le k, l \le m',$$

which is zero for elements $T \notin \mathcal{T}_{\mathtt{v}^{(i)}}$ due to the scaling by the tent height $\delta_i = \tau_i - \tau_{i-1}$. The nonzero entries of $\mathtt{G}^{-1}\bar{\mathtt{G}}_i\mathtt{G}^{-1}$ correspond to the entries of $\mathtt{G}_i^{-1}$ and we can rewrite (6.10) using the global space $V_h(\Omega_0)$ by

$$\mathtt{u}_i = \exp(-\mathtt{G}^{-1}\bar{\mathtt{G}}_i\mathtt{G}^{-1}\mathtt{A})\mathtt{u}_{i-1},$$

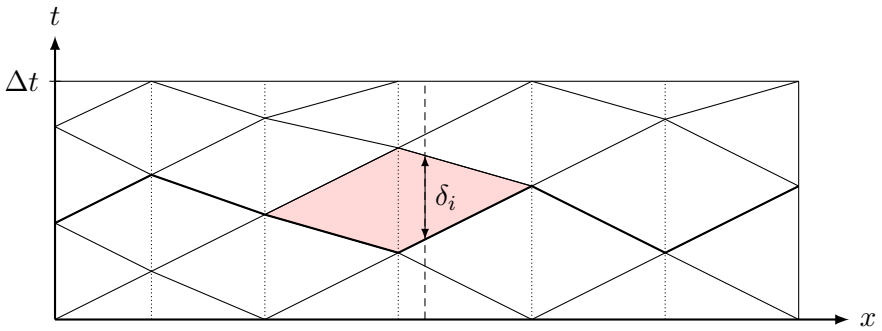


Figure 6.6: Time slab in one spatial dimension with the tent K_i filled in red.

where $\mathtt{u}_i, \mathtt{u}_{i-1} \in \mathbb{R}^{m'}$ represent the coefficient vector of the solution at the top and bottom advancing fronts τ_i and τ_{i-1} of the tent K_i. Further we observe that

$$\sum_{i=1}^{N_{\text{tents}}} \bar{\mathtt{G}}_i = \Delta t \mathtt{G}, \tag{6.11}$$

since the tent heights δ_i add up to the time slab height Δt - see Figure 6.6. Given the initial coefficients $\mathtt{u}_0 \in \mathbb{R}^{m'}$, we obtain the solution after proceeding trough all tents by

$$\begin{aligned}
\mathtt{u}_{N_{\text{tents}}} &= \prod_{i=1}^{N_{\text{tents}}} \exp(-\mathtt{G}^{-1}\bar{\mathtt{G}}_i\mathtt{G}^{-1}\mathtt{A})\ u_0 \\
&\approx \exp(-\mathtt{G}^{-1} \sum_{i=1}^{N_{\text{tents}}} (\bar{\mathtt{G}}_i)\mathtt{G}^{-1}\mathtt{A})\ u_0 \\
&= \exp(-\Delta t \mathtt{G}^{-1}\mathtt{A})\ u_0,
\end{aligned}$$

where we used (6.11) in the last step. This solution $\mathtt{u}_{N_{\text{tents}}}$ is an approximation to the solution of $\partial_t u - \operatorname{div}_x(\nu \nabla_x) = 0$ derived in (6.9) for the whole time slab.

6.2.2 Euler equations

Recall that the Euler system fits into (2.1) with

$$u = \begin{bmatrix} \rho \\ m \\ E \end{bmatrix}, \quad g(u) = u\,, \quad f(u) = \begin{bmatrix} m^\intercal \\ pI + m \otimes m/\rho \\ (E+p)m^\intercal/\rho \end{bmatrix}, \tag{6.12}$$

as discussed in (2.16). Here, the functions $\rho : \Omega_0 \to \mathbb{R}$, $m : \Omega_0 \to \mathbb{R}^2$ and $E : \Omega_0 \to \mathbb{R}$ denote the density, momentum, and total energy of a perfect gas in the spatial domain Ω_0. Furthermore, we use $p = \frac{1}{2}\rho T$ for the pressure, $T = \frac{4}{d}\big(\frac{E}{\rho} - \frac{1}{2}\frac{\|m\|^2}{\rho^2}\big)$ for the temperature and $d = 5$ denotes the degrees of freedom of the gas particles.

Convergence rates for a 2D Euler system

Now we apply SARK methods to the Euler system. Similar to the Burgers' example, which we discussed in §4.3.2, we choose smooth initial data and fix a final time before the onset of shocks so that no limiting is needed.

The initial values on the spatial domain $\Omega_0 = [0,1]^2$ are set by

$$\begin{aligned} \rho_0 &= 1 + e^{-100((x_1-0.5)^2+(x_2-0.5)^2)}, \\ m_0 &= [0,0]^\intercal, \\ P_0 &= 1 + e^{-100((x_1-0.5)^2+(x_2-0.5)^2)}, \end{aligned} \tag{6.13}$$

and the final time $t_{\max} = 0.1$.

The data shown in Figure 6.7 was generated with polynomial degree $p = 2$ in space and mesh sizes $h = 0.1 \times 2^{-i}$, for $i = 0 \dots 6$. For the tent generation $\bar{c}$ in (3.3) was set to 8 and the number of substeps to $r = 4$. Since we do not have an exact solution in closed form, we compare the numerical solution computed using $\bar{c}$ with a "reference solution" computed with the higher characteristic speed $2 \cdot \bar{c}$. The latter requires many more tents to reach the final time. Let the former and latter approximations to $u(\cdot, t_{\max})$ be denoted by u_h and u_h^{ref}, respectively. We define the error by

$$e_h := \left\| u_h - u_h^{\text{ref}} \right\|_{L^2(\Omega_0)}, \tag{6.14}$$

which is the quantity plotted in Figure 6.7.

The errors of the two-stage SARK method and the underlying Runge-Kutta method is seen to diverge already for the first refinement level in Figure 6.7a. While the SARK method shows the expected quadratic convergence, the rate of the Runge-Kutta method drops to first order. For the three-stage methods in Figure 4.5b, we see cubic convergence for both method for the first few refinements. The convergence rate of the Runge-Kutta method eventually drops to first order while the SARK converges at third order.

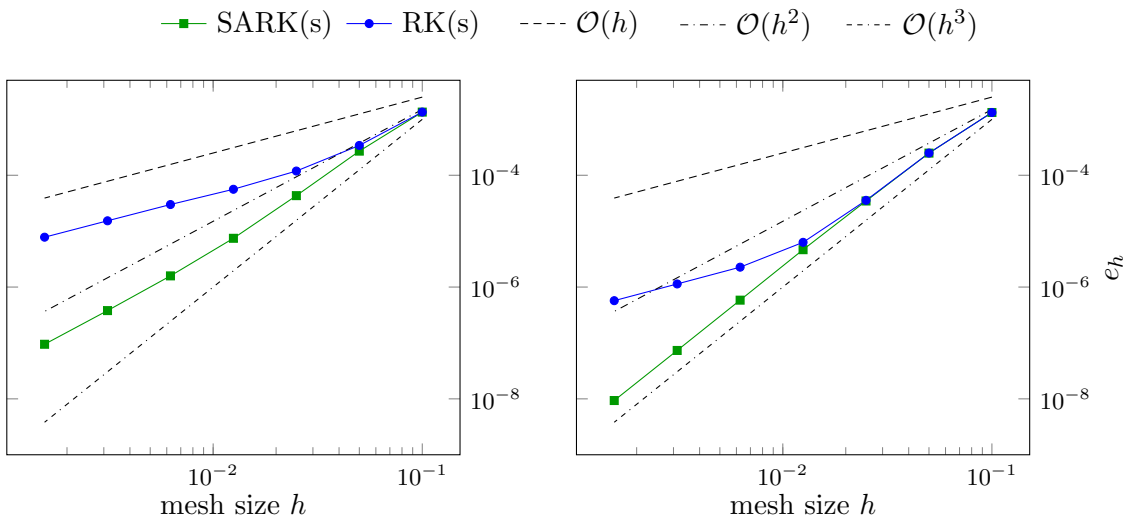
Mach 3 wind tunnel

We conclude with the well-known benchmark example [36] of the wind tunnel with a forward-facing step onto which gas flows at Mach 3, which was also discussed in [13, 14]. The situation is modeled by the already described Euler system (6.12), but now with the initial values

$$\rho_0 = 1.4, \qquad m_0 = \rho_0 (3, 0)^\intercal, \qquad P_0 = 1, \tag{6.15}$$

on a spatial domain Ω_0 with a re-entrant corner at the edge of the forward-facing step. The boundary conditions are set such that $(0, x_2)$ is an inflow boundary and $(3, x_2)$ is a free boundary, which has no effect on the flow. All other boundaries are solid walls. The domain and the boundary conditions are illustrated in Figure 6.8a.

Our numerical experience with this problem shows that it is beneficial to use high order local time stepping. As in our prior study [13], we use a spatially refined mesh near the re-entrant corner and let the tents adapt – see Figure 6.8b, providing automatic local time



(a) Convergence rates obtained from SARK(2, Ralston) method (see Table 4.1b) and the standard Ralston method.

(b) Convergence rates obtained from SARK(3, Heun) method (see Table 4.2b) and the standard Heun scheme.

Figure 6.7: Error e_h as defined in (6.14) over mesh size h for SARK and standard Runge-Kutta (RK) methods applied to the Euler equations (6.12) with the initial data (6.13).

stepping. In contrast to the standard time stepping used in [13], we now use one of the newly proposed SARK schemes.

We shall apply the SARK(3, Heun) method. Unlike in the previous convergence study, now we must handle multiple shocks that develop over time, so it is necessary to add some stabilization to the system. This is done by adding artificial viscosity based on the entropy residual as suggested by [17]. For computational convenience, we use a slight variation of the entropy viscosity regularization described in §6.2.1. Namely, the entropy viscosity coefficient on one element $T \in \mathcal{T}_\mathrm{v}$ is set by

$$\nu_{e,T} = c_X^2 \|R_h\|_{L^\infty(T)},$$

and the limiting artificial viscosity is set by

$$\nu_{*,T} = \kappa_2 \operatorname{diam}(T) \left\| \|m\| + \rho\sqrt{\gamma T} \right\|_{L^\infty(T)},$$

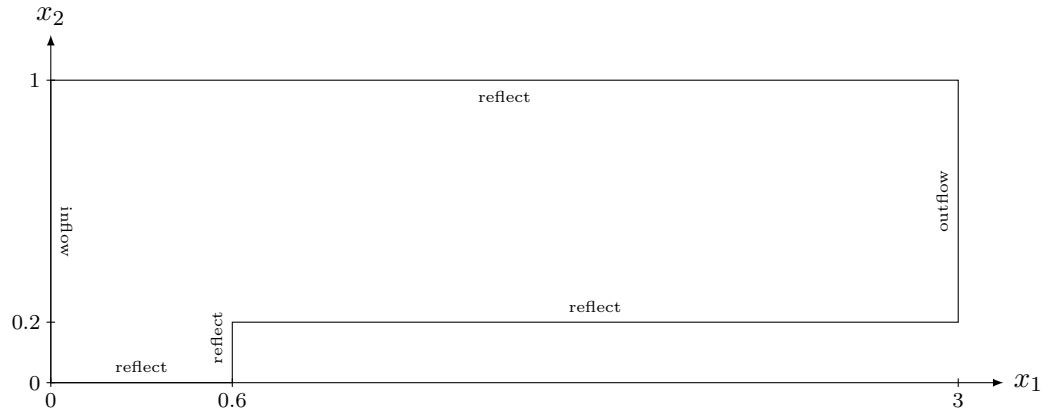
with $\gamma = \frac{d+2}{d} = 1.4$ for an ideal gas and the temperature T. The constants in the calculation of the entropy viscosity coefficient were chosen as $\kappa_1 = 1$, $\kappa_2 = \frac{1}{20}$ and the penalization parameter α in the artificial viscosity term is set to 2.

A kinetic flux (see [23]) was used for the numerical flux f_n in (3.26) while $\mathcal{F}_n$ was set by

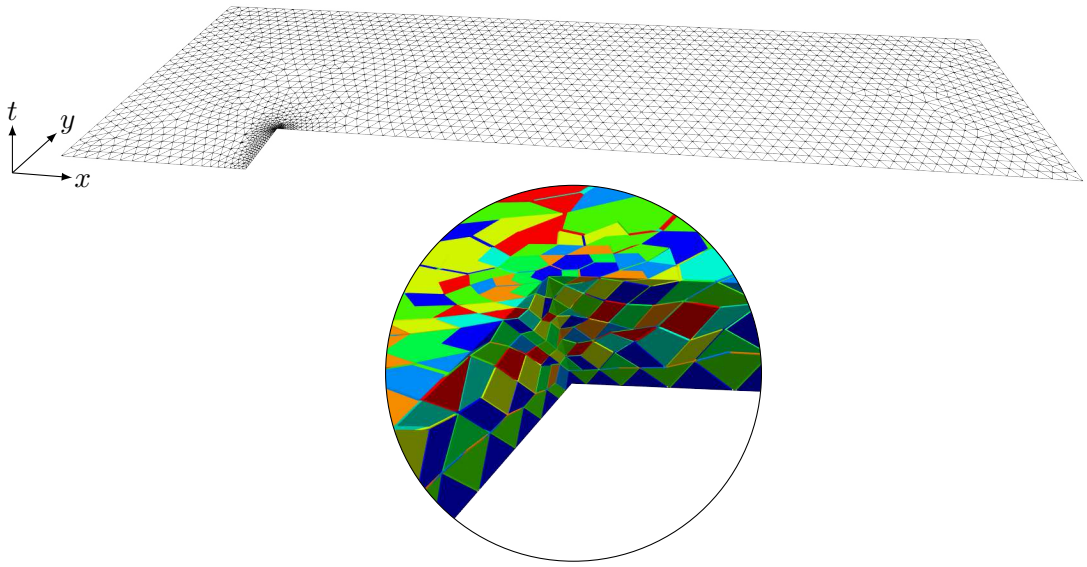
$$\mathcal{F}_n = \begin{cases} \mathcal{F}(\hat{\rho}^+, \hat{m}^+, \hat{E}^+) \cdot n, & \hat{m}^+ \cdot n \geq 0, \\ \mathcal{F}(\hat{\rho}^-, \hat{m}^-, \hat{E}^-) \cdot n, & \text{otherwise}, \end{cases}$$

where $\hat{\rho}^+$, $\hat{m}^+$ and $\hat{E}^+$ denote the traces of $\hat{\rho}$, $\hat{m}$ and $\hat{E}$, respectively, from within the element which has n as outward unit normal vector.

The logarithmic density of the computed solution is shown in Figure 6.9. This was generated with polynomial degree $p = 4$ in space, maximal characteristic speed $\bar{c} = 10$ and $r = 16$ substeps within each tent. Figure 6.8b shows the spatial mesh with the locally refined corner. The zoom in illustrates the local refinement of the tents which comes in naturally through the causality constraint while pitching the tents. The solution component (logarithmic density) shown in Figure 6.9a is comparable with the solution we previously obtained using standard methods in [13], but now due to the higher accuracy of the new SARK time integration, we obtained a similar quality solution faster (with the overall simulation time on the same processor reduced by a factor of 10). We also observed that the entropy residuals calculated off the computed solution with SARK schemes led to a significantly reduced addition of artificial viscosity. The artificial viscosity coefficients generated by the entropy residual are shown in Figure 6.9b, which is about half the size of what is shown in the corresponding plot in our earlier work [13, Figure 5].

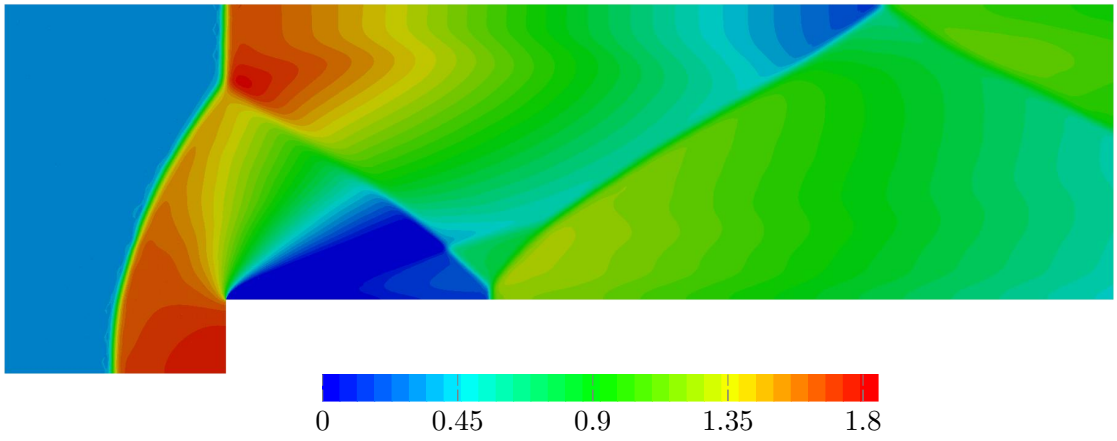


(a) Geometry and boundary conditions

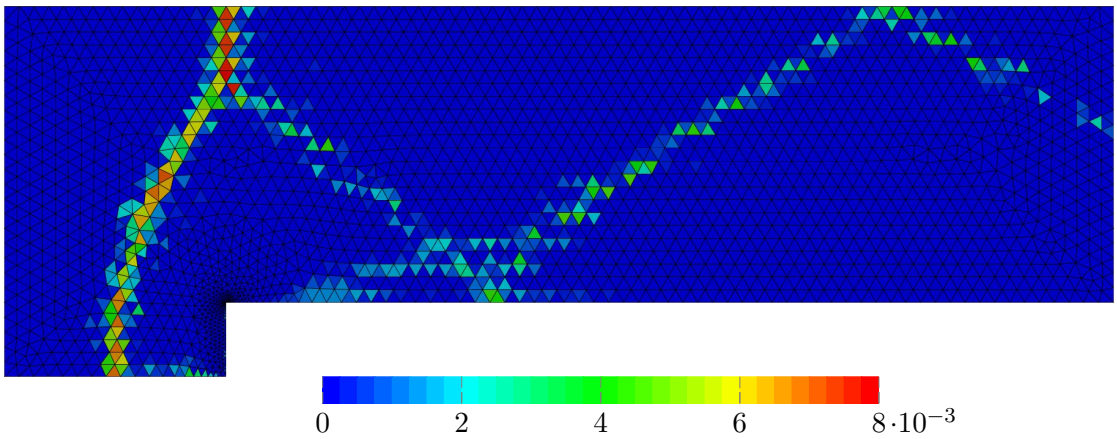


(b) Locally refined spatial mesh (top) used for the Mach 3 wind tunnel example and a zoomed in view of the spacetime tents at the refined corner showing the automatic local time stepping. (In the spacetime figure, vertical direction represents time.)

Figure 6.8: Geometry, mesh and tents of the Mach 3 wind tunnel.



(a) Logarithmic density



(b) Entropy viscosity coefficient ν

Figure 6.9: Solution of the Mach 3 wind tunnel with a forward-facing step at the final time $t_{\max} = 4$ solved on 4128 triangles with the SARK(3, Heun) scheme and spatial polynomials of degree $p = 4$.

Bibliography

[1] R. Abedi and R. B. Haber. Spacetime simulation of dynamic fracture with crack closure and frictional sliding. *Advanced Modeling and Simulation in Engineering Sciences*, 5(1):1–22, 2018.

[2] B. Bahmani and R. Abedi. Asynchronous spacetime discontinuous galerkin formulation for a hyperbolic time-delay bulk damage model. *Journal of Engineering Mechanics*, 145(10):04019075, 2019.

[3] P. G. Ciarlet. *The finite element method for elliptic problems*. North-Holland Publishing Co., Amsterdam-New York-Oxford, 1978. Studies in Mathematics and its Applications, Vol. 4.

[4] C. M. Dafermos. *Hyperbolic conservation laws in continuum physics*, volume 325 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, third edition, 2010.

[5] J. Diaz and M. J. Grote. Energy conserving explicit local time stepping for second-order wave equations. *SIAM J. Sci. Comput.*, 31(3):1985–2014, 2009.

[6] J. Erickson, D. Guoy, J. M. Sullivan, and A. Üngör. Building spacetime meshes over arbitrary spatial domains. *Engineering with Computers*, 20(4):342–353, 2005.

[7] A. Ern and J. L. Guermond. Discontinuous galerkin methods for friedrichs' systems. i. general theory. *SIAM Journal on Numerical Analysis*, 44(2):753–778, 2006.

[8] R. S. Falk and G. R. Richter. Explicit finite element methods for symmetric hyperbolic equations. *SIAM J. Numer. Anal.*, 36(3):935–952, 1999.

[9] K. O. Friedrichs. Symmetric positive linear differential equations. *Communications on Pure and Applied Mathematics*, 11(3):333–418, 1958.

[10] M. J. Gander and L. Halpern. Techniques for locally adaptive time stepping developed over the last two decades. In R. Bank, M. Holst, O. Widlund, and J. Xu, editors, *Domain Decomposition Methods in Science and Engineering XX*, pages 377–385, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[11] J. Gopalakrishnan, M. Hochsteger, J. Schöberl, and C. Wintersteiger. An explicit mapped tent pitching scheme for maxwell equations. In S. J. Sherwin, D. Moxey, J. Peiró, P. E. Vincent, and C. Schwab, editors, *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2018*, pages 359–369. Springer International Publishing, 2020.

[12] J. Gopalakrishnan, P. Monk, and P. Sepúlveda. A tent pitching scheme motivated by Friedrichs theory. *Comput. Math. Appl.*, 70(5):1114–1135, 2015.

[13] J. Gopalakrishnan, J. Schöberl, and C. Wintersteiger. Mapped tent pitching schemes for hyperbolic systems. *SIAM J. Sci. Comput.*, 39(6):B1043–B1063, 2017.

[14] J. Gopalakrishnan, J. Schöberl, and C. Wintersteiger. Structure aware Runge–Kutta time stepping for spacetime tents. *SN Partial Differential Equations and Applications*, 1(4):19, 2020.

[15] M. J. Grote, M. Mehlin, and T. Mitkova. Runge-Kutta-based explicit local time-stepping methods for wave propagation. *SIAM J. Sci. Comput.*, 37(2):A747–A775, 2015.

[16] M. J. Grote, M. Mehlin, and S. A. Sauter. Convergence analysis of energy conserving explicit local time-stepping methods for the wave equation. *SIAM J. Numer. Anal.*, 56(2):994–1021, 2018.

[17] J.-L. Guermond, R. Pasquetti, and B. Popov. Entropy viscosity method for nonlinear conservation laws. *J. Comput. Phys.*, 230(11):4248–4267, 2011.

[18] J.-L. Guermond and B. Popov. Fast estimation from above of the maximum wave speed in the Riemann problem for the Euler equations. *J. Comput. Phys.*, 321:908–926, 2016.

[19] E. Hairer, S. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Non-stiff Problems*, volume 8 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2008.

[20] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2010. Second revised edition.

[21] J. S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods*, volume 54 of *Texts in Applied Mathematics*. Springer, New York, 2008. Algorithms, analysis, and applications.

[22] M. Hochbruck, T. Pažur, A. Schulz, E. Thawinan, and C. Wieners. Efficient time integration for discontinuous Galerkin approximations of linear wave equations [Plenary lecture presented at the 83rd Annual GAMM Conference, Darmstadt, 26th–30th March, 2012]. *ZAMM Z. Angew. Math. Mech.*, 95(3):237–259, 2015.

[23] J. Mandal and S. Deshpande. Kinetic flux vector splitting for euler equations. *Computers & Fluids*, 23(2):447 – 478, 1994.

[24] S. T. Miller and R. B. Haber. A spacetime discontinuous Galerkin method for hyperbolic heat conduction. *Comput. Methods Appl. Mech. Engrg.*, 198(2):194–209, 2008.

[25] P. Monk and G. R. Richter. A discontinuous Galerkin method for linear symmetric hyperbolic systems in inhomogeneous media. *J. Sci. Comput.*, 22/23:443–477, 2005.

[26] M. Neumüller. *Space-Time Methods: Fast Solvers and Applications*. PhD thesis, Graz University of Technology, 2013.

[27] J. Palaniappan, R. B. Haber, and R. L. Jerrard. A spacetime discontinuous Galerkin method for scalar conservation laws. *Comput. Methods Appl. Mech. Engrg.*, 193(33-35):3607–3631, 2004.

[28] I. Perugia, J. Schöberl, P. Stocker, and C. Wintersteiger. Tent pitching and Trefftz-DG method for the acoustic wave equation. *Comput. Math. Appl.*, 79(10):2987–3000, 2020.

[29] G. R. Richter. An explicit finite element method for the wave equation. volume 16, pages 65–80. 1994. A Festschrift to honor Professor Robert Vichnevetsky on his 65th birthday.

[30] J. Schöberl. Netgen an advancing front 2d/3d-mesh generator based on abstract rules. *Computing and Visualization in Science*, 1(1):41–52, 1997.

[31] J. Schöberl. C++11 implementation of finite elements in ngsolve. *Institute for Analysis and Scientific Computing, Vienna University of Technology*, 2014.

[32] D. Serre. *Systems of conservation laws. 1.* Cambridge University Press, Cambridge, 1999. Hyperbolicity, entropies, shock waves, Translated from the 1996 French original by I. N. Sneddon.

[33] A. Üngör and A. Sheffer. Pitching tents in space-time: mesh generation for discontinuous Galerkin method. volume 13, pages 201–221. 2002. Volume and surface triangulations.

[34] J. J. W. van der Vegt and H. van der Ven. Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows. I. General formulation. *J. Comput. Phys.*, 182(2):546–585, 2002.

[35] L. Wang and P.-O. Persson. A high-order discontinuous Galerkin method with unstructured space-time meshes for two-dimensional compressible flows on domains with large deformations. *Comput. & Fluids*, 118:53–68, 2015.

[36] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.*, 54(1):115–173, 1984.

[37] L. Yin, A. Acharya, N. Sobh, R. B. Haber, and D. A. Tortorelli. A space-time discontinuous Galerkin method for elastodynamic analysis. In *Discontinuous Galerkin methods (Newport, RI, 1999)*, volume 11 of *Lect. Notes Comput. Sci. Eng.*, pages 459–464. Springer, Berlin, 2000.

Curriculum Vitae

Persönliche Daten

Name	**Christoph Wintersteiger**
Geburtsdatum	21.11.1989
Geburtsort	Vöcklabruck
Nationalität	Österreich
Telefon	+43 699 10968450
E-Mail	c.wintersteiger@gmx.at

Ausbildung

seit 04/2016	**Technische Universität Wien**, *Doktoratsstudium der technischen Wissenschaften*, Technische Mathematik
seit 01/2014	**Technische Universität Wien**, *Bachelorstudium*, Technische Physik
01/2014–11/2015	**Technische Universität Wien**, *Masterstudium*, Technische Mathematik
10/2010–01/2014	**Technische Universität Wien**, *Bachelorstudium*, Technische Mathematik
09/2004–06/2009	**HTL Vöcklabruck**, *Matura*, Maschinenbau und Anlagentechnik

---

Wissenschaftliche Publikationen

- J. Gopalakrishnan, J. Schöberl, and C. Wintersteiger. Structure aware Runge–Kutta time stepping for spacetime tents. *SN Partial Differential Equations and Applications*, 1(4):19, 2020
- I. Perugia, J. Schöberl, P. Stocker, and C. Wintersteiger. Tent pitching and Trefftz-DG method for the acoustic wave equation. *Comput. Math. Appl.*, 79(10):2987–3000, 2020

- J. Gopalakrishnan, M. Hochsteger, J. Schöberl, and C. Wintersteiger. An explicit mapped tent pitching scheme for maxwell equations. In S. J. Sherwin, D. Moxey, J. Peiró, P. E. Vincent, and C. Schwab, editors, *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2018*, pages 359–369. Springer International Publishing, 2020
- J. Gopalakrishnan, J. Schöberl, and C. Wintersteiger. Mapped tent pitching schemes for hyperbolic systems. *SIAM J. Sci. Comput.*, 39(6):B1043–B1063, 2017

Wien, am 20. Oktober 2020

Christoph Wintersteiger