Numerical Simulation of Relativistic Laser-Plasma Interaction

HEINRICH HEINE
UNIVERSITÄT
DÜSSELDORF

Inaugural-Dissertation

zur
Erlangung des Doktorgrades der
Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Julia Maria Schweitzer

aus Krefeld

Juni 2008

Mathematisches Institut
der Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Referent: Prof. Dr. Marlis Hochbruck
Koreferent: Prof. Dr. Karl-Heinz Spatschek

Tag der mündlichen Prüfung: 03.07.2008

Numerical Simulation of Relativistic Laser-Plasma Interaction

Abstract

In this thesis, we consider the numerical simulation of problems arising in relativistic laser-plasma physics.

In a short introduction to the physical problem we derive the model equations, which turn out to be nonlinear wave equations and nonlinear Schrödinger equations. In this thesis, we consider exponential integrators of two different types for the solution of these equations.

First we consider Gautschi-type exponential integrators to solve nonlinear wave equations. We present a short overview on the theoretical properties of such methods. Then we detail a one- and two-dimensional implementation for our particular application. To achieve an efficient implementation, we employ physical properties of the solution. In the one-dimensional case, we perform extensive comparisons to a standard method and demonstrate the superior performance of the exponential integrator for this problem. For the two-dimensional case, we consider different geometries and present a parallelized scheme. The means of parallelization are tailored to the problem and the different modifications of the integrator in use. We give some comparisons to a standard method, too. Moreover, we present a physical application, where our code was used to optimize the setup of a plasma lens to focus the laser pulse.

In the second part of this thesis, we propose and analyze exponential Rosenbrock-type integrators for the solution of stiff or oscillatory first order systems of differential equations such as the Schrödinger equation. For these methods, we present a thorough convergence and stability analysis in a semigroup framework and study the influence of perturbations on the method. Moreover, we detail a variable step size implementation employing Krylov subspace techniques to evaluate the matrix functions times some vectors. We present an extensive comparison to other methods used for such problems and demonstrate, that our implementation is competitive. Finally, we solve the nonlinear Schrödinger equation arising in the laser-plasma context with the exponential Rosenbrock-type integrator.

Numerical Simulation of Relativistic Laser-Plasma Interaction

Zusammenfassung

In dieser Arbeit betrachten wir die numerische Lösung von Problemen aus der relativistischen Laser-Plasma-Physik.

In einer kurzen Einleitung in das physikalische Problem leiten wir die Modellgleichungen her. Dabei handelt es sich um nichtlineare Wellengleichungen und nichtlineare Schrödinger-Gleichungen. Zu deren Lösung stellen wir zwei verschiedene Typen von exponentiellen Integratoren vor.

Im ersten Teil der Arbeit betrachten wir exponentielle Verfahren vom Gautschi-Typ, um nichtlineare Wellengleichungen zu lösen. In einem kurzen Überblick fassen wir die theoretischen Resultate zu diesen Verfahren zusammen. Dann stellen wir eine ein- und zweidimensionale Implementierung für unsere Anwendung im Detail vor. Um eine effiziente Implementierung zu erhalten, haben wir uns physikalische Eigenschaften der Lösung zunutze gemacht. Im eindimensionalen Fall zeigen wir ausführliche Vergleiche mit einem Standardverfahren und damit die Überlegenheit des exponentiellen Verfahrens für diese Anwendung. Im zweidimensionalen Fall betrachten wir verschiedene Koordinatensysteme und passen die Methode den verschiedenen Fällen an. Außerdem zeigen wir, wie man das Programm effizient parallelisieren kann, indem man auch hier die verschiedenen Modifizierungen berücksichtigt. Dieses Programm vergleichen wir ebenfalls mit einem Standardverfahren. Zum Schluss zeigen wir Ergebnisse unseres Programms, mit denen eine Plasmalinse zur Fokussierung eines Laserpulses optimiert wurde.

Im zweiten Teil der Arbeit stellen wir exponentielle Verfahren vom Rosenbrock-Typ vor. Diese kann man zur Lösung von steifen oder oszillatorischen Systemen von Differentialgleichungen erster Ordnung benutzen. Zu diesen gehören unter anderem auch Schrödinger-Gleichungen. Wir geben eine detailierte Konvergenz- und Stabilitätsanalyse in einem theoretischen Rahmen von Halbgruppen an. Zusätzlich wird der Einfluss von Störungen untersucht. Wir zeigen außerdem, wie man diese Methoden mit variabler Schrittweite und Krylov-Verfahren zur Approximation von Matrix-Vektor-Multiplikationen implementiert. Diese Implementierung vergleichen wir ausführlich mit anderen Methoden, die für diese Probleme benutzt werden und zeigen, dass das neue Verfahren konkurenzfähig ist. Zum Schluss lösen wir die Schrödinger-Gleichung aus der physikalischen Anwendung mit dem exponentiellen Rosenbrock-Verfahren.

Contents

Preface

The field of relativistic laser-plasma dynamics is for several reasons very interesting and active at the moment. By shooting a laser on some plasma, physicists gain the possibility to transfer a very high amount of energy from the laser pulse to other forms, e.g. particle acceleration of electrons and ions and x-ray generation. The plasma can also be used as a lens to shape a high energy pulse. For each case there is again a wide range of applications, e.g. in life science or medicine.

However, for these applications, the physics has to be really well understood and controlled. This is the point where numerics can help. It is common in physics to analyze models numerically, since for most problems the model equations are to complicated to be solved analytically. Numerical solutions are used to study experimental setups, which cannot yet be realized. In addition, very expensive experimental setups can be optimized theoretically beforehand, phenomena can often be analyzed more clearly by numerical solutions than by measurements or they can be used as prediction tools for the experimentalist to know where to look at.

Therefore, numerics is an essential part of physical research and thus there is a lot of interest in having good, robust methods, which produce reliable results. It is indispensable to minimize computational time and storage requirements, since realistic problems are typically huge.

A major part of this work results from the close collaboration between numerical mathematics and theoretical physics. For a lot of achievements in the implementation of such methods for real world problems, we use physical properties of the solution. Moreover, the results have to fit the needs of the physicists, who use the codes. Thus the communication was essential in many ways. In this thesis, we will present different numerical methods to efficiently simulate applications from laser-plasma physics. We demonstrate, that there is a long way from a theoretically understood numerical scheme to an implementation, which is efficient for a particular for application.

In addition we present numerical schemes, which potentially can be used to further improve the performance of physical simulations, but up to now have only been subject to theoretical numerical studies.

From a numerical point of view, we recall and broaden the theoretical understanding of different types of exponential integrators, a field of great interest and activity within numerical mathematics. The idea of exponential integrators already exists for quite a while. Gautschi presented the first trigonometric method for wave equations in 1961. For parabolic equations the idea of exponential integrators dates back to the middle of the last century, where Lawson combined the exponential function with Runge–Kutta schemes. Only a bit later Friedli derived the first nonstiff order conditions by using Taylor series expansion. All these methods share the motivation via the variation-of-constants formula as well as the necessity to evaluate matrix functions related to the exponential times some vector, hence their names. The latter also prevented them from being of practical use for a long time. In recent years, there was a lot of progress made in the field of matrix functions and motivated by this, exponential methods returned to researchers' focus. Theoretically, these methods are by now well understood. Now, the challenge is to prove their applicability as well as to identify applications where they are superior to standard methods.

This thesis is organized as follows: in Chapter 1 we outline the physical problem and derive a set of model equations for laser interaction with a plasma in a fluid description. The equations turn out to be a nonlinear wave equation coupled to a driven harmonic oscillator for the plasma response. For the first part of the thesis, these equations will form the main problem. A further reduction of the model leads to a nonlinear Schrödinger equation, which is the basis for the second part.

In Chapter 2, we will recall the theory of the Störmer-Verlet or leap-frog method and Gautschi-type exponential integrators for nonlinear wave equations. In Chapter 3 and 4, we will present a one-dimensional and two-dimensional, respectively, implementation of the Gautschi-type integrator for the nonlinear wave equation stated in Chapter 1. We will also present comparisons with the standard leap-frog method for the same problem.

In Chapter 5 we turn away from the wave equation and look at general stiff or oscillatory first order differential equations. For this we present a Rosenbrock-type exponential integrator. We analyze the convergence and stability of such schemes, derive stiff order conditions and give example schemes. For these we detail a variable step size implementation with a Krylov-subspace technique for the evaluation of matrix functions times some vectors, that are always an essential part of exponential integrators. Finally we apply the Rosenbrock-type exponential integrator to the laser-plasma Schrödinger equation from Chapter 1.

This thesis was supported by the Deutsche Forschungsgemeinschaft through the Transregio-SFB 18, "Relativistic Laser-Plasma Dynamics", sub project B3, "Soliton Formation during Relativistic Laser-Plasma Interaction".

Chapter 1

Physical problem

1.1 Introduction to laser-plasma physics

The interaction of high-power lasers with plasmas, the “fourth state” of matter, is recently a very active research field in physics. This is motivated by a multitude of new effects which could be observed experimentally, such as the generation of higher order harmonics of the fundamental laser frequency, or were predicted theoretically, such as the generation of extremely short pulses. For a proper understanding of these effects physicists rely to a great extend on numerical simulations. Therefore efficient and accurate numerical schemes are important, in particular for further developments such as applications in medicine or life science.

The word “laser” stems from **L**ight **A**mplification by **S**timulated **E**mission of **R**adiation. Light in the form of photons can be absorbed as well as it can be emitted by an atom. In either case an electron is simultaneously transitioning between inner-atomic states, compensating for the energy and momentum of the photon. Usually the emission of a photon happens spontaneously after some time, but it can be stimulated by another photon of the appropriate energy, too. The emitted photon then adopts properties such as frequency, phase, polarization and direction from the original one. This process, which was already predicted by Einstein in 1917, is used nowadays to generate an intense beam of coherent light. The laser frequency is determined by the laser medium, since the differences in the energy levels are unique for the material used. The medium is now “pumped” by an external energy source to create electronic excitation. The photo-induced emission is usually triggered by a seed beam whose number of photons is subsequently increased.

In principle, arbitrarily intense laser beams can be produced this way. The limiting factor, however, is the refractive index of the amplifier medium, which becomes nonlinear at high intensities. The beam begins to focus and thus the nonlinearity increases even more. This leads to intensities which potentially cause damage to the laser system. Historically this happened in particular in the context of amplifying short laser pulses and limited the laser power for almost two decades.

In 1985 Mourou et al. [35] proposed the technique of "**C**hirped **P**ulse **A**mplification". They used the fact, that laser pulses, which are short in time, possess a broad spectrum. For example using the dispersion of a set of gratings, different components of the spectrum are reflected at different angles. Thus it is possible to construct an optical setup, where the low frequency components travel a shorter way than the high frequency components. In this way, the pulse can be stretched in time by several orders of magnitude. This causes the pulse to be chirped, i.e. the spectral components of the pulse are spread along the optical axis. Thus the light intensity is significantly reduced, but the total pulse energy remains constant. Now it is possible to further amplify the pulse, since the spectral components pass an amplifier at different times and the peak intensity is much lower. After that, the pulse is recompressed in time by using another pair of gratings. Employing the CPA-technique allowed for a rapid progress in laser power.

Whenever a powerful laser pulse is focused onto matter, a plasma is formed by field ionization. In this state the electrons are no longer bound to the ions, but both kinds of particles can move freely. Since they are distributed homogeneously the plasma itself is quasi-neutral. However, the ions are much heavier and thus react much slower and to a smaller extend to external forces such as the electric field of the laser beam. They can be assumed to form a stationary background. The electrons are lighter and thus start to move earlier. Whenever a charge separation is created inside the plasma, an electrostatic restoring force is created and the electrons oscillate around the charge neutral position. The frequency of this oscillation is called the plasma frequency, which depends on the electron mass and scales with the electron number density. The light electrons respond to the part of the Lorentz force caused by the electric field of the laser pulse and start oscillating perpendicularly to the propagation direction of the laser pulse. This causes a local charge separation. If the laser frequency is higher than the plasma frequency, the electrons oscillate in the electric field of the laser pulse and thus allow the pulse to propagate inside the plasma. In the other case, the electrons can short-circuit the electric field of the laser pulse and the light is reflected at the vacuum-plasma boundary.

Since the plasma frequency is essentially coupled to the electron density, a characteristic density is determined by the equality of laser and plasma frequency, above which the laser cannot penetrate the plasma any longer. This density is called the "critical density". In practice "under-dense" or "over-dense" plasmas are realized by choosing different targets. Solid targets with many electrons ($\sim 10^{22}$ electrons per cubic centimeter) such as an aluminum foil lead to reflection of the laser pulse while gas targets ($\sim 10^{18} - 10^{20}$ electrons per cubic centimeter) allow laser propagation inside the plasma.

The interaction of the laser with the plasma is called "relativistic", if the speed of the electrons oscillating in the electric field of the laser pulse approaches a sizable fraction of the speed of light. The interaction then becomes nonlinear due to the relativistic mass increase of the electrons. The plasma frequency is altered by the mass change and thus is coupled to the intensity of the laser. In addition, the magnetic part of the Lorentz force gains importance. It is now strong enough to turn the momentum of the oscillating electrons into the direction of the laser propagation. Huge electron currents are produced which lead to enormous electric fields in laser propagation direction due to charge separation. In this regime high energy electron beams were observed experimentally during laser-gas interaction (laser wake-field acceleration, [5, 15]). Also ion beams could be produced by laser interaction with a thin metal foil (target normal sheath acceleration, [14, 32, 6]), since forces eventually become strong enough to accelerate heavy ions. The general aim of these experiments is to use the plasma to access and transfer a significant part of the energy stored in the laser pulse in order to generate intense beams of secondary particles such as electrons or ions. These new accelerator techniques are a key topic of recent research.

The pulse energy of a laser pulse traveling through under-dense plasma in the weakly relativistic regime, however, is almost conserved. This situation is ideally suited to use the laser-plasma interaction to shape the laser pulse itself in space and time. From recent theoretical studies it is known that a weakly relativistic pulse can be compressed longitudinally and focused transversally inside the plasma, see [33, 31]. Even after leaving the plasma layer, the pulse is still focusing in transversal direction. Combining these two effects, a plasma can be used as an optical device to gain very short focused pulses with very high intensities substituting conventional lenses which can be damaged by the intensities considered here. This scenario is challenging to study experimentally and up to now has only been described theoretically. For the theoretical studies of instabilities, their control and the optimization of the compression by stratified plasma-vacuum formations the numerical methods discussed in Chapter 3 and 4 were used, see [23]. An example for the pulse compression is given in Fig. 1.1, where a pulse propagating through a plasma layer and a stretch of vacuum is shown at different times. In Section 4.2.4 we explain this in more detail.

1.2 Hydro-dynamic model

We consider a high-frequency, cold-electron fluid-Maxwell model consisting of the Maxwell equations combined with a continuity and a momentum equation for the particles in the

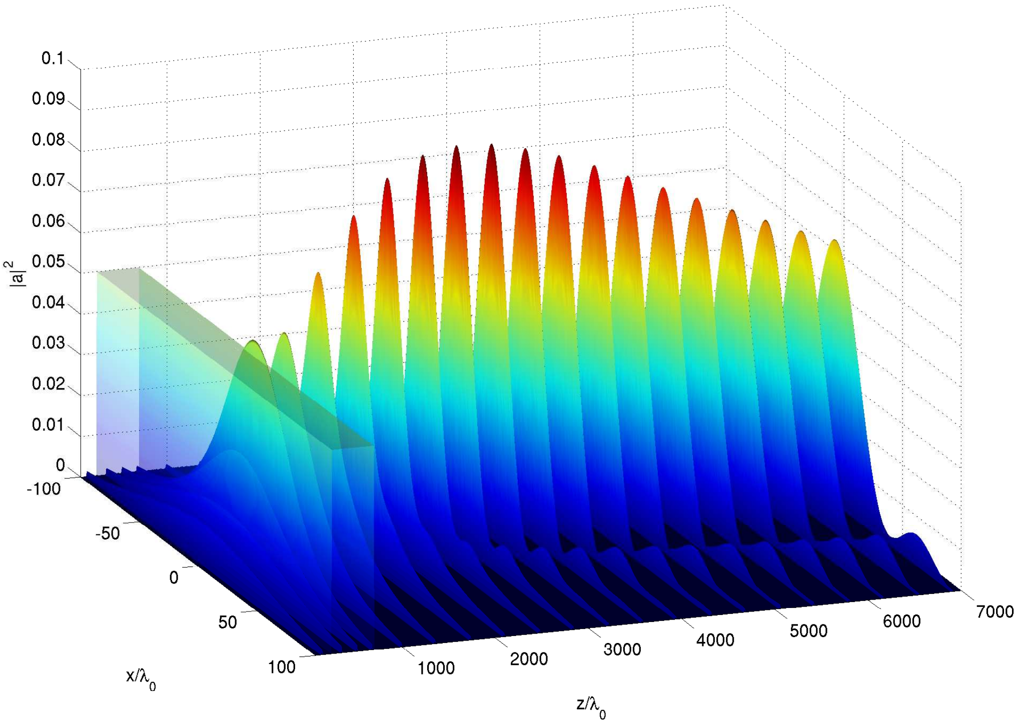


Figure 1.1: The squared amplitude of the same pulse at different times is shown. The pulse travels through a plasma layer and a stretch of vacuum and is compressed.

plasma. The Maxwell equations in cgs units are given by

$$\begin{aligned}
\nabla \cdot \mathbf{E} &= 4\pi\rho \\
\nabla \cdot \mathbf{B} &= 0 \\
\nabla \times \mathbf{E} &= -\frac{1}{c}\frac{\partial}{\partial t}\mathbf{B} \\
\nabla \times \mathbf{B} &= \frac{4\pi}{c}\mathbf{j} + \frac{1}{c}\frac{\partial}{\partial t}\mathbf{E}\,,
\end{aligned}$$

where $\mathbf{E}$ is the electric and $\mathbf{B}$ the magnetic field, ρ is the charge density, $\mathbf{j}$ is the current density and c is the speed of light in vacuum.

Within the fluid-dynamical description of a plasma consisting of different particle species s with charge q_s and mass m_s, the charge density is given by

$$\rho = \sum_s q_s n_s\,,$$

where n_s is the particle density of the species s. The current density takes the form

$$\mathbf{j} = \sum_s q_s n_s \mathbf{v}_s \,.$$

In the relativistic case we have the following relation between the velocity $\mathbf{v_s}$ and the momentum $\mathbf{p_s}$ of the particles of species s:

$$\mathbf{v}_s = \frac{\mathbf{p}_s}{m_s\gamma_s}\,, \quad \gamma_s^2 = 1 + \frac{1}{(m_s c)^2}\|\mathbf{p}_s\|^2 \,.$$

The continuity equation for the different species of particles reads

$$\frac{\partial}{\partial t} n_s + \nabla \cdot \left(\frac{n_s}{m_s\gamma_s}\mathbf{p}_s\right) = 0\,,$$

and the momentum balance is given by

$$\frac{d}{dt}\mathbf{p}_s(\mathbf{x},t) = \frac{\partial}{\partial t}\mathbf{p}_s + (\mathbf{v}_s \cdot \nabla)\mathbf{p}_s = q_s\Big(\mathbf{E} + \frac{1}{c}\mathbf{v}_s \times \mathbf{B}\Big)$$

employing the Lorentz force on the right hand side.

It is common to rescale the physical quantities in characteristic units given by the problem. This results in dimensionless equations. In our case, such units are given by the laser. We use the carrier frequency ω_0 and wave number $k_0 = \omega_0/c$ of the laser pulse in vacuum to normalize the time and space coordinate respectively,

$$\widetilde{\mathbf{r}} = \frac{\omega_0}{c}\mathbf{r}\,, \quad \widetilde{t} = \omega_0 t\,.$$

Here $\mathbf{r}$ is a vector containing the space coordinates for three dimensions. With this normalization, the laser wave-length in vacuum is $\lambda_0 = 2\pi$, the duration of a laser cycle in vacuum is $1/\nu_0 = 2\pi$, too, and the speed of light is $c = 1$.

For the plasma charges and masses are measured in terms of the (positive) elementary charge e and the electron mass m respectively. The particle densities are scaled with the maximum of the electron particle density of the initial plasma n_0. For the electric and magnetic field this results in a scaling factor $\omega_0 mc/e$ and for the momentum the factor is mc.

Using the scalings and applying the chain rule to the derivatives, we can write down the equations in dimensionless form, which we will use from now on. For simplicity, the dimensionless quantities are denoted in the same way as the original ones. The Maxwell equations in dimensionless form are given by

$$\nabla \cdot \mathbf{E} = Q\rho \tag{1.1}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{1.2}$$

$$\nabla \times \mathbf{E} = -\frac{\partial}{\partial t}\mathbf{B} \tag{1.3}$$

$$\nabla \times \mathbf{B} = Q\mathbf{j} + \frac{\partial}{\partial t}\mathbf{E}\,, \tag{1.4}$$

whereas the plasma equations turn into

$$\frac{\partial}{\partial t}n_s + \nabla \cdot \Big(\frac{1}{m_s\gamma_s}n_s\mathbf{p}_s\Big) = 0 \tag{1.5}$$

$$\frac{\partial}{\partial t}\mathbf{p}_s + \frac{1}{m_s\gamma_s}(\mathbf{p}_s \cdot \nabla)\mathbf{p}_s = q_s\Big(\mathbf{E} + \frac{1}{m_s\gamma_s}\mathbf{p}_s \times \mathbf{B}\Big) \tag{1.6}$$

with

$$\rho = \sum_s q_s n_s\,, \quad \mathbf{j} = \sum_s q_s n_s \frac{1}{m_s\gamma_s}\mathbf{p}_s \quad \text{and} \quad \gamma_s^2 = 1 + \frac{\|\mathbf{p}_s\|^2}{m_s^2}\,.$$

The constant Q, that appears in several of the equations reads

$$Q = \frac{4\pi e^2 n_0}{m\omega_0^2} = \frac{\omega_p^2}{\omega_0^2} = \frac{n_0}{n_c}$$

with the electron plasma frequency ω_p. It determines the ratio between the maximum plasma density n_0 and the critical density n_c, above which the laser cannot penetrate the plasma. Thus for under-dense plasmas, which allow the laser to propagation inside, $Q \in [0, 1)$. For the plasma lens application and thus for our numerical investigations we use $Q = 0.3$ to avoid Raman instability, which occurs for $Q < 0.25$.

1.3 Klein-Gordon equation

1.3.1 Vector and scalar potentials

The most general solution of equation (1.2) is given by

$$\mathbf{B} = \nabla \times \mathbf{A} \tag{1.7}$$

for a vector potential $\mathbf{A}$. For uniqueness, we choose $\mathbf{A}$ to fulfill the Coulomb gauge $\nabla \cdot \mathbf{A} = 0$. Inserting this ansatz into equation (1.3) gives

$$\nabla \times \Big(\mathbf{E} + \frac{\partial}{\partial t}\mathbf{A}\Big) = 0\,,$$

which also possesses a general solution given by a scalar potential φ,

$$\mathbf{E} + \frac{\partial}{\partial t}\mathbf{A} = -\nabla\,\varphi\,,$$

which is equivalent to

$$\mathbf{E} = -\nabla\,\varphi - \frac{\partial}{\partial t}\mathbf{A}\,. \tag{1.8}$$

Thus by equations (1.7) and (1.8) the magentical and electric field can both be expressed in terms of the scalar and vector potential φ and $\mathbf{A}$ and it is sufficient, to solve equations for them instead of solving the Maxwell equations.

Inserting equations (1.7) and (1.8) into (1.4) we obtain

$$\nabla \times (\nabla \times \mathbf{A}) = Q\mathbf{j} - \frac{\partial^2}{\partial t^2}\mathbf{A} - \frac{\partial}{\partial t}\nabla\,\varphi\,.$$

Using the Coulomb gauge we get

$$\nabla \times (\nabla \times \mathbf{A}) = \nabla\,(\nabla\,\cdot\,\mathbf{A}) - (\nabla\,\cdot\,\nabla)\mathbf{A} = -\Delta\,\mathbf{A}$$

for the left hand of the equation. Inserting the current density into the right hand side we obtain a wave equation,

$$\frac{\partial^2}{\partial t^2}\mathbf{A} - \Delta\,\mathbf{A} = Q\sum_s q_s n_s \frac{1}{m_s\gamma_s}\mathbf{p}_s - \frac{\partial}{\partial t}\nabla\,\varphi\,. \tag{1.9}$$

From equation (1.1) and the Coulomb gauge we get

$$-\Delta\,\varphi = Q\rho\,. \tag{1.10}$$

The momentum equation (1.6) remains to be adjusted to the potentials. For this purpose, we use the following two identities

$$(\mathbf{p}_s \cdot \nabla)\mathbf{p}_s = \frac{1}{2}\nabla\,\|\mathbf{p}_s\|^2 - \mathbf{p}_s \times (\nabla \times \mathbf{p}_s)$$

and

$$\nabla\,\gamma_s = \frac{1}{2\gamma_s}\frac{1}{m_s^2}\nabla\,\|\mathbf{p}_s\|^2\,.$$

to obtain

$$\frac{\partial}{\partial t}\mathbf{p}_s + \nabla\, m_s\gamma_s = q_s\mathbf{E} + \frac{1}{m_s\gamma_s}\mathbf{p}_s \times \Big(q_s\mathbf{B} + \nabla \times \mathbf{p}_s\Big)$$

from (1.6). Inserting the formulas for $\mathbf{B}$ and $\mathbf{E}$ and reordering terms then yields

$$\frac{\partial}{\partial t}(\mathbf{p}_s + q_s\mathbf{A}) = \nabla\,(-q_s\varphi - m_s\gamma_s) + \frac{1}{m_s\gamma_s}\mathbf{p}_s \times \Big(\nabla \times (\mathbf{p}_s + q_s\mathbf{A})\Big)\,. \tag{1.11}$$

The continuity equation (1.5) remains unchanged.

1.3.2 Reduction of the equation

First of all, we restrict ourselves to a quasi-neutral plasma consisting of electrons ($s = e$) and the same number of single-positively charged ions ($s = i$) only. Thus, the initial density profiles for electrons and ions are the same. For the charges q_s we obtain $q_e = -1$ and $q_i = 1$. For the masses m_s we get $m_e = 1$, but the scaled ion mass m_i/m is a large number, since ions are much heavier than electrons. For example, the lightest ions, which are just protons (nuclei of hydrogen), the mass ratio is approximately 1800. Since we consider the case of laser propagation in under-dense plasma, the laser-plasma interaction happens on a time scale where only the light electrons can be moved. The ions are to heavy to be accelerated on this time scale. Thus, we assume the ions to remain at rest in our model and solve only the continuity and momentum equation for the electrons. This yields $\mathbf{p}_i \equiv 0$ and $n_i(\mathbf{r}, t) = n_i(\mathbf{r}, 0) =: n_i(\mathbf{r})$. For simplicity, we write $\mathbf{p}_e = \mathbf{p}$, $\gamma_e^2 = \gamma^2 = 1 + \|\mathbf{p}\|^2$ and $n_e(\mathbf{r}, t) = n(\mathbf{r}, t) = n_i(\mathbf{r}) + \delta n(\mathbf{r}, t)$. Inserting all this into the equations, we obtain

$$\frac{\partial^2}{\partial t^2}\mathbf{A} - \Delta\,\mathbf{A} = -Q\frac{n_i + \delta n}{\gamma}\mathbf{p} - \frac{\partial}{\partial t}\nabla\,\varphi \tag{1.12}$$

$$\Delta\,\varphi = Q\delta n \tag{1.13}$$

$$\frac{\partial}{\partial t}\delta n + \nabla\,\cdot\,\Big(\frac{n}{\gamma}\mathbf{p}\Big) = 0 \tag{1.14}$$

$$\frac{\partial}{\partial t}(\mathbf{p} - \mathbf{A}) = \nabla\,(\varphi - \gamma) + \frac{1}{\gamma}\mathbf{p} \times \Big(\nabla \times (\mathbf{p} - \mathbf{A})\Big)\,. \tag{1.15}$$

To further simplify the equations, we split the vector fields $\mathbf{u} = \mathbf{u}_{cf} + \mathbf{u}_{df}$ into a curl-free part $\mathbf{u}_{cf}$ and a divergence-free part $\mathbf{u}_{df}$. Therefore we define projection operators Π_{cf} and Π_{df} with the following properties;

$$\begin{aligned}\Pi_{cf}\mathbf{u} \equiv \mathbf{u}_{cf}\,, &\quad \nabla \times \mathbf{u}_{cf} = 0\,,\\ \Pi_{df}\mathbf{u} \equiv \mathbf{u}_{df}\,, &\quad \nabla\,\cdot\,\mathbf{u}_{df} = 0\end{aligned}$$

and

$$\Pi_{cf} + \Pi_{df} = \mathbf{1}\,.$$

Clearly, $\mathbf{u}_{cf}$ is a gradient field, and $\mathbf{u}_{df}$ is a curl field. The operators can be represented as

$$\Pi_{cf} = \nabla\,\Delta^{-1}\nabla\,\cdot \quad \text{and} \quad \Pi_{df} = 1 - \nabla\,\Delta^{-1}\nabla\,\cdot\;.$$

Applying the projection operators to the momentum balance (1.15) allows to split the equation into a divergence-free and a curl-free part. The equation

$$\frac{\partial}{\partial t}(\mathbf{p}_{df} - \mathbf{A}) - \Pi_{df}\left(\frac{1}{\gamma}\mathbf{p}\times\Big(\nabla\times(\mathbf{p}_{df} - \mathbf{A})\Big)\right) = 0$$

describes the convective transport of the divergence-free part of the canonical momentum $\mathbf{p}_{\text{can}} = \mathbf{p} - \mathbf{A}$. This implies that for the initial condition $\mathbf{p}_{df} = \mathbf{A}$ the canonical momentum stays curl-free for all times, i.e.

$$\mathbf{p}_{\text{can}} = \mathbf{p}_{df} + \mathbf{p}_{cf} - \mathbf{A} = \mathbf{p}_{cf}\,.$$

This initial condition simplifies the curl-free part to

$$\frac{\partial}{\partial t}\mathbf{p}_{cf} = \nabla\,(\varphi - \gamma)\,. \tag{1.16}$$

$\mathbf{p}_{cf}$ can be written in terms of a scalar potential, $\mathbf{p}_{cf} = \nabla\psi$ and the integration of (1.16) thus leads to

$$\frac{\partial}{\partial t}\psi = \varphi - \gamma + 1\,. \tag{1.17}$$

Applying the splitting via Π_{df} and Π_{cf} to the wave equation (1.12), we obtain for the divergence-free part

$$\frac{\partial^2}{\partial t^2}\mathbf{A} - \Delta\,\mathbf{A} = -Q(1 - \nabla\,\Delta^{-1}\nabla\,\cdot)(\frac{n}{\gamma}(\mathbf{A} + \nabla\,\psi))$$

and for the curl-free part

$$\frac{\partial}{\partial t}\nabla\,\varphi = -Q\nabla\,\Delta^{-1}\nabla\,\cdot\,(\frac{n}{\gamma}(\mathbf{A} + \nabla\,\psi))\,.$$

Straightforward manipulations of the right-hand sides lead to

$$\begin{aligned}\frac{\partial^2}{\partial t^2}\mathbf{A} - \Delta\mathbf{A} &= -Q\left(\frac{n}{\gamma}\mathbf{A} - \Delta^{-1}\Big(\nabla\,(\mathbf{A}\cdot\nabla\,\frac{n}{\gamma}) + \nabla\times\big((\nabla\,\frac{n}{\gamma})\times(\nabla\,\psi)\big)\Big)\right)\\ \frac{\partial}{\partial t}\nabla\,\varphi &= -Q\left(\frac{n}{\gamma}\nabla\,\psi + \Delta^{-1}\Big(\nabla\,(\mathbf{A}\cdot\nabla\,\frac{n}{\gamma}) + \nabla\times\big((\nabla\,\frac{n}{\gamma})\times(\nabla\,\psi)\big)\Big)\right)\,.\end{aligned}$$

The direction of laser propagation always locally distinguishes one direction, the longitudinal direction, from the two transversal directions, thus it is reasonable to look at the parallel direction and the perpendicular direction separately. In the following, the spatial coordinate z always denotes the direction of the laser propagation.

To further simplify the equations in the weakly relativistic regime, we scale the dependence on the perpendicular coordinates with a parameter α and introduce the smallness parameters ε, μ, β, ϱ, and δ for the amplitudes of the physical variables:

$$\begin{aligned}
\mathbf{A}(\mathbf{r},t) &= \varepsilon\mathbf{A}^1(\mathbf{r},t)\varepsilon\big(\mathbf{A}_\perp^1(z,\alpha\mathbf{r}_\perp,t) + \mu\mathbf{e}_z A_\parallel^1(z,\alpha\mathbf{r}_\perp,t)\big)\\
n(\mathbf{r},t) &= n_i + \beta\delta n^1(z,\alpha\mathbf{r}_\perp,t)\\
\varphi(\mathbf{r},t) &= \varrho\varphi^1(z,\alpha\mathbf{r}_\perp,t)\\
\psi(\mathbf{r},t) &= \delta\psi^1(z,\alpha\mathbf{r}_\perp,t)\,.
\end{aligned}$$

The different smallness parameters are of course interrelated. In the following we justify some relations between these parameters. From the Coulomb gauge we get

$$\nabla\,\cdot\,\mathbf{A} = \varepsilon(\alpha\nabla_\perp\cdot\mathbf{A}_\perp^1 + \mu\frac{\partial}{\partial z}A_\parallel^1) = 0$$

and thus $\alpha = \mu$. The Laplace equation for φ yields

$$\varrho\Delta\varphi^1 = Q\beta\delta n^1$$

and therefore ϱ and β are equal. Using the series expansion for γ, the reduced momentum balance reads

$$\begin{aligned}
\delta\frac{\partial}{\partial t}\psi^1 &= -\varrho\varphi^1 - (\gamma - 1)\\
&= -\varrho\varphi^1 - \frac{1}{2}\|\mathbf{A} + \nabla\,\psi\|^2 + \mathcal{O}(\|\mathbf{A} + \nabla\,\psi\|^4)\\
&= -\varrho\varphi^1 - \frac{1}{2}\Big(\varepsilon^2\|\mathbf{A}_\perp^1\|^2 + 2\varepsilon\alpha\delta\mathbf{A}_\perp^1\cdot\nabla_\perp\psi^1 + \alpha^2\delta^2\|\nabla_\perp\psi^1\|^2\\
&\qquad + \varepsilon^2\alpha^2(A_\parallel^1)^2 + 2\varepsilon\alpha\delta A_\parallel^1\frac{\partial}{\partial z}\psi^1 + \delta^2(\frac{\partial}{\partial z}\psi^1)^2\Big) + \text{ h.o.t. }\,.
\end{aligned}$$

Here, the lowest order terms are those of order δ, ϱ and ε^2. If we assume, that the laser pulse is initially the only driving force for plasma oscillations, we get $\varepsilon^2 = \delta = \varrho = \beta$. The amplitude of the laser pulse scales with ε. Since we are interested in the weakly relativistic regime inside the plasma, $\varepsilon \ll 1$. In the parallel direction, the spatial dependence is given by the carrier wavelength of the laser pulse, which is rather short. In transversal direction the spatial dependence is smooth compared to the longitudinal direction. Therefore we assume $\alpha \ll 1$.

Using these relations between the scaling parameters, we can further simplify the equations and still obtain consistent approximations. For this reason, we will now look at the scalings of several terms appearing in the equations: first we expand

$$\begin{aligned}\frac{n}{\gamma} &= (n_i + \varepsilon^2 \delta n^1)(1 - \frac{1}{2}\varepsilon^2 \|\mathbf{A}_\perp^1\|^2) + \text{ h.o.t.} \\ &= n_i + \varepsilon^2(\delta n^1 - \frac{1}{2} n_i \|\mathbf{A}_\perp^1\|^2) + \text{ h.o.t. }\ .\end{aligned}$$

For a sufficiently smooth or piecewise constant density profile we assume $\nabla n_i \equiv 0$ and therefore

$$\nabla \frac{n}{\gamma} = \varepsilon^2(\nabla \delta n^1 - \frac{1}{2} n_i \nabla \|\mathbf{A}_\perp^1\|^2) + \text{ h.o.t. }\ .$$

We then have

$$(\nabla \frac{n}{\gamma}) \times (\nabla \psi) = \mathcal{O}(\varepsilon^4)$$

and using

$$\mathbf{A} \cdot \nabla = \varepsilon(\mathbf{A}_\perp^1 \cdot (\alpha \nabla_\perp) + \alpha A_\parallel^1 \frac{\partial}{\partial z})$$

we get

$$\mathbf{A} \cdot \nabla \frac{n}{\gamma} = \mathcal{O}(\alpha \varepsilon^3)\,.$$

Note, that the inverse Laplacian does not change the order of the dominating terms since

$$\Delta^{-1} = \mathcal{F}^{-1} \frac{1}{k_\parallel^2 + \alpha^2 k_\perp^2} \approx \mathcal{F}^{-1} \frac{1}{k_\parallel^2}\left(1 - \alpha^2 \frac{k_\perp^2}{k_\parallel^2}\right) = \Big(\frac{\partial^2}{\partial z^2}\Big)^{-1} + \mathcal{O}(\alpha^2)\,,$$

where $\mathcal{F}^{-1}$ denotes the inverse Fourier-transform.

Starting with the expansion of the divergence-free part of the wave equation,

$$\begin{aligned}\varepsilon\,\big(\frac{\partial^2}{\partial t^2}\mathbf{A}^1 - \frac{\partial^2}{\partial z^2}\mathbf{A}^1 - \alpha^2 \Delta_\perp \mathbf{A}^1\big) = -\varepsilon\Big(Q(n_i + \varepsilon^2 \delta n^1)\big(1 - \frac{\varepsilon^2}{2}\|\mathbf{A}_\perp^1\|^2\big)\mathbf{A}^1\Big) + \text{ h.o.t.} \\ + Q\Delta^{-1}\Big(\nabla\,(\mathbf{A} \cdot \nabla \frac{n}{\gamma}) + \nabla \times \big((\nabla \frac{n}{\gamma}) \times (\nabla \psi)\big)\Big)\,,\end{aligned}$$

we decide on the order of approximation. We choose the lowest order approximation, that includes nonlinear effects, thus we have to keep terms of order ε^3 or larger. Any higher order terms are neglected, such as $\alpha\varepsilon^3$ and $\alpha^2\varepsilon^4$ as appear in the inverse Laplacian term. It remains to decide whether a one-dimensional model is sufficient or not. This decision is based upon the spatial dependence on the perpendicular direction, namely the size of α. To obtain a one-dimensional model, terms of order $\alpha^2\varepsilon$ are neglected, otherwise they are kept. In the three-dimensional case the equation reads

$$\frac{\partial^2}{\partial t^2}\mathbf{A} - \Delta\,\mathbf{A} = -Q\Big(n_i\big(1 - \frac{1}{2}\|\mathbf{A}_\perp\|^2\big) + \delta n\Big)\mathbf{A}\,.$$

From this we can see immediately, that for initial conditions satisfying $A_{\|} = 0$, $A_{\|}$ stays zero.

Now we expand the continuity equation and apply the same approximations as above.

$$\begin{aligned}
\varepsilon^2 \frac{\partial}{\partial t}\delta n^1 &= -\nabla \cdot \Big(\frac{n}{\gamma}(\mathbf{A} + \nabla \psi)\Big) \\
&= -\frac{n}{\gamma}\Delta \psi - \mathbf{A} \cdot \nabla \frac{n}{\gamma} - \nabla \psi \cdot \nabla \frac{n}{\gamma} \\
&= -\Big(n_i + \varepsilon^2(\delta n^1 - \frac{1}{2}n_i \|\mathbf{A}_\perp^1\|^2) + \text{ h.o.t. }\Big)\varepsilon^2 \Delta \psi^1 + \mathcal{O}(\alpha\varepsilon^3) + \mathcal{O}(\varepsilon^4)
\end{aligned}$$

The only terms left in this equation are those of order ε^2,

$$\frac{\partial}{\partial t}\delta n = -n_i \Delta \psi \,.$$

Differentiating once more with respect to t and inserting (1.17) and (1.13) yields

$$\frac{\partial^2}{\partial t^2}\delta n = -n_i \Delta \frac{\partial}{\partial t}\psi = -n_i \Delta\,(\varphi - \gamma + 1) = -Q n_i \delta n + n_i \Delta\,\gamma \,.$$

Expanding γ and neglecting the higher order terms we get

$$\frac{\partial^2}{\partial t^2}\delta n + Q n_i \delta n = \frac{n_i}{2}\Delta\,\|\mathbf{A}_\perp\|^2 \,.$$

For the curl-free part of the wave equation, the approximations lead to the same equation as above, thus it is omitted.

We now have a consistent system of two coupled equations for the transversal vector potential $\mathbf{A}_\perp$ and the electron density variation δn. But it turned out, that results from simulations involving the complete γ-factor instead of the truncated series expansion only neglecting the dependence on ψ were closer to one-dimensional PIC-simulations [30]. This might be due to large constants involved in the higher order terms from the γ-expansion, which could cause the higher order terms to be more important than others with lower orders but small constants. The resulting equations are still consistent to the order ε^3, but some higher order terms are also kept.

Due to the initial condition satisfying $\mathbf{A}_{\|} = A_z \equiv 0$, we change notation for ease of presentation and implementation,

$$a(\mathbf{r}, t) = A_x(\mathbf{r}, t) + i A_y(\mathbf{r}, t) \,.$$

We end up with a system of two equations, a wave equation for the vector potential,

$$\frac{\partial^2}{\partial t^2}a - \Delta\, a = -Q\frac{n}{\gamma}a \tag{1.18}$$

and the plasma response

$$\frac{\partial^2}{\partial t^2}\delta n + Q n_i \delta n = n_i \Delta\,\gamma \tag{1.19}$$

where

$$\gamma^2 = 1 + |a|^2 \quad \text{and} \quad n = n_i + \delta n\,.$$

These equations are nonlinear and coupled via the relativistic γ-factor and δn. Note that in plasma they hold only in the weakly relativistic regime. In vacuum, however, the density is zero, thus the density equation is obsolete. It remains a linear homogeneous wave equation for the vector potential, which is exact in vacuum.

1.3.3 Different geometries

As mentioned above, depending on the size of the scaling parameter α, we can further simplify the equations by reducing the number of space dimensions. Neglecting all terms of order higher than ε^3 there is only the transversal part of the Laplacian left involving α in its scaling factor. If α is very small, which describes a very weakly focused pulse, where the spot size is much bigger than the longitudinal length of the pulse, we neglect the spatial dependence on the transversal coordinates,

$$\Delta\, = \frac{\partial^2}{\partial z^2}\,.$$

This results in one-dimensional model equations.

If a one-dimensional reduction is too restrictive, we can split α and look at different spatial geometries separately. If, for example, the pulse is only focused in one transversal direction, and shows a very small dependence in the other direction, we can look at different scaling parameters α_1 and α_2 for the two transversal directions and only neglect the smaller one of the parameters. This results in two-dimensional Cartesian coordinates:

$$\Delta\, = \frac{\partial^2}{\partial z^2} + \frac{\partial^2}{\partial x^2}\,.$$

The third case is an almost circularly shaped pulse in transversal direction. Here, we choose cylindrical coordinates and assume the scaling parameter for the angle coordinate to be small enough to be neglected. This leads to

$$\Delta\, = \frac{\partial^2}{\partial z^2} + \frac{1}{r}\frac{\partial}{\partial r}\big(r\frac{\partial}{\partial r}\big)\,.$$

The different geometries are important, since the numerical methods proposed in the following are tailored to the problems and thus a change of geometry also requires a modifications of the method.

1.4 Schrödinger equation

For the one-dimensional case, we consider a further simplification, which leads to a nonlinear Schrödinger equation. This is done in two steps. First, we transform into a comoving frame. Then, we employ a slowly varying envelope approach and neglect derivatives with respect to the slow variable. In this section, we set the density variation to zero.

1.4.1 Comoving frame

To find the right comoving frame, we have to determine the (group) velocity of the pulse. In vacuum, the pulse moves with the speed of light, which is normalized to 1. More generally, the velocity of the pulse is determined by the dispersion relation $\omega^2 = c^2k^2 + \omega_p^2$. The group velocity is then defined as

$$v_{gr} = \frac{\partial}{\partial k}\omega\big|_{k_0} = \frac{1}{2\omega}2c^2k\big|_{k_0} = \frac{c^2k_0}{\omega_0} = \frac{c\sqrt{\omega_0^2 - \omega_p^2}}{\omega_0} = \sqrt{1-Q} = \beta$$

and determines the mean velocity of the wave packet. There may be also faster and slower parts of the pulse, which causes the pulse to change its shape due to dispersion.

We choose the coordinate transformation

$$\vartheta = \frac{z}{\beta} - t$$

to change into the moving system. Inserting this into the wave equation with constant density, we obtain

$$\Big(1 - \frac{1}{\beta^2}\Big)\frac{\partial^2}{\partial\vartheta^2}a - \frac{2}{\beta}\frac{\partial^2}{\partial\vartheta\partial z}a - \frac{\partial^2}{\partial z^2}a = -Q\frac{n_i}{\gamma}a\,. \tag{1.20}$$

1.4.2 Slowly-varying envelope approach

Next, we assume, that the solution possesses a slowly varying amplitude, and that only the real and imaginary components oscillate fast. We define the slowly varying envelope ansatz as

$$a(z,\vartheta) = \widetilde{a}(z,\vartheta)e^{i\vartheta + i(\beta - \frac{1}{\beta})z}$$

and insert this into equation (1.20) to obtain

$$\Big(1 - \frac{1}{\beta^2}\Big)\frac{\partial^2}{\partial\vartheta^2}\widetilde{a} - \frac{2}{\beta}\frac{\partial^2}{\partial\vartheta\partial z}\widetilde{a} - \frac{\partial^2}{\partial z^2}a - 2i\beta\frac{\partial}{\partial z}\widetilde{a} = (\beta^2 - 1)\Big(1 - \frac{n_i}{\gamma}\Big)\widetilde{a}\,.$$

Since this transformation removes the fast dependence of a on z from $\widetilde{a}$ to the exponential term, we can neglect the higher derivatives with respect to z by setting $\partial_\vartheta \partial_z$ and ∂_z^2 to zero. Thus we obtain a nonlinear Schrödinger equation

$$2i\beta \frac{\partial}{\partial z}\widetilde{a} = \Big(1 - \frac{1}{\beta^2}\Big)\frac{\partial^2}{\partial \vartheta^2}\widetilde{a} + (\beta^2 - 1)\Big(\frac{n_i}{\gamma} - 1\Big)\widetilde{a}\,. \tag{1.21}$$

Chapter 2

Numerical integrators for wave equations

In this chapter we will recall numerical schemes to solve nonlinear differential equations of the form

$$y(t)'' = -\Omega^2 y(t) + g\big(y(t)\big) =: f\big(y(t)\big)\,, \quad 0 \le t \le T\,, \quad y_0 = y(t_0)\,, \quad y_0' = y'(t_0) \tag{2.1}$$

with a symmetric positive definite matrix Ω with arbitrary large norm and $\|g\|$, $\|g_y\|$ and $\|g_{yy}\|$ bounded. The set of equations for the laser-plasma interaction derived in the previous chapter leads to such a system when discretized in space. For a reasonable physical problem, the total energy is finite, thus we have the physically motivated bound

$$\|y'(t)\|^2 + \|\Omega y(t)\|^2 \le C\,. \tag{2.2}$$

Solving wave equations numerically is a challenge, because their solution is often highly oscillatory possibly in space and in time. For standard explicit schemes the large norm of Ω causes stability problems. To avoid this, the temporal step size has to be chosen roughly as $\tau \le 1/\|\Omega\|$. Moreover, error bounds containing derivatives of the solution are useless for oscillatory problems.

We will first give a short review on the well known Störmer-Verlet or leap-frog method, which is a classical explicit scheme, see [12] and references therein. Then we will recall the construction and properties of Gautschi-type exponential integrators in two different formulations [17, 7, 9, 12].

2.1 Störmer-Verlet leap-frog method

We start with the standard Störmer-Verlet or leap-frog method. Since in physics the name "leap-frog method" is more common, we will only use this name in the following. For a second order differential equation of the form (2.1) we use a second order difference scheme to approximate y'' and thus obtain the leap-frog method

$$y_{k+1} - 2y_k + y_{k-1} = \tau^2 f(y_k)\,. \tag{2.3}$$

This yields approximations

$$y_k \approx y(t_k) \qquad \text{for} \qquad t_k = t_0 + k\tau\,.$$

For the first step we use

$$y_1 = y_0 + \tau y_0' + \frac{\tau^2}{2} f(y_0)\,.$$

If desired, approximations to the derivatives $y_k' \approx y'(t_k)$ can be computed from

$$y_k' = \frac{y_{k+1} - y_{k-1}}{2\tau}\,.$$

There is also an equivalent one-step formulation of the leap-frog method.

This method possesses several desirable properties, such as second order accuracy, symmetry and symplecticity, see references in [12]. Another important property of a numerical scheme is, whether it conserves physical quantities such as the total energy. For highly oscillatory Hamiltonian problems the leap-frog method conserves the total energy up to order τ over long times. Also the scheme is very easy to implement.

However, there are also some problems with this method. The error bound for the leap-frog method is of the form

$$\|y(t_n) - y_n\| \le Ce^{(t_n - t_0)L} \tau^2 \max_{t\in[t_0,t_n]} \|y'''(t)\| \tag{2.4}$$

where L is the Lipschitz constant of f, which causes problems for oscillatory solutions and for large Lipschitz constants $L \sim \|\Omega\|$.

Investigating the method applied to linear problems, we observe a very stringent step size restriction due to stability problems. We have to choose step sizes $\tau\omega \le 2$ where $\omega \sim \|\Omega\|$ is the largest frequency of the linear problem. For spatially discretized wave equations, $\tau\omega$ is the Courant-Friedrichs-Lewy number, which is typically chosen as 1. For the energy conservation usually even $\tau\omega \le 1/2$ is required.

If we assume Ω^2 to approximate the (negative) Laplacian for the wave equation, ω can be very large and thus we have to choose very small step sizes to ensure stability and energy conservation. The constant in the error bound (2.4) is very large, too, see [12].

Gautschi 1961	**Gautschi's method** – two-step – exact for constant g – large errors for certain resonant step sizes τ – energy almost conserved for nonresonant $\tau\omega$
Deuflhard 1979	**Deuflhard's method** – two-step – large errors for certain resonant step sizes τ – energy almost conserved for nonresonant $\tau\omega$
García-Archilla, Sanz-Serna, Skeel 1999	**Mollified Impulse method** – two-step – order 2 independent of Ω – energy almost conserved for nonresonant $\tau\omega$
Hochbruck, Lubich 1999 Grimm 2005	**Gautschi-type exponential integrator** – two-step – order 2 independent of Ω – exact for constant g – energy almost conserved for nonresonant $\tau\omega$
Hairer, Lubich 2000	**Gautschi-type exponential integrator** – one-step – large errors for certain resonant step sizes τ – energy conserved up to order τ
Grimm, Hochbruck 2006	**Gautschi-type exponential integrator** – one-step – order 2 independent of Ω – energy conserved up to order τ

Table 2.1: Time line and people involved in the development of Gautschi-type exponential integrators and the properties of the different versions.

2.2 Gautschi-type exponential integrators

The exact solution of (2.1) is given by the variation-of-constants formula

$$\begin{aligned}\begin{pmatrix} y(t+\tau) \\ y'(t+\tau) \end{pmatrix} &= \begin{pmatrix} \cos(\tau\Omega) & \tau\ \mathrm{sinc}(\tau\Omega) \\ -\Omega\sin(\tau\Omega) & \cos(\tau\Omega) \end{pmatrix} \begin{pmatrix} y(t) \\ y'(t) \end{pmatrix} \\ &\quad + \int_0^\tau \begin{pmatrix} \Omega^{-1}\sin\big((\tau-\sigma)\Omega\big) \\ \cos\big((\tau-\sigma)\Omega\big) \end{pmatrix} g\big(y(t+\sigma)\big)\,\mathrm{d}\sigma \end{aligned} \tag{2.5}$$

applied to the second order problem rewritten as a system of first order differential equations. Gautschi-type exponential integrators are constructed using (2.5) where only the function g, which is assumed to be bounded, is replaced by an approximation. Gautschi first proposed such a method in 1961. It was constructed to integrate (2.1) with constant g exactly. In the following years a lot of people worked on this kind of methods, see Table 2.1.

The methods involve trigonometric functions of the matrix Ω. The applicability of such methods thus depends to a great extent on an efficient computation of matrix functions times some vectors. For our applications, we will detail this in Chapters 3 and 4. For more general problems, a Krylov approximation to trigonometric matrix functions times vectors is proposed in [10].

In this thesis, we only consider wave equations with constant Ω. For problems, where Ω depends on time or on the solution, there is an extension of exponential integrators, see [8].

2.2.1 Two-step formulation

First, we consider the two-step formulation of the Gautschi-type exponential integrator proposed by Hochbruck and Lubich in [17]. For the construction, we use (2.5) for $t_{k\pm 1}$ to see, that the exact solution of (2.1) satisfies

$$\begin{aligned} & y(t_{k+1}) - 2\cos(\tau\Omega)y(t_k) + y(t_{k-1}) \\ & \quad = \int_0^\tau \Omega^{-1}\sin\big((\tau-\sigma)\Omega\big)\Big(g\big(y(t_k+\sigma)\big) + g\big(y(t_k-\sigma)\big)\Big)\,\mathrm{d}\sigma\,. \end{aligned} \tag{2.6}$$

For constant $g(y(t)) \equiv g$ we integrate exactly and obtain Gautschi's method

$$y(t_{k+1}) - 2\cos(\tau\Omega)y(t_k) + y(t_{k-1}) = \tau^2\psi(\tau\Omega)g$$

with

$$\psi(\xi) = 2\int_0^1 \xi^{-1}\sin\big((1-\sigma)\xi\big)\,\mathrm{d}\sigma = 2\frac{1-\cos(\xi)}{\xi^2}\,.$$

For non-constant g Gautschi proposed to approximate $g\big(y(t+\sigma)\big) + g\big(y(t-\sigma)\big) \approx 2g(y_k)$, where $y_k \approx y(t_k)$. However, it turned out, that for certain resonant time steps the error of the method was large. In [17], Hochbruck and Lubich proposed to use $g\big(\varphi(\tau\Omega)y_k\big)$ instead of $g(y_k)$ with a real function φ, which is bounded on the positive real axis and satisfies

$$\varphi(0) = 1\,, \quad \varphi(k^2\pi^2) = 0 \text{ for } k = 1,2,3,\dots \quad \text{and} \quad |\varphi(\xi)| \le 1 \text{ for } \xi \ge 0\,.$$

φ can for example be chosen as

$$\varphi(\xi) = \ \mathrm{sinc}^2(\xi)\Big(1 + \frac{1}{2}\big(1-\cos(\xi)\big)\Big)\,.$$

The scheme then reads

$$y_{k+1} - 2\cos(\tau\Omega)y_k + y_{k-1} = \tau^2\psi(\tau\Omega)g\big(\varphi(\tau\Omega)y_k\big) \tag{2.7}$$

or equivalently

$$y_{k+1} - 2y_k + y_{k-1} = \tau^2 \psi(\tau\Omega)\Big(-\Omega^2 + g\big(\varphi(\tau\Omega)y_k\big)\Big)\,.$$

For the two-step method, a second starting value is computed by

$$y_1 = \cos(\tau\Omega)y_0 + \tau\ \mathrm{sinc}(\tau\Omega)y_0' + \frac{\tau^2}{2}\psi(\tau\Omega)g\big(\varphi(\tau\Omega)y_0\big)$$

and approximations to the derivatives of y can be obtained by

$$y_{k+1}' = y_{k-1}' + 2\tau\ \mathrm{sinc}(\tau\Omega)\Big(-\Omega^2 y_k + g\big(\varphi(\tau\Omega)y_k\big)\Big)\,.$$

Hochbruck and Lubich proved almost second order in [17]. Here, the error bound involved a term, which slowly grows with the number of time steps and the size of the system. In [7], Grimm completed the proof of the following

Theorem 2.1. *The numerical solution obtained by* (2.7) *fulfills the error bound*

$$\|y_k - y(t_k)\| \le Ce^{l(t_k - t_0)}\tau^2$$

where l is the Lipschitz constant of g. The constant C only depends on the norms of g, g_y and g_{yy} and the energy bound (2.2)*, but is independent of derivatives of y, the norm of Ω, k, τ and the size of the system.*

A similar bound

$$\|y_k' - y'(t_k)\| + \big\|\Omega\big(y_k - y(t_k)\big)\big\| \le Ce^{l(t_k - t_0)}\tau$$

holds for the derivatives.

For higher regularity of the solution, such as bounded $\|\Omega^2 y(t)\|$ and $\|\Omega y'(t)\|$, the approximations to the derivatives are of second order accuracy.

2.2.2 One-step formulation

The one-step formulation is directly motivated by the variation-of-constants formula (2.5).

$$\begin{aligned} y_{k+1} &= \cos(\tau\Omega)y_k + \tau\ \mathrm{sinc}(\tau\Omega)y_k' + \frac{\tau^2}{2}\psi(\tau\Omega)g_k \\ y_{k+1}' &= -\Omega\sin(\tau\Omega)y_k + \cos(\tau\Omega)y_k' + \frac{\tau}{2}\Big(\psi_0(\tau\Omega)g_k + \psi_1(\tau\Omega)g_{k+1}\Big) \end{aligned} \tag{2.8}$$

with $g_k = g\big(\varphi(\tau\Omega)y_k\big)$. Here, the approximation of the derivative of y is directly included. We assume the functions ψ_0 and ψ_1 to be even and to satisfy $\psi_0(0) = \psi_1(0) = 1$. The method is symmetric if and only if

$$\psi(\xi) = \ \mathrm{sinc}(\xi)\psi_1(\xi) \quad \text{and} \quad \psi_0(\xi) = \cos(\xi)\psi_1(\xi)\,.$$

To get a symplectic method, we have to choose

$$\psi(\xi) = \operatorname{sinc}(\xi)\varphi(\xi)\,.$$

In [11], Hairer and Lubich proved linear energy conservation up to order τ, if the functions satisfy

$$\psi(\xi) = \operatorname{sinc}^2(\xi)\varphi(\xi)\,.$$

Obviously, these methods cannot be symplectic and have good energy conservation at the same time. However, for physical applications, the energy conservation is considered more important.

To obtain a method, which is of second order independent of Ω, in [9] Grimm and Hochbruck derived the following set of criteria for the functions:

$$\max_{\xi\geq 0}|\chi(\xi)| \leq C_1 \text{ for } \chi = \psi, \psi_0, \psi_1, \varphi$$

$$\max_{\xi\geq 0}\left|\frac{\varphi(\xi)-1}{\xi}\right| \leq C_2$$

$$\max_{\xi\geq 0}\left|\frac{1}{\sin(\xi/2)}\big(\operatorname{sinc}^2(\xi/2) - \psi(\xi)\big)\right| \leq C_3$$

$$\max_{\xi\geq 0}\left|\frac{1}{\xi\sin(\xi)}\big(\operatorname{sinc}(\xi/2) - \chi(\xi)\big)\right| \leq C_4 \text{ for } \chi = \psi_0, \psi_1, \varphi$$

To obtain first order accuracy for the derivatives of y we need additional conditions:

$$\max_{\xi\geq 0}|\xi\psi(\xi)| \leq C_5$$

$$\max_{\xi\geq 0}\left|\frac{\xi}{\sin(\xi/2)}\big(\operatorname{sinc}^2(\xi/2) - \psi(\xi)\big)\right| \leq C_6$$

$$\max_{\xi\geq 0}\left|\frac{1}{\sin(\xi/2)}\big(\operatorname{sinc}(\xi/2) - \psi_i(\xi)\big)\right| \leq C_7 \text{ for } i = 0, 1$$

Theorem 2.2. *The numerical solution obtained by* (2.8) *employing functions that satisfy the bounds given above, for the problem* (2.1) *fulfills the error bounds*

$$\|y_k - y(t_k)\| \leq C\tau^2 \quad \textit{and} \quad \|y_k' - y'(t_k)\| \leq C\tau$$

where the constants in are independent of Ω, the size of the system, k, τ and the derivatives of y.

Hochbruck and Grimm also proposed a choice of functions,

$$\psi(\xi) = \operatorname{sinc}^3(\xi) \quad \text{and} \quad \varphi(\xi) = \operatorname{sinc}(\xi)\,.$$

These functions combined with the conditions for symmetry and energy conservation result in a set of functions ψ, ψ_0, ψ_1 and φ satisfying the conditions for the error bounds. We follow their choice for our implementation, see Chapter 4. In contrast to the two-step method, this scheme only solves problems with $g \equiv 0$ exactly.

Chapter 3

Numerical simulation of one-dimensional laser-plasma interaction

In this chapter, we return to the physical problem described in Chapter 1. For several reasons, it is interesting to study the “simple” one-dimensional case first. It is necessary to get acquainted with the physics of the problem. Moreover, the one-dimensional problem is small enough to try several methods and find out, what works best. We identify physical properties of the solution, which can be exploited to speed up our integrator. The smallness of the problem permits us to extensively compare our new method with the leap-frog method, which is the standard integrator for such problems.

This work was done in close collaboration with theoretical physics and was published in [24].

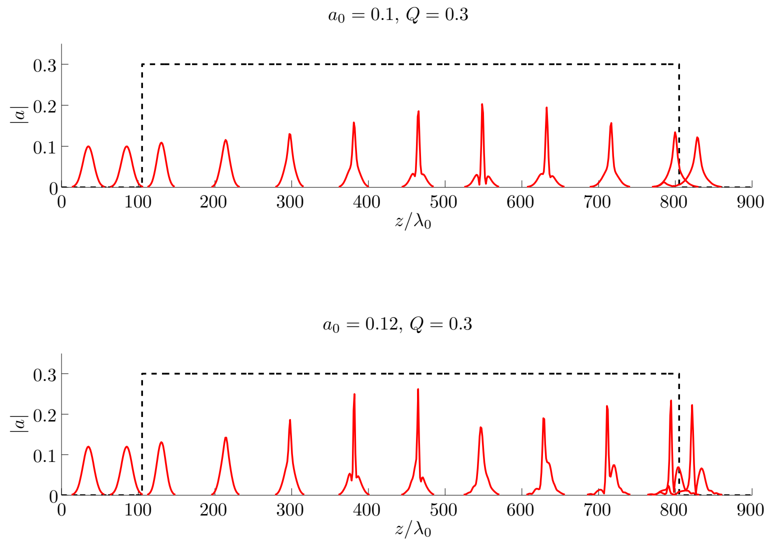
3.1 Physical example in one space dimension

The one-dimensional wave equation reads

$$\frac{\partial^2}{\partial t^2}a - \frac{\partial^2}{\partial z^2}a = -Q\frac{n}{\gamma}a \tag{3.1}$$

and the plasma response is given by

$$\frac{\partial^2}{\partial t^2}\delta n + Qn_i\delta n = n_i\frac{\partial^2}{\partial z^2}\gamma \tag{3.2}$$


$a_0 = 0.1$, $Q = 0.3$

$a_0 = 0.12$, $Q = 0.3$

Figure 3.1: Pulses (red) at different times moving through the plasma (black)

with

$$\gamma^2 = 1 + |a|^2 \quad \text{and} \quad n = n_i + \delta n\,.$$

For our test problem we choose $z \in [0, 1000\lambda_0]$. Recall that $\lambda_0 = 2\pi$ is the normalized wave length of the laser pulse in vacuum. We set $Q = 0.3$ and choose the initial density profile $n_i(z) = 1$ for $z \in [105\lambda_0, 805\lambda_0]$. At the borders of the plasma we either assume a density jump to zero or linear increase or decrease over a stretch of $5\lambda_0$. As an initial condition for the vector potential we choose

$$a(z,t) = a_0 e^{-\frac{(z-z_0-t)^2}{w_0^2}} e^{i(z-z_0-t)}\,, \tag{3.3}$$

where $z_0 = 35\lambda_0$ is the initial position of the center of the pulse, $w_0 = 10\lambda_0$ the initial width of the pulse and $a_0 = 0.1$ or $a_0 = 0.12$ is the initial amplitude of the pulse. The initial condition is basically a Gaussian pulse multiplied with a term oscillating with the carrier frequency. In vacuum, (3.3) is an exact solution of (3.1).

An overview of the pulse moving through a plasma layer for the initial amplitudes $a_0 = 0.1$ (top) and $a_0 = 0.12$ (bottom) is given in Fig. 3.1. The same pulse is shown at different times (red). The density profile n_i is drawn in black. In both cases, first a compression and then a widening of the pulse can be observed. If we calculate the amplitude for the single soliton state of the Schrödinger model for a $\operatorname{sech}(z/w_0)$ pulse with $w_0 = 10\lambda_0$ (see [33]), we get $a_0 \approx 0.038$. A simulation of such a pulse verifies that the soliton state of our model equations is close to this. For the two amplitudes above, this implies that we are well within the nonlinear regime. It also suggests that the initial condition with $a_0 = 0.1$ is close to a bound two-soliton state, while for $a_0 = 0.12$ it is clearly above. In the latter case the pulse compresses more and earlier, and more energy is radiated away from the core of the pulse after the compression.

3.2 Numerical schemes

3.2.1 Spatial discretization

Due to the finite energy assumption on the physical solution it is possible to consider periodic boundary conditions for the discretization as long as the simulation box is big enough and one takes care of the reflected parts of the pulses at the boundaries. For long time simulations this can be combined with a moving window technique. This is explicited for the two-dimensional case in Section 4.1.5.

Semi-discretization in space is done by a pseudo-spectral method with N Fourier modes on a space interval $z \in z_0 + [-L, L]$. This leads to the following system of coupled ordinary differential equations in time (the prime denotes time derivative):

$$\mathbf{a}'' + \Omega_1^2 \mathbf{a} = g_1(\mathbf{a}, \delta\mathbf{n}) , \qquad g_1(\mathbf{a}, \delta\mathbf{n}) = -Q(n_i + \delta\mathbf{n})\frac{1}{\gamma}\mathbf{a} , \tag{3.4}$$

$$\delta\mathbf{n}'' + \Omega_2^2 \delta\mathbf{n} = g_2(\mathbf{a}) , \qquad g_2(\mathbf{a}) = -n_i \Omega_1^2 \sqrt{1 + |\mathbf{a}|^2} . \tag{3.5}$$

Here, $\Omega_1^2 = -D_N^2$ with $D_N = \mathcal{F}_N^{-1} \mathcal{D}_N \mathcal{F}_N$, where $\mathcal{F}_N$ is the discrete Fourier-transform operator, and

$$\mathcal{D}_N = \frac{i\pi}{L} \operatorname{diag}(-\frac{N}{2}, \ldots, \frac{N}{2} - 1)$$

and $\Omega_2^2 = Q n_i$. The jth component of the vectors $\mathbf{a}(t)$ and $\delta\mathbf{n}(t)$ are approximations to $a(z_j, t)$ and $\delta n(z_j, t)$ at $z_j = z_0 + j\frac{2L}{N}$.

3.2.2 Time discretization

In general, these equations can be written in the form (2.1) with the properties stated in Chapter 2. Here $\Omega = \operatorname{diag}(\Omega_1, \Omega_2)$ is a block diagonal matrix containing the linear parts of the two equations in the diagonal blocks. For this kind of systems we discussed Gautschi-type exponential integrators in Section 2.2. Note, that the block diagonal structure of Ω is inherited by the matrix functions of the exponential integrator. We propose to solve these equations with a modification of the two-step Gautschi-type exponential integrator from Section 2.2.1. We will use

$$\mathbf{a}_{k+1} = 2\mathbf{a}_k - \mathbf{a}_{k-1} + \tau^2 \psi(\tau\Omega_1)\Big(-\Omega_1^2 \mathbf{a}_k + g_1\big(\varphi(\tau\Omega_1)\mathbf{a}_k, \delta\mathbf{n}_k\big)\Big) \tag{3.6}$$

with

$$\psi(\xi) = 2\frac{1-\cos(\xi)}{\xi^2} \quad \text{and} \quad \varphi(\xi) = \operatorname{sinc}^2(\xi)\Big(1 + \frac{1}{2}\big(1-\cos(\xi)\big)\Big)\,,$$

and if desired

$$\mathbf{a}'_{k+1} = \mathbf{a}'_{k-1} + 2\tau \operatorname{sinc}(\tau\Omega_1)\Big(-\Omega_1^2\mathbf{a}_k + g_1\big(\varphi(\tau\Omega_1)\mathbf{a}_k, \delta\mathbf{n}_k\big)\Big) \tag{3.7}$$

for the vector potential equation (3.4), where $\mathbf{a}_k$, $\mathbf{a}'_k$ and $\delta\mathbf{n}_k$, $\delta\mathbf{n}'_k$ are approximations to the vectors containing the coefficients of the spatial discretization and their time derivatives of the vector potential and the density variation, respectively.

The accuracy of the integrator may be further improved if approximations to the inhomogeneity are available at additional times. This is only true if we solve the equations (3.5) for the density variation because there the inhomogeneity only depends on $\mathbf{a}$. If we solve the equation for $\mathbf{a}$ first, we have approximations $\mathbf{a}_j \approx \mathbf{a}(t_j)$ for $j = k-1, k$, and $k+1$. We then replace $g_2(\mathbf{a})$ by an interpolation polynomial of degree two interpolating in $(t_{k-1}, g_2(\mathbf{a}_{k-1}))$, $(t_k, g_2(\mathbf{a}_k))$, and $(t_{k+1}, g_2(\mathbf{a}_{k+1}))$. Note that we consider the circular polarized case, in which g_2 is a smooth function. Using this interpolation polynomial instead of $g(y(t \pm \sigma))$ in (2.6) yields

$$\begin{aligned}\delta\mathbf{n}_{k+1} = 2\delta\mathbf{n}_k - \delta\mathbf{n}_{k-1} &+ \tau^2\psi(\tau\Omega_2)\big(-\Omega_2^2\delta\mathbf{n}_k + g_2(\mathbf{a}_k)\big) \\ &+ \tau^4\chi(\tau\Omega_2)\big(g_2(\mathbf{a}_{k+1}) - 2g_2(\mathbf{a}_k) + g_2(\mathbf{a}_{k-1})\big)\end{aligned} \tag{3.8}$$

for (3.5), where

$$\chi(\xi) = 2\frac{\cos\xi - 1 + \frac{1}{2}\xi^2}{\xi^4}\,.$$

The scheme (3.8) is of order four, if $\mathbf{a}_j$, $j = k-1, k, k+1$ are exact or sufficiently accurate approximations of $\mathbf{a}(t_j)$. However, the coupled scheme (3.6), (3.8) cannot be better than second order.

Since the spatial discretization was done with a pseudo-spectral method, Ω_1 can be diagonalized via fast Fourier transforms and the matrix functions can be computed for the diagonalized matrix. Ω_2 is already diagonal, thus for (3.8) the computation of the matrix functions is straight forward.

3.2.3 Choice of operators

For solving (3.4) the obvious choice would be using (3.6) with Ω_1 from the previous section. By construction, the Gautschi-type integrator then solves equations of the form

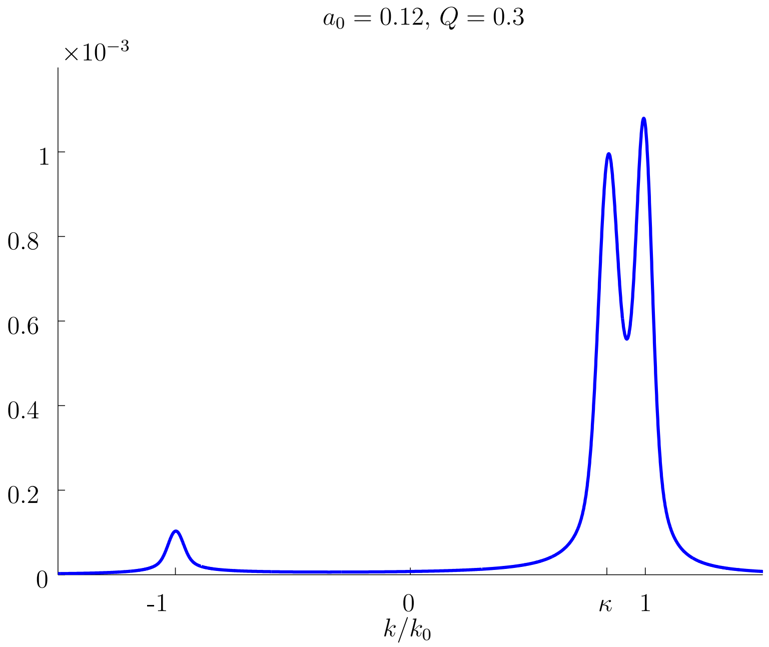


Figure 3.2: Spectrum of the vector potential while entering the plasma, $\kappa = \sqrt{1-Q}$.

$y'' = -\Omega^2 y + g$ with constant g exactly. Due to the special form of the nonlinearity g, we can enlarge the part which is integrated exactly by writing

$$g_1(\mathbf{a}, \delta\mathbf{n}) = -\alpha\mathbf{a} + \widetilde{g}_1(\mathbf{a}, \delta\mathbf{n})$$

and setting $\widetilde{\Omega}_1^2 = -D_N^2 + \alpha$ for a suitable α. If the pulse is inside the plasma, the dominant term of g_1 is $-Qn_i\mathbf{a}$ and is thus linear in $\mathbf{a}$. However, for a nonconstant plasma profile n_i, this cannot be simply added to Ω_1, since it would destroy the favorable diagonalization property. We suggest to choose $\alpha = Q$ if the pulse is inside the plasma. Outside the plasma (where $n_i = 0$) the nonlinearity is negligible so that one should set $\alpha = 0$. For varying background densities, a medium value can be used. Thus we can still use `fft`'s to evaluate the matrix functions.

3.2.4 Quasi-envelope approach

The motivation behind the quasi-envelope approach (QEA) is illustrated by the numerical result shown in Fig. 3.2: the spectrum of the vector potential splits into two parts. The important part is concentrated around a certain characteristic wave number depending on whether the pulse propagates inside or outside of the plasma. In addition there is another peak resulting from reflection which is not of interest to our physical application. Therefore, it is sufficient to resolve the main pulse only. The number of spatial grid points required can be reduced significantly by shifting the spectrum appropriately, i.e. we rewrite the vector potential a as

$$a(z,t) = \widetilde{a}(z,t)e^{i\kappa z}$$

and solve (3.1) for $\widetilde{a}$ instead of a. This yields

$$\frac{\partial^2}{\partial t^2}\widetilde{a} - \frac{\partial^2}{\partial z^2}\widetilde{a} - 2i\kappa\frac{\partial}{\partial z}\widetilde{a} + \kappa^2\widetilde{a} = -Q(n_i + \delta n)\frac{1}{\gamma}\widetilde{a}\,, \quad \gamma^2 = 1 + |\widetilde{a}|^2\,.$$

In Section 1.4 we used a similar ansatz to derive the nonlinear Schrödinger equation. Note that in contrast to the "classical" slowly-varying envelope approximation we do not neglect higher derivatives of the slow variables for the QEA. In the spatially discretized equation (3.4), $\mathcal{D}_N^2$ has to be replaced by $(\mathcal{D}_N + i\kappa)^2$. The value of κ can be varied for different positions of the pulse (inside/outside of the plasma or at the boundary), so we choose $\kappa = \sqrt{1-Q}$ or $\kappa = 1$ or the mean value of both.

3.2.5 Multilevel approach

Obviously, the spatial grid size is determined by the necessity of resolving reflections arising at jumps of the plasma density. If we have a sharp jump (for instance in the case of a rectangular density profile shown in Fig. 3.1), then the reflections require small spatial step sizes only when the pulse enters or leaves the plasma. This can be exploited in a standard way by using two (or more) different grids. In our case we used a fine grid in transitions between vacuum and plasma and a coarse one in the remaining simulation. Switching between coarse and fine grid is done by interpolation and from the fine to the coarse grid by restriction (both in Fourier space). Note that this switch requires to recompute the differential operator and hence the matrix operators required for the Gautschi-type integrator.

3.2.6 Overall numerical method

We suggest to combine the strategies described above. This requires the computation of three or more sets of operators: one in vacuum ($\alpha_\mathrm{v} = 0, \kappa_\mathrm{v} = 1$, coarse grid), one in plasma ($\alpha_\mathrm{p} = Q, \kappa_\mathrm{p} = \sqrt{1-Q}$, coarse grid), and one in the transition region ($\alpha_\mathrm{t} = Q/2, \kappa_\mathrm{t} = 0$, fine grid), and possibly additional sets if the pulse becomes too steep to be resolved on the coarse grid in plasma due to nonlinear pulse compression. If the background density is small (so that the difference between vacuum and plasma wavelength is also small) and the density profile has no sharp jump (so that no reflection occurs), it may be sufficient to use the same set of operators for both the transition region and the plasma region on the same coarse grid, with a κ equal to the mean of vacuum and plasma wave number. Recall that in vacuum, there is no nonlinearity, and thus the Gautschi-type integrator solves the problem exactly for arbitrary time steps. Obviously, it is not necessary to compute filter functions in this case.

3.3 Exemplary results

3.3.1 Description of the simulated problem

We consider the example setup from Section 3.1 for our numerical experiments.

For runtime comparisons we chose the piecewise linear density profile. In this case, the multilevel approach is not necessary, because nearly no reflections occur at the plasma boundaries. To further simplify the problem for the runtime comparisons, for methods with the QEA, only one set of operators is used with a mean value of vacuum and plasma wavelength. With an additional set of operators for the plasma part, the results discussed below would be even better. But for a low background density like $Q = 0.3$, which we used, the results are already very good. For denser plasmas (e.g. $Q = 0.6$), switching of operators between vacuum, plasma boundary and plasma parts of the density profile becomes a necessity. For the multilevel tests we used a rectangular density profile starting at $105\lambda_0$ and ending at $805\lambda_0$, cf. Fig. 3.1, and we included the different operators discussed in Section 3.2.6.

As benchmarks for the accuracy of the different numerical schemes, we used two physically relevant error measures, namely the position error and the amplitude error of the maximum of the pulse with an emphasis on the latter. Since we do not have an analytical solution of the nonlinear model equations, we computed a reference solution on a very fine grid ($N = 2^{17}$) with very small time steps. We then used it to measure the error in maximum amplitude squared (amplitude error) and its position (phase error) at different times of the simulation results. Since the simulations were computed on coarser grids (especially the QEA solutions) we first Fourier interpolated to the same number of grid points as the reference solution.

3.3.2 Effect of different time integration schemes

If the vector potential is held in Fourier space and only transformed back for the evaluation of the nonlinearity/inhomogeneity, one has to compute four fast Fourier transforms per time step for the leap-frog method (two for the nonlinearity of the wave equation and two for the inhomogeneity of the plasma response). There is one more Fourier transform needed for the Gautschi-type integrator since in each step the filtered as well as the nonfiltered vector

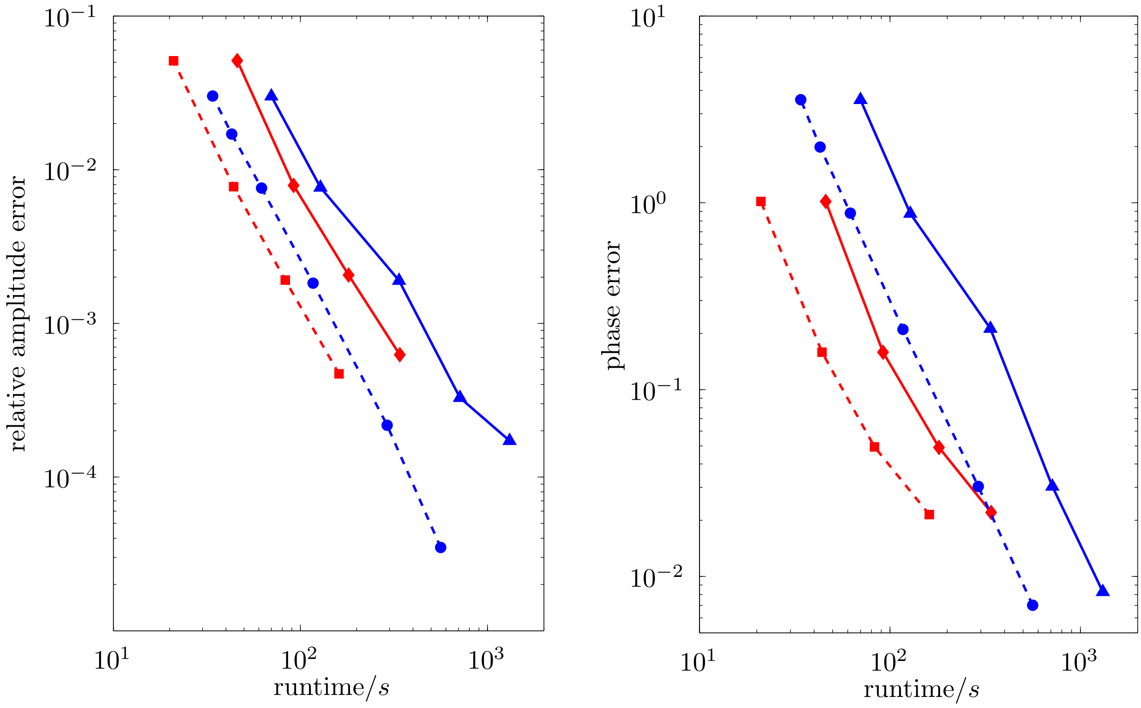


Figure 3.3: Maximum amplitude and phase error versus runtime ($a_0 = 0.1$) for varying τ for leap-frog (solid blue), Gautschi (solid red), leap-frog + QEA (dashed blue) and Gautschi + QEA (dashed red). We used $N = 2^{12}$ for methods without the QEA and $N = 2^{11}$ for methods with the QEA (see also Table 3.1).

potential is required in real space. In addition, one has to compute the products with the matrix functions ψ, φ, χ and possibly sinc. Obviously computing a single time step with the Gautschi-type integrator is more expensive than one time step with the leap-frog method. But it turns out that the Gautschi-type method allows larger time steps in order to achieve the same accuracy.

In Fig. 3.3 and Fig. 3.4 the maximum relative amplitude error (left) and the maximum phase error in λ_0 (right) are plotted over computational time. Each curve represents one integrator on one spatial grid with different time steps.

For a given tolerance for the relative amplitude error the leap-frog method (solid blue) needs two times smaller time steps than the Gautschi-type integrator (solid red) on the same spatial grid ($N = 2^{12}$). In our examples this reduces the computational time by a factor of 1.5 (see Table 3.1). If the phase error is taken into account, too, the gain in computational time is even greater.

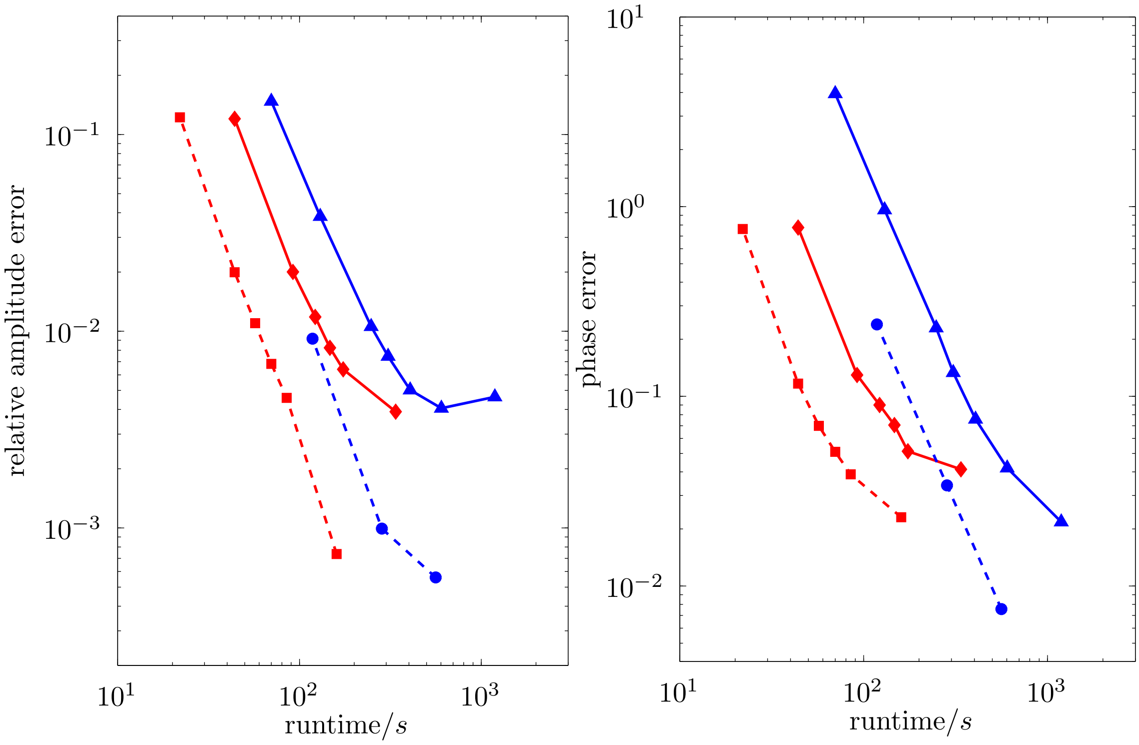


Figure 3.4: Same as Fig. 3.3, but for $a_0 = 0.12$.

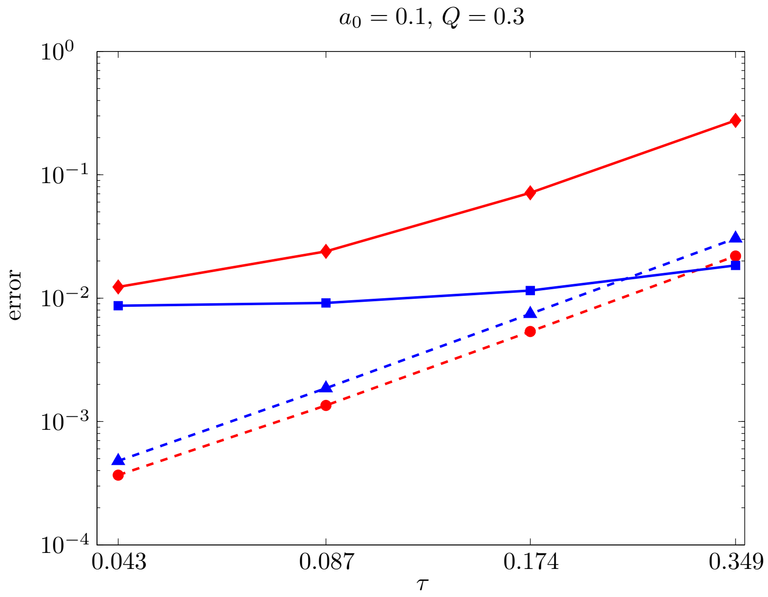


Figure 3.5: Amplitude and phase error plotted over the time-step size τ for the Gautschi-type integrator including quasi-envelope approach with and without the variant described in Section 3.2.3. Phase (solid) and amplitude (dashed) error with $\alpha = 0$ (red) and $\alpha = Q$ (blue) within the plasma for $a_0 = 0.1$.

3.3.3 Effect of choice of operators

The effect of the choice of operators is illustrated in Fig. 3.5 for the case $a_0 = 0.1$. It is observed that the choice of $\alpha = Q$ within the plasma reduces the phase error significantly while the error in the amplitude is only slightly larger. However, for $a_0 = 0.12$ switching between the operators did not pay off. The reason for this might be the increased density variation compared to the smaller amplitude. The results in Fig. 3.5 were computed including the QEA of Section 3.2.4, but the method showed the same behavior when combined with other variants described above. The phase error is given in terms of λ_0 whereas the amplitude error is given relatively compared to the reference amplitude. In both cases the error is averaged over pulses at 100 different positions spread evenly over the computation interval.

3.3.4 Effect of quasi-envelope approach

By applying the QEA to the leap-frog method as well as to the Gautschi-type integrator, the number of spatial grid points can be significantly reduced without loss of accuracy (see curves with and without the QEA in Fig. 3.3 and 3.4). Since the major part of

	$a_0 = 0.1$			$a_0 = 0.12$		
	N	τ	time/min.	N	τ	time/min.
LF	2^{12}	0.1	2:10	2^{12}	0.04	5:07
LF + QEA	2^{11}	0.1	1:03	2^{11}	0.05	1:57
Gautschi	2^{12}	0.2	1:32	2^{12}	0.12	2:28
Gautschi + QEA	2^{11}	0.2	0:44	2^{11}	0.12	1:10

Table 3.1: Runtimes for maximum one percent relative amplitude error. N is the number of spatial grid points, τ is the time-step size. Computational details: Pentium 4, 3.0 GHz, Intel C++ 8.1, FFT routines from Intel Math Kernel Library 7.2.

computational time is spent on fast Fourier transforms, which cost $\mathcal{O}(N \log N)$ operations, the reduction of grid points by a factor of 2 again leads to a saving in computational time of more than a factor of 2. Another reason for a more than linear reduction in computational time is that smaller arrays are more likely to fit into the cache of the processor. For small enough arrays, a whole time step can run from CPU cache. We observed that the QEA is more effective in reducing the amplitude error, while the Gautschi-type method is more effective in reducing the phase error.

The parameters for the discretization needed to achieve a maximum relative amplitude error of 10^{-2} are summarized in Table 3.1. Exemplary runtimes for one specific hardware/software setup are also given.

If one compares the standard leap-frog method to the new variant of the Gautschi-type integrator combined with the QEA, the computational time is reduced by a factor of 3 in the first and even by a factor of 4.5 in the second example. If we set a bound lower than 10^{-2} for the amplitude error, we see that without the QEA this error bound cannot be reached by only reducing τ. This is because the error due to the coarse spatial resolution limits the accuracy that can be reached. Thus a finer grid is needed, which results in a corresponding increase of computational time, while the discretization for the QEA can stay the same (see Fig. 3.6).

3.3.5 Effect of two-level approach

The benefit of the two-level approach suggested in Section 3.2.5 is illustrated in Fig. 3.7. The reference solution as well as the simulation results are shown at $t = 700 \cdot 2\pi$ for a plasma jump and $a_0 = 0.12$. It can be seen that in this case it is possible to work on a coarse grid ($N = 2^{11}$) in the major part of the simulation but it is not possible to do the whole simulation on the coarse grid. In the transition we interpolated to 2^{13} grid points.

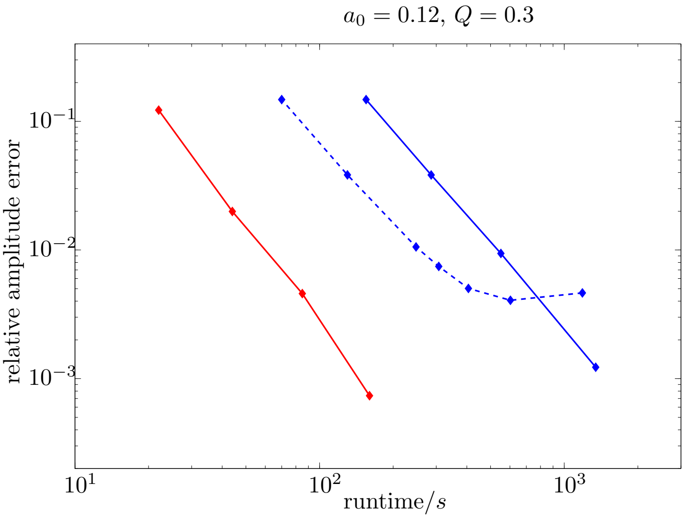


Figure 3.6: Maximum amplitude error versus runtime ($a_0 = 0.12$) for constant N and varying τ for leap-frog with $N = 2^{13}$ (blue solid), leap-frog with $N = 2^{12}$ (blue dashed) and Gautschi + QEA with $N = 2^{11}$ (red).

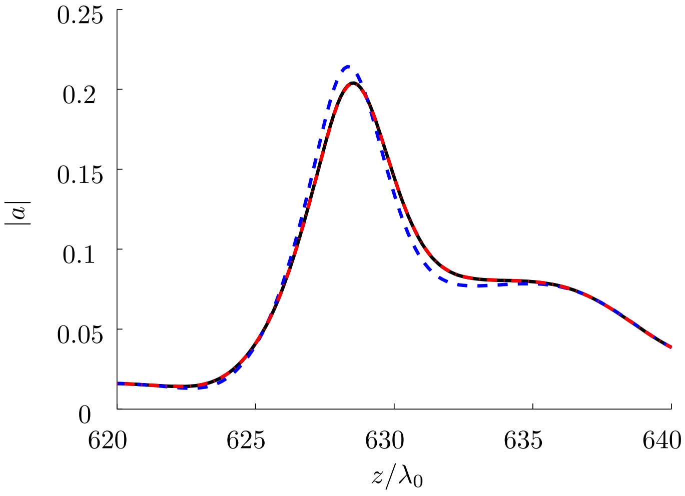


Figure 3.7: Results of simulations using the two-level approach compared to a one-level simulation on the (same) coarse grid only for $a_0 = 0.12$. Black: reference solution, blue: solution computed on a coarse grid only, red: two-level approach (curve coincides with the solid one).

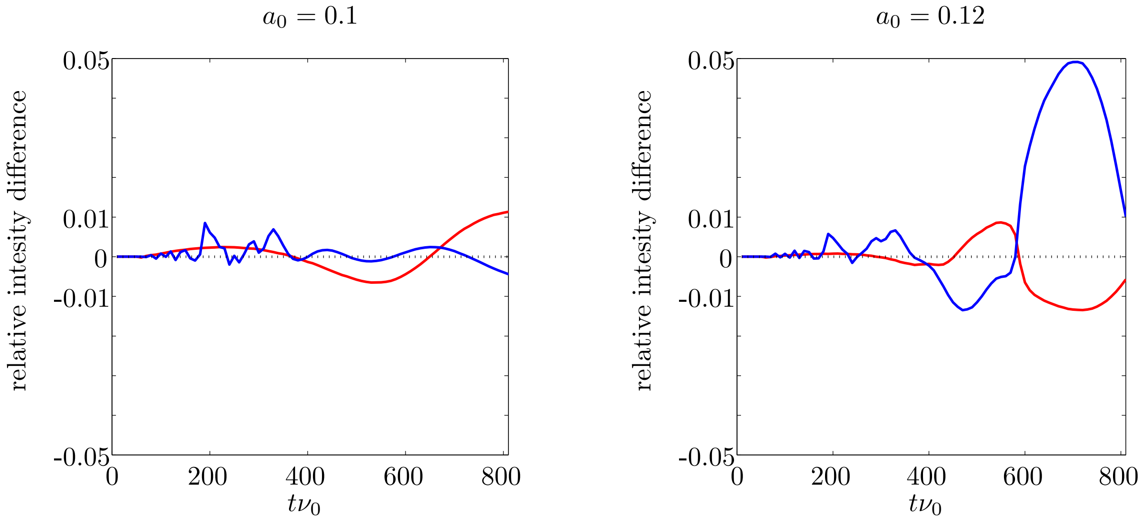


Figure 3.8: Relative difference in maximum intensity to the reference solution of the reduced model for $a_0 = 0.1$ (left) and $a_0 = 0.12$ (right). Gautschi + QEA (see Table 3.1, red) and PIC (blue) with $N = 2 \cdot 10^5$, $\tau = \Delta z(N)$ and 3 particles per cell, runtime around $5 : 30$ h.

3.3.6 Comparison with PIC

Finally, we compare our results with PIC simulations performed with VLPL [30]. Since PIC simulates $\mathbf{E}$ and $\mathbf{B}$ instead of $\mathbf{A}$, we base our comparison on intensities, calculated by

$$I = \frac{1}{2}\left(|\mathbf{E}|^2 + |\mathbf{B}|^2\right) = \frac{1}{2}\Big(\Big|\frac{\partial}{\partial t}\mathbf{A}\Big|^2 + \Big|\frac{\partial}{\partial z}\mathbf{A}\Big|^2\Big).$$

For the Gautschi-type method, one has to use (3.7) for the time derivative, and for the QEA we replace $\partial/\partial z$ by $\partial/\partial z + i\kappa$. The difference in amplitudes between the reference solution for the reduced model and PIC are of the same order than the error of the Gautschi-type method with the QEA for the parameters given in Table 3.1, see Fig. 3.8. This implies that, even with a relatively coarse discretization, the error of the simulations with our fastest solver is within the accuracy of the reduced model, which seems to be at the border of applicability at $a_0 = 0.12$.

We also noticed, that there is a systematic difference in group velocity between PIC solutions and ours. To understand whether this is due to numerical error in PIC and/or our solvers, we made simulations with both for a very small amplitude ($a_0 = 0.0001$). The combination of small amplitude and a cold plasma allows to test the phase error of the numerical simulations against the known linear analytical solution. The results in Fig. 3.9 show that PIC (blue) produces a slight error in group velocity even on a fine grid, whereas Gautschi + QEA (red) with coarse discretization is close to the exact solution.

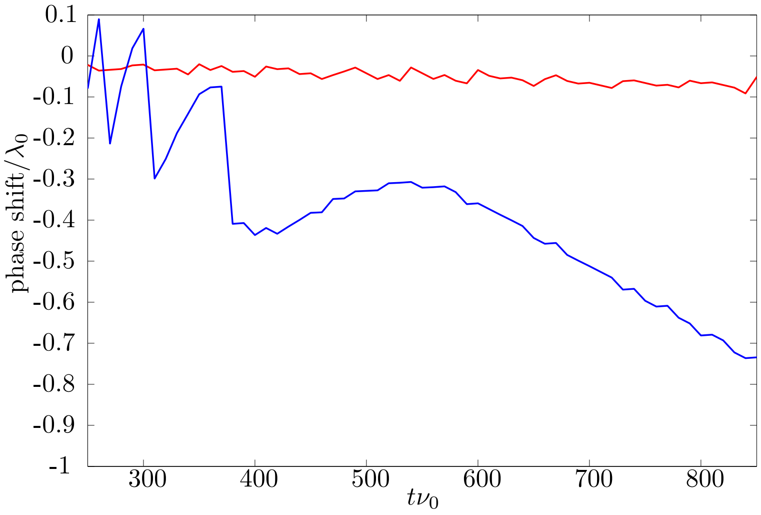


Figure 3.9: Phase-difference to the exact linear solution for PIC (blue) and Gautschi + QEA (red), both with $a_0 = 0.0001$.

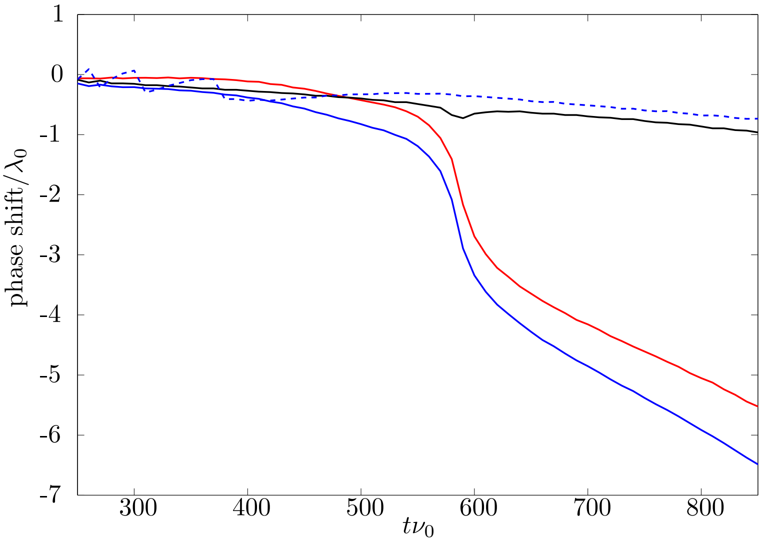


Figure 3.10: Phase-difference to the exact linear solution for PIC ($a_0 = 0.12$: blue solid and $a_0 = 0.0001$: blue dashed) and Gautschi + QEA ($a_0 = 0.12$, red), difference between PIC and Gautschi + QEA for $a_0 = 0.12$ (black).

In Fig. 3.10 we compare the phase shift (with respect to the exact linear solution) of VLPL (blue solid) and the Gautschi + QEA simulation from Table 3.1 (red) in the nonlinear case ($a_0 = 0.12$). The difference between the two (black) is consistent with the linear phase error of PIC (blue dashed). This shows that the difference in phase between nonlinear PIC and Gautschi + QEA is mostly linear phase error of PIC, which could also influence the accuracy of the amplitude calculation.

Chapter 4

Numerical simulation of two-dimensional laser-plasma interaction

In Chapter 3 we demonstrated that we successfully implemented a Gautschi-type exponential integrator combined with the QEA for the one-dimensional Klein-Gordon equation coupled to a plasma response. Now we extend the implementation to the two-dimensional case. Here we consider Cartesian and cylindrical geometries. It will be shown, that the ideas of Chapter 3 can be used for the two-dimensional case, too. However, we also need new ideas, that are specifically developed for the two-dimensional case, also with regard to parallelization of the code. We will again use physical properties of the solution to efficiently realize a two-dimensional code.

Again, this work was a collaboration with theoretical physics. The results were published in [25].

4.1 Numerical schemes

4.1.1 Gautschi-type exponential integrator for time-discretization

Similar to the one-dimensional case semi-discretization in space (cf. Sec. 4.1.3) leads to a system of second order ordinary differential equations of the form (2.1). For the solution we now suggest the one-step Gautschi-type integrator, described in Section 2.2.2,

$$
\begin{aligned}
y_{k+1} &= \cos(\tau\Omega)y_k + \tau \ \mathrm{sinc}(\tau\Omega)y_k' + \frac{\tau^2}{2}\psi(\tau\Omega)g\big(\varphi(\tau\Omega)y_k\big) \\
y_{k+1}' &= -\Omega\sin(\tau\Omega)y_k + \cos(\tau\Omega)y_k' + \frac{\tau}{2}\Big(\psi_0(\tau\Omega)g\big(\varphi(\tau\Omega)y_k\big) + \psi_1(\tau\Omega)g\big(\varphi(\tau\Omega)y_{k+1}\big)\Big)
\end{aligned}
$$

with

$$\psi(\xi) = \ \mathrm{sinc}^3(\xi)\,, \quad \varphi(\xi) = \ \mathrm{sinc}(\xi)\,,$$

the condition for symmetry,

$$\psi(\xi) = \ \mathrm{sinc}(\xi)\psi_1(\xi) \quad \text{and} \quad \psi_0(\xi) = \cos(\xi)\psi_1(\xi)$$

and the energy conservation condition

$$\psi(\xi) = \ \mathrm{sinc}^2(\xi)\,.$$

Note, that linear problems with $g \equiv 0$ are solved exactly by this scheme. This allows to use arbitrarily large time steps for the propagation in vacuum. For the propagation inside of the plasma layers, smaller time steps have to be used to obtain the desired accuracy. Note that this change of time steps would be much more complicated for the two-step method discussed in the one-dimensional case.

4.1.2 Implementation of exponential integrators

For a Gautschi-type time integration scheme, the main effort per time step is the evaluation or approximation of the products of certain matrix functions of the discretized Laplacian Ω with vectors. It is indispensable to do this in an efficient way. The computational cost of each time step is thus closely related to the spatial discretization.

For one-dimensional problems with periodic boundary conditions, the method of choice is using pseudo-spectral spatial discretization, in which case the matrix Ω is diagonalizable via one-dimensional Fourier transformations. The computational cost of these transformations is $O(N_z \log N_z)$ operations for N_z spatial grid points.

The situation is slightly different in two space dimensions. Recall that a two-dimensional Fourier transformation on a grid consisting of $N_z \times N_x$ grid points can be evaluated using $O(N_zN_x(\log N_z + \log N_x))$ operations. For large grids, this may become too expensive. In addition, on parallel machines, such transformations become inefficient due to the large communication effort.

In general, the diagonalization of a large matrix Ω resulting from a finite difference or finite element discretization is impossible. An alternative is to use Krylov subspace methods such as the symmetric Lanczos process [3, 16]. However, for the applications considered here such techniques were not competitive to the methods that are implied by the solution itself.

Therefore, we propose to use different spatial discretizations in different regimes depending on physical properties of the solution. Moreover, we alter the splittings in (2.1) during the time integration, i.e. we move parts of the discretized Laplacian into the function g. This allows for an efficient evaluation of the matrix functions.

4.1.3 Spatial discretization

We consider the Laplacian in Cartesian coordinates as well as in cylindrical coordinates.

In the Cartesian case we assume periodic boundary conditions in both directions. This is possible as long as reflected waves are taken care of at the boundaries, since the physical solution satisfies a finite energy condition.

For the cylindrical case, we impose periodic boundary conditions only for the longitudinal direction and homogeneous Dirichlet boundary conditions for $r = R$.

For both geometries we solve the density equation and evaluate the γ-factor only on grid points which are inside the plasma.

Cartesian coordinates in vacuum

In vacuum we only need to solve the linear wave equation

$$\frac{\partial^2}{\partial t^2}a - \frac{\partial^2}{\partial z^2}a - \frac{\partial^2}{\partial x^2}a = 0\,. \tag{4.1}$$

For periodic boundary conditions the semi-discretization in space is done by a pseudo-spectral method with N_z Fourier modes on the interval $z \in z_0 + [-L_z, L_z]$ in propagation direction and N_x modes on the interval $x \in [-L_x, L_x]$ in perpendicular direction.

Let $\mathbf{a} = \mathbf{a}(t) \in \mathbb{C}^{N_z \times N_x}$ and $\mathbf{a}' = \mathbf{a}'(t) \in \mathbb{C}^{N_z \times N_x}$ be complex matrices containing approximations to the vector potential and its time derivative on the grid,

$$\mathbf{a}_{i,j} \approx a(x_j, z_i, t)\,, \quad \mathbf{a}'_{i,j} \approx \frac{\partial}{\partial t}a(x_j, z_i, t)\,.$$

The Laplacian is approximated by

$$\Delta a \approx \mathcal{F}_{N_z}^{-1}\mathcal{D}_z^2\mathcal{F}_{N_z}\mathbf{a} + \mathbf{a}\mathcal{F}_{N_x}^{T}\mathcal{D}_x^2\mathcal{F}_{N_x}^{-T}$$

where

$$\mathcal{D}_k = \frac{i\pi}{L_k}\mathrm{diag}\left(-\frac{N_k}{2}, \ldots, \frac{N_k}{2} - 1\right), \quad k = x, z\,,$$

and $\mathcal{F}_N$ denotes the discrete Fourier transform for N Fourier modes.

Formally, the matrices $\mathbf{a}$ and $\mathbf{a}'$ can be reorganized by writing them column-wise into long vectors. Then the spatially discretized equation (4.1) can be written as a system of differential equations (2.1), where Ω is a matrix which can be diagonalized via two-dimensional fast Fourier transforms and $g \equiv 0$. However, for the implementation, the matrix notation is more efficient.

In the first time step, where the initial data is given in the spatial domain, we start by performing a two-dimensional Fourier transform by applying fast (one-dimensional) Fourier transforms to all columns and rows of $\mathbf{a}$ and $\mathbf{a}'$. Then we evaluate the functions arising in the Gautschi-type integrator at the diagonalized operator. The resulting operator can be applied to the matrices $\mathbf{a}$ and $\mathbf{a}'$ by pointwise multiplication. (If desired, subsequent time steps in vacuum can be computed in frequency space by diagonal operations only.) At times, where the solution is required in the spatial domain, inverse Fourier transforms have to be applied to all rows and columns of $\mathbf{a}$ and $\mathbf{a}'$ again.

Due to the Gautschi-type integrator being exact in vacuum, in the best case we only have to compute one time step. The total cost amounts to two two-dimensional Fourier transforms and in addition four scalar multiplications per grid point. Storage is required for two arrays for $\mathbf{a}$ and $\mathbf{a}'$ plus four arrays for the diagonalized matrix functions of the same size. If a reduction of storage is necessary, the matrix functions can be computed on demand. From the computational point of view, this is a rather small overhead compared to the two-dimensional Fourier transforms.

Cartesian coordinates in plasma

In plasma layers we have to solve the full, nonlinear system of equations

$$\frac{\partial^2}{\partial t^2}a - \frac{\partial^2}{\partial z^2}a - \frac{\partial^2}{\partial x^2}a = -Q\frac{n_i + \delta n}{\gamma}a\,, \tag{4.2a}$$

$$\frac{\partial^2}{\partial t^2}\delta n + Q n_i \delta n = n_i \Delta\gamma\,. \tag{4.2b}$$

After space discretization, the linear part is represented by a 2 × 2 block diagonal matrix, whose upper diagonal block contains the discretized Laplacian and whose lower diagonal block contains the linear operator of the second equation, which is already diagonal. Hence, the matrix operators required for the time integration scheme can be computed separately for both equations. Note that due to the nonlinearity, we need to compute (and store) more

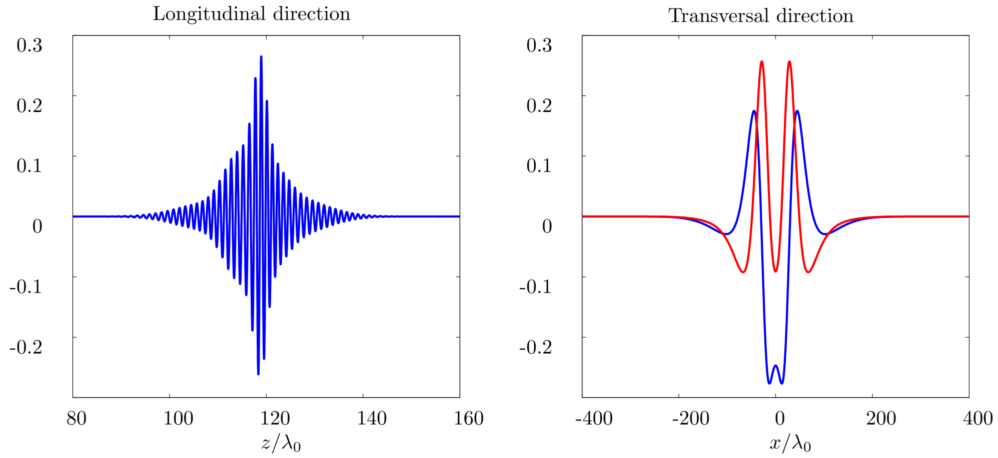


Figure 4.1: **Left:** The spatial distribution of the real part of the solution in longitudinal direction through the center of the pulse is highly oscillatory. **Right:** The spatial distribution of the real (blue) and imaginary (red) part of the solution in transversal direction through the maximum of the pulse is smooth.

matrix operators than in vacuum. The main cost of one time step in frequency domain amounts to two two-dimensional Fourier transformations.

Due to the nonlinearity, the time integration scheme does not solve the discretized system exactly anymore. However, the time step size is only limited by accuracy, not by stability, see Section 2.2.

This straightforward implementation turns out to be quite expensive with respect to computational cost and storage. Fortunately, it is possible to increase the efficiency considerably by exploiting properties of the solution.

In the left graph of Fig. 4.1 we show the longitudinal distribution of the real part of the vector potential a along the optical axis of the pulse. On the right, we show the transversal distribution of the real (blue) and the imaginary (red) part of a at the point z, where the maximum of the pulse is attained. The transversal distribution is obviously much smoother than the longitudinal. Therefore, we can discretize the transversal direction on a much coarser grid. Moreover, we propose to split the Laplacian and only treat the longitudinal part of it exactly ($\Omega_1 \approx \Delta_{\|}$) whereas the transversal part is added to the nonlinearity. To avoid the expensive two-dimensional Fourier transformations, we propose to use fourth order finite differences in this direction,

$$\frac{\partial^2}{\partial x^2}\mathbf{a}(x_j, z_i, t) \approx \frac{1}{12\Delta x^2}\big(-\mathbf{a}_{j-2,i}(t) + 16\mathbf{a}_{j-1,i}(t) - 30\mathbf{a}_{j,i}(t) + 16\mathbf{a}_{j+1,i}(t) - \mathbf{a}_{j+2,i}(t)\big)$$

with the spatial grid size Δx in transversal direction.

Due to this splitting, the longitudinal part of the Laplacian, can be diagonalized by N_x one-dimensional Fourier transforms (of length N_z). Moreover, we only have to compute (and store) matrix operators of length N_z. For the computation we keep the vector potential and its derivative in Fourier space only in longitudinal direction. In transversal direction the arrays are not transformed.

For the density equation the application of the exponential integrator is straight forward in the spatial domain. Since the density profile only depends on z here the storage requirements are again only of the order of vectors of length N_z. The inhomogeneity contains the Laplacian of the relativistic factor γ which depends on the absolute value of the vector potential. This is a smooth function for circularly polarized laser beams. Thus it is sufficient to use fourth order finite differences in *both* directions to approximate the inhomogeneity of the density equation.

Cylindrical coordinates

For the equations formulated in cylindrical coordinates

$$\frac{\partial^2}{\partial t^2}a - \frac{\partial^2}{\partial z^2}a - \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}a\right) = -Q\frac{n_i + \delta n}{\gamma}a \tag{4.3a}$$

$$\frac{\partial^2}{\partial t^2}\delta n + Qn_i\delta n = n_i\Delta\gamma \tag{4.3b}$$

we basically use the same ideas as for Cartesian coordinates in plasma regions, i.e., we use $\Omega \approx \Delta_{\|}$ and treat the transversal direction as part of the nonlinearity. For the longitudinal direction, we use pseudo-spectral discretization while for the transversal direction, we suggest to use second order finite differences.

Since in cylindrical coordinates it is hard to diagonalize the complete Laplacian in a fast and stable way we use the same implementation in vacuum as within the plasma. Note, that for this discretization, we do not obtain the exact solution in vacuum any more. Thus, we have to choose small time steps in vacuum, too.

Quasi-envelope approach

The quasi-envelope approach (QEA) is motivated by the fact that the important part of the spectrum of the operator in longitudinal direction is concentrated around a certain characteristic wave number depending on whether the pulse propagates inside or outside of the plasma, see Fig. 4.2, left. The idea of QEA is to shift the spectrum appropriately, see Section 3.2.4. In the two-dimensional case, the situation in longitudinal direction is exactly the same as in the one-dimensional case but no shift is necessary for the transversal

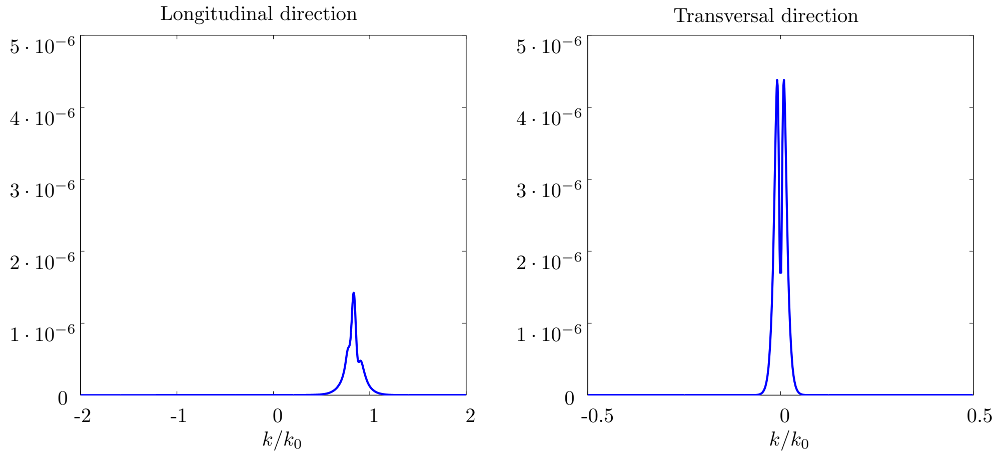


Figure 4.2: The spectrum of the longitudinal spatial distribution (left) is not centered around 0, other than that of the transversal spatial distribution (right).

direction, as can be seen in Fig. 4.2, right. Note, that this only reduces the number of grid points needed to resolve the solution, but the large norm of the approximation of the parallel part of the Laplacian remains unchanged, thus the Gautschi-type time integrator is still essential.

Here again, we replace the vector potential a by

$$a(x,z,t) = \widetilde{a}(x,z,t)e^{i\kappa z} \ ,$$

which leads to a new equation for $\widetilde{a}$

$$\frac{\partial^2}{\partial t^2}\widetilde{a} - \left(\frac{\partial}{\partial z} + i\kappa\right)^2 \widetilde{a} - \Delta_\perp \widetilde{a} = -Q\frac{n_i + \delta n}{\gamma}\widetilde{a} \quad , \quad \gamma^2 = 1 + |\widetilde{a}|^2 \ .$$

The value of κ is chosen depending on the position of the pulse, namely $\kappa = \sqrt{1-Q}$ or $\kappa = 1$ or the mean value of both.

4.1.4 Adaptivity

In order to apply all the different variations of our scheme at the appropriate time we have to determine the location of the pulse. This is done by physically motivated means. At the beginning we know the location of the maximum amplitude and the exact width of the pulse. Additionally we know the approximate group velocity of the pulse at any time. This allows to determine the approximate speed of the maximum of the pulse and to estimate the time when the pulse hits the next region of the simulation domain.

With this method we can switch between the different integration schemes in vacuum and plasma for Cartesian coordinates as well as adapt the values of κ for the QEA. The latter can be done by a simple shift in the position of the Fourier coefficients which also ensures periodicity of the shift function $e^{i\kappa z}$ with regard to the box length $2L_z$.

Additionally we can change the spatial grid, which becomes necessary for very narrow pulses as they occur in the simulation of pulse compression. Also for hard plasma boundaries, where reflections are no longer negligible, it becomes necessary to interpolate to a finer grid and invert the QEA shift, as was already shown for the one-dimensional case in Chapter 3. For pseudo-spectral discretization this only requires a larger array in Fourier space where extra entries are filled with zeros. Since the computation is much more expensive for the finer grid, interpolation is avoided unless absolutely necessary. Therefore, we also use a rather tight estimate for the pulse to be nonzero.

4.1.5 Moving simulation window

There are a lot of interesting applications where the full simulation domain is very large and it is not at all feasible to use the complete spatial domain during the whole simulation. To avoid this we use a moving window technique.

Using the group velocity as described above we estimate the time when the pulse comes close to the right boundary of the simulation box. For this purpose we slightly overestimate the domain on which we consider the pulse to be nonzero. This increases robustness while the computational overhead is negligible.

The shift is implemented by transforming the vector potential to physical space, cutting off the left part and extrapolating to the right by adding zeros for $\mathbf{a}$ and $\delta\mathbf{n}$. n_i is calculated from the known profile function.

There are two difficulties to be mentioned in this context due to the periodic boundary conditions. First, if reflections occur at plasma boundaries we have to cut them off entirely when shifting the simulation box. Secondly, in vacuum this limits the time-step size because otherwise the pulse would move periodically through the box instead of moving on continuously. This would result in spatial shifts of the solution.

4.1.6 Parallelization

Even though we already reduced computational costs significantly, for large problems it can be useful to have a parallel version of the method. Here we have to tailor the means of parallelization to the different cases described above.

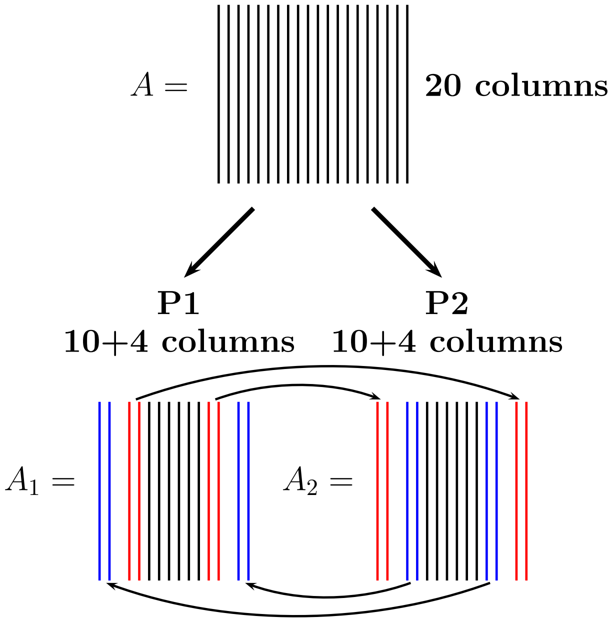


Figure 4.3: Example parallelization scheme for two processors, Cartesian coordinates in plasma, periodic boundary conditions and 20 grid points in transversal direction. The colored columns have to be communicated between the processors for the evaluation of the transversal Laplacian with finite differences and are stored twice.

Vacuum

For Cartesian coordinates in vacuum we first distribute the columns of the arrays uniformly over the processors to perform the one-dimensional fast Fourier transforms for each column. We then do a parallel transposition of the array and distribute the rows over the processors for the second part of the two-dimensional Fourier transform[1]. Of course the application of the matrix function is also spread over the processors involved.

Plasma

In plasma we basically use the same strategy for parallelization for both kinds of geometries. Here we again distribute all the columns of the arrays over the processors. But since we only need one-dimensional Fourier transforms we can avoid transposing the arrays and therefore save a lot of communication time between different processors.

The only communication between processors is due to the transversal part of the Laplacian,

[1] We use the MPI based transpose routine from FFTW version 2 and serial FFT routines from FFTW version 3.

which is discretized by fourth and second order finite differences in plasma for Cartesian and cylindrical coordinates, respectively. Thus we have to exchange at most two columns at each side of the distributed array slices. In Fig. 4.3 this is demonstrated for a matrix divided to two processors for Cartesian coordinates and periodic boundary conditions. In this case we have to store four extra columns per processor which are copied from the neighboring array.

Each processor first sends the boundary columns to the neighboring processors. Then the next time step is performed for the inner part of the array. At the end, the information sent from the neighboring arrays is used to calculate the finite difference approximation at the boundaries. This results in a parallelization which hardly suffers from communication overhead between processors, because latencies and transmission times are almost completely hidden by the asynchronous communication.

4.2 Exemplary results

4.2.1 Laplacian splitting

In this section we will demonstrate, that the error introduced by the splitting of the Laplacian is negligible. For this, we use a rather small example, where it is possible to have a high resolution reference solution to compare with. We also reduce the model and only consider the wave equation with constant density and cubic nonlinearity

$$\frac{\partial^2}{\partial t^2}a - \Delta a = -Q(1 - \frac{1}{2}|a|^2)a \ , \quad Q = 0.3 \ . \tag{4.4}$$

This is sufficient, since the splitting only affects the wave equation and does neither depend on the kind of nonlinearity nor on the density equation.

The initial conditions are chosen from

$$a(x,z,t) = a_0 e^{\frac{-(z-z_0-k_0 t)^2}{l_0^2}} e^{\frac{-x^2}{w_0^2}} e^{i(k_0 z - z_0 - t)} \tag{4.5}$$

where $a_0 = 0.15$ is the initial amplitude, $z_0 = 35\lambda_0$ the initial pulse position in longitudinal direction, $l_0 = 10\lambda_0$ the length, $w_0 = 100\lambda_0$ the width of the pulse and $k_0 = \sqrt{1-Q}$ the plasma wave length.

Equation (4.4) is solved for Cartesian coordinates $(x,z) \in [-300\lambda_0, 300\lambda_0] \times [0\lambda_0, 300\lambda_0]$ and $t \in [0/\nu_0, 300/\nu_0]$. We use 1024 grid points in z-direction and 512 grid points in x-direction. The time-step size is chosen as $0.2\Delta z$ for the spatial grid size Δz in longitudinal direction. For the reference solution we use twice as many points in both spatial directions,

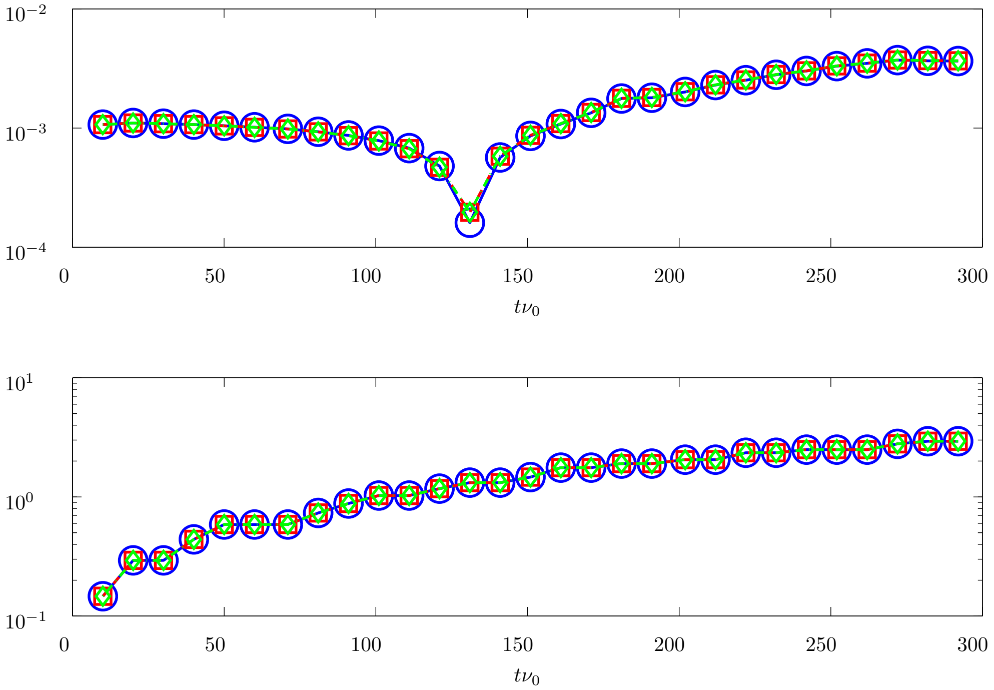


Figure 4.4: The relative error of the maximum squared amplitude is shown in the upper picture and the absolute error of the position of the maximum in wave lengths is drawn in the lower picture. The curves marked by blue circles are the errors of the Gautschi-type method applied to the full Laplacian, the red squares are the errors of the splitted method with Fourier spectral discretization in both directions and the green diamonds are those for the splitted method with finite differences in transversal direction.

whereas for the time discretization we choose one fourth of the original time step. For the error calculation we Fourier interpolate the solutions to the finer grid.

In Fig. 4.4 we can see the error in two different measures, in the upper picture the relative error in the maximum squared amplitude is shown and the lower one shows the absolute error of the position of the maximum in wave lengths. For each type of error there are three different curves. The blue circular marks show the error of the Gautschi-type method applied to the full Laplacian, discretized via Fourier spectral method in both directions. The red square marks illustrate the errors of the Gautschi-type method applied to the parallel Laplacian only and the transversal part treated as nonlinearity. Here we still use Fourier spectral methods for the discretization in both directions. The green diamond marks represent the error of the splitting method, but this time with fourth order finite differences in transversal direction. We can see, that the three error curves are nearly indistinguishable, which indicates that the splitting does not degrade accuracy.

4.2.2 Effect of different time integration schemes

We next compare our new implementation of the Gautschi-type integrator with the leap-frog scheme, which is the standard scheme used for the solution of second order wave equations.

Here, we solve the full system of equations for the two-dimensional Cartesian case (4.2). The density layer starts at $250\lambda_0$ with a linear increase up to $Q = 0.3$ over $5\lambda_0$, then it stays constant over $500\lambda_0$ until there is a linear decrease between $755\lambda_0$ and $760\lambda_0$ again.

The initial conditions are again taken from (4.5) with $a_0 = 0.12$, $z_0 = 150\lambda_0$ and $k_0 = 1$, since the pulse starts in vacuum. The remaining coefficients are the same as above.

The simulation is run up to $t = 1240/\nu_0$, thus the pulse propagates through the plasma layer and travels through vacuum afterwards for some time. For the runtime comparisons we used the moving window technique, since the simulation domain is quite long.

The solution for this example can be seen in Fig. 4.5.

In vacuum there is no need to compare the leap-frog scheme with the exact solution which the Gautschi-type integrator computes, thus we include only the time steps done inside of the plasma in the runtime comparison.

As a measure for the quality of the solution we choose the relative error of the maximum amplitude. As a sensible error threshold we use a value of 1%. Since the reference solution was computed on a finer grid, we interpolated the solution to the reference grid and then computed the maximum amplitude.

In Fig. 4.6 the amplitude error of the Gautschi-type method (red) and the leap-frog method (blue) is plotted against computation time spent in plasma regions. The dashed line represents a coarse spatial discretization with 1024 grid points in longitudinal direction, where Δz

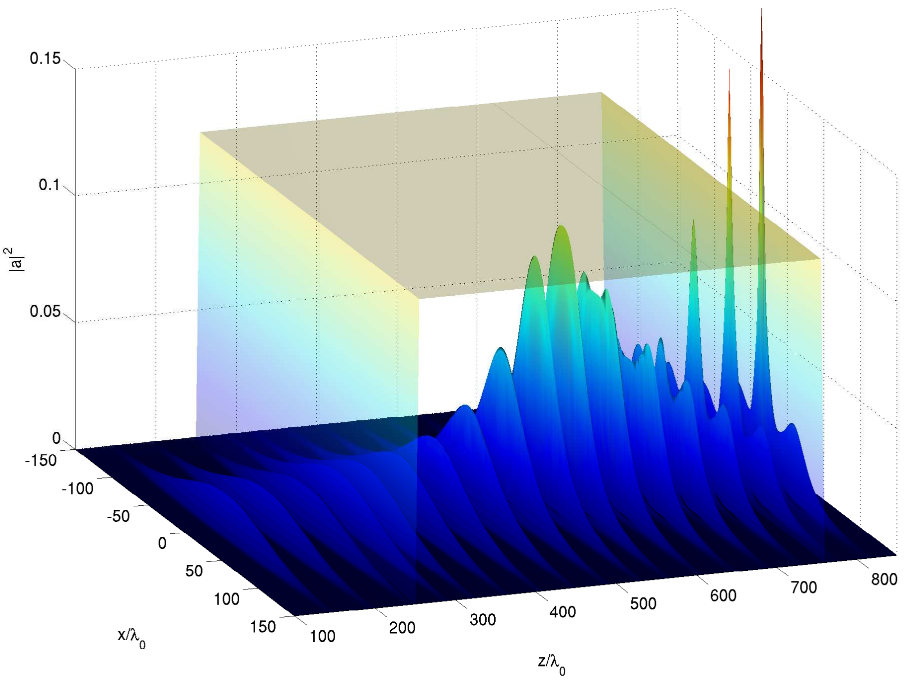


Figure 4.5: Example solution for the runtime comparison. The pulse is shown at different times of the computation.

is chosen to be $0.352\lambda_0$ and 400 grid points in transversal direction with $\Delta x = 2\lambda_0$. The continuous line gives the errors for a fine spatial discretization with $N_z = 2048$, $\Delta z = 0.176\lambda_0$, $N_x = 800$ and $\Delta x = \lambda_0$. In both cases the resulting simulation box of approximately $360\lambda_0$ in longitudinal direction and $800\lambda_0$ in transversal direction is moved along with the solution.

For the same error the temporal step size for the leap-frog method has to be about twice as small as for the Gautschi-type integrator. This is in agreement with the results in the one-dimensional case from Chapter 3. Moreover, in the two-dimensional case the advantage of the leap-frog method in terms of computational time per time step is smaller than in the one-dimensional case, because simulation times are more strongly affected by memory bandwidth limitations, see Fig. 4.7. Thus it is even more efficient to invest in a more sophisticated algorithm and benefit from the larger time steps.

4.2.3 Parallelization

To demonstrate the efficiency of the parallelized version of our code we simulated the same problem as for the runtime comparison with one, two, four, six and eight processors on a

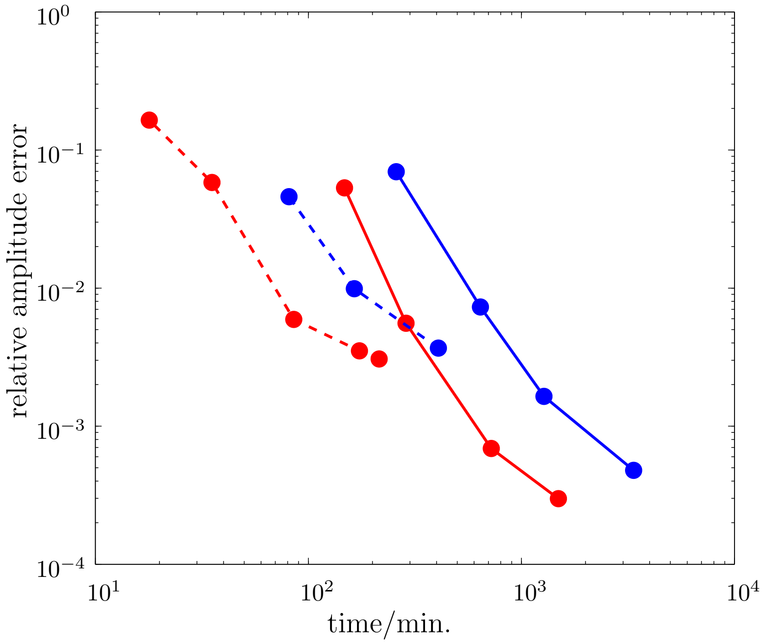


Figure 4.6: The relative error in the maximum amplitude is plotted over the runtime in minutes. Red: Gautschi-type integrator. Blue: leap-frog. Dashed: coarse spatial discretization. Solid: fine spatial discretization. Along each curve the value of τ varies.

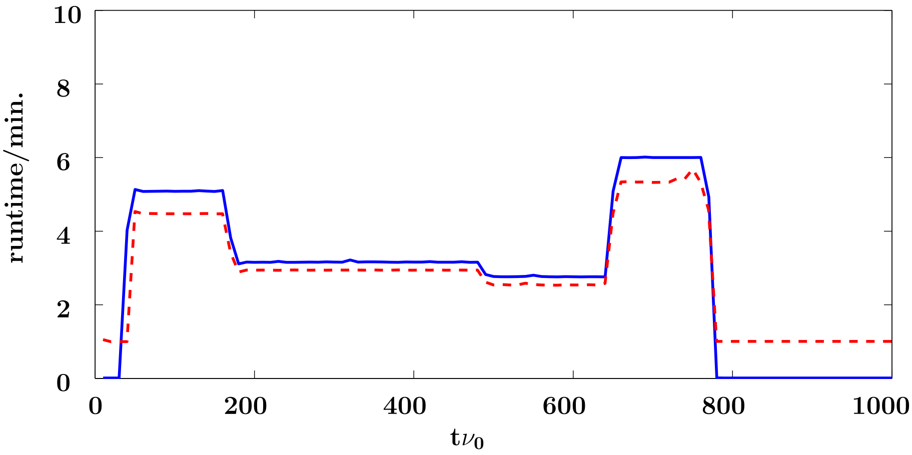


Figure 4.7: For the Gautschi-type method (blue, solid) and the leap-frog method (red, dashed) the runtime between outputs (i.e. 228 time steps, except for vacuum steps with the Gautschi-type method) is shown. The space- *and* time-step size is the same for both schemes except in vacuum.

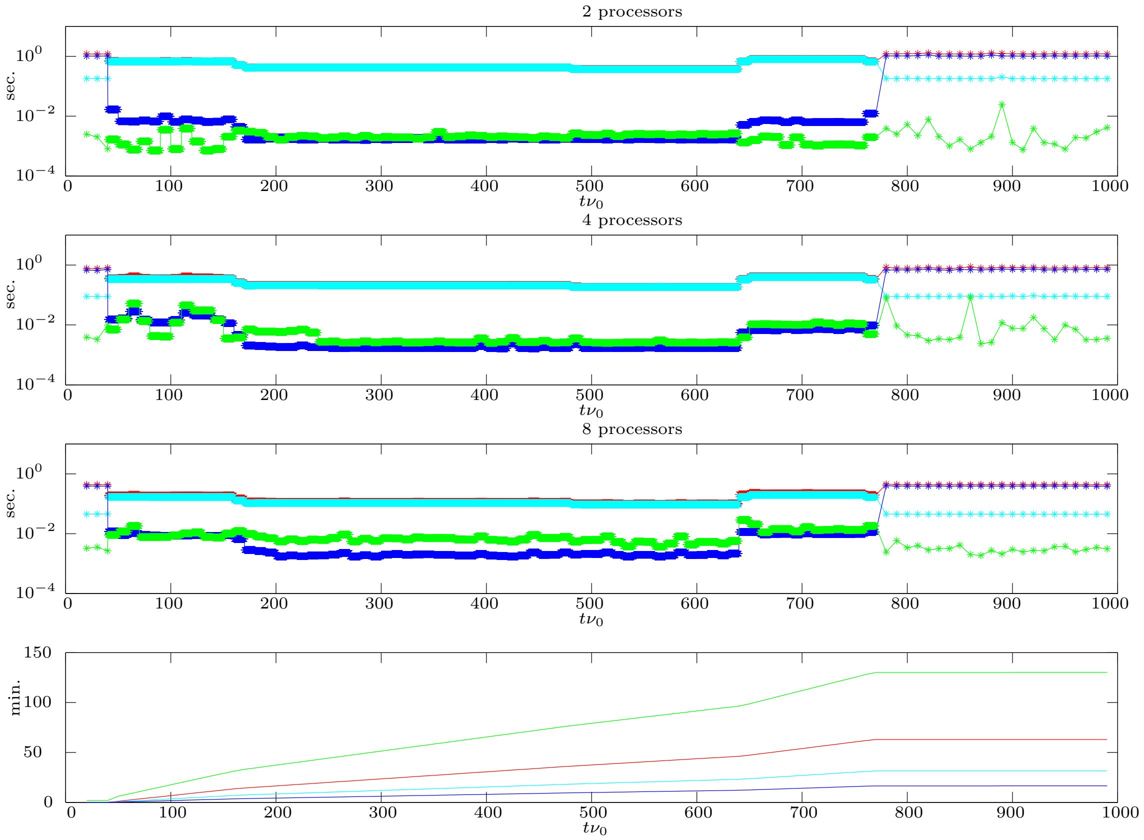


Figure 4.8: The upper three pictures show the full time (red), pure number crunching time (cyan), data receive time (blue) and synchronization time (green) per time step for two, four and eight processors respectively. The fourth picture shows the accumulated full integration time for one, two, four and eight processors (curves from top to bottom).

cluster of single CPU P4 nodes with standard Gigabit Ethernet interconnects. We used the finer one of the two spatial discretizations.

In the upper three pictures of Fig. 4.8 full time (red), pure number crunching time (cyan), data receive time (blue) and synchronization time[2] (green) per time step for two, four and eight processors, respectively, is shown. In each case we can distinguish between three different behaviors of the code. First of all there is the vacuum step region. Here, the crunching time is quite low, since we neither evaluate the nonlinearity nor solve the density equation. However, due to the matrix transpositions the communication time is rather long.

The other two cases are the plasma and the transition regions. The only difference is the spatial resolution which is higher in the transition region. However in both cases the full equations are solved and the Laplace splitting is applied. The first results in higher crunching times whereas the latter reduces the communication time significantly.

Another nice property is the very short synchronization time given by the middle gray curves. Thus independent of the number of processors used, the work is evenly balanced over the processors.

In comparison we can see, that a single vacuum time step takes longer than a single time step for the full equations, even with the higher spatial resolution in transition regions. This is compensated by the fact that the time steps in vacuum are 200 times larger than the time steps we use for solving the full set of equations. We illustrate this in the fourth picture of Fig. 4.8, where the accumulated full integration time is shown for a single processor and for two, four and eight parallel processors (curves from top to bottom). The strongest increase of computational time is in the transition region, where we use the higher spatial resolution followed directly by the plasma regions. We can also see, that the integrator spends hardly any time in vacuum regions. Note, that the length of the time steps in vacuum is only limited by points of data output and the shifting of the simulation box.

The runtime per output step is shown in Fig. 4.9. Here again the different regions of the simulation are visible. The drop in simulation time towards the end of the plasma region is due to the remaining length of the plasma layer inside the simulation box, since the density equation is only solved on those grid points which lie inside the plasma.

Another point to emphasize is the good scaling of the accumulated full integration times with the number of processors used, even for this relatively small problem. Using two processors reduces the runtime by a factor of 1.97. The runtimes for four, six and eight processors scale with 3.88, 5.65 and 7.08 respectively (see Fig. 4.10).

[2] The synchronization time is due to `MPI_Barrier()` calls after each time step.

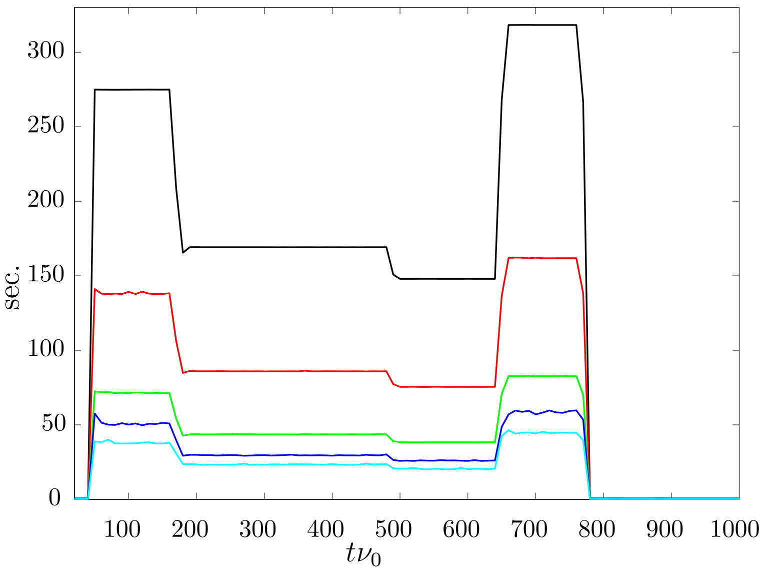


Figure 4.9: The runtime per output step for two, four, six and eight processors (red, green, blue and cyan curves) respectively compared to single processor runtime (black).

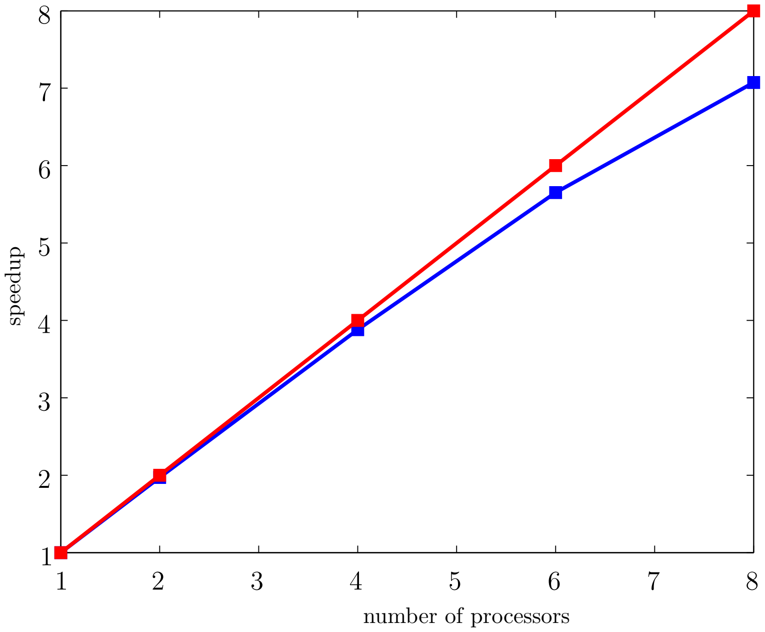


Figure 4.10: Speedup factor for 2, 4, 6 and 8 processors (blue) versus ideal scaling (red).

4.2.4 Example for a physical application

One physical application for our code is the simulation of layered plasma-vacuum structures to study the longitudinal compression and especially the transversal focusing properties of such structures. For a controllable and efficient longitudinal compression, the laser amplitude has to be subrelativistic, i.e. $a_0^2 < 1$ (where $a_0 = 1$ corresponds to $10^{18} W/cm^2$), otherwise the energy loss inside the plasma would be too large. Moreover, the spot size has to be much larger than the pulse length, otherwise the pulse would directly show collapse behavior. This implies that a high power laser pulse has to be only weakly focused initially to be in the right amplitude and spot size range. Inside the plasma the pulse is longitudinally compressed from ten or more wave-lengths to just one or two cycles [33]. To reach high subrelativistic or even relativistic intensities, the pulse has to be transversally focused as well. Fortunately the plasma induces a negative curvature of the phase front of the pulse, which leads to focusing of the pulse behind the plasma layer [31]. This focusing can be enhanced by slicing a plasma layer of optimal length for longitudinal compression into multiple shorter layers with vacuum sections in between.

An example for two layers with different lengths is shown in Fig. 4.11; the amplitude curve and the radial profiles at different times are shown in Fig. 4.12. In the vacuum regions between the layers additional transversal focusing occurs and thus the transversal focusing potential of the pulse can be fully exploited.

This leads to a much smaller spot size in the focus behind the last plasma layer (Fig. 4.14). Furthermore the pulse enters the next layer with a higher amplitude and thus the nonlinear self interaction is enhanced, too. As can be seen in Fig. 4.13, both effects combined lead to a much larger achievable intensity that increases with the number of layers. But the optimum configuration seems to be two layers, where the second layer is much thinner than the first.

An additional advantage of a layered structure is the control of transversal filamentation [33]. For this more than two layers can be necessary. Results on filamentation control and a thorough review of the focusing properties of multiple plasma layers can be found in [23].

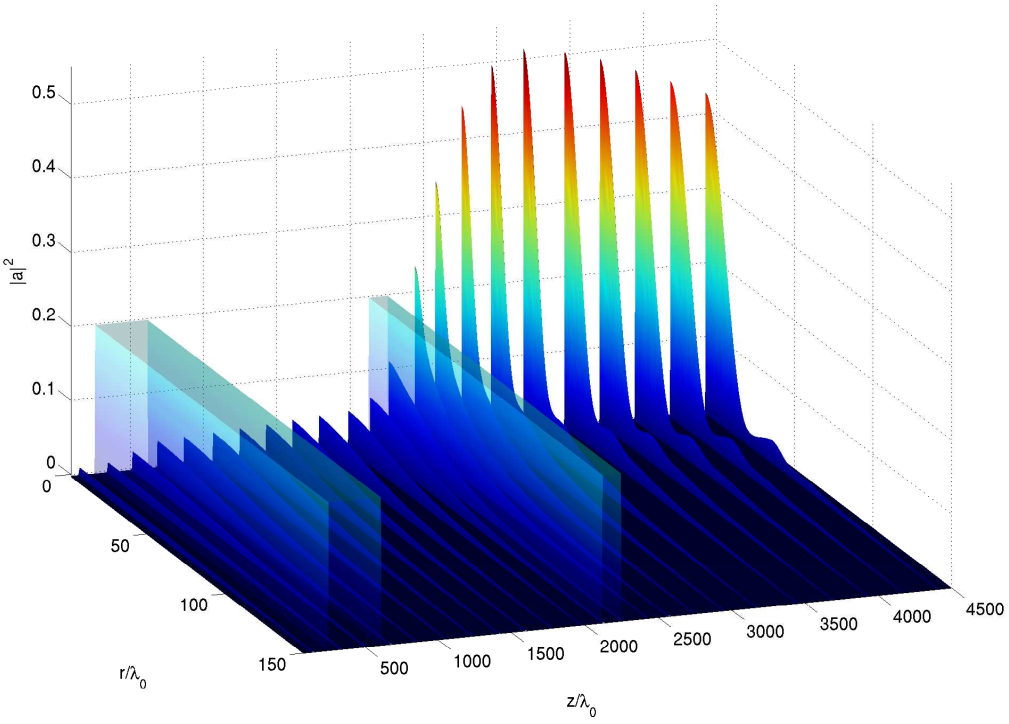


Figure 4.11: An initially (in both directions) Gaussian pulse with $a_0 = 0.1$, $L_0 = 10\lambda_0$ and $W_0 = 150\lambda_0$ propagates through two plasma layers of density $Q = 0.3$ and different lengths. The first layer is $330\lambda_0$ long and the second $125\lambda_0$ with $1500\lambda_0$ vacuum in between.

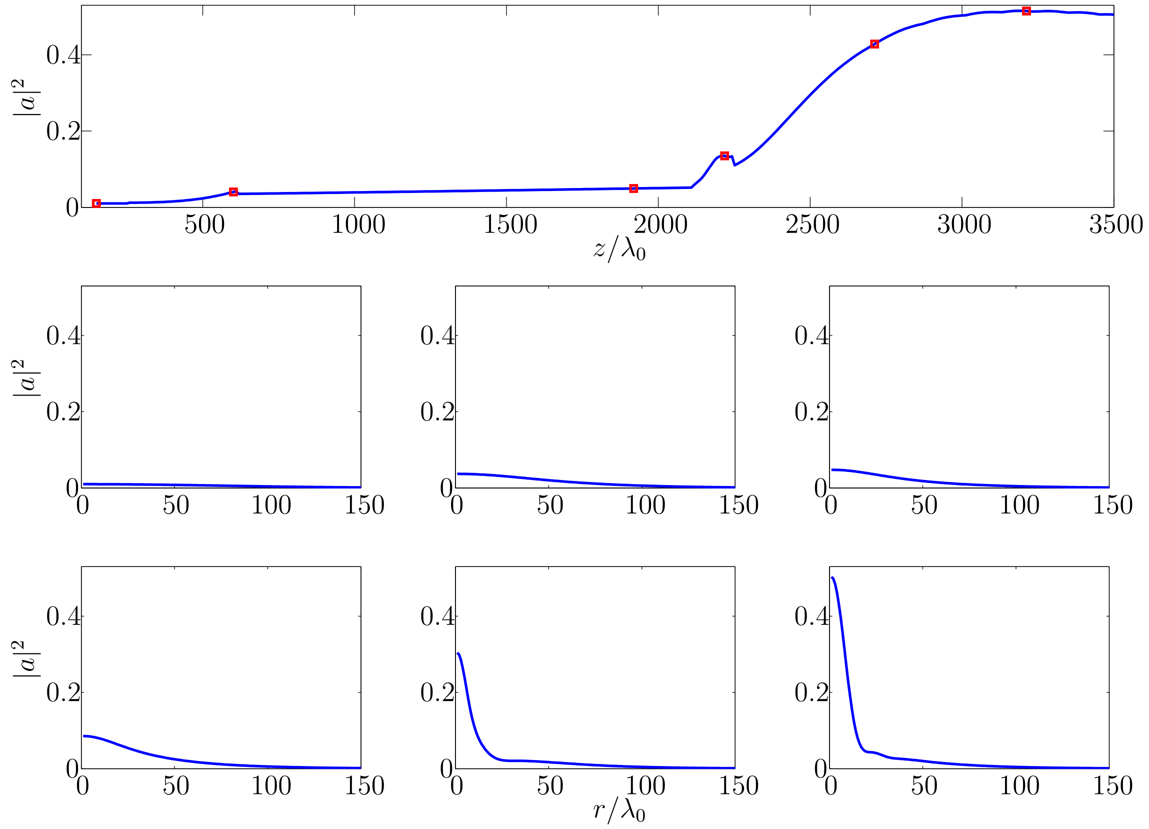


Figure 4.12: Plots of the radial profile of Fig. 4.11 at six different times/locations, marked with red squares in the plot of the maximum amplitude (top). Time evolution from left to right, top to bottom.

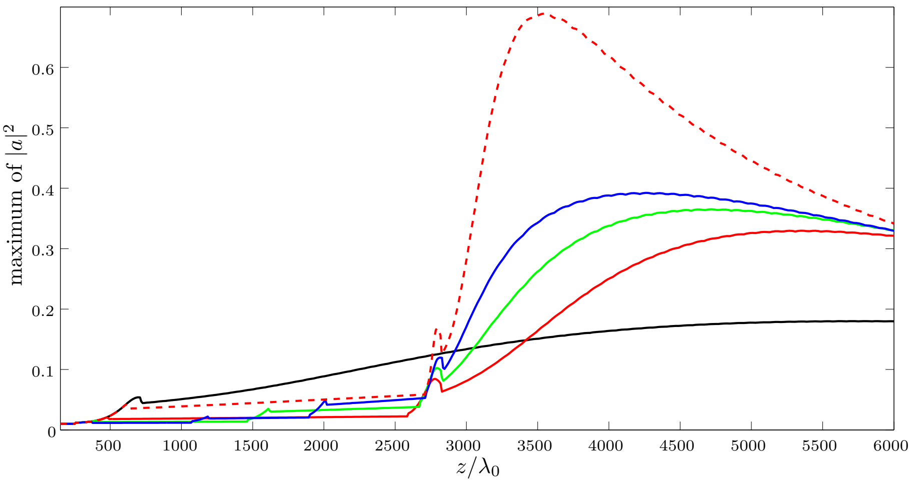


Figure 4.13: Maximum amplitude of the same initial pulse as in Fig. 4.11 as it propagates through one to four plasma layers of density $Q = 0.3$. The curves show the maximum amplitude for one layer ($460\lambda_0$, black), two layers ($227\lambda_0$ each, red solid), three layers ($150\lambda_0$ each, green), four layers ($112\lambda_0$ each, blue) and again two layers ($355\lambda_0$ and $100\lambda_0$, red dashed). In each case there is a total of $2100\lambda_0$ vacuum between the layers.

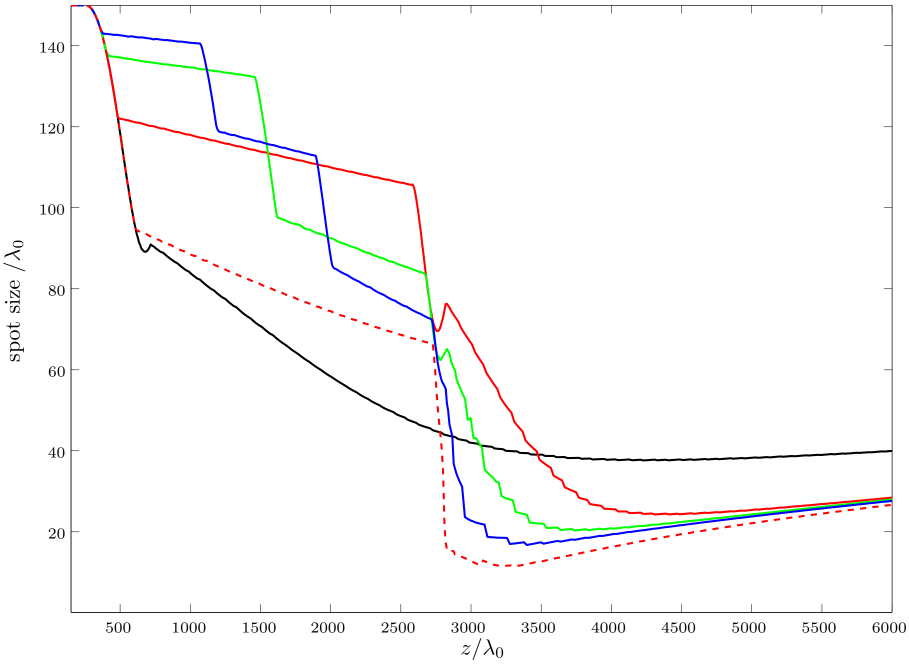


Figure 4.14: Spot size of the same initial pulse as in Fig. 4.11 propagating through the same plasma / vacuum configurations as in Fig. 4.13.

Chapter 5

Exponential Rosenbrock-type methods

Now we leave the wave equation and turn to first order systems of differential equations. We propose a new class of numerical methods for the time integration of large systems of stiff or oscillatory differential equations

$$u'(t) = F(t, u(t)) \,, \qquad u(t_0) = u_0 \,. \tag{5.1}$$

Such equations typically arise from spatial discretizations of nonlinear time dependent partial differential equations such as the Schrödinger equation, see Chapter 1.4

This chapter is based on [21] and [22].

5.1 Exponential integrators for first order systems

In this section we consider the time discretization of (possibly abstract) differential equations in autonomous form

$$u'(t) = F(u(t)) \,, \qquad u(t_0) = u_0 \,. \tag{5.2}$$

The precise assumptions on the problem class will be stated in Section 5.2 below. The numerical schemes considered are based on a continuous linearization of (5.2) along the numerical solution. For a given point u_n in the state space, this linearization is given by

$$u'(t) = J_n u(t) + g_n(u(t)) \,, \tag{5.3a}$$

$$J_n = \mathrm{D}F(u_n) = \frac{\partial F}{\partial u}(u_n) \,, \qquad g_n(u(t)) = F(u(t)) - J_n u(t) \tag{5.3b}$$

with J_n denoting the Jacobian of F evaluated at u_n, and g_n the nonlinear remainder, respectively. The numerical schemes will make *explicit* use of these quantities.

5.1.1 Method class

Let u_n denote the numerical approximation to the solution of (5.2) at time t_n. Its value at t_0 is given by the initial condition. Applying an explicit exponential Runge–Kutta scheme [20] to (5.3a), we obtain the following class of explicit one-step methods

$$U_{ni} = e^{c_i\tau_n J_n}u_n + \tau_n \sum_{j=1}^{i-1} a_{ij}(\tau_n J_n) g_n(U_{nj})\,, \qquad 1 \le i \le s\,, \tag{5.4a}$$

$$u_{n+1} = e^{\tau_n J_n}u_n + \tau_n \sum_{i=1}^{s} b_i(\tau_n J_n) g_n(U_{ni})\,, \tag{5.4b}$$

which henceforth will be called *exponential Rosenbrock methods*. Here, $\tau_n > 0$ denotes a time step, and u_{n+1} is the numerical approximation to the exact solution at time $t_{n+1} = t_n + \tau_n$.

The method is built on s internal stages U_{ni} that approximate the solution at $t_n + c_i\tau_n$. The real numbers c_i are called nodes of the method. The method is fully explicit and does not require the solution of linear or nonlinear systems of equations. As usual in exponential integrators, the weights $b_i(z)$ are linear combinations of the entire functions

$$\varphi_k(z) = \int_0^1 e^{(1-\sigma)z} \frac{\sigma^{k-1}}{(k-1)!}\,\mathrm{d}\sigma\,, \quad k \ge 1\,. \tag{5.5}$$

These functions satisfy the recurrence relations

$$\varphi_k(z) = \frac{\varphi_{k-1}(z) - \varphi_{k-1}(0)}{z}\,, \qquad \varphi_0(z) = e^z\,. \tag{5.6}$$

The coefficients $a_{ij}(z)$ will be chosen as linear combinations of the related functions $\varphi_k(c_i z)$. Without further mentioning, we will assume throughout the paper that the methods fulfill the following simplifying assumptions

$$\sum_{j=1}^{s} b_j(z) = \varphi_1(z), \qquad \sum_{j=1}^{i-1} a_{ij}(z) = c_i\varphi_1(c_i z), \quad 1 \le i \le s. \tag{5.7}$$

Note that (5.7) implies $c_1 = 0$ and consequently $U_{n1} = u_n$.

Proposition 5.1. *Exponential Rosenbrock methods* (5.4) *satisfying the simplifying assumptions* (5.7) *preserve equilibria.*

Proof. We consider an initial condition $u_0 = u^\star$ satisfying $u(t) \equiv u^\star$ for all $t > t_0$ and hence $F\big(u(t)\big) \equiv 0$. Since $U_{11} = u_0 = u^\star$ induction and the simplifying assumptions yield

$$\begin{aligned} U_{1i} &= e^{c_i\tau_1 J_1}u^\star + \tau_1 \sum_{j=1}^{i-1} a_{ij}(\tau_1 J_1)\Big(F(u^\star) - J_1 u^\star\Big) \\ &= e^{c_i\tau_1 J_1}u^\star - \tau_1 c_i \varphi_1(c_i\tau_1 J_1) J_1 u^\star\,. \end{aligned}$$

Using the recurrence relation for φ_1 we obtain $U_{1i} = u^\star$. Now we can repeat the calculation for u_1 and obtain the desired result. □

Methods that satisfy the simplifying assumptions (5.7) possess several other interesting features:

- The methods allow a reformulation for efficient implementation (Section 5.1.2).
- The inner stages have small defects and this leads to simple order conditions for stiff problems (Section 5.2.1).
- The methods can easily be extended to non-autonomous problems (Section 5.5).

5.1.2 Reformulation of the method

For the implementation of an exponential Rosenbrock method, it is crucial to approximate the multiplication of matrix functions and vectors efficiently. In contrast to the implementation of the Gautschi-type methods of the previous chapters, we cannot rely on the use of Fourier-transforms, since the linearization may lead to more complicated Jacobians than the discretized Laplacian. Moreover ideas such as the Laplacian splitting cannot be applied here, since the convergence analysis shows the necessity to use the complete linearization. Otherwise order reduction will arise.

For large matrices where fast diagonalization is not possible the method of choice is Krylov-subspace approximation. We therefore suggest to express the vectors $g_n(U_{nj})$ as

$$g_n(U_{nj}) = g_n(u_n) + D_{nj}\,, \qquad D_{nj} = g_n(U_{nj}) - g_n(u_n)\,, \qquad 2 \le j \le s\,.$$

Due to the simplifying assumptions (5.7), the method (5.4) is equivalent to

$$U_{ni} = u_n + c_i\tau_n\varphi_1(c_i\tau_n J_n)F(u_n) + \tau_n \sum_{j=2}^{i-1} a_{ij}(\tau_n J_n)D_{nj}\,, \tag{5.8a}$$

$$u_{n+1} = u_n + \tau_n\varphi_1(\tau_n J_n)F(u_n) + \tau_n \sum_{i=2}^{s} b_i(\tau_n J_n)D_{ni}\,. \tag{5.8b}$$

The main motivation for this reformulation is that the vectors D_{ni} are expected to be small, usually $\mathcal{O}(\tau_n^2)$. When computing the application of matrix functions to these vectors with some Krylov subspace method, this should be possible with very few Krylov steps. Consequently, only one computationally expensive Krylov approximation will be required in each time step, namely that involving $F(u_n)$. A similar idea has also been used in [36] and in [18] to make the code `exp4` efficient. A detailed description of the implementation is given in Section 5.7.

5.2 Analytic framework and preliminary error analysis

For the error analysis of (5.4), we work in a semigroup framework. Background information on semigroups can be found in the textbooks [4, 29]. Here, we only recall the elementary definitions.

Definition 5.2.

1. *A* strongly continuous *or* $\mathcal{C}_0$-semigroup *on a Banach space* X *is a one-parameter family* $\Gamma(t)$, $0 \le t < \infty$, *of bounded linear operators from* X *into* X *such that*
   - $\Gamma(0) = I$ *is the identity operator on* X,
   - $\Gamma(t+s) = \Gamma(t)\Gamma(s)$ *for every* $t, s \ge 0$ *and*
   - $\|\Gamma(t)x - x\| \to 0$ *as* $t \searrow 0$ *for all* $x \in X$.
2. *The* infinitesimal generator J *of a* $\mathcal{C}_0$*-semigroup* Γ *is defined by*

$$Jx = \lim_{t \searrow 0} \frac{1}{t}\big(\Gamma(t) - I\big)x$$

   whenever the limit exists. The semigroup is then denoted by e^{Jt}.

3. *The* domain of J, $D(J)$, *is the set of* $x \in X$ *for which this limit exists.*

Remark. Let J be the infinitesimal generator of a $\mathcal{C}_0$-semigroup $\Gamma(t) = e^{Jt}$. There exist constants $\omega \ge 0$ and $C \ge 1$ such that

$$\|\Gamma(t)\|_{X \leftarrow X} \le Ce^{\omega t} \qquad \text{for} \qquad 0 \le t < \infty\,. \tag{5.9}$$

Here $\|\cdot\|_{X\leftarrow X}$ denotes the norm of an operator from X to X.

Let

$$J = J(u) = \mathrm{D}F(u) = \frac{\partial F}{\partial u}(u) \tag{5.10}$$

be the Fréchet derivative of F in a neighborhood of the exact solution of (5.2). For our analysis we consider the following assumptions.

Assumption A.1. *The linear operator* J *is the generator of a strongly continuous semigroup* e^{tJ} *on a Banach space* X. *More precisely, we assume that there exist constants* $C \ge 1$ *and* $\omega \le 0$ *such that* (5.9) *holds uniformly in a neighborhood of the exact solution of* (5.2).

Recall that the analytic functions $b_i(z)$ and $a_{ij}(z)$ are linear combinations of $\varphi_k(z)$ and $\varphi_k(c_i z)$, respectively. These functions are related to the exponential function through (5.5). Assumption A.1 thus guarantees that the coefficients $b_i(\tau_n J)$ and $a_{ij}(\tau_n J)$ of the method are bounded operators. This property is crucial in our proofs.

In the subsequent analysis we restrict our attention to semilinear problems

$$u'(t) = F(u(t))\,, \qquad F(u) = Au + f(u)\,, \qquad u(t_0) = u_0\,. \tag{5.11}$$

Then J takes the form

$$J = J(u) = A + \frac{\partial f}{\partial u}(u)\,. \tag{5.12}$$

This restriction simplifies the assumptions considerably, especially the stability of discrete evolution operators (c.f. Lemma 5.6 and Theorem 5.9 below) in case of variable step sizes would be more complicated to proof in general. In many cases this assumption is not really a restriction, since problems arising form discretized parabolic partial differential equations often involve a large linear part A and a rather well behaved nonlineartity f. Nevertheless it is necessary to work with the full Jacobian.

For our case this implies that (5.3b) takes the form

$$J_n = A + \frac{\partial f}{\partial u}(u_n)\,, \qquad g_n(u(t)) = f(u(t)) - \frac{\partial f}{\partial u}(u_n)u(t)\,. \tag{5.13}$$

Our main hypothesis on the nonlinearity f is the following:

Assumption A.2. *We suppose that* (5.11) *possesses a sufficiently smooth solution* $u : [0,T] \to X$ *with derivatives in* X*, and that* $f : X \to X$ *is sufficiently often Fréchet differentiable in a strip along the exact solution. All occurring derivatives are supposed to be uniformly bounded,* $\|\cdot\|$ *denotes the norm on* X*.*

By Assumption A.2, the Jacobian (5.12) satisfies the Lipschitz condition

$$\|J(u) - J(v)\|_{X \leftarrow X} \le C\|u - v\| \tag{5.14}$$

in a neighborhood of the exact solution.

Remark. If the semigroup generated by J is even analytic, more general nonlinearities can be analyzed. To keep our presentation simple, we restrict ourselves for the moment to strongly continuous semigroups and sketch the possible extensions to analytic semigroups in Section 5.10.

5.2.1 Defects

For brevity, we write $G_n(t) = g_n(u(t))$. Inserting the exact solution into the numerical scheme gives

$$u(t_n + c_i\tau_n) = e^{c_i\tau_n J_n}u(t_n) + \tau_n \sum_{j=1}^{i-1} a_{ij}(\tau_n J_n)G_n(t_n + c_j\tau_n) + \Delta_{ni}\,, \tag{5.15a}$$

$$u(t_{n+1}) = e^{\tau_n J_n}u(t_n) + \tau_n \sum_{i=1}^{s} b_i(\tau_n J_n)G_n(t_n + c_i\tau_n) + \delta_{n+1} \tag{5.15b}$$

with defects Δ_{ni} and δ_{n+1}. The specific form of the defects is calculated by expressing the left-hand side of (5.15) by the variation-of-constants formula

$$u(t_n + c_i\tau_n) = e^{c_i\tau_n J_n}u(t_n) + \int_0^{c_i\tau_n} e^{(c_i\tau_n-\sigma)J_n}G_n(t_n+\sigma)\,\mathrm{d}\sigma$$

and then expanding G_n into a Taylor series at t_n. This yields

$$\begin{aligned}
\Delta_{ni} &= \int_0^{c_i\tau_n} e^{(c_i\tau_n-\sigma)J_n}G_n(t_n+\sigma)\,\mathrm{d}\sigma - \tau_n\sum_{j=1}^{i-1} a_{ij}(\tau_n J_n)G(t_n+c_j\tau_n)\\
&= \sum_{k=0}^{q}\Big(\int_0^{c_i\tau_n} e^{(c_i\tau_n-\sigma)J_n}\frac{\sigma^k}{k!}\,\mathrm{d}\sigma - \tau_n\sum_{j=1}^{i-1} a_{ij}(\tau_n J_n)\frac{(c_j\tau_n)^k}{k!}\Big)G_n^{(k)}(t_n) + \Delta_{ni}^{[q+1]}\\
&= \sum_{k=0}^{q}\tau_n^{k+1}\Big(c_i^{k+1}\int_0^1 e^{(1-\sigma)c_i\tau_n J_n}\frac{\sigma^k}{k!}\,\mathrm{d}\sigma - \sum_{j=1}^{i-1} a_{ij}(\tau_n J_n)\frac{c_j^k}{k!}\Big)G_n^{(k)}(t_n) + \Delta_{ni}^{[q+1]}\\
&= \sum_{k=0}^{q}\tau_n^{k+1}\psi_{k+1,i}(\tau_n J_n)G_n^{(k)}(t_n) + \Delta_{ni}^{[q+1]} \qquad (5.16)
\end{aligned}$$

with

$$\psi_{k,i}(z) = \varphi_k(c_i z)c_i^k - \sum_{j=1}^{i-1} a_{ij}(z)\frac{c_j^{k-1}}{(k-1)!} \tag{5.17}$$

and remainders

$$\begin{aligned}
\Delta_{ni}^{[q+1]} = {}& \int_0^{c_i\tau_n} e^{(c_i\tau_n-\sigma)J_n}\int_0^s \frac{(\sigma-s)^q}{q!}G_n(t_n+s)\,\mathrm{d}s\,\mathrm{d}\sigma\\
&- \tau_n\sum_{j=1}^{i-1} a_{ij}(\tau_n J_n)\int_0^{c_j\tau_n}\frac{(c_j\tau_n-s)^q}{q!}G_n(t_n+s)\,\mathrm{d}s\,.
\end{aligned}$$

Small defects in the internal stages facilitate our convergence proofs considerably. This gives a further reason for requiring (5.7) which implies $\psi_{1,i}(z) \equiv 0$. Unfortunately, explicit methods *cannot* have $\psi_{2,i}(z) \equiv 0$ for all i. Nevertheless, the second term on the right-hand side of (5.16) turns out to be small. This is seen from the identity

$$G_n'(t_n) = \frac{\partial g_n}{\partial u}\big(u(t_n)\big)u'(t_n) = \left(\frac{\partial f}{\partial u}\big(u(t_n)\big) - \frac{\partial f}{\partial u}(u_n)\right) u'(t_n)\,,$$

which itself is a consequence of linearizing at each step, cf. (5.13). By Assumption A.2 this relation implies

$$\|G_n'(t_n)\|_{X \leftarrow X} \le C\|e_n\| \tag{5.18}$$

with $e_n = u_n - u(t_n)$. $\Delta_{ni}^{[2]}$ satisfies

$$\big\|\Delta_{ni}^{[2]}\big\| \le C\tau_n^3\,. \tag{5.19}$$

and the defects of the internal stages thus obey the bound

$$\|\Delta_{ni}\| \le C\tau_n^2\|e_n\| + C\tau_n^3\,. \tag{5.20}$$

Similarly, we get for the defects δ_{n+1} at time t_{n+1}

$$\delta_{n+1} = \sum_{k=0}^{q} \tau_n^{k+1}\psi_{k+1}(\tau_n J_n)G_n^{(k)}(t_n) + \delta_{n+1}^{[q+1]}\,, \tag{5.21}$$

with

$$\psi_k(z) = \varphi_k(z) - \sum_{i=1}^{s} b_i(z)\frac{c_i^{k-1}}{(k-1)!} \tag{5.22}$$

and remainders $\delta_{n+1}^{[q]}$ satisfying

$$\big\|\delta_{n+1}^{[q]}\big\| \le C\tau_n^{q+1}\,. \tag{5.23}$$

Again, small defects are desirable. Due to (5.7), we have $\psi_1(z) \equiv 0$. To obtain higher order bounds for δ_{n+1} first observe that the τ^2-term in (5.21) is small due to (5.18). Additional terms vanish if $\psi_j = 0$, $j \ge 3$.

All conditions encountered so far are collected in Table 5.1. They will later turn out to be the order conditions for methods up to order 4.

Lemma 5.3. *If the order conditions of Table* 5.1 *are satisfied up to order* $p \le 4$, *we obtain*

$$\|\delta_{n+1}\| \le C\tau_n^2\|e_n\| + C\tau_n^{p+1}\,. \tag{5.24}$$

Proof. This follows at once from (5.21). □

No.	condition in defect	order condition	order
1	$\psi_1(z) \equiv 0$	$\sum_{i=1}^{s} b_i(z) = \varphi_1(z)$	1
2	$\psi_{1,i}(z) \equiv 0$	$\sum_{j=1}^{i-1} a_{ij}(z) = c_i\varphi_1(c_i z), \quad 2 \le i \le s$	2
3	$\psi_3(z) \equiv 0$	$\sum_{i=2}^{s} b_i(z)c_i^2 = 2\varphi_3(z)$	3
4	$\psi_4(z) \equiv 0$	$\sum_{i=2}^{s} b_i(z)c_i^3 = 6\varphi_4(z)$	4

Table 5.1: Stiff order conditions for exponential Rosenbrock methods applied to autonomous problems.

5.2.2 Preliminary error bounds

Let

$$e_n = u_n - u(t_n) \quad \text{and} \quad E_{ni} = U_{ni} - u(t_n + c_i\tau_n)$$

denote the differences between the numerical solution and the exact solution. Subtracting (5.15) from the numerical method (5.4) gives the error recursion

$$E_{ni} = e^{c_i\tau_n J_n} e_n + \tau_n \sum_{j=1}^{i-1} a_{ij}(\tau_n J_n)\Big(g_n(U_{nj}) - G_n(t_n + c_j\tau_n)\Big) - \Delta_{ni}\,, \tag{5.25a}$$

$$e_{n+1} = e^{\tau_n J_n} e_n + \tau_n \sum_{i=1}^{s} b_i(\tau_n J_n)\Big(g_n(U_{ni}) - G_n(t_n + c_i\tau_n)\Big) - \delta_{n+1}\,. \tag{5.25b}$$

In the following we will derive bounds for these errors.

Lemma 5.4. *If Assumption* A.2 *is satisfied, we have*

$$\Big\| \frac{\partial g_n}{\partial u}\big(u(t_n)\big) \Big\|_{X \leftarrow X} \le C \left\| e_n \right\| \,, \tag{5.26a}$$

$$\left\| g_n(u_n) - G_n(t_n) \right\| \le C \left\| e_n \right\|^2 \,, \tag{5.26b}$$

$$\left\| g_n(U_{ni}) - G_n(t_n + c_i\tau_n) \right\| \le C\big(\tau_n + \|e_n\| + \|E_{ni}\|\big) \left\| E_{ni} \right\| \,, \tag{5.26c}$$

as long as the errors E_{ni} and e_n remain in a sufficiently small neighborhood of 0.

Proof. The bound for (5.26a),

$$\frac{\partial g_n}{\partial u}\big(u(t_n)\big) = \frac{\partial f}{\partial u}\big(u(t_n)\big) - \frac{\partial f}{\partial u}(u_n)\,,$$

is a direct consequence of the linearization and a Lipschitz condition.

Using Taylor series expansion at $u(t_n + c_i\tau_n)$, we get

$$\begin{aligned} g_n(U_{ni}) - G_n(t_n + c_i\tau_n) = {} & \frac{\partial g_n}{\partial u}\big(u(t_n + c_i\tau_n)\big)E_{ni} \\ & + \int_0^1 (1-\sigma)\frac{\partial^2 g_n}{\partial u^2}\big(u(t_n + c_i\tau_n) + \sigma E_{ni}\big)(E_{ni}, E_{ni})\,\mathrm{d}\sigma\,. \end{aligned}$$

Setting $i = 1$ and using (5.26a) proves (5.26b) at once. To derive (5.26c), we expand the first term on the right-hand side once more at t_n,

$$\begin{aligned} \frac{\partial g_n}{\partial u}\big(u(t_n + c_i\tau_n)\big)E_{ni} = {} & \frac{\partial g_n}{\partial u}\big(u(t_n)\big)E_{ni} \\ & + \int_0^1 c_i\tau_n(1-\sigma)\frac{\partial^2 g_n}{\partial u^2}\big(u(t_n + \sigma c_i\tau_n)\big)\big(u'(t_n + \sigma c_i\tau_n), E_{ni}\big)\,\mathrm{d}\sigma\,, \end{aligned}$$

and use (5.26a) again to finally prove (5.26c). □

Using this result, we can establish an error bound for the internal stages.

Lemma 5.5. *Under Assumptions* A.1 *and* A.2 *we have*

$$\|E_{ni}\| \le C\,\|e_n\| + C\tau_n^3\,,$$

as long as the global errors e_n remain in a bounded neighborhood of 0.

Proof. The assertion follows from (5.25a). For $i = 1$ we have

$$\|E_{n1}\| = \|e_n\|\,.$$

This yields

$$\|E_{n2}\| \le C\|e_n\| + C\tau_n\|e_n\|^2 + C\tau_n^2\|e_n\| + C\tau_n^3\,.$$

For $i = 3, \dots, s$ we insert $\|E_{nj}\|$ recursively and obtain

$$\begin{aligned} \|E_{ni}\| &\le \|e^{c_i\tau_n J_n} e_n\| + \tau_n \sum_{j=1}^{i-1} \Big\| a_{ij}(\tau_n J_n)\big(g_n(U_{nj}) - G_n(t_n + c_j\tau_n)\big)\Big\| + \|\Delta_{ni}\| \\ &\le C\|e_n\| + C\tau_n \sum_{j=1}^{i-1} \big(\tau_n + \|e_n\| + \|E_{nj}\|\big)\|E_{nj}\| + C\tau_n^2\|e_n\| + C\tau_n^3 \\ &\le C\|e_n\| + C\tau_n^3 \end{aligned}$$

using Lemma 5.4 and (5.20). □

5.2.3 Stability bounds

In order to establish convergence bounds, we have to solve recursion (5.25b). For this purpose, stability bounds for the discrete evolution operators are crucial. In a first step we will show stability along the exact solution.

We start with two auxiliary results.

Lemma 5.6. *Let the initial value problem* (5.11) *satisfy Assumptions* A.1 *and* A.2, *and let* $\widehat{J}_n = \mathrm{D}F(u(t_n))$. *Then, for any* $\widehat{\omega} > \omega$, *there exists a constant* C_{L} *independent of* τ_{n-1} *such that*

$$\big\| e^{t\widehat{J}_n} - e^{t\widehat{J}_{n-1}} \big\|_{X \leftarrow X} \le C_{\mathrm{L}} \tau_{n-1} e^{\widetilde{\omega} t}, \qquad t \ge 0. \tag{5.27}$$

Proof. Applying the variation-of-constants formula to the initial value problem

$$v'(t) = \widehat{J}_n v(t) = \widehat{J}_{n-1} v(t) + \big(\widehat{J}_n - \widehat{J}_{n-1}\big) v(t)$$

with $v(0) = v_0$ yields

$$v(t) = e^{t\widehat{J}_n} v_0 = e^{t\widehat{J}_{n-1}} v_0 + \int_0^1 e^{t\widehat{J}_{n-1}(1-\sigma)} \big(\widehat{J}_n - \widehat{J}_{n-1}\big) v(\sigma t) \, \mathrm{d}\sigma .$$

This implies the representation

$$e^{t\widehat{J}_n} - e^{t\widehat{J}_{n-1}} = \int_0^1 t e^{(1-\sigma)t\widehat{J}_{n-1}} \big(\widehat{J}_n - \widehat{J}_{n-1}\big) e^{\sigma t \widehat{J}_n} \, \mathrm{d}\sigma . \tag{5.28}$$

The required estimate now follows from (5.14) and the smoothness of $u(t)$. □

Lemma 5.7. *Under the assumptions of Lemma* 5.6, *the relation*

$$|||x|||_n = \sup_{t \ge 0} e^{-\widetilde{\omega} t} \big\| e^{t\widehat{J}_n} x \big\| , \qquad x \in X \tag{5.29}$$

defines a norm on X *for any* $n = 0, 1, 2, \ldots$. *This norm is equivalent to* $\|\cdot\|$ *and satisfies the bound*

$$|||x|||_n \le (1 + C_{\mathrm{L}} \tau_{n-1}) \, |||x|||_{n-1} , \qquad n \ge 1 . \tag{5.30}$$

Proof. Obviously, we have

$$\|x\| = e^{-\widetilde{\omega}\cdot 0} \big\| e^{0\cdot \widehat{J}_n} x \big\| \le \sup_{t \ge 0} e^{-\widetilde{\omega} t} \big\| e^{t\widehat{J}_n} x \big\| = |||x|||_n .$$

On the other hand, the bound (5.9) yields $|||x|||_n \le C\,\|x\|$. Thus, the two norms are equivalent.

For arbitrary $x \in X$, we have

$$\begin{aligned}
|||x|||_n &= \sup_{t\ge 0} e^{-\widetilde{\omega}t}\big\|\big(e^{t\widehat{J}_n} - e^{t\widehat{J}_{n-1}} + e^{t\widehat{J}_{n-1}}\big)x\big\| \\
&\le |||x|||_{n-1} + \sup_{t\ge 0} e^{-\widetilde{\omega}t}\big\|e^{t\widehat{J}_n} - e^{t\widehat{J}_{n-1}}\big\|_{X\leftarrow X}\|x\| \\
&\le (1 + C_{\mathrm{L}}\tau_{n-1})\,|||x|||_{n-1}
\end{aligned}$$

by Lemma 5.6 and the equivalence of the norms. □

The following lemma proves the stability of the discrete evolution operators along the exact solution.

Lemma 5.8. *Under the assumptions of Lemma* 5.6*, there exists a constant C such that*

$$\big\|e^{\tau_n\widehat{J}_n}\cdot\ldots\cdot e^{\tau_0\widehat{J}_0}\big\|_{X\leftarrow X} \le C\,e^{\widehat{\omega}(\tau_0+\ldots+\tau_n)} \tag{5.31}$$

with $\widehat{\omega} = C_{\mathrm{L}} + \widetilde{\omega}$.

Proof. By (5.29) and (5.9) we have

$$\begin{aligned}
\Big|\Big|\Big|e^{\tau_n\widehat{J}_n}\cdot\ldots\cdot e^{\tau_0\widehat{J}_0}x\Big|\Big|\Big|_n &= \sup_{t\ge 0} e^{-\widetilde{\omega}t}\big\|e^{t\widehat{J}_n}e^{\tau_n\widehat{J}_n}\cdot\ldots\cdot e^{\tau_0\widehat{J}_0}x\big\| \\
&= \sup_{t\ge 0} e^{-\widetilde{\omega}t}\big\|e^{t\widehat{J}_n}e^{\widetilde{\omega}\tau_n}e^{-\widetilde{\omega}\tau_n}e^{\tau_n\widehat{J}_n}\cdot\ldots\cdot e^{\tau_0\widehat{J}_0}x\big\| \\
&\le C\sup_{t\ge 0} e^{-\widetilde{\omega}t}\big\|e^{t\widehat{J}_n}e^{\widetilde{\omega}\tau_n}e^{\tau_{n-1}\widehat{J}_{n-1}}\cdot\ldots\cdot e^{\tau_0\widehat{J}_0}x\big\| \\
&= e^{\widetilde{\omega}\tau_n}\Big|\Big|\Big|e^{\tau_{n-1}\widehat{J}_{n-1}}\cdot\ldots\cdot e^{\tau_0\widehat{J}_0}x\Big|\Big|\Big|_n \\
&\le e^{\widetilde{\omega}\tau_n}(1 + C_{\mathrm{L}}\tau_{n-1})\Big|\Big|\Big|e^{\tau_{n-1}\widehat{J}_{n-1}}\cdot\ldots\cdot e^{\tau_0\widehat{J}_0}x\Big|\Big|\Big|_{n-1},
\end{aligned}$$

where the last inequality follows from Lemma 5.7. Thus, the estimate $1 + C_{\mathrm{L}}\tau_{n-1} \le e^{C_{\mathrm{L}}\tau_{n-1}}$ together with an induction argument proves the lemma. □

We now turn our attention to the operators $J_n = \mathrm{D}F(u_n)$ that result from the linearization process (5.3). These operators constitute an essential component of the numerical scheme (5.4). We now repeat the above estimations with J_n in the role of $\widehat{J}_n$ to state the following stability result for the discrete evolution operators on X.

Theorem 5.9. *Let the initial value problem* (5.11) *satisfy Assumptions* A.1 *and* A.2. *Then, for any $\widetilde{\omega} > \omega$, there exist constants C and $C_{\mathtt{E}}$ such that*

$$\left\| e^{\tau_n J_n} \cdot \ldots \cdot e^{\tau_0 J_0} \right\|_{X \leftarrow X} \le C e^{\widehat{\omega}(\tau_0 + \ldots + \tau_n) + C_{\mathtt{E}} \sum_{j=1}^{n} \|e_j\|} \tag{5.32}$$

with $\widehat{\omega} = C_{\mathtt{L}} + \widetilde{\omega}$. The bound holds as long as the numerical solution u_n stays in a sufficiently small neighborhood of the exact solution of (5.11).

Proof. By (5.14) we have

$$\|J_n - J_{n-1}\|_{X \leftarrow X} \le C \|u_n - u_{n-1}\| .$$

The triangle inequality shows that

$$\|u_n - u_{n-1}\| = \|u_n - u(t_n) + u(t_n) - u(t_{n-1}) + u(t_{n-1}) - u_{n-1}\| \le C\tau_{n-1} + \|e_n\| + \|e_{n-1}\| . \tag{5.33}$$

Following the proof of Lemma 5.6 with J_{n-1} and J_n instead of $\widehat{J}_{n-1}$ and $\widehat{J}_n$, we have

$$\left\| e^{tJ_n} - e^{tJ_{n-1}} \right\|_{X \leftarrow X} \le C_{\mathtt{L}}(\tau_{n-1} + \|e_n\| + \|e_{n-1}\|) e^{\widetilde{\omega} t} , \qquad t \ge 0 .$$

Obviously $|||\cdot|||_n$ with respect to J_n instead of $\widehat{J}_n$ is still a norm which is also equivalent to $\|\cdot\|$. Only the last estimate of Lemma 5.7 changes to

$$|||x|||_n \le (1 + C_{\mathtt{L}}(\tau_n + \|e_n\| + \|e_{n-1}\|))\, |||x|||_{n-1}$$

which in turn leads to the modified stability estimate for the discrete evolution operators along the numerical solution using the same induction argument as in the proof of Lemma 5.8. □

The stability bound (5.32) requires some attention. Strictly spoken, stability is only guaranteed if the term $\sum_{j=1}^{n} \|e_j\|$ is uniformly bounded in n for $t_0 \le t_n \le T$. This condition can be considered as a (weak) restriction on the employed step size sequence, see the discussion in Section 5.3 below.

5.3 Error bounds

We will now show that the conditions of Table 5.1 are sufficient to obtain convergence up to order 4 under a mild restriction on the employed step size sequence.

Theorem 5.10. *Let the initial value problem* (5.11) *satisfy Assumptions* A.1 *and* A.2. *Consider for its numerical solution an explicit exponential Rosenbrock method* (5.4) *that fulfills the order conditions of Table* 5.1 *up to order* p *for some* $2 \le p \le 4$. *Further, let the step size sequence* τ_j *satisfy the condition*

$$\sum_{k=1}^{n} \sum_{j=0}^{k-1} \tau_j^{p+1} \le C_{\mathrm{T}} \tag{5.34}$$

with a constant C_{T} *that is uniform in* $t_0 \le t_n \le T$. *Then, for* C_{T} *sufficiently small, the numerical method converges with order* p. *In particular, the numerical solution satisfies the error bound*

$$\|u_{n+1} - u(t_{n+1})\| \le C \sum_{j=0}^{n} \tau_j^{p+1} \tag{5.35}$$

uniformly on $t_0 \le t_{n+1} \le T$. *The constant* C *is independent of the chosen step size sequence satisfying* (5.34).

Proof. From (5.25b) we obtain the error recursion

$$e_{n+1} = e^{\tau_n J_n} e_n + \tau_n \varrho_n - \delta_{n+1}\,, \qquad e_0 = 0 \tag{5.36}$$

with

$$\varrho_n = \sum_{i=1}^{s} b_i(\tau_n J_n) \Big(g_n(U_{ni}) - G_n(t_n + c_i \tau_n) \Big)\,.$$

Solving this recursion and using $e_0 = 0$ yields

$$e_{n+1} = \sum_{j=0}^{n} \tau_j e^{\tau_n J_n} \cdot \ldots \cdot e^{\tau_{j+1} J_{j+1}} \big(\varrho_j - \tau_j^{-1} \delta_{j+1} \big)\,. \tag{5.37}$$

Employing the Lemmas 5.3, 5.4 and 5.5, we obtain the bound

$$\begin{aligned}
\|\varrho_j\| + \tau_j^{-1} \|\delta_{j+1}\| &\le C \sum_{i=1}^{s} \big(\tau_j + \|e_j\| + \|E_{ji}\| \big) \|E_{ji}\| + C\tau_j \|e_j\| + C\tau_j^p \\
&\le C\big(\tau_j + \|e_j\| + C\|e_j\| + C\tau_j^3 \big) \big(C\|e_j\| + C\tau_j^3 \big) + C\tau_j \|e_j\| + C\tau_j^p \\
&\le C\big(\tau_j \|e_j\| + \|e_j\|^2 + \tau_j^p \big)\,.
\end{aligned}$$

Inserting this into (5.37) and using the stability estimate (5.32) yields

$$\|e_{n+1}\| \le C \sum_{j=0}^{n} \tau_j \big(\tau_j \|e_j\| + \|e_j\|^2 + \tau_j^p \big)\,. \tag{5.38}$$

The constant in this estimate is uniform as long as

$$\sum_{j=1}^{n} \|e_j\| \le C_{\mathtt{A}} \tag{5.39}$$

uniformly holds on $t_0 \le t_{n+1} \le T$. The application of the discrete Gronwall Lemma 5.12 below to (5.38) then shows the desired bound (5.35).

It still remains to verify that condition (5.39) holds with a uniform bound $C_{\mathtt{A}}$. This follows now recursively from summing over (5.35) for $k \le n$,

$$\sum_{k=1}^{n} \|e_k\| \le C \sum_{k=1}^{n} \sum_{j=0}^{k-1} \tau_j^{p+1} \,,$$

and our assumption on the step size sequence (5.34) with $C_{\mathtt{T}}$ sufficiently small. □

Lemma 5.11. (Gronwall-Lemma) *Let φ be a continuous function with*

$$\varphi(t) \le \alpha + \beta \int_{t_0}^{t} \varphi(x)\,\mathrm{d}x\,, \quad t_0 \le t \le t_1$$

and constants α, $\beta \ge 0$, then

$$\varphi(t) \le \alpha e^{\beta(t-t_0)}\,.$$

Proof. For

$$\psi(t) = \alpha + \beta \int_{t_0}^{t} \varphi(x)\,\mathrm{d}x$$

we have $\psi'(t) = \beta\varphi(t)$ and using the assumption on φ we obtain

$$\psi'(t) \le \beta\psi(t)\,.$$

From this we can deduce

$$\big(\psi(t)e^{-\beta t}\big)' = e^{-\beta t}\big(\psi'(t) - \beta\psi(t)\big) \le 0\,.$$

The monotonic decrease of $\psi(t)e^{-\beta t}$ then yields

$$\varphi(t)e^{-\beta t} \le \psi(t)e^{-\beta t} \le \psi(t_0)e^{-\beta t_0} = \alpha e^{-\beta t_0}$$

which is the desired inequality. □

Lemma 5.12. (Discrete Gronwall-Lemma) *Let $\{\varepsilon_n\}_{n\geq 1}$ and $\{\tau_n\}_{n\geq 1}$ be sequences of non negative numbers satisfying*

$$\varepsilon_{n+1} \leq \alpha + \beta \sum_{k=1}^{n} \tau_k \varepsilon_k$$

and constants α, $\beta \geq 0$, then

$$\varepsilon_{n+1} \leq \alpha e^{\beta \sum_{k=1}^{n} \tau_k} .$$

Proof. For this proof, we choose a piecewise linear function $\varphi(t)$ satisfying the assumptions of Lemma 5.11, $\varphi(T) = \varepsilon_{n+1}$ for $T = t_0 + \sum_{k=1}^{n} \tau_k$ and

$$\int_{t_0}^{T} \varphi(x) \, \mathrm{d}x = \sum_{k=1}^{n} \tau_k \varepsilon_k .$$

For $t_k = t_0 + \sum_{j=1}^{k} \tau_j$, $k = 1, \ldots, n$, $\mathcal{T}_k = \min\{\tau_{k-1}, \tau_k\}/2$, $k = 2, \ldots, n-1$ and $\mathcal{T}_0 = \min\{\tau_1, \tau_n\}/2$ the function

$$\varphi(t) = \begin{cases} \varepsilon_k, & t_{k-1} + \frac{\mathcal{T}_{k-1}}{2} \leq t \leq t_k - \frac{\mathcal{T}_k}{2} \\ \frac{\varepsilon_{k+1} - \varepsilon_k}{\mathcal{T}_k}(t - t_k) + \frac{\varepsilon_{k+1} + \varepsilon_k}{2}, & t_k - \frac{\mathcal{T}_k}{2} \leq t \leq t_k + \frac{\mathcal{T}_k}{2} \\ \frac{2(\varepsilon_{n+1} - \varepsilon_n)}{\mathcal{T}_0}(t - t_0) + \varepsilon_1 + \varepsilon_{n+1} - \varepsilon_n, & t_0 \leq t \leq t_0 + \frac{\mathcal{T}_0}{2} \\ \frac{2(\varepsilon_{n+1} - \varepsilon_n)}{\mathcal{T}_0}(t - T) + \varepsilon_{n+1}, & T - \frac{\mathcal{T}_0}{2} \leq t \leq T \end{cases}$$

can be used. □

In the remainder of this section, we discuss the encountered restriction (5.34) on the step size sequence.

Lemma 5.13. *If the step size sequence fulfills one of the following conditions, then* (5.34) *holds.*

1. *For* constant *step size $\tau_j = \tau$* (5.34) *holds with*

$$C_{\mathrm{T}} = \frac{\tau^{p-1}}{2} (t_{n+1} - t_0)^2 .$$

2. *For a* quasi-uniform *step size sequence where the ratio C between the maximal and minimal step length is uniformly bounded* (5.34) *holds with*

$$C_{\mathrm{T}} = C^{p+1} \frac{\tau_{\min}^{p-1}}{2} (t_{n+1} - t_0)^2 .$$

3. *For sequences with increasing step sizes $\tau_0 \le \tau_1 \le \ldots \le \tau_{n-1}$, condition* (5.34) *is fulfilled with*
$$C_{\mathrm{T}} = (t_{n+1} - t_0)^2 \tau_{n-1}^{p-1}\,.$$

Since $p \ge 2$, C_{T} tends to zero for $n \to \infty$ in cases 1 and 2. In the third case C_{T} is still a constant.

Proof.

1. For constant step size $\tau_j = \tau$ (5.34) takes the form
$$\sum_{k=1}^{n}\sum_{j=0}^{k-1} \tau^{p+1} = \frac{n(n+1)}{2}\tau^{p+1} \le \frac{(n+1)^2}{2}\tau^{p+1} = \frac{\tau^{p-1}}{2}(t_{n+1}-t_0)^2\,.$$

2. For a *quasi-uniform* step size sequences, we have $\tau_j \le C\tau_{\min}$, which yields
$$\sum_{k=1}^{n}\sum_{j=0}^{k-1} \tau_j^{p+1} \le \sum_{k=1}^{n}\sum_{j=0}^{k-1} C^{p+1}\tau_{\min}^{p+1} = \frac{n(n+1)}{2}C^{p+1}\tau_{\min}^{p+1} \le C^{p+1}\frac{\tau_{\min}^{p-1}}{2}(t_{n+1}-t_0)^2\,.$$

3. For sequences with increasing step sizes $\tau_0 \le \tau_1 \le \ldots \le \tau_{n-1}$, we have
$$\begin{aligned}\sum_{k=1}^{n}\sum_{j=0}^{k-1} \tau_j^{p+1} &\le \sum_{k=1}^{n}\sum_{j=0}^{k-1} \tau_j \tau_{k-1}^{p} \le (t_{n+1}-t_0)\sum_{k=1}^{n}\tau_{k-1}^{p}\\ &\le (t_{n+1}-t_0)\sum_{k=1}^{n}\tau_{k-1}\tau_{n-1}^{p-1} \le (t_{n+1}-t_0)^2\tau_{n-1}^{p-1}\,.\end{aligned}$$

In cases 1 and 2, the step size involved in C_{T} tends to zero for infinitely many steps. □

In practice, a problem with (5.34) might occur when the step size suddenly drops by several orders of magnitude. In that case, however, it is possible to modify the above stability analysis and to relax the condition on the step sizes. We shortly explain the idea, but we do not work out all details. If the error at time t_j, say, is large compared to the actual step length, one should rather compare the numerical solution with a smooth trajectory that passes closeby u_j. Although u_j might be a non-smooth initial value, such trajectories exist. Then the previous stability proof can be applied once more. As long as one does not switch too often between trajectories, stability in (5.32) is still guaranteed.

5.4 Methods of order up to four

The well known exponential Rosenbrock–Euler method is given by

$$\begin{aligned} u_{n+1} &= e^{\tau_n J_n} u_n + \tau_n \varphi_1(\tau_n J_n) g_n(u_n) \\ &= u_n + \tau_n \varphi_1(\tau_n J_n) F(u_n)\,. \end{aligned} \tag{5.40}$$

It is computationally attractive since it requires only one matrix function per step. The method obviously satisfies Condition 1 of Table 5.1, while Condition 2 is void. Therefore, it is second-order convergent for problems satisfying Assumptions A.1 and A.2. A possible error estimator for (5.40) is described in [1].

From the order conditions of Table 5.1, it is straightforward to construct pairs of embedded methods of order 3 and 4. For our variable step size implementation, we consider (5.4b) together with an embedded approximation

$$\widehat{u}_{n+1} = u_n + \tau_n \varphi_1(\tau_n J_n) F(u_n) + \tau_n \sum_{i=2}^{s} \widehat{b}_i(\tau_n J_n) D_{ni} \tag{5.41}$$

which relies on the same stages U_{ni}. The methods given below were first introduced in [21]. They will be used in the numerical experiments in Section 5.8.

The scheme exprb32 consists of a third-order exponential Rosenbrock method with a second-order error estimator (the exponential Rosenbrock–Euler method). Its coefficients are

$$\begin{array}{c|cc} c_1 & & \\ c_2 & a_{21} & \\ \hline & b_1 & b_2 \\ & \widehat{b}_1 & \end{array} \quad = \quad \begin{array}{c|cc} 0 & & \\ 1 & \varphi_1 & \\ \hline & \varphi_1 - 2\varphi_3 & 2\varphi_3 \\ & \varphi_1 & \end{array}\,.$$

The scheme exprb43 is a fourth-order method with a third-order error estimator. Its coefficients are

$$\begin{array}{c|ccc} c_1 & & & \\ c_2 & a_{21} & & \\ c_3 & a_{31} & a_{32} & \\ \hline & b_1 & b_2 & b_3 \\ & \widehat{b}_1 & \widehat{b}_2 & \widehat{b}_3 \end{array} \quad = \quad \begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2}\varphi_1(\frac{1}{2}\,\cdot) & & \\ 1 & 0 & \varphi_1 & \\ \hline & \varphi_1 - 14\varphi_3 + 36\varphi_4 & 16\varphi_3 - 48\varphi_4 & -2\varphi_3 + 12\varphi_4 \\ & \varphi_1 - 14\varphi_3 & 16\varphi_3 & -2\varphi_3 \end{array}\,.$$

Note that the internal stages of the above methods are just exponential Rosenbrock–Euler steps. This leads to simple methods that can cheaply be implemented.

Evidently, the order conditions of Table 5.1 imply that the weights of any third-order method have to depend on φ_3, whereas that of any fourth-order method depend on φ_3 and φ_4 (in addition to φ_1).

5.5 Non-autonomous problems

The proposed method can easily be extended to non-autonomous problems

$$u' = F(t,u)\,, \qquad u(t_0) = u_0 \tag{5.42}$$

by rewriting the problem in autonomous form

$$U' = \mathcal{F}(U)\,, \quad U = \begin{pmatrix} t \\ u \end{pmatrix}\,, \quad \mathcal{F}(U) = \begin{pmatrix} 1 \\ F(t,u) \end{pmatrix} \tag{5.43a}$$

with Jacobian

$$\mathcal{J}_n = \begin{pmatrix} 0 & 0 \\ v_n & J_n \end{pmatrix}\,, \quad v_n = \frac{\partial}{\partial t} F(t_n,u_n)\,, \quad J_n = \frac{\partial}{\partial u} F(t_n,u_n)\,. \tag{5.43b}$$

In order to apply our method to this autonomous system, we have to compute matrix functions of $\mathcal{J}_n$. Using Cauchy's integral formula and exploiting the special structure of $\mathcal{J}$, we get

$$\varphi(\tau\mathcal{J}) = \begin{pmatrix} \varphi(0) & 0 \\ \tau\widehat{\varphi}(\tau J)v & \varphi(\tau J) \end{pmatrix}\,, \quad \widehat{\varphi}(z) = \frac{\varphi(z)-\varphi(0)}{z}\,.$$

For the particular functions in our method, we obtain from (5.6) the relation

$$\widehat{\varphi}_i(\tau J) = \varphi_{i+1}(\tau J)\,. \tag{5.44}$$

In our formulation, we will work again with the smaller quantities

$$D_{nj} = g_n(t_n + c_j\tau_n, U_{nj}) - g_n(t_n,u_n)$$

where

$$g_n(t,u) = F(t,u) - J_n u - v_n t\,. \tag{5.45}$$

Applying method (5.8) to the autonomous formulation (5.43), we get

$$U_{ni} = u_n + \tau_n c_i \varphi_1(c_i\tau_n J_n)F(t_n,u_n) + \tau_n \sum_{j=2}^{i-1} a_{ij}(\tau_n J_n) D_{nj} + \tau_n^2 c_i^2 \varphi_2(c_i\tau_n J_n)v_n \tag{5.46a}$$

$$u_{n+1} = u_n + \tau_n \varphi_1(\tau_n J_n)F(t_n,u_n) + \tau_n \sum_{i=2}^{s} b_i(\tau_n J_n) D_{ni} + \tau_n^2 \varphi_2(\tau_n J_n)v_n\,. \tag{5.46b}$$

Thus, the only difference between the exponential Rosenbrock method for non-autonomous problems and the original scheme is the definition of g_n in (5.45), where we now use the full linearization also with respect to the time, and the additional correction terms involving $\varphi_2(\cdot)v_n$ in (5.46).

5.6 Error bounds for perturbed methods

In general the implementation of such a method introduces perturbations resulting from Krylov approximations to the matrix functions multiplied with vectors or from approximations to the Jacobian. Therefore, it is essential to understand the influence of such errors to our method. We consider general perturbations first. Then we study the influence of an approximated Jacobian in more detail.

5.6.1 General perturbations

We will now investigate the error of the exponential Rosenbrock method

$$\begin{aligned} \widetilde{U}_{ni} &= e^{c_i\tau_n J_n}\widetilde{u}_n + \tau_n \sum_{j=1}^{i-1} a_{ij}(\tau_n J_n) g_n(\widetilde{U}_{nj}) + P_{ni}\,, \qquad 1 \le i \le s \\ \widetilde{u}_{n+1} &= e^{\tau_n J_n}\widetilde{u}_n + \tau_n \sum_{i=1}^{s} b_i(\tau_n J_n) g_n(\widetilde{U}_{ni}) + p_{n+1} \end{aligned} \tag{5.47}$$

with small perturbations P_{ni} and p_{n+1} which for example can result from the approximation of the matrix functions by Krylov methods.

Theorem 5.14. *Under the assumptions of Theorem 5.10 the solution of* (5.47) *satisfies the error bound*

$$\|\widetilde{u}_{n+1} - u(t_{n+1})\| \le C \sum_{j=0}^{n} \Big(\tau_j^2 \sum_{i=1}^{s} \|P_{ji}\| + \|p_{j+1}\| \Big) + C \sum_{j=0}^{n} \tau_j^{p+1}$$

uniformly on $t_0 \le t_{n+1} \le T$. The constants C are independent of the chosen step size sequence.

Proof. We expand the error,

$$\|\widetilde{u}_{n+1} - u(t_{n+1})\| \le \|\widetilde{u}_{n+1} - u_{n+1}\| + \|u_{n+1} - u(t_{n+1})\|\,,$$

where the second term on the right hand side is the error of the exact method known from Theorem 5.10. For the difference of the perturbed and the unperturbed method we obtain the recursion

$$\varepsilon_{n+1} = \widetilde{u}_{n+1} - u_{n+1} = e^{\tau_n J_n}\varepsilon_n + \tau_n \sum_{i=1}^{s} b_i(\tau_n J_n)\big(g_n(\widetilde{U}_{ni}) - g_n(U_{ni}) \big) + p_{n+1}$$

which is of a similar form as the error recursion above. Analogously we define the differences at the internal stages,

$$\varepsilon_{ni} = \widetilde{U}_{ni} - U_{ni} = e^{c_i\tau_n J_n}\varepsilon_n + \tau_n \sum_{j=1}^{i-1} a_{ij}(\tau_n J_n)\big(g_n(\widetilde{U}_{nj}) - g_n(U_{nj})\big) + P_{ni}\,.$$

Using Taylor-series expansion we obtain

$$g_n(\widetilde{U}_{ni}) - g_n(U_{ni}) = \frac{\partial g_n}{\partial u}(U_{ni})\varepsilon_{ni} + \int_0^1 (1-\sigma)\frac{\partial^2 g_n}{\partial u^2}(U_{ni}+\sigma\varepsilon_{ni})(\varepsilon_{ni},\varepsilon_{ni})\,\mathrm{d}\sigma$$

and the special form of g_n yields

$$\begin{aligned}\frac{\partial g_n}{\partial u}(U_{ni})\varepsilon_{ni} &= \Big(\frac{\partial f}{\partial u}(U_{ni}) - \frac{\partial f}{\partial u}(u_n)\Big)\varepsilon_{ni}\\ &= \int_0^1 (1-\sigma)\frac{\partial^2 f}{\partial u^2}\Big(u_n + \sigma\tau_n\sum_{j=1}^{i-1} a_{ij}(\tau_n J_n) g_n(U_{nj})\Big)\Big(\tau_n\sum_{j=1}^{i-1} a_{ij}(\tau_n J_n) g_n(U_{nj}), \varepsilon_{ni}\Big)\,\mathrm{d}\sigma\,.\end{aligned}$$

Thus we can state the following estimate,

$$\big\|g_n(\widetilde{U}_{ni}) - g_n(U_{ni})\big\| \le C\big(\tau_n + \|\varepsilon_{ni}\|\big)\|\varepsilon_{ni}\|\,.$$

Using this, we obtain recursively

$$\begin{aligned}\|\varepsilon_{ni}\| &\le C\|\varepsilon_n\| + \tau_n C\sum_{j=1}^{i-1}\big\|g_n(\widetilde{U}_{nj}) - g_n(U_{nj})\big\| + \|P_{ni}\|\\ &\le C\|\varepsilon_n\| + \tau_n C\sum_{j=1}^{i-1}\big(\tau_n + \|\varepsilon_{nj}\|\big)\|\varepsilon_{nj}\| + \|P_{ni}\|\\ &\le C\|\varepsilon_n\| + \|P_{ni}\| + \tau_n^2\sum_{j=1}^{i-1}\|P_{nj}\|\,.\end{aligned}$$

Solving the recursion and using $\varepsilon_0 = 0$ yields

$$\varepsilon_{n+1} = \sum_{j=0}^{n}\tau_j e^{\tau_n J_n}\cdot\ldots\cdot e^{\tau_{j+1}J_{j+1}}\big(\widetilde{\varrho}_j - \tau_j^{-1}p_{j+1}\big)$$

with

$$\widetilde{\varrho}_j = \sum_{i=1}^{s} b_i(\tau_j J_j)\big(g_j(\widetilde{U}_{ji}) - g_j(U_{ji})\big)\,.$$

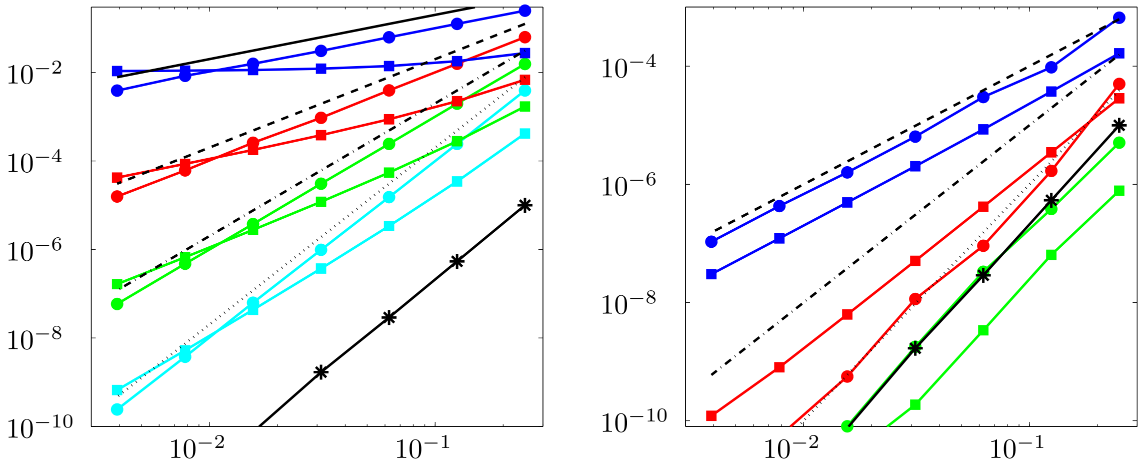


Figure 5.1: The error of the `exprb43` method with different types of perturbations are plotted versus the step size. **Left:** Only the final stages are perturbed with $p_{j+1} = \mathcal{O}(\tau^l)$, $P_{ji} = 0$. **Right:** Only the internal stages are perturbed with $P_{ji} = \mathcal{O}(\tau^l)$, $p_{j+1} = 0$. In both pictures $l = 1, 2, 3$ and 4 for the blue, red, green and cyan curves, respectively, where the squares represent the worst case perturbations and the circles the random perturbation. The black curve with stars is the error of the unperturbed solution. The black lines represent order curves (solid: order 1, dashed: order 2, dash-dotted: order 3, dotted: order 4).

Combining the stability result from Theorem 5.9 with the previous estimates and again applying the Gronwall-Lemma 5.12 bounds the difference of the perturbed and the unperturbed method to

$$\|\varepsilon_{n+1}\| \le C \sum_{j=0}^{n} \Big(\tau_j^2 \sum_{i=1}^{s} \|P_{ji}\| + \|p_{j+1}\| \Big) ,$$

which yields the desired result. □

This estimate is a worst case estimate. If we consider a linear parabolic problem, we have

$$\varepsilon_{n+1} = \sum_{j=0}^{n} e^{(\tau_n + \ldots + \tau_{j+1})A} p_{j+1} ,$$

hence most perturbations are damped.

To demonstrate this, we consider the test equation

$$\frac{\partial}{\partial t} u = \frac{\partial^2}{\partial x^2} u + \frac{1}{1+u^2} + k(t) , \quad x \in [0,1] , \quad t \in [0, 0.5]$$

with homogeneous Dirichlet boundary conditions and $k(t)$ chosen such that the exact solution is

$$u(t,x) = x(x-1)e^t .$$

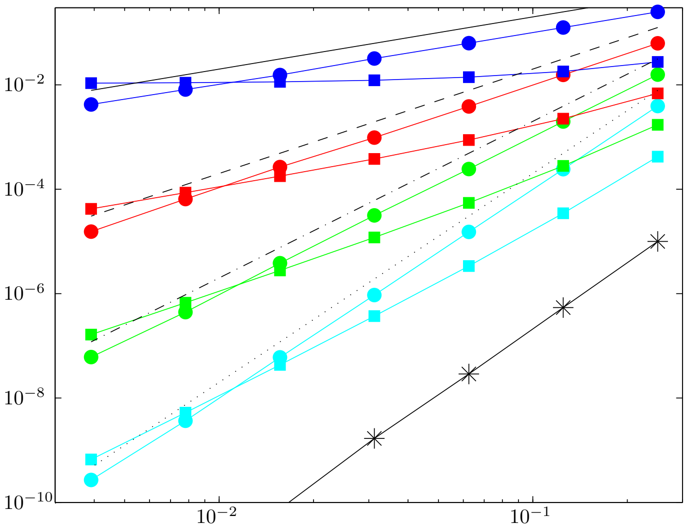


Figure 5.2: The error of the `exprb43` method with different types of perturbations P_{ji}, $p_{j+1} = \mathcal{O}(\tau^l)$ in the inner as well as the final stages are plotted versus the step size. In both pictures $l = 1, 2, 3$ and 4 for the blue, red, green and cyan curves, respectively, where the squares represent the worst case perturbations and the circles the random perturbation. The black curve with stars is the error of the unperturbed solution. The black lines represent order curves (solid: order 1, dashed: order 2, dash-dotted: order 3, dotted: order 4).

The problem is discretized in space by finite differences with 200 nodes and solved with `exprb43` employing constant step sizes.

In the left picture of Fig. 5.1, we choose $P_{ji} = 0$ and $p_{j+1} = \tau^l r_j$ for a normalized random vector r_j (curves with circles) and $p_{j+1} = \tau^l w_j$, where w_j is the eigenvector to the largest eigenvalue of the discretized linear operator plus a small perturbation (curves with squares), $l = 1, 2, 3$ and 4 for the blue, red, green and cyan curves, respectively. In the right picture, we perturbed the inner stages in the same way as above and do not change the final stage. It can be seen, that for the random perturbation, we almost always obtain better results than the estimate predicts. In the worst case, where the perturbations are chosen to be the eigenvector to the largest eigenvalue of the linear part plus a random perturbation of the size of the machine precision, the errors are not damped, but sum up in the predicted way. In Fig. 5.2 results are shown where the inner stages and the final stages are perturbed. The errors show the same behavior than the errors of the method with perturbations only in the final stage. As predicted by Theorem 5.14, it is more important to compute the final stage more accurately than the inner stages.

5.6.2 Inexact Jacobian

It might be convenient to use the scheme with an inexact Jacobian, for instance if it is computed via numerical differentiation. Replacing $J_n = \mathrm{D}F(u_n)$ by an approximation $\widetilde{J}_n$ yields

$$\begin{aligned} U_{ni} &= e^{c_i\tau_n\widetilde{J}_n}u_n + \tau_n\sum_{j=1}^{i-1} a_{ij}(\tau_n\widetilde{J}_n)\widetilde{g}_n(U_{nj})\,, \qquad 1 \le i \le s \\ u_{n+1} &= e^{\tau_n\widetilde{J}_n}u_n + \tau_n\sum_{i=1}^{s} b_i(\tau_n\widetilde{J}_n)\widetilde{g}_n(U_{ni})\,. \end{aligned} \tag{5.48}$$

The function $\widetilde{g}_n$ is given by

$$\widetilde{g}_n\big(u(t)\big) = F\big(u(t)\big) - \widetilde{J}_n u(t)\,.$$

Theorem 5.15. *Let the assumptions of Theorem 5.10 be satisfied. Let* $\Delta\widetilde{J}_n = J_n - \widetilde{J}_n$ *be sufficiently small and*

$$\sum_{j=0}^{n}\Bigg(\sum_{k=0}^{j-1}\big(\tau_k^2\|\Delta\widetilde{J}_k\|_{X\leftarrow X}\big) + \|\Delta\widetilde{J}_j\|_{X\leftarrow X}\Bigg) \le C_{\mathrm{J}}\,. \tag{5.49}$$

If $\widetilde{J}_n$ *also satisfies Assumption* A.1*, the solution of* (5.48) *satisfies the error bound*

$$\|u_{n+1} - u(t_{n+1})\| \le C\sum_{j=0}^{n}\big(\tau_j^{p+1} + \tau_j^2\|\Delta\widetilde{J}_j\|_{X\leftarrow X}\big)$$

uniformly on $t_0 \le t_{n+1} \le T$.

The analysis follows the lines of Sections 5.2 and 5.3.

Proof. For a representation of the new defects, we apply the variation-of-constants formula to

$$u'(t) = \widetilde{J}_n u(t) + \widetilde{G}_n(t)\,,$$

where $\widetilde{G}_n(t) = \widetilde{g}_n(u(t))$. This combined with equation (5.18) yields the estimate

$$\|\widetilde{G}_n'(t_n)\|_{X\leftarrow X} \le \|\Delta\widetilde{J}_n\|_{X\leftarrow X} + \|G_n'(t_n)\|_{X\leftarrow X} \le \|e_n\| + \|\Delta\widetilde{J}_n\|_{X\leftarrow X}$$

which leads to the following estimates for the new defects,

$$\begin{aligned} \|\widetilde{\Delta}_{ni}\| &\le C\tau_n^2\big(\|e_n\| + \|\Delta\widetilde{J}_n\|_{X\leftarrow X}\big) + C\tau_n^3\,, \\ \|\widetilde{\delta}_{n+1}\| &\le C\tau_n^2\big(\|e_n\| + \|\Delta\widetilde{J}_n\|_{X\leftarrow X}\big) + C\tau_n^{p+1}\,. \end{aligned}$$

Next, we adjust the estimate from Lemma 5.5,

$$\begin{aligned}\|E_{ni}\| &\le C\|e_n\| + C\tau_n \sum_{j=1}^{i-1}\big(\tau_n + \|e_n\| + \|E_{nj}\|\big)\|E_{nj}\| + \|\widetilde{\Delta}_{ni}\| \\ &\le C\|e_n\| + C\tau_n^3 + C\tau_n^2\|\Delta\widetilde{J}_n\|_{X\leftarrow X}\,.\end{aligned}$$

Since we use $\widetilde{J}_n$ to evaluate the matrix functions, we have to derive a stability bound for the discrete evolution operator involving $\widetilde{J}_n$ instead of J_n. Therefore we repeat the proof of Theorem 5.9 for the approximate Jacobian. Here we have

$$\begin{aligned}\|\widetilde{J}_n - \widetilde{J}_{n-1}\|_{X\leftarrow X} &= \|\widetilde{J}_n - J_n + J_n - J_{n-1} + J_{n-1} - \widetilde{J}_{n-1}\|_{X\leftarrow X} \\ &\le \|\Delta\widetilde{J}_n\|_{X\leftarrow X} + \|J_n - J_{n-1}\|_{X\leftarrow X} + \|\Delta\widetilde{J}_{n-1}\|_{X\leftarrow X} \\ &\le C\big(\tau_n + \|e_n\| + \|e_{n-1}\|\big) + \|\Delta\widetilde{J}_n\|_{X\leftarrow X} + \|\Delta\widetilde{J}_{n-1}\|_{X\leftarrow X}\end{aligned}$$

which leads to

$$\Big\|e^{-t\widetilde{J}_n} - e^{-t\widetilde{J}_{n-1}}\Big\|_{X\leftarrow X} \le C_{\mathtt{L}}\big(\tau_{n-1} + \|e_n\| + \|e_{n-1}\| + \|\Delta\widetilde{J}_n\|_{X\leftarrow X} + \|\Delta\widetilde{J}_{n-1}\|_{X\leftarrow X}\big)e^{\widetilde{\omega}t}$$

and

$$\big\|e^{\tau_n\widetilde{J}_n}\cdot\ldots\cdot e^{\tau_0\widetilde{J}_0}\big\|_{X\leftarrow X} \;\le\; Ce^{\widehat{\omega}(\tau_0+\ldots+\tau_n)+C_{\mathtt{E}}\sum_{j=1}^{n}(\|e_j\|+\|\Delta\widetilde{J}_j\|)}\,.$$

Solving the error recursion and applying the modified stability bound yields the desired bound. The constant depends on

$$\sum_{j=0}^{n}\big(\|e_j\| + \|\Delta\widetilde{J}_j\|_{X\leftarrow X}\big) \le \sum_{j=0}^{n}\left(\sum_{k=0}^{n-1}\big(\tau_k^{p+1} + \tau_k^2\|\Delta\widetilde{J}_k\|_{X\leftarrow X}\big) + \|\Delta\widetilde{J}_j\|_{X\leftarrow X}\right),$$

which is bounded uniformly using (5.34) and (5.49). □

It is obvious that $\|\Delta\widetilde{J}_j\|_{X\leftarrow X} \le C\tau_j$ should be satisfied. Otherwise stability cannot be guaranteed any more. To get the same order as for the exact Jacobian, $\|\Delta\widetilde{J}_j\|_{X\leftarrow X}$ has to be of the order of τ_j^{p-1}.

For the same example as in the previous section, the errors of the perturbed method are shown in Fig. 5.3. We can see, that the order is reduced in the predicted way.

5.7 Implementation

5.7.1 Step size control

We have implemented the exponential Rosenbrock methods `exprb32` and `exprb43` for autonomous as well as non-autonomous problems in Matlab with adaptive time stepping.

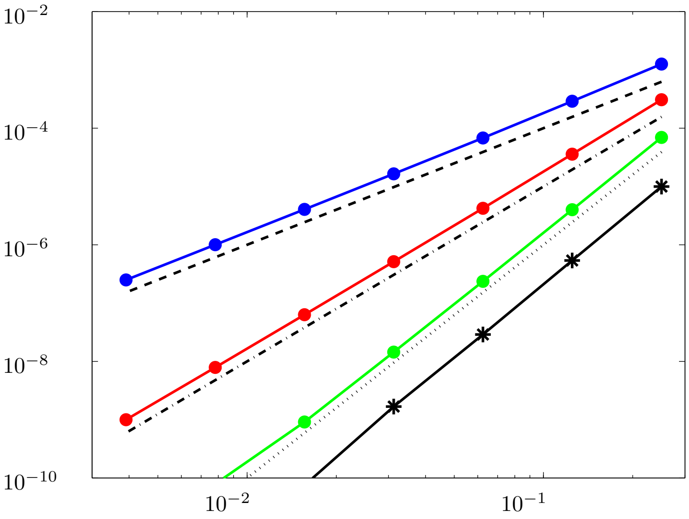


Figure 5.3: The errors of the `exprb43` method with $\Delta\widetilde{J}_j = \mathcal{O}(\tau_j^l)$ are plotted versus the step size for $l = 1, 2$ and 3 for the blue, red and green curves, respectively. The black curve with stars is the error of the unperturbed solution. The black lines represent order curves (dashed: order 2, dash-dotted: order 3, dotted: order 4).

For the step size control we used the embedded methods (5.41) to estimate the local error in each time step,

$$\widehat{e}_{n+1} = \tau_n \sum_{i=2}^{s} \bigl(b_i(\tau_n J_n) - \widehat{b}_i(\tau_n J_n)\bigr) D_{ni}\,.$$

A time step is accepted if the scaled norm satisfies

$$\|\widehat{e}_n\|_{\mathrm{sc}} = \left(\frac{1}{d}\sum_{i=1}^{d}\left|\frac{\widehat{e}_n(i)}{\mathrm{sc}(i)}\right|^2\right)^{\frac{1}{2}} \le 1\,, \qquad \mathrm{sc} = \mathrm{ATol} + \max\{|u_n|, |u_{n-1}|\}\mathrm{RTol}\,.$$

We then employed a combination of a classical error estimation and a Gustafsson method to select a new step size using factors

$$f_{\mathrm{C}} = \|\widehat{e}_{n+1}\|_{\mathrm{sc}}^{\frac{1}{p}} \tag{5.50}$$

and

$$f_{\mathrm{G}} = \frac{\tau_n}{\tau_{n-1}}\left(\frac{\|\widehat{e}_n\|_{\mathrm{sc}}}{\|\widehat{e}_{n+1}\|_{\mathrm{sc}}^2}\right)^{\frac{1}{p}}.$$

The new time-step size is then determined by

$$\tau_{n+1} = \tau_n \max\bigl\{\min\{f_{\mathrm{s}} f_{\mathrm{C}}, f_{\mathrm{s}} f_{\mathrm{G}}, f_{\max}\}, f_{\min}\bigr\}$$

for an accepted time step and in case of rejection by

$$\tau_{n+1} = \tau_n \max\bigl\{\min\{f_{\mathrm{s}} f_{\mathrm{C}}, f_{\max}\}, f_{\min}\bigr\}$$

with a safety factor $f_{\mathrm{s}} = 0.9$. The factors $f_{\min} = 0.2$ and $f_{\max} = 5$ prevent the method from changing the step size to fast. For details see [13].

5.7.2 Matrix functions

Our implementation involves two different options for dealing with the matrix φ-functions: for small examples, we employ diagonalization for the explicit computation of the matrix functions. For large problems, Krylov subspace methods are used for approximating the product of the matrix functions with the corresponding vectors. We use a standard Arnoldi algorithm to compute the basis of the Krylov space and then again use diagonalization for the evaluation of the small matrix functions, see [27] and references therein.

We use the results from Section 5.6.1, to construct a stopping criterion for the Krylov process. For now, we omit the indices of J_n and τ_n to avoid confusion with the Krylov indices. In the mth step of the Arnoldi-Krylov process for $J \in C^{N,N}$ and a starting vector $v = \|v\| v_1$ we have the relation

$$JV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T$$

with an upper Hessenberg matrix H_m of size $m \times m$ and $V_m \in \mathbb{C}^{N,m}$ with unitary columns v_k forming the basis of $\mathcal{K}_m(J, v) = \operatorname{span}\{v, Jv, \dots, J^{m-1}v\}$. The matrix function is then approximated by

$$\varphi(\tau J)v \approx \|v\| V_m \varphi(\tau H_m) e_1 \,.$$

A popular stopping criterion is based on the generalized residual defined as

$$\mathrm{res}_m = \tau \|v\| h_{m+1,m} \bigl(\varphi(\tau H_m)\bigr)_{m,1} v_{m+1} \,,$$

see [18]. Using the reformulation of the method (5.8), the perturbation in the new stage is then given by

$$\begin{aligned} p_{n+1} = &\Bigl(\|\tau_n F(u_n)\| V^{(1)}_{m^{(1)}} \varphi_1(\tau_n H^{(1)}_{m^{(1)}}) e_1 - \tau_n \varphi_1(\tau_n J_n) F(u_n)\Bigr) \\ &+ \sum_{i=2}^{s} \Bigl(\|\tau_n D_{ni}\| V^{(i)}_{m^{(i)}} b_i(\tau_n H^{(i)}_{m^{(i)}}) e_1 - \tau_n b_i(\tau_n J_n) D_{ni}\Bigr) \end{aligned}$$

where $V^{(1)}_{m^{(1)}}$ and $H^{(1)}_{m^{(1)}}$ result from the Krylov space $\mathcal{K}_{m^{(1)}}(J_n, F(u_n))$ and $V^{(i)}_{m^{(i)}}$ and $H^{(i)}_{m^{(i)}}$ from $\mathcal{K}_{m^{(i)}}(J_n, D_{ni})$, $2 \le \ldots \le s$. To obtain an error of the same order of the method, the perturbation has to be of order τ_n^{p+1} or smaller.

We now use the stopping criterion

$$\|\mathrm{res}_m\|_{\mathrm{sc}} \le \tau_n \,.$$

The scaled norm ensures that the perturbation is of the desired tolerance. If we ask this to be smaller than τ, we do not loose accuracy even if the perturbations sum up. We use the same Krylov spaces for the inner and the outer stages, thus we always use the same stopping criterion.

For autonomous problems, we use the reformulation (5.8), which requires one Krylov subspace with the vector $F(u_n)$ and $s-1$ Krylov subspaces with the vectors D_{ni}, $i = 2, \ldots, s$. Due to $\|D_{ni}\| = \mathcal{O}(\tau_n^2)$, these approximations can be computed in very low-dimensional subspaces since the norm of the starting vector enters the stopping criterion. For non-autonomous problems, the format (5.46) requires one additional Krylov subspace with the vector v_n. Since the term involving v_n is multiplied by τ_n^2 (compared to τ_n for the other vectors), this subspace will be low-dimensional, as well.

We also included a maximum dimension for the Krylov spaces. If the desired tolerance is not reached within these steps, the time-step size is reduced.

5.8 Numerical Experiments

Example 1. As a first example we consider a two-dimensional advection-diffusion-reaction equation for $u = u(x, y, t)$

$$(5.51)\quad \frac{\partial}{\partial t}u = \varepsilon\Big(\frac{\partial^2}{\partial x^2}u + \frac{\partial^2}{\partial y^2}u\Big) - \alpha\Big(\frac{\partial}{\partial x}u + \frac{\partial}{\partial y}u\Big) + \gamma u\big(u - \tfrac{1}{2}\big)(1-u)\,, \qquad (x,y) \in (0,1)^2$$

with homogeneous Neumann boundary conditions and the initial value

$$u(x,y,0) = 256\big((1-x)x(1-y)y\big)^2 + 0.3\,,$$

where $\varepsilon = 1/100$, $\alpha = -10$, and $\gamma = 100$. The spatial discretization was done with finite differences using 101 grid points in each direction.

This example is taken from [1], where Fortran implementations of `exprb43`, combined with the real Leja point method [2], and of the Runge–Kutta–Chebyshev method `RKC` from [34] were compared. Here we compare Matlab implementations of `RKC`, `exprb43`, `exp4` from [18], and Krogstad's method [26]. The latter three make use of Krylov subspace approximations. To improve the efficiency of the Krogstad method, we reused information from previously computed Krylov subspaces, an approach proposed in [19]. Since an adaptive step-size control based on embedding is not possible for Krogstad's method, we ran

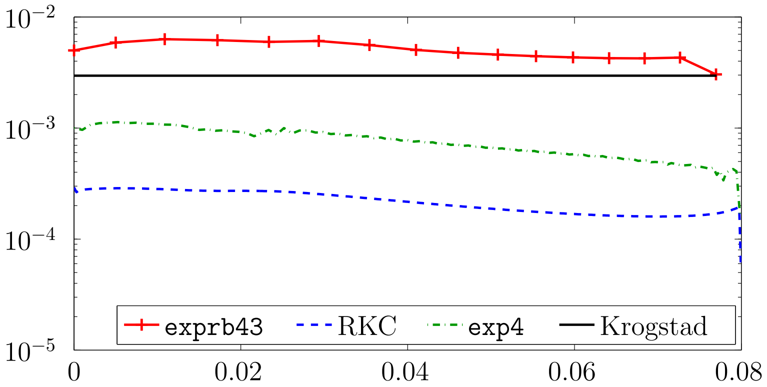


Figure 5.4: Step sizes for the advection-diffusion-reaction equation (5.51) for $t \in [0, 0.08]$

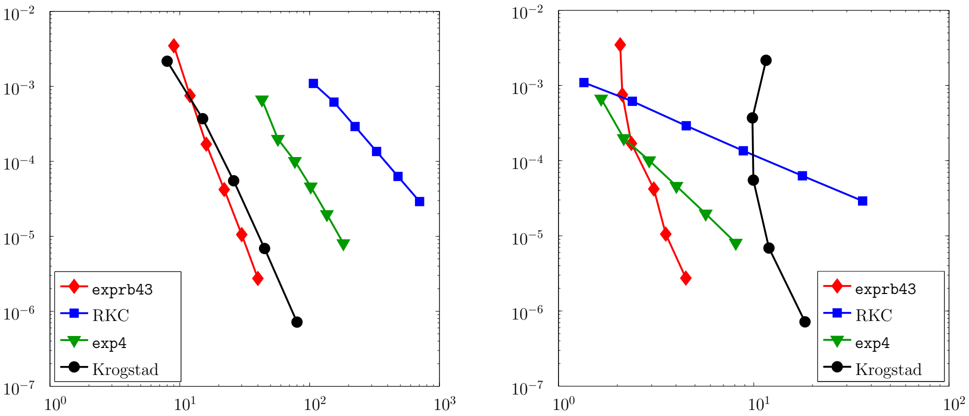


Figure 5.5: Number of time steps versus accuracy (left) and CPU time versus accuracy (right) for the advection-diffusion-reaction example (5.51) for $t = 0.08$

this method with constant step size. For this particular example, the step-size control of the other schemes also lead to almost constant steps sizes, see Fig. 5.4. All simulations achieved a final accuracy of about 0.004 at $t = 0.08$. It can be seen that, due to the large advection part, the exponential methods can take much larger steps than `RKC` with `exprb43` taking the largest ones. In total, `exprb43` takes only 18 steps, Krogstad's method takes 27 steps, `exp4` takes 119 steps, while `RKC` uses 383 steps.

In Fig. 5.5, we compare the performance of the Krylov implementations of `exp4`, `exprb43` and Krogstad's method with a Matlab implementation of `RKC`. Our implementations of `exp4` and `exprb43` allow a maximum dimension of the Krylov subspaces of 36. The codes were run with tolerances ATol = RTol = $10^{-4}, 10^{-4.5}, \dots, 10^{-6.5}$ (except for Krogstad's method, which was used with constant step size). In the left diagram, we plot the achieved accuracy as a function of the required number of steps. It turns out that, for a given accuracy, the exponential Rosenbrock method `exprb43` uses significantly larger time steps

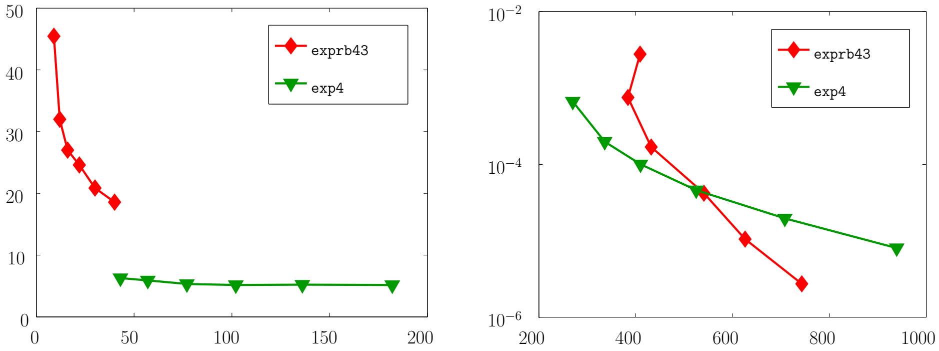


Figure 5.6: Number of time steps versus average number of Krylov steps (left) and number of Krylov steps versus accuracy (right) for the advection-diffusion-reaction example (5.51) for $t = 0.08$

than `exp4` and `RKC`. The number of time steps required for Krogstad's method is about the same as for `exprb43`.

The situation changes, if we consider the achieved accuracy as a function of the required CPU time, cf. Fig. 5.5. It can be seen that for moderate tolerances, `exp4` is faster than `exprb43` while for more stringent tolerances `exprb43` requires less CPU time. This can be explained by considering the number of Krylov steps used by these methods. In the left diagram in Fig. 5.6 we plotted the average number of Krylov steps over the total number of time steps. Since `exprb43` uses significantly larger time steps, we know from the convergence analysis of Krylov subspace methods [3, 16] that this requires more Krylov steps. The right diagram of Fig. 5.6 shows the achieved accuracy versus the total number of Krylov steps. Since the Krylov approximations dominate the computational cost, this explains the right diagram of Fig. 5.5. Note that it is impossible to give a reformulation of Krogstad's method in such a way that only one expensive Krylov subspace is required in each step. The gain achieved by reusing previously computed Krylov subspaces [19] does not compensate this disadvantage. Moreover, Krogstad's method has four stages and uses even more matrix functions than `exprb43`.

Example 2. As a second example, we consider the one-dimensional Schrödinger equation [18] for $\psi = \psi(x, t)$

$$i\frac{\partial}{\partial t}\psi = H(x,t)\psi \tag{5.52a}$$

with the time dependent Hamiltonian

$$H(x,t) = -\frac{1}{2}\frac{\partial^2}{\partial x^2} + \kappa\frac{x^2}{2} + \mu(\sin t)^2 x\,. \tag{5.52b}$$

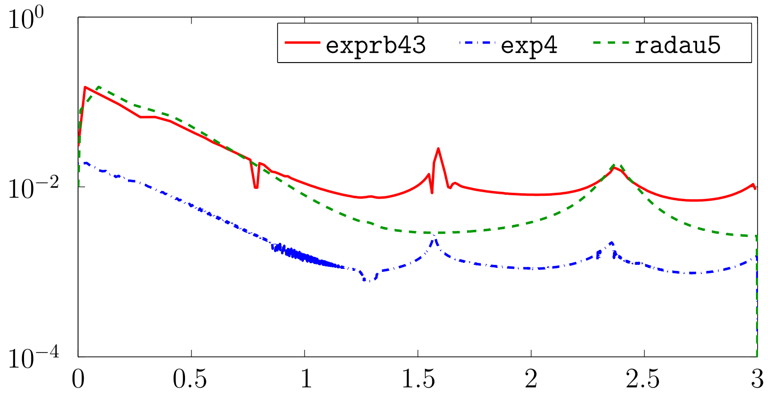


Figure 5.7: Step sizes taken by `exp4`, `radau5`, and `exprb43` for the laser example (5.52) for $t \in [0, 3]$

We used the parameter values $\kappa = 10$ and $\mu = 100$. The initial value was chosen as $\psi(x, 0) = e^{-\sqrt{\kappa}x^2/2}$, which corresponds to the ground state of the unforced harmonic oscillator. Semi-discretization in space was done by a pseudo-spectral method with 512 Fourier modes on the interval $[-10, 10]$ with periodic boundary conditions.

It was shown in [18] that the Matlab implementation of `exp4` outperforms Matlab's standard nonstiff `ode45` method and matrix-free implementations of the stiff solvers `radau5` and `ode15s`. We refer to [18] for details. Here, we use exactly the same spatial discretization but run the simulation until $t = 3$.

In Fig. 5.7, we display the step sizes chosen by the adaptive step-size control for `exp4`, `radau5`, and `exprb43`. The tolerances were set in such a way that all methods achieved a final accuracy of about 0.05. As illustrated in Fig. 5.7, `exprb43` advances with larger step sizes than the other two methods. In total `exprb43` uses 256 steps, `exp4` uses 1906 steps, and `radau5` uses 537 steps. In our implementation of `radau5`, the linear systems arising within the Newton iteration are solved directly while `exp4` and `exprb43` are used with Krylov subspace approximations. Therefore, the `radau5` code takes more than 10 times longer than `exprb43`. Since it has been shown in [18] that a much more efficient W-version of `radau5` was still slower than `exp4`, we did not include `radau5` into our run time comparisons.

In Fig. 5.8, we compare the performance of the Krylov implementations of `exp4` and `exprb43`. Both codes were run with tolerances ATol = RTol = 10^{-4}, $10^{-4.5}$, $\ldots, 10^{-6.5}$. The diagrams show that the exponential Rosenbrock method `exprb43` uses significantly larger step sizes than `exp4`. Moreover, it is also much faster in terms of total CPU time.

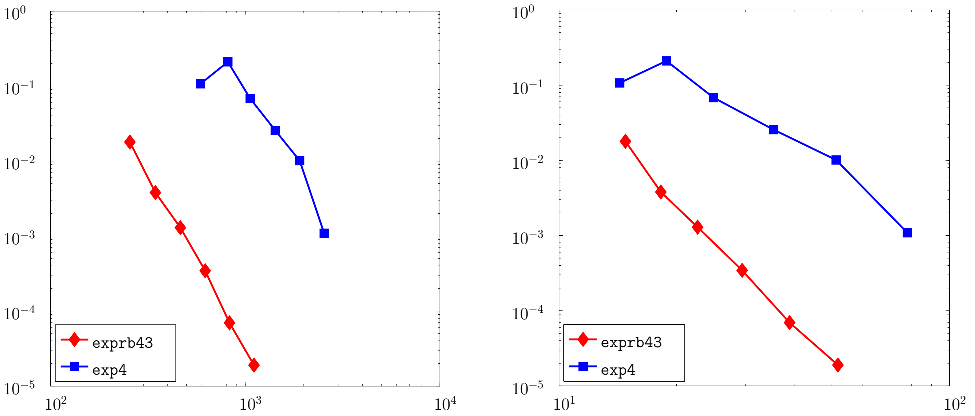


Figure 5.8: Number of time steps versus accuracy (left) and CPU time versus accuracy (right) for the laser example (5.52) for $t = 3$

5.9 Nonlinear Schrödinger equation

Now, we return to the laser-plasma application and solve the Schrödinger equation (1.21) from Section 1.4 with the exponential Rosenbrock-type method. As in Chapters 3 and 4 we choose $Q = 0.3$. The initial conditions are computed via (3.3). We apply the same transformations to (3.3) as in Section 1.4 to derive the Schrödinger equation. Then we see, that the initial condition takes the form

$$\widetilde{a}(\vartheta) = a_0 e^{\frac{(\beta\vartheta - z_0)^2}{w_0^2}} e^{-z_0}$$

for $\beta = \sqrt{1-Q}$ and $a_0 = 0.1$. We choose $\vartheta \in [-50/\nu_0, 50/\nu_0]$ and periodic boundary conditions. The problem is discretized in space via a speudo-spectral method. Then, we integrate from $z = 0\lambda_0$ to $z = 500\lambda_0$. The result is shown in Fig. 5.9. We can see a similar behavior of the solution to that of the top picture of Fig. 3.1. In the Schrödinger case, we start the simulation at the plasma boundary. We can see, that the pulse compresses and develops the pre and post pulses of the two-soliton state similar to the pulse from the full wave equation. However, the slowly varying envelope approach breaks down in the regime of sub-cycle pulses, where the pulse length is only one or two λ_0.

5.10 Analytic semigroups

So far we restricted our attention to strongly continuous semigroups. This framework, however, limits the class of possible nonlinearities due to Assumption A.2. If the semigroup is even analytic, we can allow more general nonlinearities. In this section we sketch how

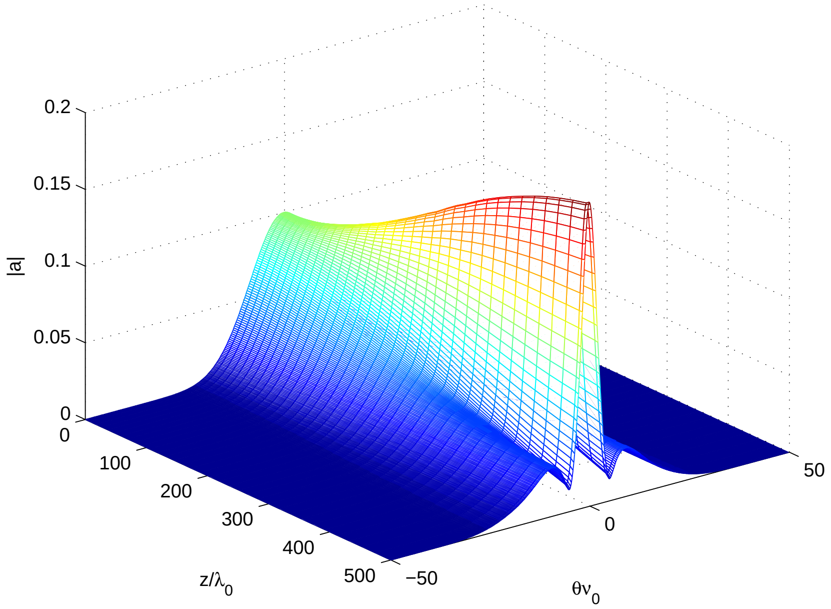


Figure 5.9: Solution of the nonlinear Schrödinger equation (1.21) computed with the exponential Rosenbrock-type method `exprb43`.

to extend our analysis to this case. We first recall the basic definitions. For the theoretical background of analytic semigroups, we refer to [4, 29].

Definition 5.16. *Let $\Delta = \{z : \varphi_1 < \arg(z) < \varphi_2,\ \varphi_1 < 0 < \varphi_2\}$ and for $z \in \Delta$ let $\Gamma(z)$ be a bounded linear operator. The family $\Gamma(z)$, $z \in \Delta$ is an* analytic semigroup *in Δ if*

- *$z \to \Gamma(z)$ is analytic in Δ,*
- *$\Gamma(0) = I$ and $\lim\limits_{\substack{z\to 0\\ z\in\Delta}} \Gamma(z)x = x$ for all $x \in X$ and*
- *$\Gamma(z_1 + z_2) = \Gamma(z_1)\Gamma(z_2)$ for all $z_1,\ z_2 \in \Delta$.*

Remark. An analytic semigroup is an extension of strongly continuous semigroups to sectors in the complex plane containing the positive real axis.

Assumption A.3. *The linear operator A in* (5.11) *is the generator of an analytic semigroup.*

Without loss of generality, we can assume that A is invertible (otherwise we shift it by an appropriate multiple of the identity). Therefore, fractional powers of A are well defined.

We choose $0 \le \alpha < 1$ and define $V = \mathcal{D}(A^\alpha) \subset X$. The linear space V is a Banach space with norm $\|v\|_V = \|A^\alpha v\|$.

Our basic assumptions on f are the following:

Assumption A.4. *We suppose that* (5.11) *possesses a sufficiently smooth solution* $u : [0,T] \to V$ *with derivatives in* V*, and that* $f : V \to X$ *is sufficiently often Fréchet differentiable in a strip along the exact solution. All occurring derivatives are supposed to be uniformly bounded.*

A consequence of Assumption A.3 is that there exist constants C and ω such that

$$\left\| e^{tJ} \right\|_{V \leftarrow V} + \left\| t^\alpha e^{tJ} \right\|_{V \leftarrow X} \le C e^{\omega t}, \qquad t \ge 0 \tag{5.53}$$

holds in a neighborhood of the exact solution.

With these assumptions at hand, we derive once more the bounds of Section 5.2. Instead of (5.20), we now get

$$\|\Delta_{ni}\|_X + \tau_n^\alpha \, \|\Delta_{ni}\|_V \le C\tau_n^2 \|e_n\|_V + C\tau_n^3, \tag{5.54}$$

and (5.24) is replaced by

$$\|\delta_{n+1}\|_X + \tau_n^\alpha \|\delta_{n+1}\|_V \le C\tau_n^2 \|e_n\|_V + C\tau_n^{p+1}. \tag{5.55}$$

The same arguments as in the proofs of Lemma 5.4 and 5.5 show the following refined estimates.

Lemma 5.17. *Under Assumptions* A.3 *and* A.4*, we have*

$$\left\| \frac{\partial g_n}{\partial u}\big(u(t_n)\big) \right\|_{X \leftarrow V} \le C \, \|e_n\|_V \,, \tag{5.56a}$$

$$\|g_n(u_n) - G_n(t_n)\|_X \le C \|e_n\|_V^2 \,, \tag{5.56b}$$

$$\|g_n(U_{ni}) - G_n(t_n + c_i\tau_n)\|_X \le C\big(\tau_n + \|e_n\|_V + \|E_{ni}\|_V\big) \|E_{ni}\|_V \,, \tag{5.56c}$$

and

$$\|E_{ni}\|_V \le C\|e_n\|_V + C\tau_n^{3-\alpha} \,, \tag{5.56d}$$

as long as the errors E_{ni} *and* e_n *remain in a sufficiently small neighborhood of 0.*

Further, Assumption A.4 implies

$$\|\widehat{J}_n - \widehat{J}_{n-1}\|_{X \leftarrow V} \le C\tau_{n-1} \,, \qquad n \ge 1 \tag{5.57}$$

with a constant C that is independent of τ_{n-1}. The same arguments as in the proof of Lemma 5.6 with (5.9) replaced by (5.53) now show that

$$\|e^{t\widehat{J}_n} - e^{t\widehat{J}_{n-1}}\|_{V\leftarrow V} \le C_{\mathrm{L}}\tau_{n-1}e^{\widetilde{\omega}t}\,. \tag{5.58}$$

This implies the desired stability estimate in V. For the convergence proof, we need an additional stability result that reflects the parabolic smoothing.

Lemma 5.18. *Let the initial value problem* (5.11) *satisfy Assumptions* A.3 *and* A.4, *and let $\widehat{J}_n = \mathrm{D}F(u(t_n))$. Then, for any $\widetilde{\omega} > \omega$, there exists a constant C independent of τ_{n-1} such that*

$$\big\|e^{\tau_n\widehat{J}_n}\cdot\ldots\cdot e^{\tau_0\widehat{J}_0}\big\|_{V\leftarrow X} \le C\frac{e^{\widehat{\omega}(\tau_0+\ldots+\tau_n)}}{(\tau_0+\ldots+\tau_n)^\alpha}\,, \tag{5.59}$$

with $\widehat{\omega} = C_{\mathrm{L}} + \widetilde{\omega}$ and C_{L} from (5.58).

Proof. Using the same arguments as in [28, Sec. 5] shows this bound. □

We are now in the position to state the convergence proof for exponential Rosenbrock methods in the framework of analytic semigroups. For notational simplicity, we formulate the result for constant step sizes only.

Theorem 5.19. *Let the initial value problem* (5.11) *satisfy Assumptions* A.3 *and* A.4 *and consider for its numerical solution an explicit exponential Rosenbrock method* (5.4) *with constant step size τ. Assume that the order conditions of Table* 5.1 *hold up to order p with $p = 2$ or $p = 3$. Then, for h sufficiently small, the numerical method converges with order p. In particular, the numerical solution u_n satisfies the uniform error bound*

$$\|u_n - u(t_n)\|_V \le C\tau^p\,.$$

The constant C depends on T, but it is independent of n and τ for $0 \le n\tau \le T - t_0$.

Proof. We proceed as in the proof of Theorem 5.10. Due to (5.55) and (5.56), we can bound

$$\|\varrho_n\|_X + \tau^{-1}\|\delta_{n+1}\|_X \le C\big(\tau\|e_n\|_V + \|e_n\|_V^2 + \tau^p\big)\,. \tag{5.60}$$

By the stability estimate, we now have

$$\|e_n\|_V \le C\sum_{j=0}^{n-1}\frac{\tau}{(t_n - t_{j+1})^\alpha}\Big(\tau\|e_j\|_V + \|e_j\|_V^2 + \tau^p\Big)\,.$$

The desired error bound thus follows from the application of a discrete Gronwall lemma with weakly singular kernel. □

Remark. For $p \geq 4$, the analysis is much more delicate. Due to (5.56d), the bound (5.60) now contains a term of the order $\tau^{4-\alpha}$. Under additional assumptions on f, this order reduction can be avoided. For exponential Runge–Kutta methods, this has been detailed in [20]. We do not elaborate this point here.

Bibliography

[1] M. Caliari and A. Ostermann, *Implementation of exponential Rosenbrock-type integrators.* to appear in Appl. Numer. Math., 2008.

[2] M. Caliari, M. Vianello, and L. Bergamaschi, *Interpolating discrete advection-diffusion propagators at Leja sequences*, J. Comp. Appl. Math., 172 (2004), pp. 79–99.

[3] V. Druskin and L. Knizhnerman, *Krylov subspace approximation of eigenpairs and matrix functions in exact and computer arithmetic*, Numer. Lin. Algebra Appl., 2 (1995), pp. 205–217.

[4] K.-J. Engel and R. Nagel, *One-parameter Semigroups for Linear Evolution Equations*, Springer New York, 2000.

[5] J. Faure, Y. Glinec, A. Pukhov, S. Kiselev, S. Gordienko, E. Lefebvre, J.-P. Rousseau, F. Burgy, and V. Malka, *A laser-plasma accelerator producing monoenergetic electron beams*, Nature, 431 (2004), pp. 541–544.

[6] J. Fuchs, C. A. Cecchetti, M. Borghesi, T. Grismayer, E. d'Humieres, P. Antici, S. Atzeni, P. Mora, A. Pipahl, L. Romagnani, A. Schiavi, Y. Sentoku, T. Toncian, P. Audebert, and O. Willi, *Laser-foil acceleration of high-energy protons in small-scale plasma gradients*, Phys. Rev. Lett., 99 (2007).

[7] V. Grimm, *A note on the Gautschi-type method for oscillatory second-order differential equations*, Numer. Math., 102 (2005), pp. 61–66.

[8] ——, *On error bounds for the Gautschi-type exponential integrator applied to oscillatory second-order differential equations*, Numer. Math., 100 (2005), pp. 71–89.

[9] V. Grimm and M. Hochbruck, *Error analysis of exponential integrators for oscillatory second-order differential equations*, J. Phys. A: Math. Gen., 39 (2006), pp. 5495–5507.

[10] ——, *On the computation of certain trigonometric operator functions.* preprint submitted to BIT, 2008.

[11] E. Hairer and C. Lubich, *Long-time energy conservation of numerical methods for oscillatory differential equations*, SIAM J. Numer. Anal., 38 (2000), pp. 414–441.

[12] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations*, vol. 31 of Springer Series in Computational Mathematics, Springer, 2nd ed., 2006.

[13] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, vol. 14 of Springer Series in Computational Mathematics, Springer, 2nd ed., 1996.

[14] B. M. Hegelich, B. J. Albright, J. Cobble, K. Flippo, S. Letzring, M. Paffett, H. Ruhl, J. Schreiber, R. K. Schulze, and J. C. Fernández, *Laser acceleration of quasi-monoenergetic MeV ion beams*, Nature, 439 (2006), pp. 441–444.

[15] B. Hidding, K.-U. Amthor, B. Liesfeld, H. Schwoerer, S. Karsch, M. Geissler, L. Veisz, K. Schmid, J. G. Gallacher, S. P. Jamison, D. Jaroszynski, G. Pretzler, and R. Sauerbrey, *Generation of quasi-monoenergetic electron bunches with 80-fs laser pulses*, Phys. Rev. Lett., 96 (2006).

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

[17] ——, *A Gautschi-type method for oscillatory second-order differential equations*, Numer. Math., 83 (1999), pp. 403–426.

[18] M. Hochbruck, C. Lubich, and H. Selhofer, *Exponential integrators for large systems of differential equations*, SIAM J. Sci. Comp., 19 (1998), pp. 1552–1574.

[19] M. Hochbruck and J. Niehoff, *Approximation of matrix operators applied to multiple vectors.* to appear in J. Math. Comp. Simul., 2008.

[20] M. Hochbruck and A. Ostermann, *Explicit exponential runge-kutta methods for semilinear parabolic problems*, SIAM J. Numer. Anal., 43 (2005), pp. 1069–1090.

[21] ——, *Exponential integrators of Rosenbrock-type*, Oberwolfach Reports, 3 (2006), pp. 1107–1110.

[22] M. Hochbruck, A. Ostermann, and J. Schweitzer, *Exponential Rosenbrock-type methods.* preprint submitted to SIAM J. Numer. Anal., 2008.

[23] C. Karle, *Relativistic laser pulse compression and focusing in stratified plasma-vacuum systems*, Ph.D. Thesis, Institut für Theoretische Physik I, Heinrich-Heine-Universität, Düsseldorf, Germany, 2008.

[24] C. Karle, J. Schweitzer, M. Hochbruck, E. W. Laedke, and K.-H. Spatschek, *Numerical solution of nonlinear wave equations in stratified dispersive media*, J. Comp. Phys., 216 (2006), pp. 138–152.

[25] C. Karle, J. Schweitzer, M. Hochbruck, and K.-H. Spatschek, *A parallel implementation of a two-dimensional fluid laser-plasma integrator for stratified plasma-vacuum systems*, J. Comp. Phys., 227 (2008), pp. 7701–7719.

[26] S. Krogstad, *Generalized integrating factor methods for stiff PDEs*, J. Comp. Phys., 203 (2005), pp. 72–88.

[27] C. Moler and C. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Review, 45 (2003), pp. 3–49.

[28] A. Ostermann and M. Thalhammer, *Convergence of Runge-Kutta methods for nonlinear parabolic equations*, J. Appl. Numer. Math., 42 (2002), pp. 367–380. Numerical Solution of Differential and Differential-Algebraic Equations, 4-9 September 2000, Halle, Germany.

[29] A. Pazy, *Semigroups of Linear Operators and Applications to Partial Differential Equations*, Springer New York, 1983.

[30] A. Pukhov, *Three-dimensional electromagnetic relativistic particle-in-cell code VLPL (virtual laser plasma lab)*, J. Plasma Phys., 61 (1999), pp. 425–433.

[31] C. Ren, B. Duda, R. Hemker, W. Mori, T. Katsouleas, T. Antonsen, and P. Mora, *Compressing and focusing a short laser pulse by a thin plasma lens*, Phys. Rev. E, 63 (2001).

[32] H. Schwoerer, S. Pfotenhauer, O. Jäckel, K.-U. Amthor, B. Liesfeld, W. Ziegler, R. Sauerbrey, K. W. D. Ledingham, and T. Esirkepov, *Laser-plasma acceleration of quasi-monoenergetic protons from microstructured targets*, Nature, 439 (2006), pp. 445–448.

[33] O. Shorokhov, A. Pukhov, and I. Kostyukov, *Self-compression of laser pulses in plasma*, Phys. Rev. Lett., 91 (2003).

[34] B. P. Sommeijer, L. F. Shampine, and J. G. Verwer, *RKC: An explicit solver for parabolic PDEs*, J. Comp. Appl. Math., 88 (1998), pp. 315–326.

[35] D. Strickland and G. Mourou, *Compression of amplified chirped optical pulses*, Opt. Commun., 56 (1985), pp. 219–221.

[36] M. Tokman, *Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods*, J. Comp. Phys., 213 (2006), pp. 748–776.

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die zum Gelingen dieser Dissertation beigetragen haben.

An erster Stelle gilt mein Dank Prof. Dr. Marlis Hochbruck für die Vergabe dieses interessanten Themas, für die sehr gute Betreuung dieser Arbeit und dafür, dass sie die Arbeit in ihrer Gruppe stets interessant und angenehm gestaltet.

Außerdem bedanke ich mich bei Prof. Dr. Karl-Heinz Spatschek für die Übernahme des Koreferats.

Dr. Christoph Karle hat mir durch geduldige Erklärungen den Einstieg in die Physik sehr erleichtert. Dir danke ich ganz besonders für die produktive Zusammenarbeit, für die vielen inspirierenden Gespräche und für die Umsetzung unserer Programme in C++.

Prof. Dr. Alexander Ostermann und Dr. Ernst Wolfgang Laedke danke ich ebenfalls für unsere Zusammenarbeit.

Mein besonderer Dank gilt meinen Eltern Christa und Michael Schweitzer und meinem Lebensgefährten Karsten Hax. Ihr habt mit mir alle Höhen und Tiefen durchlebt und mich immer wieder motiviert und unterstützt.

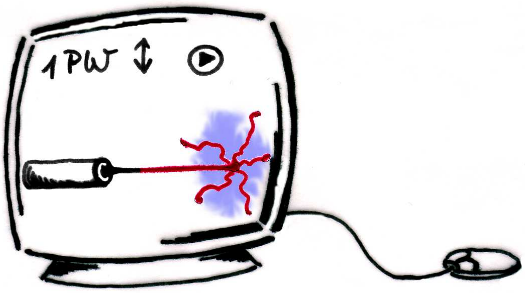
Erklärung

Die hier vorgelegte Dissertation habe ich eigenständig und ohne unerlaubte Hilfe angefertigt. Die Dissertation wurde in der vorgelegten oder in ähnlicher Form noch bei keiner anderen Institution eingereicht. Ich habe bisher keine erfolglosen Promotionsversuche unternommen.

I do herewith declare that the material contained in this dissertation is an original work performed by myself without illegitimate help. The material in this thesis has not been previously submitted for a degree at any university.

Düsseldorf, den 12.06.2008

(Julia Schweitzer)


1PW ↕ ▷