

A parallel implementation of a two-dimensional fluid laser-plasma integrator for stratified plasma-vacuum systems

Ch. Karle^a, J. Schweitzer^b, M. Hochbruck^b, K.H. Spatschek^{a,*}

^a*Institut für Theoretische Physik, Heinrich–Heine–Universität Düsseldorf,
D–40225 Düsseldorf, Germany*

^b*Mathematisches Institut, Heinrich–Heine–Universität Düsseldorf,
D–40225 Düsseldorf, Germany*

Abstract

A two-dimensional fluid laser-plasma integrator for stratified plasma-vacuum systems is presented. Inside a plasma, a laser pulse can be longitudinally compressed from ten or more wave lengths to one or two cycles. However, for physically realistic simulations, transversal effects have to be included, because transversal instabilities can destroy the pulse and transversal compression in the plasma as well as focusing in vacuum allows much higher intensities to be reached. In contrast to the one-dimensional case, where a two-step implementation of the Gautschi-type exponential integrator with constant step size turned out to be sufficient, it is essential to enable changes of the time step-size for the two-dimensional case. The use of a one-step version of the Gautschi-type integrator, being accurate of second order independent of the highest frequencies arising in the system, is proposed. In vacuum this allows to take arbitrarily large time steps. To optimize runtime and memory requirements within the plasma, a splitting of the Laplacian is suggested. This splitting allows to evaluate the matrix functions arising in the Gautschi-type method by one-dimensional Fourier transforms. It is also demonstrated how the different variants of the scheme can be parallelized. Numerical experiments illustrate the superior performance and accuracy of the integrator compared to the standard leap-frog method. Finally, we discuss the simulation of a layered plasma vacuum structure using the new method.

Key words: Nonlinear wave propagation, relativistic plasma dynamics, Gautschi–type integrator, quasi–envelope approach

PACS: 02.70.-c; 02.30.Jr; 05.45.Yv; 52.27.Ny

* Corresponding author.

Email address: khs@tp1.uni-duesseldorf.de (K.H. Spatschek).

1 Introduction

The huge progress in laser technology enabled a new area of relativistic laser-plasma-interaction with many fundamental applications. The so called relativistic optics [1] is still rapidly growing. When a relativistically intense laser pulse is propagating through a plasma, many new physical effects appear [2,3]. Among the latter are relativistic optical guiding, relativistic self-compression, filamentation, harmonic excitation, wake-field generation, and laser pulse frequency variations (photon acceleration). Electrons may be accelerated up to GeV energies and the ions can gain multi-MeV energies.

The CPA (chirped pulse amplification) technology [1] allowed to considerably compress the laser pulse in the longitudinal direction. It became possible to build short-pulse lasers with high powers and large focused intensities. A significant new effect is that short and intense electromagnetic pulses create intensive wakes of plasma oscillations behind themselves. The electrostatic oscillations have a frequency close to the plasma frequency and a wave-number such that the phase velocity of the oscillations is close to the speed of light. Particles being properly injected into the wake-field [4] may be trapped in there and can be accelerated to high energies. Wake-field generation [5] became a very interesting phenomenon of short pulse laser physics, with a huge potential for practical applications. The so called bubble acceleration [6] turned out to be a very effective process. Laser-plasma accelerators have been proposed as a next generation of compact accelerators because of the large electric fields they can sustain over short distances compared to conventional accelerator technology [7–9].

In the present paper we use a reduced fluid description with the aim of modeling laser pulse propagation in layered plasma-vacuum structures at weakly relativistic intensities (Section 2). At these intensities, efficient longitudinal pulse compression is possible with low energy losses to the plasma, because the relativistic mass nonlinearity dominates over ponderomotive and other nonlinear effects. Inside a plasma a laser pulse can be longitudinally compressed from ten or more wavelengths to one or two cycles [10], much shorter than currently possible with CPA alone. In [11] highly accurate and efficient methods were developed to simulate this one-dimensional pulse compression with much less numerical noise and much shorter runtimes than e.g. PIC simulations. But for physically realistic simulations, transversal effects have to be included, because transversal instabilities can destroy the pulse [10] and transversal compression in the plasma as well as focusing in vacuum [12] allows much higher intensities to be reached. In the present paper we develop the implementation of a two-dimensional integrator based on a one step Gautschi-type exponential integration scheme to study the longitudinal compression and especially the transversal focusing properties of layered plasma-vacuum systems. For two space dimensions the numerical efficiency becomes truly important, especially for the intended application. This is because a large number of pa-

rameters influences the compression and focusing of the pulse, like plasma density, pulse lengths, width and amplitude, number of plasma layers, relative length of the layers, amount of vacuum between the layers and thus a large number of simulation runs is required for the optimization.

In [11], we suggested to use a Gautschi-type exponential integrator (see [13,14] and references given there) for the time-integration of the one-dimensional problem and we presented comparisons which showed that such trigonometric integrators outperform the standard leap-frog method for this particular application. For the spatial discretization we introduced the so-called quasi-envelope approach (QEA) to reduce the number of spatial grid points significantly.

The results achieved with this one-dimensional code motivated us to generalize the techniques to the two-dimensional case (Section 3). However, in contrast to the one-dimensional case, where a two-step implementation of the Gautschi-type exponential integrator with constant step size turned out to be sufficient, it is essential to enable changes of the time-step-size for the two-dimensional case. In Section 3.1, we therefore suggest to use a one-step version of the Gautschi-type integrator [13,15]. An error analysis for the whole family of these methods is given in [16]. In particular, it was shown that these methods are second order accurate independent of the highest frequencies arising in the system.

The main computational effort for one time-step with an exponential integrator arises from the computation or approximation of the product of a matrix function with a vector. In the one-dimensional case, the proposed pseudospectral discretization enabled the use of fast Fourier transformation. This lead to an implementation where the overhead compared to the leap-frog method was quite low. The situation changes completely in two space dimensions. We thus suggest to use the full two-dimensional Laplacian within the matrix functions only in vacuum, where huge time-steps can be performed and higher costs pay off. During propagation in plasma, we split the Laplacian into a transversal and a longitudinal direction and use only the (one-dimensional) longitudinal direction with the matrix functions. This splitting is justified by physical properties of the solution (Section 3.2–3.5).

Nevertheless, for large problems it can be necessary to parallelize the scheme. A key observation is that the parallelization has to be adapted to the different variations of the integrator being applied in different regions during the simulation (e.g. vacuum, plasma, and transition regions). We will discuss our implementation in Section 3.6.

In Section 4 we present numerical experiments to illustrate the performance of our scheme and we discuss a physical application.

2 Physical model

In this paper we consider the following two-dimensional model

$$\frac{\partial^2}{\partial t^2} a = \Delta a - \frac{n_0}{n_c} \frac{(n_e^0 + n_e^1)}{\gamma} a, \quad (1)$$

$$\frac{\partial^2}{\partial t^2} n_e^1 = -\frac{n_0}{n_c} n_e^0 n_e^1 + n_e^0 \Delta \gamma \quad (2)$$

where a is the normalized vector potential, γ the relativistic factor,

$$\gamma^2 = 1 + |a|^2 \quad (3)$$

and n_e^1 denotes the electron density variation. Δ is the two-dimensional Laplacian operator, $\frac{n_0}{n_c}$ is a density parameter where $n_0 < n_c$ to allow for pulse propagation inside of the plasma. n_e^0 is the normalized, time-independent density profile. To avoid reflections at plasma boundaries we only consider densities changing linearly at plasma boundaries.

This model has been introduced in [11] for the one-dimensional case and is valid in the weakly relativistic regime, i.e. $|\mathbf{p}|_e \ll m_e c$ for the electrons moving in the laser field. Thus the initial amplitude scales as $a_0 \sim \varepsilon \ll 1$ and $n_e^1 \sim \varepsilon^2$.

A derivation of this model including details of the scaling can be found in Appendix A.

Note, that in vacuum $n_e^0 \equiv 0$ and therefore $n_e^1 \equiv n_e^1(0) = 0$. Thus, in vacuum, only a two-dimensional linear wave equation for the vector potential remains to be solved.

3 Numerical schemes

3.1 Gautschi-type exponential integrator for time-discretization

After semi discretization in space (cf. Sec. 3.3), we obtain a system of second order ordinary differential equations of the form

$$\ddot{y}(t) = -\Omega^2 y(t) + F(y(t)) \quad (4)$$

where Ω^2 is a symmetric, positive semi-definite matrix and $\|F\|$, $\|F_y\|$, $\|F_{yy}\|$ and $\|y'\|^2 + \|\Omega y\|^2$ are bounded. For the solution we suggest to apply the following

family of numerical schemes [15, 17]

$$y_{n+1} = \cos(h\Omega) y_n + h \operatorname{sinc}(h\Omega) \dot{y}_n + \frac{1}{2} h^2 \Psi F(\Phi y_n) \quad (5a)$$

$$\dot{y}_{n+1} = -\Omega \sin(h\Omega) y_n + \cos(h\Omega) \dot{y}_n + \frac{1}{2} h \left(\Psi_0 F(\Phi y_n) + \Psi_1 F(\Phi y_{n+1}) \right). \quad (5b)$$

Here,

$$\Phi = \phi(h\Omega), \quad \Psi = \psi(h\Omega), \quad \Psi_0 = \psi_0(h\Omega), \quad \Psi_1 = \psi_1(h\Omega),$$

where $\phi, \psi, \psi_0, \psi_1$ are even and analytical functions which are bounded on the non-negative real axis satisfying

$$\phi(0) = \psi(0) = \psi_0(0) = \psi_1(0) = 1.$$

To obtain a scheme with certain desirable properties imposes constraints on the choice of these functions. For instance, a scheme is symmetric if and only if

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

and symplecticity is equivalent to

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

Moreover, Hairer and Lubich [15] proved that for $\Omega = \omega I$, $\omega > 0$ and $F(y) = By$ with constant B , the energy is conserved up to $O(h)$ for all values of $h\omega$ if and only if

$$\psi(\xi) = \operatorname{sinc}^2(\xi) \phi(\xi) \quad (6)$$

Thus, there is no scheme in this family which is symplectic *and* gives good energy conservation.

In [16], Grimm and Hochbruck derived criteria, which guarantee second order accuracy independent of the eigenvalues of Ω . They suggested to choose

$$\psi(\xi) = \operatorname{sinc}^3(\xi), \quad \phi(\xi) = \operatorname{sinc}(\xi), \quad (7)$$

which results in a symmetric second order scheme satisfying (6). We use this choice of functions for our implementation.

Note, that linear problems with $F \equiv 0$ are solved exactly by all these schemes. 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 [11], which does not have the favorable energy conservation property. The only advantage of the two-step method is that problems with $F \equiv c$ are solved exactly even for $c \neq 0$, but this case is not relevant for our application.

3.2 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 spectral 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 $O(N_z N_x (\log N_z + \log N_x))$ 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, diagonalization of a large matrix Ω resulting from finite difference or finite element discretization is impossible. An alternative is to use Krylov subspace methods such as the symmetric Lanczos process [18, 19]. However, for the applications considered here such techniques were not competitive.

Therefore, we propose to use different spatial discretizations in different regimes depending on physical properties of the solution. Moreover, we alter the splittings in (4) during the time integration, i.e. we move parts of the discretized Laplacian into the function F . This allows for an efficient evaluation of the matrix functions.

3.3 Spatial discretization

We consider the Laplacian in Cartesian coordinates as well as cylindrical coordinates. The z -coordinate always resembles the propagation direction of the laser pulse whereas x or r , respectively, denote the transversal direction. The dependence on y or the angular coordinate, respectively, are omitted in the two-dimensional case.

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.

3.3.1 Cartesian coordinates in vacuum

In vacuum we only need to solve the linear wave equation

$$\frac{\partial^2}{\partial t^2} a = \frac{\partial^2}{\partial z^2} a + \frac{\partial^2}{\partial x^2} a. \quad (8)$$

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 $a = a(t) \in \mathbb{C}^{N_z \times N_x}$ and $\dot{a} = \dot{a}(t) \in \mathbb{C}^{N_z \times N_x}$ be complex matrices containing approximations to the vector potential and its time derivative on the grid,

$$a_{i,j} \approx a(x_j, z_i, t), \quad \dot{a}_{i,j} \approx \frac{\partial}{\partial t} a(x_j, z_i, t).$$

The Laplacian is approximated by

$$\Delta a \approx \mathcal{F}_{N_z}^{-1} D_z^2 \mathcal{F}_{N_z} a + a \mathcal{F}_{N_x}^T D_x^2 \mathcal{F}_{N_x}^{-T}$$

where

$$D_k = \frac{2\pi i}{L_k} \text{diag} \left(-\frac{N_k}{2}, \dots, \frac{N_k}{2} - 1 \right), \quad k = x, z,$$

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

Formally, the matrices a and \dot{a} can be reorganized by writing them column-wise into long vectors. Then the spatially discretized equation (8) can be written as a system of differential equations (4), where Ω is a matrix which can be diagonalized via two-dimensional fast Fourier transforms and $F \equiv 0$. However, for the implementation, the matrix notation is more efficient.

In the first time-step, where the initial data is given in physical space, we start by performing a two-dimensional Fourier transform by applying fast (one-dimensional) Fourier transforms to all columns and rows of a and \dot{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 a and \dot{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 physical space, inverse Fourier transforms have to be applied to all rows and columns of a and \dot{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 a and \dot{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.

3.3.2 Cartesian coordinates in plasma

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

$$\frac{\partial^2}{\partial t^2} a - \frac{\partial^2}{\partial z^2} a - \frac{\partial^2}{\partial x^2} a = -\frac{n_0}{n_c} \frac{n_e^0 + n_e^1}{\gamma} a \quad (9a)$$

$$\frac{\partial^2}{\partial t^2} n_e^1 + \frac{n_0}{n_c} n_e^0 n_e^1 = n_e^0 \Delta \gamma. \quad (9b)$$

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 diagonal operator of the second equation. 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 matrix operators than in vacuum. The main costs 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 the discretized system exactly anymore. However, the time-step size is only limited by accuracy, not by stability. This is in contrast to the well known leap-frog method, where stability requires to use time-steps proportional to the inverse of the largest eigenvalue of the linear part. We will present comparisons in Sec. 4.2. For more detailed comparisons for one-dimensional problems we refer to [11].

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. 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 (solid) and the imaginary (dashed) 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 ($\Omega \approx \Delta_{\parallel}$) whereas the transversal part is added to the nonlinearity $F(y)$. To avoid the expensive two-dimensional Fourier transformations, we propose to use fourth-order finite

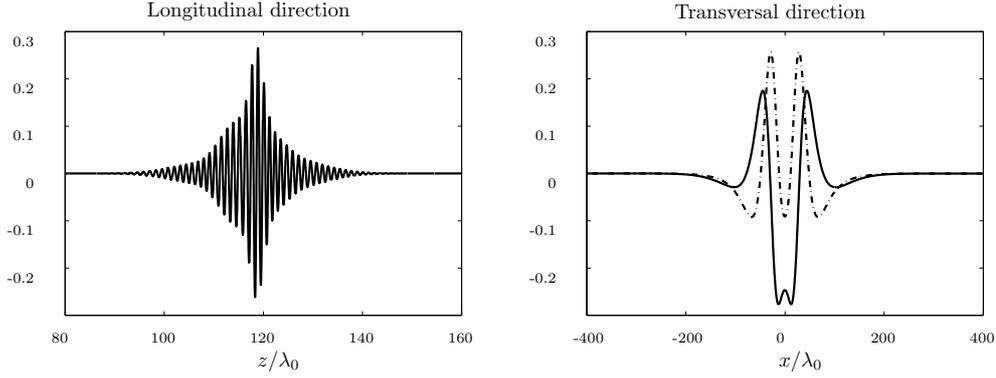


Fig. 1. **Left:** The spatial distribution of the real part of the solution in longitudinal direction through the center of the pulse is highly oscillatory. **Right:** The spatial distribution of the real (solid) and imaginary (dashed) part of the solution in transversal direction through the maximum of the pulse is smooth.

differences in this direction.

Due to this splitting, the longitudinal part of the Laplacian, can be diagonalized by 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 physical space. 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 circular polarized laser beams. Thus it is sufficient to use fourth order finite differences in *both* directions to approximate the inhomogeneity of the density equation.

3.3.3 Cylindrical coordinates

For the equations formulated in cylindrical coordinates

$$\frac{\partial^2}{\partial t^2} a = \frac{\partial^2}{\partial z^2} a + \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} a \right) - \frac{n_0}{n_c} \frac{n_e^0 + n_e^1}{\gamma} a \quad (10a)$$

$$\frac{\partial^2}{\partial t^2} n_e^1 = -\frac{n_0}{n_c} n_e^0 n_e^1 + n_e^0 \Delta \gamma \quad (10b)$$

we basically use the same ideas as for Cartesian coordinates in plasma regions, i.e., we use $\Omega \approx \Delta_{\parallel}$ 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.

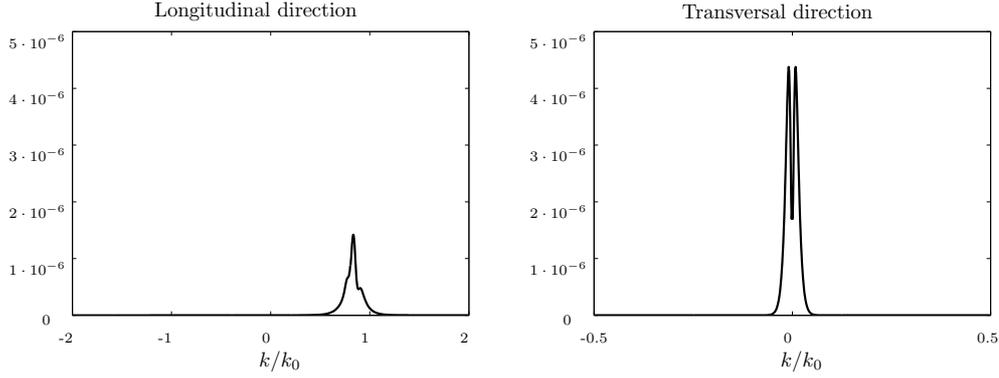


Fig. 2. The spectrum of the longitudinal spatial distribution (left) is not centered around 0, other than that of the transversal spatial distribution (right).

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.

3.3.4 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. 2, left. The idea of QEA is to shift the spectrum appropriately, see [11] for details. 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 direction, as can be seen in Fig. 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

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

which leads to a new equation for \tilde{a}

$$\frac{\partial^2}{\partial t^2} \tilde{a} = \left(\frac{\partial}{\partial z} + i\kappa \right)^2 \tilde{a} + \Delta_{\perp} \tilde{a} - \frac{n_0 n_e^0 + n_e^1}{n_c \gamma} \tilde{a} \quad , \quad \gamma^2 = 1 + |\tilde{a}|^2 .$$

The value of κ is chosen depending on the position of the pulse, namely $\kappa = \sqrt{1 - \frac{n_0}{n_c}}$ or $\kappa = 1$ or the mean value of both [11].

3.4 *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 [11]. For pseudo-spectral discretization this only requires a larger array in Fourier space where extra entries are filled with zeros. But the computation is much more expensive for the finer grid, thus interpolation is avoided unless absolutely necessary. Therefore, we also use a rather tight estimate for the pulse to be nonzero.

3.5 *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 cost 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 a and n_e^1 . n_e^0 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.

3.6 *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.

3.6.1 *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 transform¹. Of course the application of the matrix function is also spread over the processors involved.

3.6.2 *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, 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. 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

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

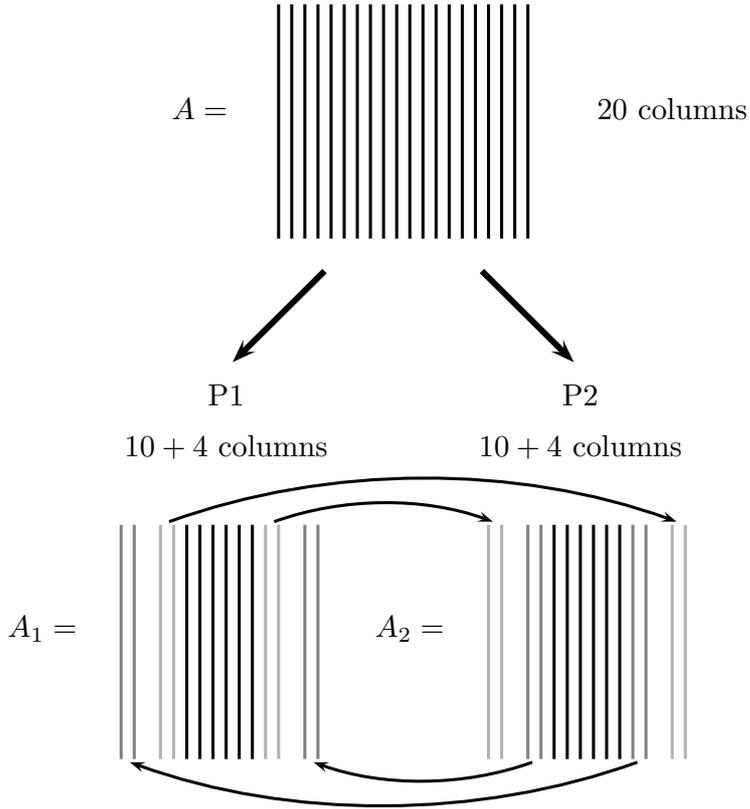


Fig. 3. Example parallelization scheme for two processors, Cartesian coordinates in plasma, periodic boundary conditions and 20 grid points in transversal direction. The colored columns have to be communicated between the processors for the evaluation of the transversal Laplacian with finite differences and are stored twice.

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 Exemplary results

4.1 Splitting of the Laplacian

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

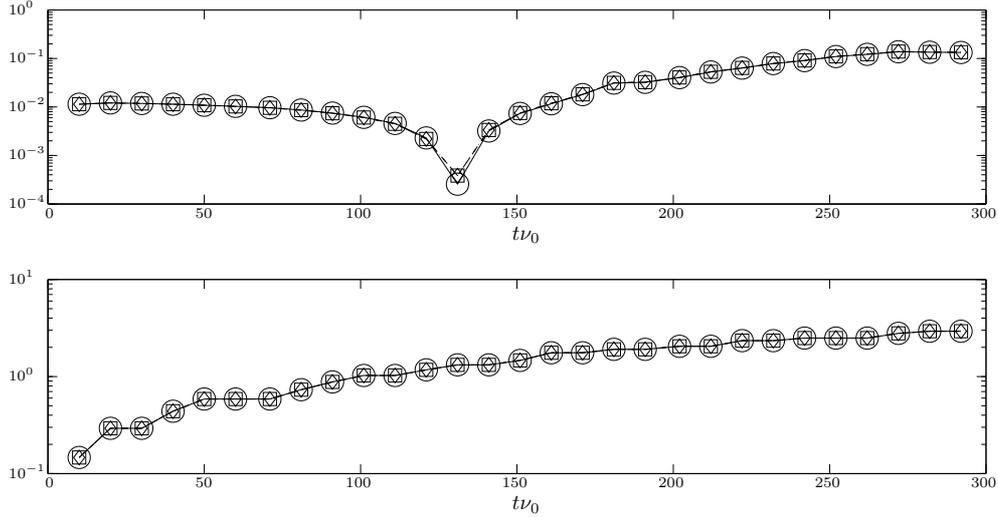


Fig. 4. The relative error of the maximum squared amplitude is shown in the upper picture and the absolute error of the position of the maximum in wave-lengths is drawn in the lower picture. The curves marked by circles are the errors of the Gautschi-type method applied to the full Laplacian, the squares are the errors of the splitted method with Fourier spectral discretization in both directions and the diamonds are those for the splitted method with finite differences in transversal direction.

cubic nonlinearity

$$\frac{\partial^2}{\partial t^2} a = \Delta a - \frac{n_0}{n_c} \left(1 - \frac{1}{2} |a|^2\right) a, \quad \frac{n_0}{n_c} = 0.3. \quad (11)$$

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

$$a(x, z, t) = a_0 e^{-\frac{(z-z_0-k_0t)^2}{l_0^2}} e^{-\frac{x^2}{w_0^2}} e^{i(k_0z-z_0-t)} \quad (12)$$

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 $k_0 = \sqrt{1 - \frac{n_0}{n_c}}$ the plasma wave-length.

This is solved for Cartesian coordinates $(x, z) \in [-300\lambda_0, 300\lambda_0] \times [0\lambda_0, 300\lambda_0]$ and $t \in [0/v_0, 300/v_0]$. We use 1024 grid points in z -direction and 512 grid points in x -direction. The time-step size is chosen as $0.2 dz$. For the reference solution we use twice as many points in both spatial directions, whereas for the time-discretization we choose a fourth of the original time-step. For the error calculation we Fourier interpolate the solutions to the finer grid.

In Fig. 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

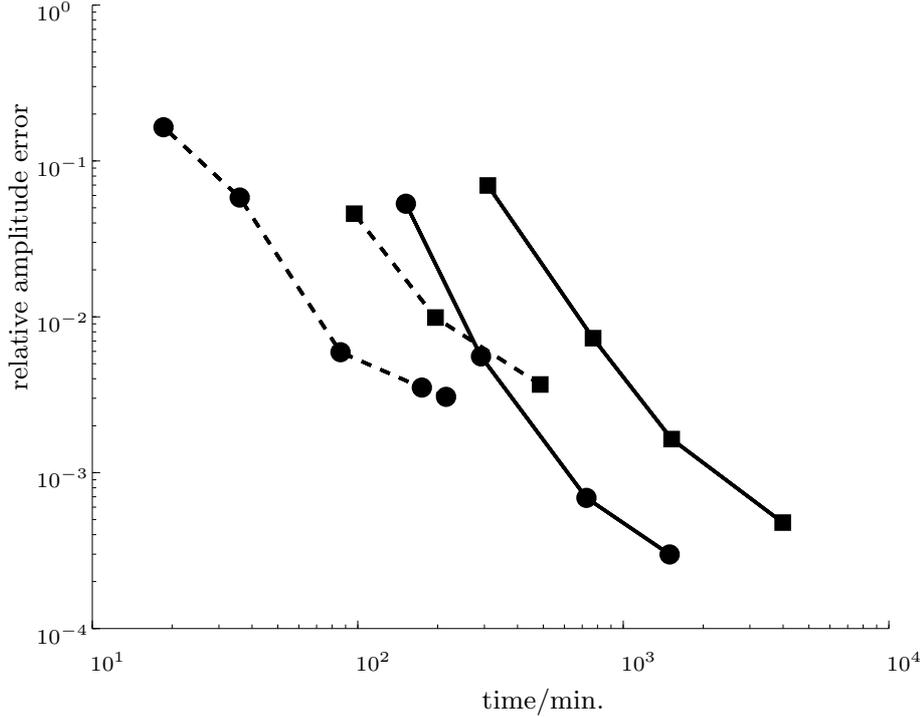


Fig. 5. The relative error in the maximum amplitude is plotted over the run time in minutes. Circles: Gautschi-type integrator. Squares: leap-frog. Dashed: coarse spatial discretization. Line: fine spatial discretization. Along each curve the value of τ varies.

the absolute error of the position of the maximum in wave-lengths. For each type of error there are three different curves. The circular marks show the error of the Gautschi-type method applied to the full Laplacian, discretized via Fourier spectral method in both directions. The square marks are 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 diamond marks represent the error of the splitted method, but this time with fourth order finite differences in transversal direction. We can see, that the three error curves are nearly indistinguishable, which proves that the splitting does not degrade accuracy.

4.2 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 (9). The density layer starts at $250\lambda_0$ with a linear increase up to $\frac{n_0}{n_c} = 0.3$ 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 (12) 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 $t = 1240/v_0$, 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.

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. 5 the amplitude error of the Gautschi-type method (circles) and the leap-frog method (squares) is plotted against computation time in plasma. The dashed line represents a coarse spatial discretization with 1024 grid points in longitudinal direction, where dz is chosen to be $0.352\lambda_0$ and 400 grid points in transversal direction with $dx = 2\lambda_0$. The continuous line gives the errors for a fine spatial discretization with $N_z = 2048$, $dz = 0.176\lambda_0$, $N_x = 800$ and $dx = \lambda_0$. 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 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 [11]. But 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. 6. Thus it is even more efficient to invest in a more sophisticated algorithm and benefit from the larger time-steps.

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

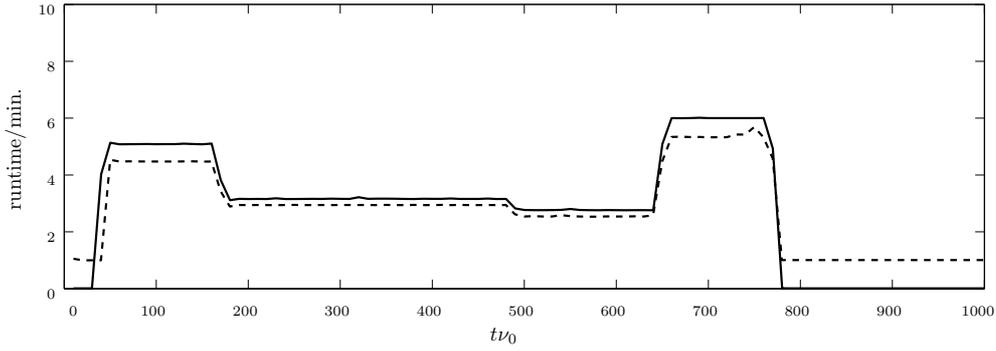


Fig. 6. For the Gautschi-type method (solid) and the leap-frog method (dashed) the runtime between outputs (i.e. 228 time-steps, except for vacuum steps with the Gautschi-type method) is shown. The spatial *and* time-discretization is the same for both schemes except

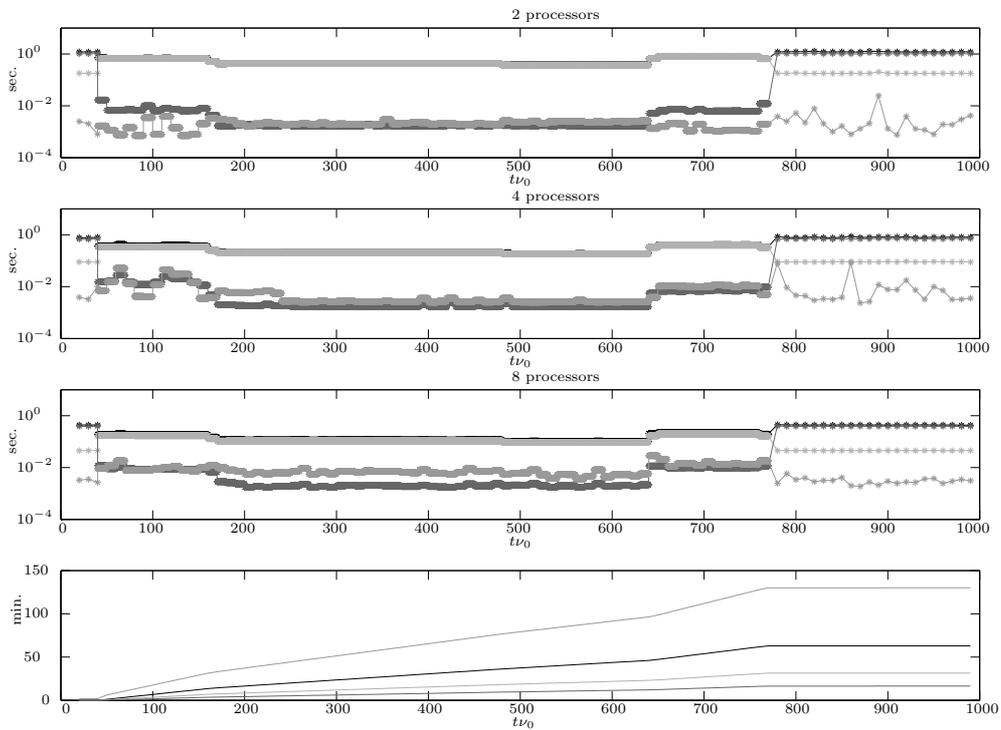


Fig. 7. The upper three pictures show the full time (black), pure number crunching time (light gray), data receive time (dark gray) and synchronization time (middle gray) per time-step for two, four and eight processors respectively. The fourth picture shows the accumulated full integration time for one, two, four and eight processors (curves from top to bottom).

In the upper three pictures of Fig. 7 full time (black), pure number crunching time (light gray), data receive time (dark gray) and synchronization time² (middle gray) 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

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

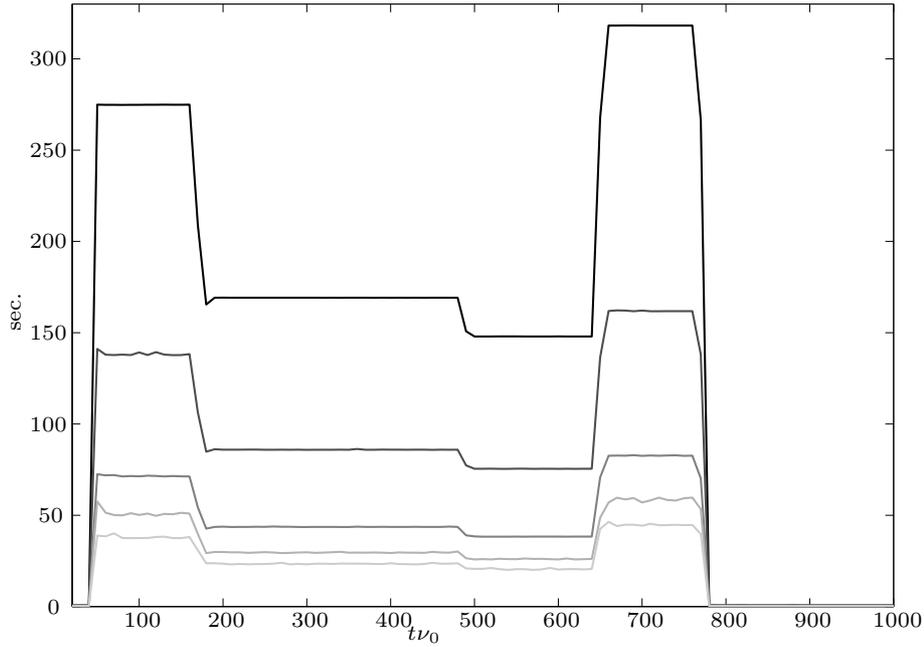


Fig. 8. The runtime per output step for two, four, six and eight processors (curves from dark to light gray) respectively compared to single processor runtime (black).

is the vacuum step region. Here, the crunching time is quite low, since we neither calculate the nonlinearity nor the density equation. However, due to the matrix transpositions the communication time is rather long.

The other two cases are the plasma and 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 times 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. This is illustrated in the fourth picture of Fig. 7, 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 directly followed by the plasma regions. We can also see, that in comparison 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.

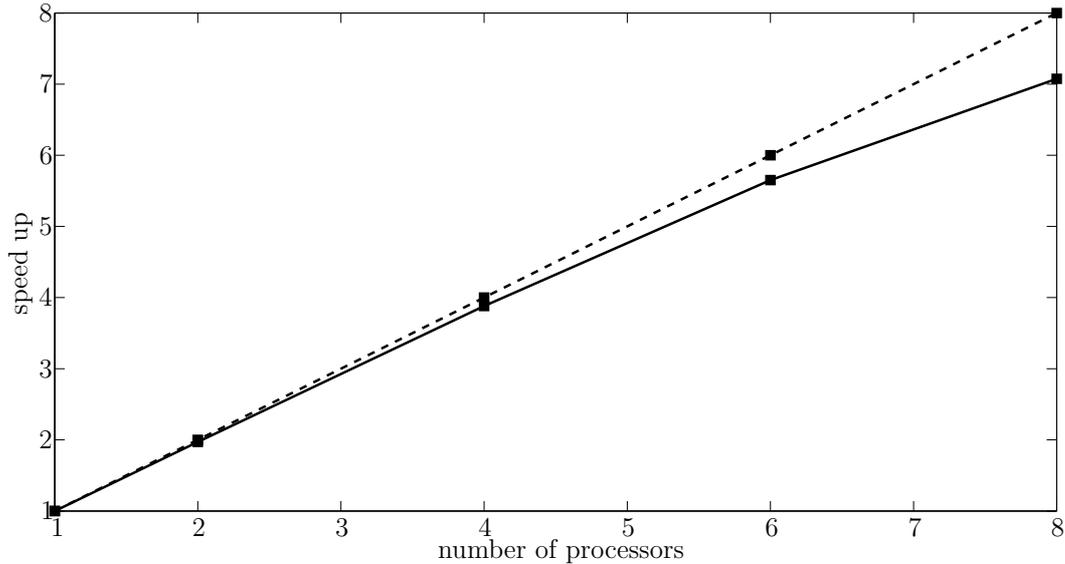


Fig. 9. Speed up factor for 2, 4, 6 and 8 processors (solid) vs. ideal scaling (dashed).

The runtime per output step is shown in Fig. 8. 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. 9).

4.4 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 [10]. To reach high subrelativistic or even relativistic intensities, the pulse has to be transversely focused as well. Fortunately the plasma induces a negative curvature of the phase front of the

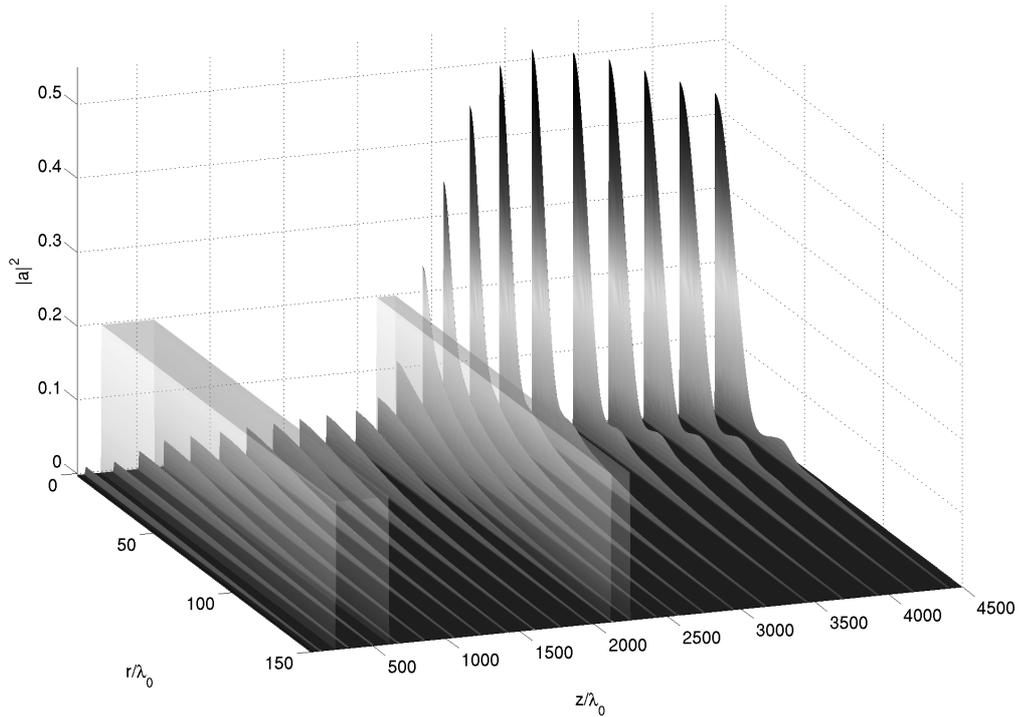


Fig. 10. An initially (in both directions) Gaussian pulse with $a_0 = 0.1$, $L_0 = 10\lambda_0$ and $W_0 = 150\lambda_0$ propagates through two plasma layers of density $n_0 = 0.3n_c$ and different lengths. The first layer is $330\lambda_0$ long and the second $125\lambda_0$ with $1500\lambda_0$ vacuum in between.

pulse, which leads to focusing of the pulse behind the plasma layer [12]. 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. 10; the amplitude curve and the radial profiles at different times are shown in Fig. 11. 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. 13). 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. 12, 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 [10]. 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

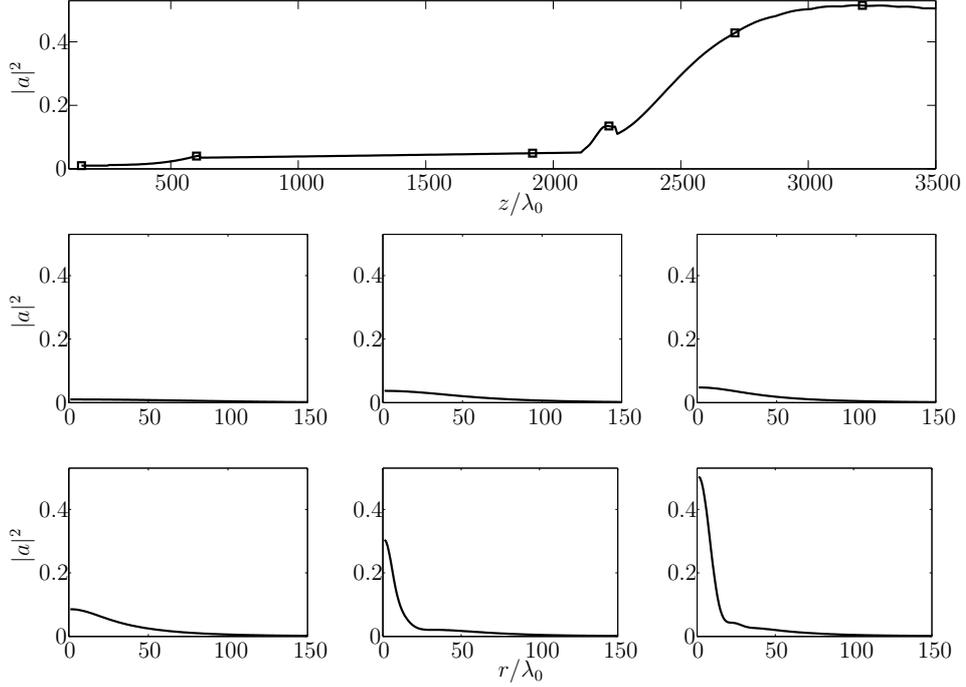


Fig. 11. Plots of the radial profile of Fig. 10 at six different times/locations, marked with squares in the plot of the maximum amplitude (top). Time evolution from left to right, top to bottom.

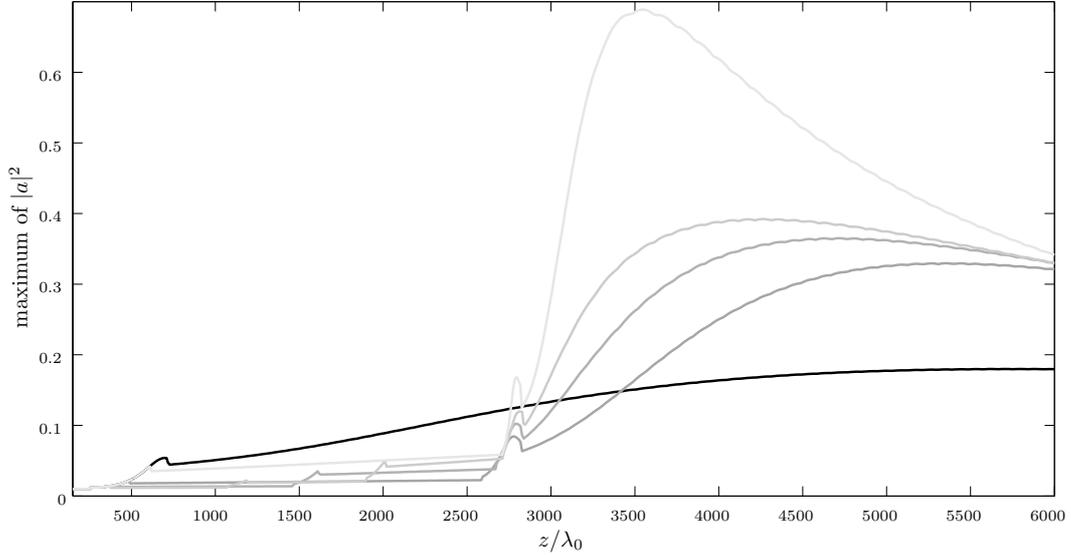


Fig. 12. Maximum amplitude of the same initial pulse as in Fig. 10 as it propagates through one to four plasma layers of density $n_0 = 0.3n_c$. The curves are in increasing lightness: one layer ($460\lambda_0$), two layers ($227\lambda_0$ each), three layers ($150\lambda_0$ each), four layers ($112\lambda_0$ each) and again two layers ($355\lambda_0$ and $100\lambda_0$). In each case there is a total of $2100\lambda_0$ vacuum between the layers.

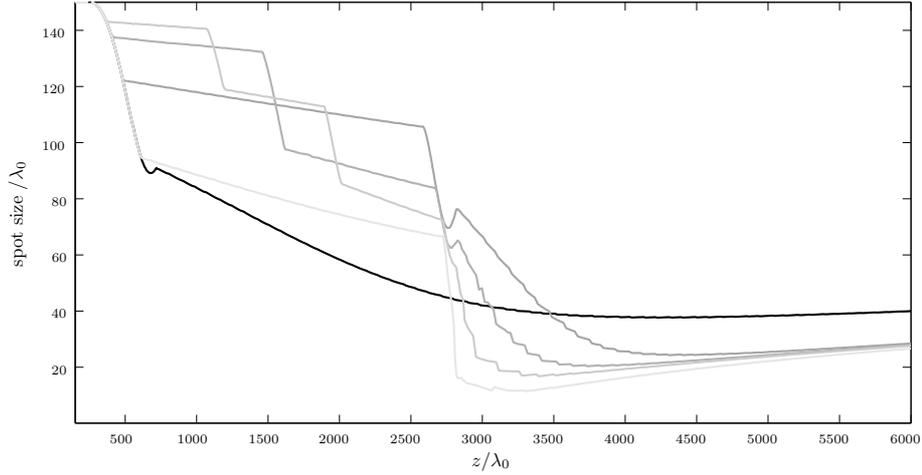


Fig. 13. Spot size of the same initial pulse as in Fig. 10 propagating through the same plasma / vacuum configurations as in Fig. 12.

can be found in [20].

5 Conclusion

We have detailed the implementation of a two-dimensional integrator for weakly relativistic laser-plasma interaction. Special attention was paid to efficiently simulate systems with large amounts of vacuum between the plasma layers, by selecting a time-integration method that is in principle able to use arbitrarily large steps in vacuum. This is necessary for the study of transversal focusing of a laser pulse by layered plasma and vacuum structures. We have shown this integrator to be highly accurate and to be distinctly superior to the standard leap-frog method in terms of stability, allowed step size and the integrator runtimes for the same accuracy. We also discussed how other methods to reduce the computational complexity, like QEA which was introduced in [11] and a moving simulation window, can be combined with this time-integration scheme.

The proposed implementation can be directly extended to the three-dimensional case. Then parallelization becomes strictly necessary and even for large numbers of processors we expect the scaling to be almost linear.

6 Acknowledgment

This work has been supported by the Deutsche Forschungsgemeinschaft through the Transregio-SFB 18 and the Research Training Group GRK 1203.

A Details of the physical model

The high-frequency fluid-Maxwell model consists of the wave equation, Poisson's equation, and the electron continuity and momentum equations. In dimensionless form, these equations are

$$\frac{\partial^2}{\partial t^2} \mathbf{A} - \Delta \mathbf{A} - \frac{\partial}{\partial t} \nabla \phi = -\frac{n_0}{n_c} n_e \frac{\mathbf{P}}{\gamma} \quad (\text{A.1})$$

$$\Delta \phi = \frac{n_0}{n_c} (n_i - n_e)$$

$$\frac{\partial}{\partial t} n_e + \nabla \cdot \left(n_e \frac{\mathbf{P}}{\gamma} \right) = 0 \quad (\text{A.2})$$

$$\frac{\partial}{\partial t} (\mathbf{P} - \mathbf{A}) - \frac{\mathbf{P}}{\gamma} \times \{ \nabla \times (\mathbf{P} - \mathbf{A}) \} = \nabla (\phi - \gamma) \quad (\text{A.3})$$

where $\gamma = \sqrt{1 + \mathbf{P}^2}$ is the relativistic factor. We use the Coulomb gauge condition $\nabla \cdot \mathbf{A} = 0$. The vector potential \mathbf{A} is measured in the unit e/mc^2 ; the same is used for the non-dimensional electrostatic potential ϕ . The unit for the density is the maximum value n_0 of the ion background density, whereas n_c is the so called critical density. Since we are interested in laser propagation in an underdense plasma, $n_0/n_c < 1$ holds. The momentum \mathbf{P} is measured in mc , where m , e , c are electron mass, elementary charge, and speed of light, respectively.

The momentum balance (A.3) can be simplified further by choosing an initial gauge condition which does not restrict generality. We define the projection operators Π_c and Π_g such that any field \mathbf{u} is decomposed into $\mathbf{u} = \mathbf{v} + \mathbf{w}$ with the following properties

$$\begin{aligned} \Pi_c \mathbf{u} = \mathbf{v} &\equiv \mathbf{u}_c, \quad \nabla \times \mathbf{v} = 0, \quad \text{but generally } \nabla \cdot \mathbf{v} \neq 0, \\ \Pi_g \mathbf{u} = \mathbf{w} &\equiv \mathbf{u}_g, \quad \nabla \cdot \mathbf{w} = 0, \quad \text{but generally } \nabla \times \mathbf{w} \neq 0, \end{aligned}$$

with

$$\Pi_c + \Pi_g = \mathbf{1}.$$

Clearly, \mathbf{v} is a gradient field, and \mathbf{w} is a rotation field. The operators can be represented as

$$\Pi_c = \nabla \Delta^{-1} \nabla \cdot \quad \text{and} \quad \Pi_g = 1 - \nabla \Delta^{-1} \nabla \cdot .$$

Applying the projection operators to the momentum balance (A.3) allows to split the equation in a divergence-free and a curl-free part. The equation

$$\frac{\partial}{\partial t} (\mathbf{P}_g - \mathbf{A}) - \Pi_g \left[\frac{\mathbf{P}}{\gamma} \times \{ \nabla \times (\mathbf{P}_g - \mathbf{A}) \} \right] = 0$$

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 $\mathbf{P}_g = \mathbf{A}$ the canon-

ical momentum stays curl-free for all times, i.e.

$$\mathbf{P}_{\text{can}} = \mathbf{P}_g + \mathbf{P}_c - \mathbf{A} = \mathbf{P}_c .$$

This initial condition simplifies the curl-free part to

$$\frac{\partial}{\partial t} \mathbf{P}_c = \nabla(\phi - \gamma) .$$

Since $\nabla \times \mathbf{P}_c = 0$, \mathbf{P}_c can be written in terms of a Clebsch-potential, $\mathbf{P}_c = \nabla\psi$. The integration of (A.3) thus leads to

$$\frac{\partial}{\partial t} \psi = \phi - \gamma + 1 .$$

Applying the splitting via Π_g and Π_c to the wave equation (A.1) for \mathbf{A} , we obtain for the divergence-free part

$$\frac{\partial^2}{\partial t^2} \mathbf{A} - \Delta \mathbf{A} = -\frac{n_0}{n_c} (1 - \nabla \Delta^{-1} \nabla \cdot) \left\{ \frac{n_e}{\gamma} (\mathbf{A} + \nabla \psi) \right\}$$

and for the curl-free part

$$\frac{\partial}{\partial t} \nabla \phi = -\frac{n_0}{n_c} \nabla \Delta^{-1} \nabla \cdot \left\{ \frac{n_e}{\gamma} (\mathbf{A} + \nabla \psi) \right\} .$$

Straightforward evaluations of the right-hand-sides lead to

$$\begin{aligned} \frac{\partial^2}{\partial t^2} \mathbf{A} - \Delta \mathbf{A} &= -\frac{n_0}{n_c} \left[\frac{n_e}{\gamma} \mathbf{A} - \Delta^{-1} \left\{ \nabla \left(\mathbf{A} \cdot \nabla \frac{n_e}{\gamma} \right) + \nabla \times \left[\left(\nabla \frac{n_e}{\gamma} \right) \times (\nabla \psi) \right] \right\} \right] \\ \frac{\partial}{\partial t} \nabla \phi &= -\frac{n_0}{n_c} \left[\frac{n_e}{\gamma} \nabla \psi + \Delta^{-1} \left\{ \nabla \left(\mathbf{A} \cdot \nabla \frac{n_e}{\gamma} \right) + \nabla \times \left[\left(\nabla \frac{n_e}{\gamma} \right) \times (\nabla \psi) \right] \right\} \right] . \end{aligned}$$

To further simplify the equations in the weakly relativistic regime, we scale the perpendicular variation with respect to the parallel changes (scaling parameter α) and introduce smallness parameters ε , μ , β , ρ , and δ for the amplitudes of the physical variables:

$$\begin{aligned} \mathbf{A}(\mathbf{r}, t) &= \varepsilon \{ \mathbf{A}_\perp(z, \alpha \mathbf{r}_\perp) + \mu \mathbf{e}_z A_\parallel(z, \alpha \mathbf{r}_\perp) \} \\ n_e(\mathbf{r}, t) &= n_e^0 + \beta n_e^1(z, \alpha \mathbf{r}_\perp) \\ \phi(\mathbf{r}, t) &= \rho \phi^1(z, \alpha \mathbf{r}_\perp) \\ \psi(\mathbf{r}, t) &= \delta \psi^1(z, \alpha \mathbf{r}_\perp) \\ \gamma(\mathbf{r}, t) &= 1 + \frac{\varepsilon^2}{2} |\mathbf{A}|^2 . \end{aligned}$$

The different smallness parameters are of course interrelated. In the following we justify the relations

$$\mu = \alpha \ll 1 \quad \text{and} \quad \delta = \rho = \beta = \varepsilon^2 \ll 1$$

between these parameters. From the Coulomb gauge we get

$$\nabla \cdot \mathbf{A} = \varepsilon \{ \alpha \nabla_{\perp} \cdot \mathbf{A}_{\perp} + \mu \partial_z A_{\parallel} \} = 0 \Rightarrow \mu = \alpha .$$

The Laplace-equation for ϕ yields

$$\rho \Delta \phi^1 = \frac{n_0}{n_c} (n_e - n_e^0) = \frac{n_0}{n_c} \beta n_e^1 \Rightarrow \rho = \beta .$$

The reduced momentum balance

$$\delta \partial_t \psi^1 = \rho \phi^1 - (\gamma - 1) = \rho \phi^1 + o(\varepsilon^2)$$

and the continuity equation

$$\rho \nabla \cdot \partial_t \nabla \phi^1 = \beta \partial_t n_e^1 = -\frac{n_0}{n_c} n_e^0 \delta \Delta \psi^1 + \dots$$

are consistent with $\delta = \rho = \beta = \varepsilon^2$. This scaling is also compatible with the wave equation for \mathbf{A} , i.e.

$$\varepsilon \{ \partial_t^2 \mathbf{A} - \partial_z^2 \mathbf{A} - \alpha^2 \Delta_{\perp} \mathbf{A} \} = -\varepsilon \left\{ \frac{n_0}{n_c} (n_e^0 + \beta n_e^1) \left[1 - \frac{\varepsilon^2}{2} |\mathbf{A}|^2 \right] \mathbf{A} \right\} + \dots .$$

Then, because of

$$\nabla \frac{n_e}{\gamma} = \varepsilon^2 (\nabla n_e^1 + n_e^0 \nabla \frac{|\mathbf{A}|^2}{2})$$

and

$$\mathbf{A} \cdot \nabla = \mathbf{A}_{\perp} \cdot (\alpha \nabla_{\perp}) + \alpha A_{\parallel} \partial_z$$

we have

$$\nabla \left(\mathbf{A} \cdot \nabla \frac{n_e}{\gamma} \right) = o(\varepsilon^3 \alpha) ,$$

and

$$\nabla \times \left[\left(\nabla \frac{n_e}{\gamma} \right) \times (\nabla \psi) \right] = o(\varepsilon^4 \alpha) .$$

The inverse Laplace-operator does not change the order of the dominating terms since

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

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

To get consistent equations, we include all terms up to order ε^3 and $\varepsilon\alpha^2$, while neglecting terms of order $\varepsilon^3\alpha$ and higher.

Linearizing and differentiating in time the continuity equation (A.2), i.e.

$$\frac{\partial^2}{\partial t^2} n_e^1 = -\frac{n_0}{n_c} n_e^0 \frac{\partial}{\partial t} \Delta \psi^1,$$

and eliminating on the right-hand-side $\Delta \psi^1$ by taking the Laplacian of the equation for ψ^1 , we arrive at

$$\frac{\partial^2}{\partial t^2} n_e^1 + \frac{n_0}{n_c} n_e^0 n_e^1 = \frac{n_e^0}{2} \Delta |\mathbf{A}|^2. \quad (\text{A.4})$$

Together with the lowest order wave equation

$$\frac{\partial^2}{\partial t^2} \mathbf{A} - \Delta \mathbf{A} = -\frac{n_0}{n_c} \left\{ n_e^0 \left(1 - \frac{1}{2} |\mathbf{A}|^2 \right) + n_e^1 \right\} \mathbf{A} \quad (\text{A.5})$$

we obtain a closed system of two equations. Instead of a series expansion we may retain the full γ -factor, keeping in mind at the end the valid accuracy of the results. For the initial condition $A_{\parallel} = 0$, the parallel component will stay zero during time evolution. It is then possible to use the scalar complex-valued field a to describe the vector potential via

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

Making use of the latter, we obtain (1) being coupled to (A.4), where the latter agrees with (2).

References

- [1] G. Mourou, Optics in the relativistic regime, *Rev. Mod. Phys.* 78 (2006) 303–371.
- [2] P. Sprangle, E. Esarey, A. Ting, Nonlinear interaction of intense laser pulses in plasmas, *Phys. Rev. A* 41 (1990) 4463.
- [3] A. Pukhov, Strong field interaction of laser radiation, *Rep. Prog. Phys.* 66 (2003) 47–101.
- [4] R. Bingham, U. de Angelis, M. Amin, R. Cairns, B. McNamara, Relativistic Langmuir waves generated by ultra-short pulse lasers, *Plasma Phys. Control. Fusion* 34 (1992) 557–567.
- [5] L. Gorbunov, V. Kirsanov, Excitation of plasma waves by an electromagnetic wave packet, *Zh. Eksp. Teor. Fiz.* 93 (1987) 509.
- [6] A. Pukhov, J. M. ter Vehn, Laser wake field acceleration: The highly non-linear broken-wave regime, *Appl. Phys. B* 74 (2002) 355.

- [7] J. Faure, Y. Glinec, A. Pukhov, S. Kiselev, S. Gordienko, E. Lefebvre, J.-P. Rousseau, F. Burgy, V. Malka, A laser-plasma accelerator producing monoenergetic electron beams, *Nature* 431 (2004) 541.
- [8] J. Faure, C. Rechatin, A. Norlin, A. Lifschitz, Y. Glinec, V. Malka, Controlled injection and acceleration of electrons in plasma wakefields by colliding laser pulses, *Nature* 444 (2006) 737.
- [9] V. Malka, J. Faure, Y. Glinec, A. Lifschitz, Laser-plasma accelerators: A new tool for science and for society, *Plasma Phys. Control. Fusion* 47 (2005) 481.
- [10] O. Shorokov, A. Pukhov, I. Kostyukov, Self-compression of laser pulses in plasma, *Phys. Rev. Lett.* 91 (2003) 265002–1–4.
- [11] C. Karle, J. Schweitzer, M. Hochbruck, E. Laedke, K. Spatschek, Numerical solution of nonlinear wave equations in stratified dispersive media, *J. Comp. Phys.* 216 (2006) 138–152.
- [12] C. Ren, B. Duda, R. Hemker, W. Mori, T. Katsouleas, T. Antonsen, P. Mora, Compressing and focusing a short laser pulse by a thin plasma lens, *Phys. Rev. E* 63 (2001) 026411.
- [13] E. Hairer, C. Lubich, G. Wanner, *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations*, Springer, Berlin, 2002.
- [14] M. Hochbruck, C. Lubich, A Gautschi-type method for oscillatory second-order differential equations., *Numer. Math.* 83 (1999) 403–426.
- [15] E. Hairer, C. Lubich, Long-time energy conservation of numerical methods for oscillatory differential equations, *SIAM Journal on Numerical Analysis* 38 (2) (2000) 414–441.
- [16] V. Grimm, M. Hochbruck, Error analysis of exponential integrators for oscillatory second-order differential equations, *J. Phys. A: Math. Gen.* 39 (2006) 5495–5507.
- [17] E. Hairer, C. Lubich, G. Wanner, Geometric numerical integration illustrated by the Störmer/Verlet method, *Acta Numerica* (2003) 1–51.
- [18] V. Druskin, L. Knizherman, Krylov subspace approximation of eigenpairs and matrix functions in exact and computer arithmetic, *Numer. Linear Algebra Appl.* 2 (3) (1995) 205–217.
- [19] M. Hochbruck, C. Lubich, On Krylov subspace approximations to the matrix exponential operator, *SIAM Journal on Numerical Analysis* 34 (5) (1997) 1911–1925.
- [20] Ch. Karle, Relativistic laser pulse compression and focusing in stratified plasma-vacuum systems, PhD Thesis, Institut für Theoretische Physik, Universität Düsseldorf, Germany, 2008