

Hyperbolic relaxation method for elliptic equations

Hannes R. Rüter,¹ David Hilditch,^{1,2} Marcus Bugner,¹ and Bernd Brügmann¹

¹*Theoretical Physics Institute, University of Jena, 07743 Jena, Germany*

²*CENTRA, University of Lisbon, 1049 Lisboa, Portugal*



(Received 15 November 2017; published 26 October 2018)

We show how the basic idea of parabolic Jacobi relaxation can be modified to obtain a new class of hyperbolic relaxation schemes that are suitable for the solution of elliptic equations. Some of the analytic and numerical properties of hyperbolic relaxation are examined. We describe its implementation as a first order system in a pseudospectral evolution code, demonstrating that certain elliptic equations can be solved within a framework for hyperbolic evolution systems. Applications include various initial data problems in numerical general relativity. In particular we generate initial data for the evolution of a massless scalar field, a single neutron star, and binary neutron star systems.

DOI: [10.1103/PhysRevD.98.084044](https://doi.org/10.1103/PhysRevD.98.084044)

I. INTRODUCTION

The solution of elliptic partial differential equations (elliptic PDEs) is an important problem in many areas of physics. Correspondingly large is the variety of analytic and numerical methods dealing with the solution of elliptic PDEs. The starting point for many methods is a discretization and (if required) a linearization, which for typical problems arising in physics leads to a sparse system of linear equations for a large but finite number of degrees of freedom. A key role in the solution of linear systems is played by iterative methods, e.g., [1,2]. Among the basic iterative methods are relaxation methods, in particular Gauss-Seidel and Jacobi relaxation methods, and the family of Krylov subspace methods. Closely connected are strategies to accelerate these methods, such as preconditioners and multigrid methods.

Motivated by the need to solve elliptic systems as initial data within our time evolution code, we study a modification of the classic Jacobi method, which is closely linked to physical relaxation problems. For concreteness, consider as a minimal example the Laplace equation,

$$\Delta\phi = 0, \quad (1)$$

for a function $\phi(x, y, z)$ on a regular subset of \mathbb{R}^3 together with appropriate boundary conditions. The Jacobi relaxation scheme can be obtained by introducing a time parameter t and considering instead of (1) the parabolic diffusion equation [3]

$$\partial_t\phi = \Delta\phi. \quad (2)$$

As time approaches infinity, any initial data for ϕ “relaxes” to a stationary state, where $\partial_t\phi = 0$ and hence Eq. (1) is

satisfied as well. The Jacobi iteration method is obtained by discretizing the diffusion equation (2). In essence, we introduce an “unphysical” time dependence, which is not part of the original problem, and obtain the solution to the time-independent problem by means of a fixed point iteration.

Here we investigate a similar strategy, which however relies on a different type of evolution equation. Instead of embedding the elliptic equation (1) in a parabolic equation (2), we consider a *hyperbolic* wave equation with damping,

$$\partial_t^2\phi + \partial_t\phi = \Delta\phi. \quad (3)$$

The relationship between hyperbolic equations and parabolic diffusion equations has already been investigated in some special cases [4–6]. In particular it can be shown that for large times t the solution of the hyperbolic equation (3) will tend toward the solution of the parabolic PDE, (2). The hyperbolic equation can be cast in first order form by introducing the reduction variables $\psi = \partial_t\phi + \phi$ and $r_i = \partial_i\phi$, yielding the system

$$\partial_t\psi = \psi - \phi, \quad (4)$$

$$\partial_t\psi = \delta^{ij}\partial_i r_j, \quad (5)$$

$$\partial_t r_i = \partial_i\psi - r_i. \quad (6)$$

The first of these equations is an ordinary differential equation, and it is directly evident that ϕ will tend toward ψ exponentially. Thus we can eliminate (4) and obtain the solution directly from ψ .

Combining time derivatives as in (3) adds strong damping to the pure wave equation while maintaining the

hyperbolic character of the PDE. The idea is that deviations from the stationary state satisfying $\Delta\phi = 0$ are damped to zero or are propagated away, and furthermore it can be advantageous to perform hyperbolic as opposed to parabolic evolutions.

In the limit of vanishing damping we obtain the plain wave equation. If a stationary state is reached, we again have solved (1). This undamped approach has, e.g., been used to solve the Poisson equation in the context of self-gravity [7]. Experimenting with (3) however, we found that the damping is the main desirable feature, while propagating waves off the grid is far less relevant for the reduction of the residual.

In [8] the authors analyzed a specific hyperbolized version of the Navier-Stokes equations, that exhibits hyperbolic relaxation. Although the type of equations that are considered are similar, our perspective is different. The elliptic equation is for us the fundamental problem, and we add a hyperbolic, damped time-dependence to obtain an iterative scheme for the solution of the elliptic equation. Since there does not seem to be an established name for this idea, we refer to the method as *hyperbolic relaxation for elliptic equations* (HypRelax), as opposed to parabolic relaxation that is at the heart of the Jacobi method.

With regard to previous literature on hyperbolic relaxation for elliptic equations, some aspects have been explored in [9] in the context of “gauge drivers” for numerical relativity. In particular, [9] introduced one of the most used gauge conditions for certain black hole evolutions, the Gamma-driver for the shift vector, which employs a hyperbolic equation related to the elliptic equation for a minimal distortion shift. They likewise discussed a similar approach to the lapse in which the associated elliptic equation corresponds to maximal slicing. Also see [10] on gauge drivers, where however only parabolic relaxation is considered.

The goal of the present paper is to develop hyperbolic relaxation given by the prototype in (3) into a method to solve a general class of systems of second order, nonlinear elliptic equations. The problem of immediate interest to us is defined by the constraint equations of general relativity, which we solve as a system of nonlinear elliptic equations to obtain initial data for evolution in numerical relativity. However, the formalism is quite independent of this particular problem.

The main result of this paper is that hyperbolic relaxation can be formulated for nontrivial equations, and numerical experiments for some specific test problems were indeed successful. Specific examples include the Poisson equation, the conformal-thin-sandwich equations, scalar field initial data, and some simple configurations of single and binary neutron stars.

Considering (3), let us collect some basic observations here in order to introduce the main questions we want to address. First of all, we have to address the well-posedness

of the hyperbolic PDEs. Given a self-adjoint, elliptic operator, the hyperbolicity of equations of type (3) should be clear. We demonstrate this below for a general class of equations. There exists a rich theoretical background regarding well-posedness and numerical stability for hyperbolic PDEs [11–13], which helps to find relaxation schemes that are well suited for numerical applications.

Second, in addition to the boundary conditions of the original elliptic equation we have to choose boundary conditions for the hyperbolic equations that are compatible with the asymptotic elliptic problem. This choice is not unique, but of great importance to obtain successful evolutions. In particular, we consider maximally dissipative boundary conditions.

Third, assuming feasibility and stability of hyperbolic relaxation, a key question concerns the efficiency of the method. In both parabolic and hyperbolic relaxation methods the time parameter is unrelated to the elliptic equation, i.e., the time evolution is of no interest as long as the stationary state is reached efficiently. This is the basis for different acceleration strategies. For hyperbolic relaxation, there is a finite propagation speed, and in contrast to the diffusion equation it is not clear how to bypass that speed to accelerate the method.

Beyond the intrinsic interest in a new method, we have to ask whether hyperbolic relaxation, after some significant further development that is beyond the scope of this paper, might become an interesting alternative to the highly developed standard methods. For example, Jacobi relaxation is a fundamental building block of many advanced methods, but it is essentially never used on its own because of its slow convergence (for long wavelengths). However, multigrid methods with parabolic relaxation as smoother (for short wave lengths) are highly efficient, reducing the computational complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log(n))$. It remains to be seen how efficient hyperbolic relaxation can be, with and without acceleration methods.

As it stands, there are pragmatic considerations that can make hyperbolic relaxation methods interesting, in particular when solving elliptic equations as part of a larger project. For example, elliptic PDEs are often solved to provide initial data for evolution systems that are subject to certain constraint equations, e.g., the Maxwell equations or the Einstein equations. However, the main work load is the actual evolution of the data by integrating a hyperbolic PDE. In such a case the hyperbolic relaxation method does not have to compete with optimized standard methods in terms of efficiency as long as solving the elliptic equation is only a small part of the entire work load. On the other hand, a hyperbolic relaxation method may be easy to implement using the existing infrastructure of a numerical evolution code, avoiding the need for and the complications of an external elliptic solver. Using the same infrastructure also has the advantage that interpolation errors can be avoided by using the same grid discretization. Considering our

research in numerical relativity, a sophisticated infrastructure for evolutions is indeed available, but we were looking for alternative elliptic solvers. Hence we implemented hyperbolic relaxation in the pseudospectral hyperbolic evolution code BAMPs [14,15], which only required minor modifications once the formalism itself was established.

The paper is structured as follows. In Sec. II we derive and motivate the evolution equations on which our relaxation method is based and discuss some of its properties. In Sec. III we give details on the numerical implementation and state the methods used in the BAMPs code. In Secs. IV and V we apply our hyperbolic relaxation method to some test cases and to the construction of initial data for numerical relativity. We conclude in Sec. VI.

Throughout the paper we use the Einstein summation convention, i.e., we sum over indices that occur once as an upper index and once as a lower index, e.g., $s^i t_i = \sum_i s^i t_i$. Latin letters i, j, k, \dots denote coordinate components and they are lowered and raised by an arbitrary metric with positive signature. An index s denotes a contraction with a vector s_i , in particular $\partial_s = s^i \partial_i = s^i \frac{\partial}{\partial x^i}$. Greek letter indices α, β, γ denote components of a field and they are lowered and raised by the Euclidean metric $\delta_{\alpha\beta}$.

II. THE HYPERBOLIC RELAXATION EQUATIONS

A. Evolution system

In the following we present the principal ideas of the hyperbolic relaxation method and derive the equations that follow for the iteration scheme. Although it is possible to write down hyperbolic relaxation schemes in second order form like Eq. (3), we will focus on a first order formulation. This is primarily done because our evolution code BAMPs only handles first derivatives, but we will see that the reduction does allow considerable freedom in the problem setup.

In this paper we consider only systems of elliptic equations given in second order form, i.e.,

$$(L\psi)_\alpha = a(x^k)^{ij\beta}{}_\alpha \partial_i \partial_j \psi_\beta + F_\alpha(x^k, \psi_\beta, \partial_i \psi_\beta) = 0, \quad (7)$$

where the ψ_α are the N unknown solution variables and F is a continuous function of the solution variables, their derivatives and the D coordinates x^k . We take $a^{ij\beta}{}_\alpha$ to be a smooth function of the coordinates and suppress this dependence in the notation. Every elliptic system will be accompanied by a set of boundary conditions on the variables ψ_α and we discuss their treatment in Sec. II E. In the following we consider *classically elliptic* systems [16] only, i.e., systems with

$$\det(a^{ij\beta}{}_\alpha s_i s_j) \neq 0 \quad \forall s \in \mathbb{R}^D \setminus \{0\}, \quad (8)$$

where the determinant is understood to be taken on the indices α and β .

To solve the second order equation (7) one could employ the Jacobi method, which can be motivated by evolving the parabolic partial differential equation:

$$\partial_t \psi_\alpha = (L\psi)_\alpha, \quad (9)$$

where t is some parameter that plays the role of time.

For a classically elliptic system with constant coefficients the Jacobi method can only converge if $a^{ij\beta}{}_\alpha$ is positive definite on the whole domain, i.e., there exists an $\epsilon > 0$

$$a^{ij\beta}{}_\alpha t_{j\beta} t_i^\alpha \geq \epsilon t^\alpha t_{i\alpha} \quad \forall t \in \mathbb{R}^{D \times N} \setminus \{0\}, \quad (10)$$

which we assume in the rest of this paper. This condition corresponds to the notion of *strong ellipticity*, which defines an important subclass of classically elliptic systems [16]. Note that we have the freedom to multiply the elliptic equation with an invertible matrix $d^\beta{}_\alpha$, yielding $(\tilde{L}\psi)_\alpha = d^\beta{}_\alpha (L\psi)_\beta = 0$, which has the same solutions as the original equation. This freedom allows the transformation of some systems that are classically but not strongly elliptic into strongly elliptic form.

To construct the hyperbolic relaxation equations we can reduce the second order elliptic system to first order by introducing the *reduction variables* $r_{i\alpha}$:

$$0 = a^{ij\beta}{}_\alpha \partial_i r_{j\beta} + F_\alpha(\psi_\beta, r_{i\beta}), \quad (11)$$

$$0 = \partial_i \psi_\alpha - r_{i\alpha}. \quad (12)$$

In analogy to the Jacobi method (9) we evolve ψ_α by taking Eq. (11) as the right-hand side, yielding

$$\partial_t \psi_\alpha = a^{ij\beta}{}_\alpha \partial_i r_{j\beta} + F_\alpha(x^i, \psi_\beta, r_{i\beta}) \quad (13)$$

and we proceed similarly with the equations for the reduction variables r_i :

$$\partial_t r_{i\alpha} = b^j{}_{i\alpha} (\partial_j \psi_\beta - r_{j\beta}), \quad (14)$$

where $b^j{}_{i\alpha}$ is arbitrary under the requirement of positive definiteness, meaning in analogy to Eq. (10)

$$b^j{}_{i\alpha} t_{j\beta} t^{i\alpha} > \epsilon t_{i\alpha} t^{i\alpha} \quad \forall t \in \mathbb{R}^{D \times N} \setminus \{0\}. \quad (15)$$

The system of Eqs. (13) and (14) forms a first order hyperbolic differential equation which we refer to as the *hyperbolic relaxation system*. Clearly the reduction constraint Eq. (12) is not enforced at all times and will indeed be violated during the relaxation process, however we are only interested in the steady state, which fulfills the reduction constraint, because Eq. (14) drives the reduction variable r_i towards $\partial_i \psi$. To see this, let us assume for

arguments sake that $\partial_t \partial_i \psi_\alpha = 0$, which is reasonable close to the steady state. The solution for $r_{i\alpha}$ then has the form

$$r_{i\alpha}(t) = \sum_{l=1}^n e^{-\lambda_l t} \sum_{k_l=0}^{m_l} x^{k_l} h_{k_l i \alpha} + \partial_i \psi_\alpha, \quad (16)$$

where h is constant and the λ_l are the n eigenvalues defined by the eigenvalue equation $b^j_{i\beta} t^{i\alpha} = \lambda_l t_l^{j\beta}$ and m_l depends on the geometric multiplicity of λ_l . From the positive definiteness of b we know that all the eigenvalues have positive real part and it follows immediately that $r_{i\alpha}$ approaches $\partial_i \psi_\alpha$ exponentially. Indeed in Sec. II C it will be shown for a simple case that the modes of the system are always damped even for $\partial_t \partial_i \psi_\alpha \neq 0$. We emphasize however that in some cases, e.g., if the elliptic system has no solution, $\partial_i \psi_\alpha$ can grow faster than $r_{i\alpha}$ and thus the reduction constraints cannot be satisfied asymptotically in time.

It is obvious that if the hyperbolic relaxation system reaches a steady state we must have obtained a solution to the first order elliptic system Eqs. (11) and (12), and hence also of the original elliptic equation (7).

It is natural to ask here why we restrict to second order elliptic systems rather than including also first order systems. The reason is that only for this class of operators have we identified a hyperbolic relaxation scheme which we expect to efficiently settle down to a steady state. The next natural query is what, if anything, is gained by insisting on taking the relaxation scheme in first order form, beyond the practical requirement that we can implement the scheme within our code. As we have seen, the reduction allows a variety of choices for $b^j_{i\beta}$, which is not directly evident in the second order form. In fact it is not clear whether a similar second order form relaxation equation like Eq. (3) can be found for generic $b^j_{i\beta}$. As will be seen in Sec. III D the first order formulation also provides a method to define a refinement criterion.

B. Residual evolution

The residuals of the first order system, Eqs. (11) and (12), are given by

$$R_\alpha = a^{ij\beta} \partial_i r_{j\beta} + F_\alpha(x^i, \psi_\beta, r_{i\beta}), \quad (17)$$

$$R_{i\alpha} = b^j_{i\beta} \alpha (\partial_j \psi_\beta - r_{j\beta}). \quad (18)$$

A simple calculation shows that the residuals will evolve according to

$$\partial_t R_\alpha = a^{ij\beta} \partial_i R_{j\beta} + \frac{\partial F_\alpha}{\partial \psi_\beta} R_\beta + \frac{\partial F_\alpha}{\partial r_{i\beta}} R_{i\beta}, \quad (19)$$

$$\partial_t R_{i\alpha} = b^j_{i\beta} \alpha (\partial_j R_\beta - R_{j\beta}). \quad (20)$$

For a working relaxation scheme, we want the residual evolution system to be stable, i.e., the first order residuals should converge to zero for $t \rightarrow \infty$, for residuals that are sufficiently close to zero. Systems of this type and stability conditions are discussed in detail in [17,18]. It is not possible for us to give general results on the stability of the hyperbolic relaxation scheme, as the multitude of possible systems is too large to be covered in a closed form, especially for elliptic systems with more than one variable. A stability analysis must therefore be done individually for the concrete problem.

C. Mode analysis

To shed some light on the behavior of solutions to the hyperbolic relaxation, we perform a simple mode analysis ignoring the issue of boundary conditions of a generalization of Eq. (3)

$$\epsilon \partial_t^2 \phi + \eta \partial_t \phi = \Delta \phi, \quad (21)$$

where ϵ and η are real, non-negative constants. A third constant in front of $\Delta \phi$ has been rescaled to one without loss of generality. With $\epsilon = \eta = 1$ as in (3) we fix the unit of time to be dimensionless (unity since $[T^2] = [T]$).

We introduce the plane-wave ansatz

$$\phi_{pw}(t, x) = e^{i(kx - \omega t)}, \quad (22)$$

where k and ω are constants. The wave number k is a real number related to the wave length, $k = 2\pi/\lambda$, while ω may be a complex number. Inserted into the hyperbolic relaxation equation (3), we obtain

$$\epsilon \omega^2 + i\eta \omega = k^2, \quad (23)$$

$$\omega_\pm(k) = -\frac{1}{2\epsilon} (\eta i \pm \sqrt{4\epsilon k^2 - \eta^2}). \quad (24)$$

Recall that for the wave equation $\omega_\pm(k) = \pm k$, while for the heat equation $\omega(k) = -ik^2$. For hyperbolic relaxation, there is a further case distinction for the sign under the square root $\sqrt{4\epsilon k^2 - \eta^2}$. The existence of a transition at a specific length scale $\lambda_{\text{crit}} = \frac{2\pi}{k_{\text{crit}}} = \frac{4\pi\sqrt{\epsilon}}{\eta}$ signals that our choice of hyperbolic relaxation equation has fixed a scale, that can be adjusted by changing the parameters η and ϵ .

For sufficiently large wave number,

$$\phi_{pw} = e^{-\frac{\eta}{2\epsilon} t} e^{i(kx \pm \frac{1}{2\epsilon} \sqrt{4\epsilon k^2 - \eta^2} t)}, \quad k \geq \frac{\eta}{2\sqrt{\epsilon}}, \quad (25)$$

which is a damped wave with phase velocity $v(k) = \frac{1}{\epsilon} \sqrt{\epsilon - \frac{\eta^2}{4k^2}}$. The damping is independent of k (as opposed to the heat equation with $e^{-k^2 t}$). The phase velocity approaches $v = 1/\sqrt{\epsilon}$ for large k , but for k approaching

the critical value $\frac{\eta}{2\sqrt{\epsilon}}$ from above the phase velocity tends towards $v = 0$. This has the consequence that the maximal possible time step scales like $\Delta t \sim \sqrt{\epsilon}\Delta x$ for large k (high resolution), but has a finite upper bound for small k . Increasing the parameter ϵ might appear to be a good idea at first, since it allows one to use larger time steps, but at the same time it reduces the damping by the same factor.

For sufficiently small wave number,

$$\phi_{pw} = e^{-\frac{1}{2\epsilon}(\eta \pm \sqrt{\eta^2 - 4\epsilon k^2})t} e^{ikx}, \quad 0 \leq k \leq \frac{\eta}{2\sqrt{\epsilon}}, \quad (26)$$

which is a nonmoving wave profile e^{ikx} times a k -dependent damping factor. For $k = \frac{\eta}{2\sqrt{\epsilon}}$, the damping is $e^{-\frac{\eta}{2\epsilon}t}$, while for k equal to zero there are two cases, e^0 or $e^{-\frac{\eta}{\epsilon}t}$. For small k , the worse (more weakly) damped case is $\exp(-\frac{1}{2\epsilon}(\eta - \sqrt{\eta^2 - 4\epsilon k^2})t) \approx \exp(-\frac{1}{\eta}k^2 t)$, which is the same damping as for the basic heat equation, when we choose $\eta = 1$.

Summarizing, the plane-wave mode analysis suggests that solutions to the hyperbolic relaxation equation exhibit a mixture of relaxation and wave propagation phenomena, see Fig. 1. For wave numbers larger than a critical value, $k \geq k_{\text{crit}}$ with $k_{\text{crit}} = \frac{\eta}{2\sqrt{\epsilon}}$, there is wave propagation with simultaneous damping. Noteworthy is that the damping is independent of k , $e^{-\frac{\eta}{2\epsilon}t}$. This is a promising feature compared to parabolic relaxation with $e^{-k^2 t}$ for intermediate values of k . For large values of k parabolic relaxation has much stronger damping, but the overall convergence rate is dominated by small k . For hyperbolic relaxation, there is no wave propagation for $k \leq k_{\text{crit}}$, but the damping persists. Interestingly, the damping factor asymptotes toward $e^{-\frac{1}{\eta}k^2 t}$ for $k \rightarrow 0$, and is never worse than parabolic relaxation for small k , when we choose $\eta = 1$.

If we mimic the first order relaxation equations (13) and (14) by

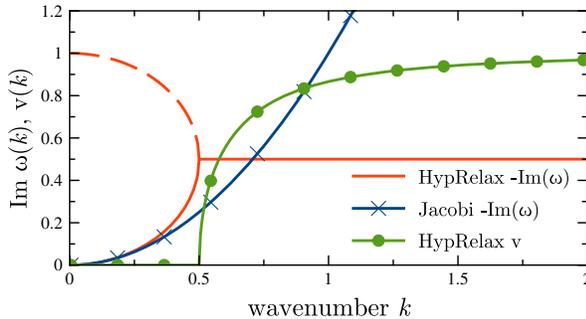


FIG. 1. Damping and propagation speed of basic hyperbolic relaxation with parameters $\epsilon = 1$ and $\eta = 1$ compared to parabolic relaxation. There is a transition at $k_{\text{crit}} = \frac{1}{2}$, which can be moved to lower values by adjusting η . The overall damping is determined by the slowest damping rate.

$$\partial_t \psi = \frac{1}{\eta} \partial_j r^j, \quad \partial_t r_i = \frac{\eta}{\epsilon} (\partial_i \psi - r_i), \quad (27)$$

we recover the modes in (23) and $D - 1$ transverse modes that have $\psi = 0$ and for which $\omega(k) = -i\frac{\eta}{\epsilon}$ is purely imaginary.

For the choice of η and ϵ we can take our motivation from the gauge driver construction [9] and set $\epsilon = 1$, η an arbitrary non-negative constant, and obtain

$$\omega_{\pm}(k) = -\frac{1}{2}(i\eta \pm \sqrt{4k^2 - \eta^2}). \quad (28)$$

We also considered $\epsilon = \eta^2$ and arbitrary η for uniform scaling of time.

To avoid small k that drop below (or too far below) k_{crit} , we can adjust ϵ or η such that the length scale of k corresponds to the physical size L of the domain, say $\lambda_{\text{crit}} = 2L$. This will decrease the damping for large k , but will also avoid the severe slow down when the damping approaches that of parabolic relaxation.

D. Hyperbolicity analysis

In this section we introduce a notation for inverse tensors. These have to be understood as the inverse of matrices with respect to the field indices. For example the inverse of the tensor c^β_α is $(c^{-1})^\alpha_\gamma$ and we have

$$c^\beta_\alpha (c^{-1})^\alpha_\gamma = \delta^\beta_\gamma. \quad (29)$$

For the hyperbolicity analysis we start with writing the hyperbolic relaxation system in matrix form

$$\partial_t \mathbf{u} = \mathbf{P}^k \partial_k \mathbf{u} + h(x^i, \mathbf{u}) \quad (30)$$

with

$$\mathbf{P}^k = \begin{pmatrix} 0 & a^{ki\beta}_\alpha \\ b^k_{i\beta}_\alpha & 0 \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} \psi_\alpha \\ r_{i\alpha} \end{pmatrix}. \quad (31)$$

The principal symbol of this system is then given by

$$\mathbf{P}^s = \mathbf{P}^k s_k = \begin{pmatrix} 0 & a^{si\beta}_\alpha \\ b^s_{i\beta}_\alpha & 0 \end{pmatrix}, \quad (32)$$

where s_i is an arbitrary unit vector, $s^i s_i = 1$. Suppose $c^\beta_\alpha = a^{sie}_\alpha b^s_{i\beta}_\epsilon$ has a complete set of eigenvectors w_γ^α with $c^\beta_\alpha w_\gamma^\alpha = \sigma_\gamma^\alpha w_\gamma^\beta$, where σ_γ^α is diagonal. If furthermore all the eigenvalues, i.e., the diagonal elements of σ_γ^α , are positive then \mathbf{P}^s has the following left eigenvectors

$$\mathbf{v}_{j\gamma}^0 = (0, \delta_\gamma^\alpha \delta_j^i - b^s_{j\epsilon} (b^s_{s^{-1}})^\alpha_\epsilon s^i), \quad (33)$$

$$\mathbf{v}_\gamma^\pm = (w_\gamma^\alpha, \pm (\rho^{-1})^\delta_\gamma w_\delta^\epsilon a^{s i \alpha}_\epsilon), \quad (34)$$

where ρ_γ^β is the root of σ_γ^β , i.e., it is a positive diagonal tensor with $\rho_\gamma^\alpha \rho_\alpha^\beta = \sigma_\gamma^\beta$. Note that of the eigenvectors $\mathbf{v}_{j\gamma}^0$ only $(D-1)N$ are linearly independent, while the \mathbf{v}_γ^\pm are $2N$ independent vectors. If there exists a constant K , independent of s^i , such that $\|\mathbf{V}\|_2 + \|\mathbf{V}^{-1}\|_2 < K$, where \mathbf{V} is an, in general s^i -dependent, square matrix constructed from a linearly independent set of the eigenvectors \mathbf{v} , then the system is strongly hyperbolic [11–13]. The characteristic variables \hat{u} and their characteristic speeds λ are thus

$$\hat{u}_{j\gamma}^0 = r_{j\gamma} - b^s j_\gamma^\epsilon (b^s s^{-1})^\alpha \epsilon^s r_{i\alpha}, \quad \lambda_{i\gamma}^0 = 0, \quad (35)$$

$$\hat{u}_\gamma^\pm = w_\gamma^\alpha \psi_\alpha \pm (\rho^{-1})_\gamma^\delta w_\delta^\epsilon a^{s i \alpha} \epsilon^s r_{i\alpha}, \quad \lambda_\gamma^\pm = \pm \rho_\gamma^\alpha e_\alpha, \quad (36)$$

where e_α denote the Cartesian basis vectors. From this we can recover the evolved variables in terms of the characteristics:

$$\psi_\alpha = \frac{1}{2} (w^{-1})_\alpha^\gamma (\hat{u}_\gamma^+ + \hat{u}_\gamma^-), \quad (37)$$

$$r_{i\alpha} = \hat{u}_{i\alpha}^0 + b^s i_\alpha^\epsilon (c^{-1})^\beta \epsilon \cdot \left((w^{-1})_\beta^\delta \rho_\delta^\gamma \frac{\hat{u}_\gamma^+ - \hat{u}_\gamma^-}{2} - a^{s j \gamma} \beta \hat{u}_{j\gamma}^0 \right). \quad (38)$$

We can use the freedom in the choice of $b^j i_\alpha^\beta$ to impose certain properties on the hyperbolic relaxation system. In the following we briefly discuss three interesting choices, that fulfill the restrictions we have set for b .

- (1) *b is the identity.* A very easy and natural choice is $b^j i_\alpha^\beta = \delta_i^j \delta_\alpha^\beta$. With this choice we have $c^\beta_\alpha = a^{s\beta\alpha}$, which has only eigenvalues with positive real part due to (10). The imaginary part however can be nonvanishing. If however $a^{s\alpha\gamma} = a^{s\gamma\alpha}$, then c is guaranteed to have a complete set of eigenvectors with purely real eigenvalues and thus system is strongly hyperbolic. If we have $a^{ij\alpha\gamma} = a^{ij\gamma\alpha}$, then the system is even symmetric hyperbolic with symmetrizer:

$$\mathbf{H} = \begin{pmatrix} \delta_\gamma^\alpha & 0 \\ 0 & a^{ij\alpha\gamma} \delta_{j\gamma} \end{pmatrix}. \quad (39)$$

- (2) *b is the transpose of a.* We can also make the system trivially symmetric hyperbolic by choosing $b^j i_\alpha^\beta = a^j i_\alpha^\beta$. The principal symbol of this system is symmetric and thus the system is symmetric hyperbolic.
- (3) *b is the inverse of a.* We can choose b to be the inverse of a in the sense that b fulfills $a_k^{i\alpha} b^j i_\alpha^\beta = \delta_k^j \delta_\alpha^\beta$. This choice is particularly interesting because we then have $c^\beta_\alpha = \delta_\alpha^\beta$ and thus all the nonzero characteristic speeds have values ± 1 . Furthermore the eigenvectors

of c become trivial: $w_\gamma^\alpha = \delta_\gamma^\alpha$. A symmetrizer for this system is

$$\mathbf{H} = \begin{pmatrix} \delta_\gamma^\alpha & 0 \\ 0 & a_m^{i\alpha} a_\omega^m l_\gamma^\omega \end{pmatrix}. \quad (40)$$

Since all traveling characteristic variables have the same speeds, we consider this the best choice for $b^j i_\alpha^\beta$. A straightforward generalization of this choice allows $b^j i_\alpha^\beta$ to be scaled by a constant factor, which will also uniformly scale the characteristic speeds.

E. Boundary conditions

The basic idea to impose boundary conditions in our method is to modify the right-hand side of the hyperbolic relaxation system. The outward pointing unit normal covector s_i to the boundary surface is naturally defined by taking the gradient of a scalar field which is increasing across, but constant in the boundary, and then normalizing this gradient to unit magnitude using our arbitrary but fixed metric. This metric is subsequently used to raise the index and form the outward pointing vector s^i . We restrict our attention to strongly hyperbolic systems, for which a regular (s^i -dependent) similarity transformation matrix \mathbf{T}_s exists which transforms between our evolved variables \mathbf{u} and a linearly independent set of the characteristic variables $\hat{\mathbf{u}}$ given in Eqs. (35) and (36)

$$\mathbf{u} = \mathbf{T}_s \hat{\mathbf{u}}. \quad (41)$$

We can then decompose our evolution equations (30) as

$$\partial_t \mathbf{u} = \mathbf{P}^s \partial_s \mathbf{u} + \mathbf{P}^k q_k^i \partial_i \mathbf{u} + h(x^i, \mathbf{u}), \quad (42)$$

where $q_k^i = \delta_k^i - s_k s^i$ is the projector onto the boundary surface. We now multiply by \mathbf{T}_s^{-1} and obtain

$$\begin{aligned} d_t \hat{\mathbf{u}} &= \mathbf{T}_s^{-1} \mathbf{P}^s \partial_s \mathbf{u} + \mathbf{T}_s^{-1} (\mathbf{P}^k q_k^i \partial_i \mathbf{u} + h(x^i, \mathbf{u})) \\ &= \mathbf{T}_s^{-1} \mathbf{P}^s \mathbf{T}_s \mathbf{T}_s^{-1} \partial_s \mathbf{u} + \mathbf{T}_s^{-1} (\mathbf{P}^k q_k^i \partial_i \mathbf{u} + h(x^i, \mathbf{u})) \\ &= \Lambda^s d_s \hat{\mathbf{u}} + \mathbf{T}_s^{-1} (\mathbf{P}^k q_k^i \partial_i \mathbf{u} + h(x^i, \mathbf{u})). \end{aligned} \quad (43)$$

Here the straight derivative symbol d denotes that the transformation matrix stands outside of the derivative, i.e., $d_t \hat{\mathbf{u}} \equiv \mathbf{T}_s^{-1} \partial_t \mathbf{u}$, and Λ^s is a diagonal matrix containing the characteristic speeds. We can now impose boundary conditions on the incoming variables, i.e., those with positive characteristic speeds, by modifying their right hand sides. After the right-hand sides have been modified we transform the system back by multiplying with \mathbf{T}_s .

1. Penalty method

In the penalty method [19–21] the boundary conditions are *weakly imposed* by modifying the right-hand sides of

the incoming characteristic variables \hat{u}_γ^+ in the following way

$$d_t \hat{u}_\gamma^+ \hat{=} D_t \hat{u}_\gamma^+ + p(\hat{u}_\gamma^{+BC} - \hat{u}_\gamma^+), \quad (44)$$

where p is the penalty parameter, u_γ^{+BC} is some given boundary data that we want \hat{u}_γ^+ to approach, $D_t \hat{u}_\gamma^+$ is the unmodified right-hand side and $\hat{=}$ denotes equality at the boundary. The penalty parameter can not be chosen arbitrarily and we refer the reader to [15] for a detailed derivation of the penalty parameters used in BAMPS.

2. Maximally dissipative boundary conditions

Maximally dissipative boundary conditions [11–13,22] will allow us to set boundary conditions of the form

$$s^i \partial_i \psi_\alpha |_{\partial\Omega} = \phi_\alpha(\psi_\beta, \partial_i \psi_\beta, q_j^i \partial_i \partial_k \psi_\beta), \quad (45)$$

where ϕ is a function that is allowed to depend on the coordinates x^i , the fields ψ_α , their derivatives and the transverse projections ($q_j^i = \delta_j^i - s^i s_j$) of their second derivatives. For brevity we suppress dependence on all the arguments in the following.

Enforcing the boundary conditions (45) would cause reflections which we wish to avoid from the outer boundary during the relaxation process. We can however construct maximally dissipative boundary conditions such that (45) is fulfilled in the steady state. Maximally dissipative boundary conditions are imposed by requiring

$$(\rho^{-1})_\gamma^\beta w_\beta^\alpha \partial_t \psi_\alpha + w_\gamma^\alpha \partial_s \psi_\alpha = w_\gamma^\alpha \phi_\alpha. \quad (46)$$

This boundary condition is actually different from Eq. (45) during the relaxation process. However, again we are only interested in the steady state at the end of the evolution, where we have $\partial_t \psi_\alpha = 0$ and thus the correct boundary condition will be imposed. For numerical stability the functions ϕ_α must not depend on normal derivatives of the evolved variables. Therefore in (45) in the arguments of ϕ_α we have to make the replacements $\partial_i \psi_\beta \rightarrow r_{i\beta}$ and $q_j^i \partial_i \partial_k \psi_\beta \rightarrow q_j^i \partial_i r_{k\beta}$. For the normal derivatives of the incoming characteristic we obtain the relation

$$d_s \hat{u}_\gamma^+ = w_\gamma^\alpha \partial_s \psi_\alpha + (\rho^{-1})_\gamma^\delta w_\delta^\epsilon a^{s\alpha} \epsilon \partial_s r_{i\alpha} \quad (47)$$

$$= w_\gamma^\alpha \phi_\alpha - (\rho^{-1})_\gamma^\delta w_\delta^\alpha (\partial_t \psi_\alpha - a^{sj\epsilon} \partial_s r_{j\epsilon}), \quad (48)$$

where in the actual implementation $\partial_t \psi_\alpha$ is to be replaced by Eq. (13). This equation is now used to impose the boundary condition by replacing the $d_s \hat{u}_\gamma^+$ terms in Eq. (43), yielding the modified right-hand side

$$d_t \hat{u}_\gamma^+ \hat{=} D_t \hat{u}_\gamma^+ - \rho_\gamma^\beta (w_\beta^\alpha \partial_s \psi_\alpha + (\rho^{-1})_\beta^\delta w_\delta^\alpha \partial_t \psi_\alpha - w_\beta^\alpha \phi_\alpha). \quad (49)$$

With the general expression at hand, we now discuss choices for ϕ_α that lead in the steady state to standard boundary conditions for elliptic equations.

- (1) *Dirichlet conditions.* Dirichlet conditions are of the form $\psi_\alpha |_{\partial\Omega} = g_\alpha$, where the g_α are some function defined on the domain boundary $\partial\Omega$. To achieve such a boundary condition in the steady state, ϕ_α has to take the form

$$\phi_\alpha = s^i r_{i\alpha} + e^\beta_\alpha (g_\beta - \psi_\beta), \quad (50)$$

where e is positive definite, i.e., $e^\beta_\alpha t_\beta t^\alpha > 0$. In the steady state we have $\partial_i \psi_\alpha = r_{i\alpha}$ and thus Eq. (46) becomes $0 = e^\beta_\alpha (g_\beta - \psi_\beta)$, which is only fulfilled for the requested boundary condition. The positive definiteness of e is important to guarantee stability at the boundary. Suppose we have $\partial_i \psi_\alpha = r_{i\alpha}$ fixed, then Eq. (46) has the form

$$(\rho^{-1})_\gamma^\beta w_\beta^\alpha \partial_t \psi_\alpha = w_\gamma^\alpha e^\beta_\alpha (g_\beta - \psi_\beta), \quad (51)$$

which would have solutions not asymptoting to g_α if e was not positive definite. Besides positive definiteness there are no further restrictions apparent on e and therefore, it can be chosen to be the identity $e^\beta_\alpha = \delta^\beta_\alpha$, which we use in our applications.

- (2) *Neumann conditions.* Neumann boundary conditions are of the form $s^i \partial_i \psi_\alpha |_{\partial\Omega} = g_\alpha$. Their implementation in our method is straightforward; one just has to take $\phi_\alpha = g_\alpha$.
- (3) *Robin conditions.* Robin boundary conditions are a mixture of Dirichlet and Neumann boundary conditions and can be written as $s^i \partial_i \psi_\alpha |_{\partial\Omega} = g_\alpha + f^\beta_\alpha \psi_\beta$, where the f^β_α are functions defined on the domain boundary. Their implementation is also straightforward choosing $\phi_\alpha = g_\alpha + f^\beta_\alpha \psi_\beta$.

III. NUMERICAL SETUP

A. Grid setup

We employ the pseudospectral hyperbolic evolution code BAMPS and refer the reader to [15], where the grid setup is explained in detail. Here we only give a short summary of the basic grid setup and numerical method. Our grid consists of different coordinate patches, a cube patch in the center, transition shell patches and outer shell patches. On each patch we have a mapping between local Cartesian coordinates to global Cartesian coordinates, where on shell patches we employ the ‘‘cubed sphere’’ construction [23]. The patches themselves can consist of smaller subpatches, which are the smallest units we use for

our parallelization scheme. On each subpatch we approximate the fields by a Chebyshev pseudospectral method, i.e., the subpatches are discretized in every direction by the Gauss-Lobatto collocation points. It is then possible to reconstruct the Chebyshev coefficients from the fields values at the collocation points.

The BAMPS code is adapted to evolutions in three dimensions. For axisymmetric and spherically symmetric problems we use the cartoon method to reduce the computational domain to two or one dimensions respectively [15].

B. Integration method

The time integration for relaxation methods does not require a high order of error convergence, since we are only interested in the steady state at the end of the evolution. More important are the efficiency and stability of the integration algorithm. For the time integration we use the method of lines. It is known that for linear hyperbolic equations the simple forward Euler-method and also explicit second-order Runge-Kutta methods are, at least without artificial dissipation, unstable (see e.g., Chapter 6.7 of [11]) and thus are not suited for the integration of the hyperbolic relaxation equations.

In the applications presented in this paper we employ the popular fourth-order Runge-Kutta scheme (RK4), which is stable for hyperbolic equations. This method needs four evaluations of the right-hand side per time step, which appears to be not very efficient. After all, we do not need a very accurate integrator, since we are only interested in approaching the stationary state. Therefore it is worthwhile to investigate other classes of integrators, e.g., multistep methods like the third- or fourth-order Adams schemes [24], which effectively only require one or two evaluations per time step and are usually also stable for hyperbolic PDEs. Some simple experiments with the Poisson equation indicated, however, that RK4 is more efficient than RK3 or a fourth order Adams scheme since RK4 allows comparatively large time steps.

Contrary to what is described in [15], we neither use nor need filtering to assure stability in the hyperbolic relaxation method, since the system usually tends towards a stable static or stationary solution automatically.

In our code we have two types of boundaries. On the one hand we have boundaries between different subpatches, and on the other hand the boundaries of our computational domain, in particular the outer boundaries. To treat boundaries between subpatches we employ the penalty method described in Sec. II E 1 setting the boundary data to be the outgoing characteristic of the neighboring subpatch.

Treating the outer boundary with this method we have to provide a function g_γ equaling u_γ^{+BC} at the boundary, i.e.,

$$\hat{u}_\gamma^{+BC} = w_\gamma^\alpha \psi_\alpha + (\rho^{-1})_\gamma^\delta w_\delta^\epsilon a^{s\iota\alpha} \epsilon r_{i\alpha} = g_\gamma. \quad (52)$$

Here the given data could be generalized to include combinations of the outgoing characteristic variables at outer boundary. This strategy could be used to implement a desired boundary condition from the original elliptic PDE, but has the undesirable side-effect of reflection from the boundary, which may serve as an obstruction to decay of the residual. On the other hand, a boundary condition of the direct form (52) would be unusual in practice for elliptic equations, as the characteristic fields of the relaxation scheme have no special meaning in the original system. Therefore the penalty method is not best suited for the treatment of our outer boundaries. Instead, as described in Sec. II E 2, we embed the desired boundary conditions for the elliptic system inside boundary conditions for the relaxation scheme that are more likely to absorb outgoing waves.

C. Initial guesses

To start the hyperbolic relaxation one has to provide an initial guess to the solution. A suitable initial guess will always depend on the specific form of the problem, in particular it should be chosen such that in the course of the relaxation the variables do not have to cross any points where the equations (e.g., terms in the nonprincipal part) become singular. In our tests we found the solver to be particularly well behaved when starting with a guess that is stationary in the interior, but not at the boundary. The whole solution then starts to relax from the boundary to the inside. For example in our applications to numerical relativity initial data, taking the flat metric everywhere lead to stable relaxations, which demonstrates a remarkably high robustness that can be achieved by the method. For the reduction variables we simply take the numerical derivative of the initial guess, i.e., ${}^{(ini)}r_{i\alpha} = \partial_i {}^{(ini)}\psi_\alpha$.

D. Refinement strategy

To speed up the relaxation process we employ a simple scheme of successive refinement. It can be assumed that the right-hand side of the solution variables $\partial_t \psi_\alpha$, Eq. (13), is a good approximation to the residual of the elliptic equation $(L\psi)_\alpha$, Eq. (7). This however is only true until a discretization limit is reached below which the norm of the residual is no longer decreasing. The norm of the $\partial_t \psi_\alpha$ will typically continue decreasing until machine precision is reached. We note that this continuing decrease will only be present if no spectral filter is applied. This observation makes it possible to construct an indicator signaling when the discretization limit is reached and thus relaxation should be continued on a higher resolution grid. In particular we choose the following criterion,

$$\int \sum_{\alpha=1}^N |\partial_t \psi_\alpha| dV < c \int \sum_{\alpha=1}^N |(L\psi)_\alpha| dV, \quad (53)$$

where c is some constant smaller than one. For our applications we found $c = 0.1$ to be a choice working reasonably well. We note however that depending on the specific problem also smaller values might be beneficial. Additionally we increase the resolution when the error of the elliptic equation reaches machine precision, i.e., when the norm of $(L\psi)_\alpha$ is smaller than 10^{-13} times the number of grid points.

We start with relaxing the system on the coarsest grid and check every 1000 relaxation steps whether to proceed relaxation on a finer grid based on whether one of the two criteria mentioned above is fulfilled. The final resolution can be determined by an error bound on the residual of the elliptic equation, or by some predetermined resolution, which may be required for the evolution of the data. For the refinement we increase the resolution on every subpatch by two collocation points in every direction, which is equivalent to adding two Chebyshev modes in every direction. We interpolate the coarse steady state solution to the new subpatches and repeat the procedure until we arrive at the desired resolution. We also find it advisable to use the interpolated values for the reduction variables instead of taking the numerical derivative of the solution variables, since the latter introduces new errors, which costs some extra effort to damp.

IV. APPLICATION TO TEST CASES

A. Poisson equation—Finite differencing

To provide a reference point independent of the specific pseudospectral methods of BAMPS, we first discuss a minimal implementation using a finite difference method to solve the Poisson equation. We consider the hyperbolic relaxation equation

$$\partial_t^2 \phi + \eta \partial_t \phi = \Delta \phi - \rho, \quad (54)$$

which we implement as a first order in time, second order in space system,

$$\partial_t \phi = \pi - \eta \phi, \quad (55)$$

$$\partial_t \pi = \Delta \phi - \rho. \quad (56)$$

We consider the fully first order version of this system in Sec. IV B. At the boundaries we use asymptotic Dirichlet conditions analogous to (50), $\partial_t \phi = g - \phi$ and $\partial_t \pi = g - \pi$.

We choose centered, second order accurate finite differences in space, and the default time integrator is the classic fourth-order Runge-Kutta method. The numerical domain is an equidistant grid of points in $[-\frac{L}{2}, \frac{L}{2}]^d$, dimension $d = 1, 2, \text{ or } 3$, with Cartesian coordinates. There are N points in each of up to three directions with a total of $V = N^d$ points.

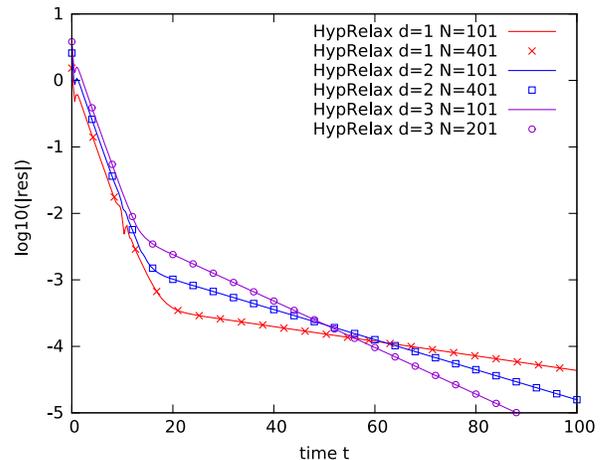


FIG. 2. Poisson equation, FD method. Convergence of the residual with time for one, two, and three dimensions. Shown is a solid line for $N = 101$ and markers for a higher resolution given by $N = 201$ or 401 points. On the scale of the plot, the convergence rate is independent of resolution for any given number of dimensions.

Let us discuss some results for vanishing source term, $\rho = 0$, and vanishing Dirichlet boundary, $g = 0$, where the method has to reduce an initial guess of $\phi = 1/(1 + x_j x^j)$ and $\pi = 0$ at $t = 0$ to the asymptotic, late-time value $\phi = \pi = 0$.

In Fig. 2, we show results for a box of size $L = 20$, damping parameter $\eta = 1$, varying the number of points and the number of dimensions. The norm is weighted by the grid spacing Δx to represent the integral of the residual, $\|f\|_2 = (\sum f^2 \Delta x^d)^{1/2}$. Convergence is exponential in time, with two distinct phases. Inspection of the evolution of ϕ and π shows that the initial phase corresponds to the damping of short wavelengths (in this example until $t \approx 20$), after which long wavelengths dominate and the convergence is slower. The convergence of the (weighted) norm of the residual with time is quite independent of the resolution. In this example the time-step is $\Delta t = \lambda \Delta x$, so the number of time steps is proportional to the number N of grid points in one direction. For efficiency we chose $\lambda = 1.4, 1.0, 0.8$ for 1d, 2d, and 3d, respectively, and obtain stable time-stepping with RK4. The work per right-hand-side evaluation is $\mathcal{O}(N^d)$, so the total work to reach a final time T is $\mathcal{O}(N^{d+1})$.

A key question is how efficient hyperbolic relaxation is compared to other methods. In Fig. 3, we show a comparison of different methods for a two-dimensional example with $N = 101$ points. The methods considered are hyperbolic relaxation as above, the standard Jacobi iteration [25], and the BiCGSTAB method as an example for a Krylov subspace method [26]. Also included are two additional variants of hyperbolic relaxation. In these examples $\Delta t = 1.0 \Delta x$ for RK4 in 2d.

Referring to Fig. 3, the Jacobi method shows the slowest convergence. Reducing the residuum of the 2d Poisson

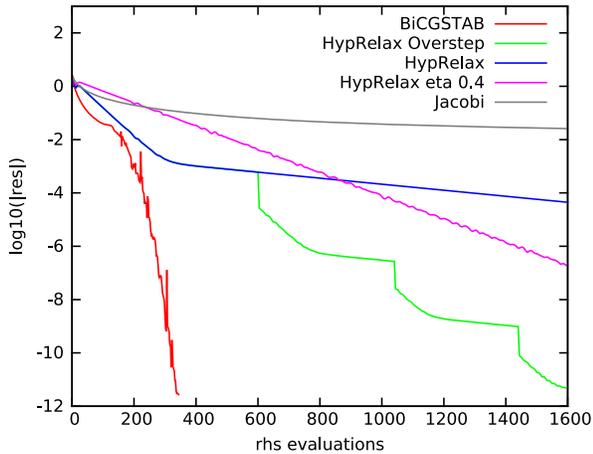


FIG. 3. Poisson equation, FD method. Convergence of the residual with the number of right-hand-side evaluations. Shown is a comparison between different methods for $N = 101$ in two dimensions.

equation by a factor 10^{-p} requires $n \approx \frac{1}{2} p N^2$ iterations on a $N \times N$ grid [25]. For a 2d grid with $V = N^2$ degrees of freedom, the operation count is therefore $\mathcal{O}(V^2) = \mathcal{O}(N^4)$, compared to $\mathcal{O}(N^3)$ for optimal SOR and $\mathcal{O}(V \log V)$ for multigrid methods. Hyperbolic relaxation with $\mathcal{O}(V) \times \mathcal{O}(N) = \mathcal{O}(N^3)$, as demonstrated in Fig. 2, is therefore a reasonable candidate for further consideration. In the concrete example, the Jacobi method is significantly slower than hyperbolic relaxation, but the Jacobi method is usually not considered as a stand-alone method.

For this simple comparison, the BiCGSTAB method is used without a preconditioner, but the Laplace operator leads to a sufficiently well conditioned operator such that convergence is fast nevertheless, compared to the other methods considered here. There is an initial phase of relatively slow convergence, but once the trial solution is sufficiently close to the final answer, convergence becomes much faster.

Remarkably, hyperbolic relaxation does about as well as BiCGSTAB during the first phase. However, convergence slows down after the shorter wavelengths have been damped and errors due to larger wavelength remain. We have considered three ideas to improve the convergence of hyperbolic relaxation for long wavelengths. Not shown here is the multi-level refinement strategy which we employ in the BAMPS code, see Sec. III D.

As an immediate application of the mode analysis of Sec. II C, we introduced the damping parameter η , which for the basic experiments so far was set to $\eta = 1$. Also shown in Fig. 3 is the result for $\eta = 0.4$, which exhibits a constant decay rate that is slower than $\eta = 1$ initially, but faster for later iterations. This effect depends on η and the size $L = 20$ of the box. With $\epsilon = 1$, for $\eta = 1$ we have $\lambda_{\text{crit}} = 4\pi < L$, and for $\eta = 0.4$ we have $\lambda_{\text{crit}} = 10\pi > L$. This indicates that for $\eta = 0.4$ all wavelengths fitting into

the box fall into the range $\lambda < \lambda_{\text{crit}}$, and we expect the constant damping rate $e^{-\frac{\eta}{2}t}$ for all wavelengths. On the other hand, for $\eta = 1$ wavelengths both smaller and larger than λ_{crit} are present initially, but damping for $\lambda < \lambda_{\text{crit}}$ is faster than for $\lambda > \lambda_{\text{crit}}$, so after an initial transient the damping rate slows down when $\lambda > \lambda_{\text{crit}}$ is the dominant contribution. There is a trade-off between reducing η in order to suppress $\lambda < \lambda_{\text{crit}}$ for a given domain size, and increasing η for a stronger damping factor $e^{-\frac{\eta}{2}t}$. In the example of Fig. 3 HypRelax with $\eta = 0.4$ overtakes HypRelax with $\eta = 1$ at about 900 RHS evaluations. It seems possible to construct a dynamically adjusted damping $\eta(t)$.

Similar results hold for the parameter ϵ in (21) as discussed in Sec. II C. At large k the velocity of the modes scales with $1/\sqrt{\epsilon}$ but is independent of η , so for optimal performance the Courant factor has to be adjusted together with ϵ but can be kept constant when η is varied.

We also experimented with a “one-step overrelaxation” method (as opposed to successive overrelaxation). This is based on the observation that after the initial propagation/damping phase of hyperbolic relaxation, the second time derivative of ϕ becomes significantly smaller than the first time derivative, $\partial_t^2 \phi \ll \partial_t \phi$. Hence it seems promising to attempt a *linear* extrapolation in time. The curve labeled “overstep” in Fig. 3 is obtained by searching every few iterations for the time step $\Delta T = \lambda \Delta t$ that minimizes the global residual of $\phi_{\text{new}} = \phi + \Delta T F(\phi)$, where F is the update suggested by the time stepping algorithm (e.g., RK4). This is similar to various other 1d step-size optimizations. For the example considered here (but also for $\rho \neq 0$ as below), the late time solution of hyperbolic relaxation is sufficiently regular that indeed an appropriate global ΔT can be found. The overstep algorithm only accepts the large step ΔT if it decreases the residual by at least a factor f , say $f = 10$ (we tried $f = 2$ to $f = 1000$). The optimal choice of f depends on different features of the problem, in particular choosing f too small can make the method less efficient. Each overstep introduces new local error modes (since ΔT is a global parameter). As can be seen in Fig. 3, the approximate solution is disturbed, but converges again with the typical speed for shorter wavelengths to a new regular state. In the optimal case the overall convergence rate seems to approach that of the fast phase of hyperbolic relaxation. It may be possible to derive a continuous variant of this method analogous to successive overrelaxation, which we leave to future research. With the *scheduled Jacobi method* [27,28] there has been a similar idea for a parabolic solver, accelerating the Jacobi method considerably by a clever schedule of over- and under-relaxation steps.

The main points regarding the convergence rate of hyperbolic relaxation as shown in Fig. 3 are that the method works out-of-the-box and that its performance falls somewhere between Jacobi and BiCGSTAB. There seems to be quite some potential for accelerating the

convergence rate of hyperbolic relaxation. From the point of view of solving elliptic equations with a code designed for hyperbolic equations, note that hyperbolic relaxation is “only” slower by a factor of about 5 (to reach a residual of 10^{-9} in this example) than a standard method like BiCGSTAB, which however may not be readily available.

B. Poisson equation—Pseudospectral method

To test the hyperbolic relaxation elliptic solver as implemented in BAMPS we start by solving Poisson’s equation,

$$\Delta\psi - \rho = 0, \quad (57)$$

in spherical symmetry, i.e., $\rho = \rho(r)$, $r = \sqrt{x^i x_i}$. To solve this equation we choose the hyperbolic relaxation system

$$\partial_t \psi = \delta^{ij} \partial_i r_j - \rho, \quad (58)$$

$$\partial_t r_i = \partial_i \psi - r_i. \quad (59)$$

For our first test we take ρ to be smooth, i.e., it is infinitely often continuously differentiable,

$$\rho = \rho_0 \left(\frac{-6}{R^2} + \frac{4r^2}{R^4} \right) e^{-r^2/R^2}, \quad (60)$$

where R and ρ_0 are nonzero parameters. For this ρ Poisson’s equation has the solution

$$\psi_{\text{analytic}} = \rho_0 e^{-r^2/R^2}. \quad (61)$$

At the boundary a falloff in ψ compatible with this solution is obtained by imposing the Robin boundary condition $\partial_r \psi = s^i \partial_i \psi = -2r\psi/R^2$.

For our second test we take a nonsmooth ρ that corresponds to a homogeneously charged sphere, which is like a toy model for stars. The density ρ is then given by

$$\rho = \begin{cases} \rho_0 & \text{if } r \leq R \\ 0 & \text{if } r > R, \end{cases} \quad (62)$$

for which the Poisson equation has the solution

$$\psi_{\text{analytic}} = \rho_0 \begin{cases} \frac{r^2}{6} - \frac{R^2}{2} & \text{if } r \leq R \\ -\frac{R^3}{3r} & \text{if } r > R. \end{cases} \quad (63)$$

Again we impose Robin boundary conditions according to the falloff of this solution, i.e., $\partial_r \psi = s^i \partial_i \psi = -\psi/r$.

In our tests we place the outer boundary at radius of 10 and we divide the grid into a total of eight subpatches, where the inner five extend over the interval $[0, 5]$ and the outer three, having a coarser resolution, extend over $[5, 10]$. The parameters determining ρ are chosen to be $R = 5$ and

$\rho_0 = 1$. For the nonsmooth case special care has to be taken to ensure convergence. In particular we chose the grid such that the discontinuity lies at a boundary of subpatch, ensuring second order convergence. In both test cases the relaxed solution converges with the number of grid points to the analytical solution. To investigate the convergence we have to make sure that the solution is completely relaxed on every resolution. This is achieved by choosing in Eq. (53) $c = 0.0001$. In Fig. 4 we report the absolute difference between the analytical and numerical solution integrated over the outermost subpatch. We note however that the convergence behavior is the same on all other subpatches. As expected we find the error of the numerical solution to decrease exponentially with the number of points for the smooth ρ from Eq. (60). For the nonsmooth ρ of Eq. (62) it is well known that convergence can only be polynomial and indeed we only get a convergence order of approximately two, which is the expected convergence order for discontinuous ρ . Of course this is not very efficient for a spectral method. In nonsmooth regions it is therefore often preferable to increase the number of subpatches (h-refinement) instead of the number of collocation points per subpatch. In numerical applications one often uses spectral methods even for nonsmooth problems, i.e., in (general relativistic) hydrodynamics, and thus it is still interesting to investigate the behaviour of our method on grids suited for these simulations.

In Fig. 5 we investigate how the L1-norm of different quantities, that can be used to approximate the error, progresses during the relaxation process. First we observe

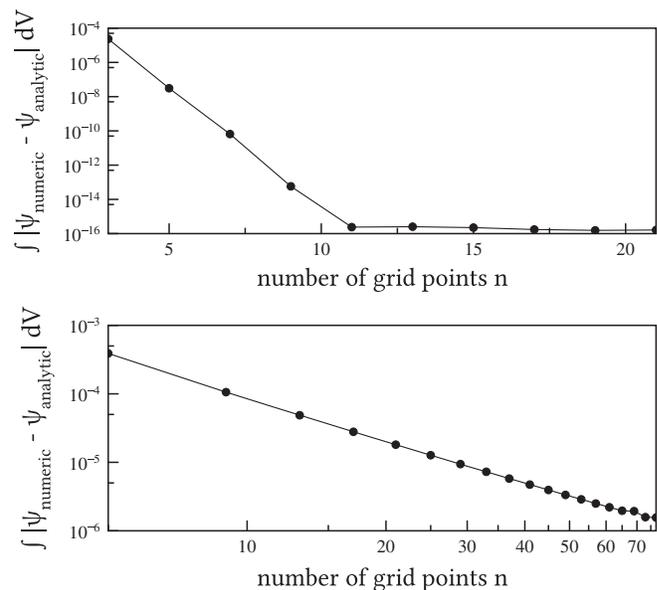


FIG. 4. Convergence of the L1-norm of the difference between the analytical and the numerical solution. Upper plot: for smooth ρ [Eq. (60)]. Lower plot: for nonsmooth ρ [Eq. (62)]. Note that in the upper plot only the error axis is logarithmic, while in the lower plot both axes are logarithmic.

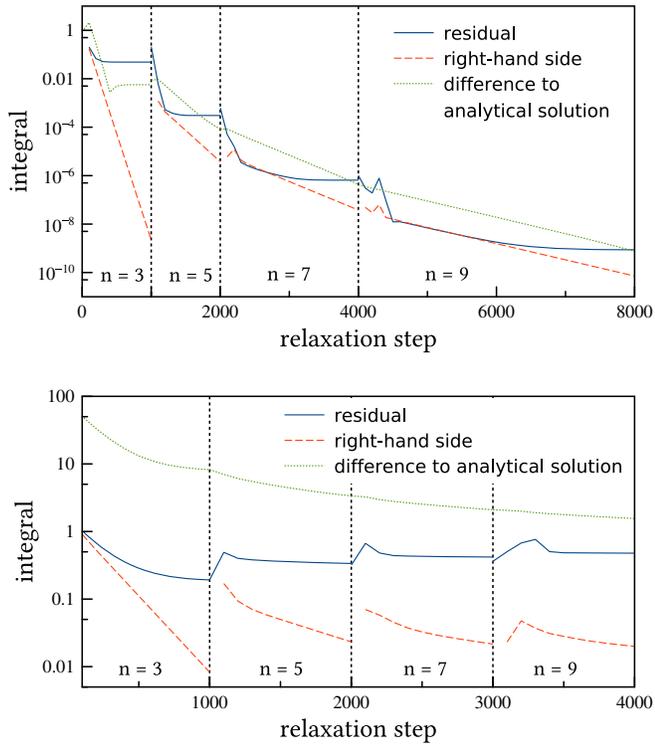


FIG. 5. Progression of the L1-norm of different error quantities during the relaxation process for the Poisson equation. Vertical dashed lines indicate transitions to a finer grid. The respective error quantities are: blue solid line: residual, defined as left-hand side of Eq. (57), orange dashed line: right-hand side of Eq. (58), green dotted line: difference to the analytical solution. Upper plot: for smooth ρ [Eq. (60)]. Lower plot: for nonsmooth ρ [Eq. (62)].

that the difference to the analytical solution decreases even when the computed residual, given by left-hand side of Eq. (57), is already leveling off. This is especially remarkable for the non-smooth case, where the residual itself is not converging at all. For the smooth case we secondly observe that after refining the grid the norm of right-hand side of Eq. (58) practically continues at the same level as before. The norm of the residual on the other hand drops quickly after refining, reaching the right-hand sides level until again the discretization limit is reached. These observations suggest that for problems with smooth solutions it is preferable to relax for longer on the coarse grid. For problems with nonsmooth solutions, however, more new error develops during each refinement and thus refining for longer on the coarse grid is not paying off. Furthermore, it is preferable to increase the grid resolution faster.

As a last simple test, we investigated the behavior in the case of nonunique solutions. For this we took the smooth ρ from Eq. (60) and imposed the Neumann boundary condition $\partial_r \psi = 0$, for which multiple solutions differing only by an additive constant exist. We find that after some relaxation the right hand side of Eq. (58) becomes approximately constant in space. From this point on the

solution is no longer improving, since only constant terms, which do not improve the residual of Eq. (57), are added.

V. APPLICATION TO INITIAL DATA FOR GENERAL RELATIVITY

A. The extended conformal thin-sandwich equations

For the formulas in this section we introduce Latin letters a, b, c, \dots, h to denote the spacetime components of general relativity. These spacetime indices are lowered and raised by the spacetime four-metric g_{ab} . In numerical relativity one usually decomposes the spacetime metric g_{ab} into a temporal and spatial part in the form

$$g_{ab} dx^a dx^b = -\alpha^2 dt^2 + \gamma_{ij} (dx^i + \beta^i dt)(dx^j + \beta^j dt), \quad (64)$$

where α is called the lapse, β^i the shift and γ_{ij} the spatial metric. The equations of motion in general relativity are subject to constraint equations, which have to be solved before the spacetime is evolved numerically. A popular formulation of the constraint equations is given by the extended conformal thin-sandwich (XCTS) equations [29,30] in which the spatial metric is decomposed into a conformal factor ψ and a spatial conformal metric $\bar{\gamma}_{ij}$ as $\gamma_{ij} = \psi^4 \bar{\gamma}_{ij}$. In the XCTS framework the constraint equations take the form

$$\bar{D}^j \bar{D}_j \psi = \frac{\psi}{8} \bar{R} - \psi^5 \left(2\pi\rho - \frac{K^2}{12} + \frac{1}{8} A_{ij} A^{ij} \right), \quad (65)$$

$$\begin{aligned} \bar{D}^j \bar{D}_j \beta^i &= -\frac{1}{3} \bar{D}^i \bar{D}_j \beta^j - \bar{R} \beta^i + 16\pi\alpha\psi^4 J^i \\ &+ \left(\bar{D}^i \beta^j + \bar{D}^j \beta^i - \frac{2}{3} \bar{\gamma}^{ij} \bar{D}_k \beta^k \right) \\ &\cdot \bar{D}_j \ln(\alpha\psi^{-6}) \\ &- \alpha\psi^{-6} \bar{D}_j (\alpha^{-1} \psi^6 \partial_i \bar{\gamma}^{ij}) + \frac{4}{3} \alpha \bar{D}^i K, \end{aligned} \quad (66)$$

$$\begin{aligned} \bar{D}^j \bar{D}_j (\alpha\psi) &= \alpha\psi^5 \left(\frac{7}{8} A_{ij} A^{ij} + \frac{5}{12} K^2 + 2\pi(\rho + 2J) \right) \\ &- \psi^5 (\partial_t - \beta^j \bar{D}_j) K + \frac{1}{8} \alpha\psi \bar{R}. \end{aligned} \quad (67)$$

Here \bar{D}_i is the covariant derivative compatible with the conformal metric $\bar{\gamma}_{ij}$, \bar{R}_{ij} is the Ricci tensor of $\bar{\gamma}_{ij}$ and \bar{R} is the corresponding Ricci scalar. The tensor A_{ij} is the tracefree part of the extrinsic curvature K_{ij} and K is the trace of K_{ij} . The matter source terms are defined as the following contractions of the energy-momentum tensor T_{ab} : $\rho = T_{ab} n^a n^b$, $J^i = -T_{ab} \gamma^{ia} n^b$ and $J = \gamma^{ab} T_{ab}$, where n^a is the timelike vector normal to the spatial hypersurface $n^a = (1/\alpha, -\beta^i/\alpha)$. For conformal quantities (denoted by a bar) the conformal spatial metric $\bar{\gamma}^{ij}$ lowers and raises

indices and for unbarred quantities the physical spatial metric γ^{ij} is used. In the XCTS equations $\bar{\gamma}_{ij}$, $\partial_i \bar{\gamma}_{ij}$, K and $\partial_t K$ are given functions, depending on the type of initial data you want to construct.

In Eq. (67) the product $\alpha\psi$ is taken as one variable. For our computations we rewrite this equation with the help of Eq. (65) as

$$\begin{aligned} \bar{D}^j \bar{D}_j \alpha = & -\frac{2}{\psi} (\bar{D}^j \alpha) (\bar{D}_j \psi) - \psi^4 (\partial_t - \beta^j \bar{D}_j) K \\ & + \alpha \psi^4 \left(A_{ij} A^{ij} + \frac{K^2}{3} + 4\pi(\rho + J) \right), \end{aligned} \quad (68)$$

so we can solve directly for α . We have tested our elliptic solver with both versions and found them to work equally well in our applications. In the following we investigate the XCTS system with our replacement for the lapse equation, but the analysis would be exactly the same for the original system.

The principal part of the XCTS equations is given by

$$\begin{pmatrix} \bar{\gamma}^{kj} & 0 & 0 \\ 0 & \bar{\gamma}^{kj} \delta_i^l + \frac{1}{3} \bar{\gamma}^{lk} \delta_i^j & 0 \\ 0 & 0 & \bar{\gamma}^{kj} \end{pmatrix} \partial_k \partial_j \begin{pmatrix} \psi \\ \beta^i \\ \alpha \end{pmatrix}. \quad (69)$$

The principal part is coupled only between the components of β^i . Thus we can carry out the hyperbolicity analysis independently for ψ , α and β^i .

The metric to lower and raise spatial indices in the hyperbolic relaxation method is in principal arbitrary, but for the XCTS equations we use the conformal metric $\bar{\gamma}_{ij}$, because this choice simplifies the following formulas considerably. Another peculiarity is the fact that spatial indices and field indices ‘‘mix,’’ but in general they are lowered and raised by different metrics. Therefore we do not lower and raise field indices and instead write the Euclidian metric explicitly.

For the conformal factor part we have in the hyperbolic relaxation system $a^{ij} = \bar{\gamma}^{ij}$ and we choose $b^i_j = \delta^i_j$. Here we have suppressed the field indices, because we only have a single field ψ . The characteristic variables and speeds of the hyperbolic relaxation system of the ψ part are then given by

$$\hat{u}_j^0 = r_j - s_j s^i r_i^{(\psi)}, \quad \lambda_i^0 = 0, \quad (70)$$

$$\hat{u}^\pm = \psi \pm s^i r_i^{(\psi)}, \quad \lambda^\pm = \pm 1, \quad (71)$$

where $r_i^{(\psi)}$ is the reduction variable for ψ .

The lapse part has exactly the same principal part, so we have identical a^{ij} and b^i_j and the characteristics are

$$\hat{u}_j^0 = r_j - s_j s^i r_i^{(\alpha)}, \quad \lambda_i^0 = 0, \quad (72)$$

$$\hat{u}^\pm = \alpha \pm s^i r_i^{(\alpha)}, \quad \lambda^\pm = \pm 1, \quad (73)$$

with $r_i^{(\alpha)}$ being the reduction variable for α .

In the derivation of the hyperbolic relaxation equations we labeled the fields with lower indices. For the shift however we use upper indices here and thus one has to be careful, not to confuse the indices. Therefore, we introduce auxiliary fields ϕ_α with

$$\beta^\alpha = \delta^{\alpha\beta} \phi_\beta. \quad (74)$$

Substituting β^α we obtain for the principal part of the shift equations

$$a^{ij\beta} \partial_i \partial_j \phi_\beta = \left(\bar{\gamma}^{ij} \delta_\alpha^\beta + \frac{1}{3} \bar{\gamma}^{i\epsilon} \delta_{\epsilon\alpha} \delta^{j\beta} \right) \partial_i \partial_j \phi_\beta. \quad (75)$$

We take $b^i_j{}^\beta{}_\alpha$ to be the inverse of a (as defined in Sec. II D), $b^i_j{}^\beta{}_\alpha = \delta_j^\beta \delta_\alpha^i - \frac{1}{6} \delta^{i\beta} \delta_{j\alpha}$. The characteristic variables and speeds for this part of the hyperbolic relaxation system are then given by

$$\hat{u}_{j\gamma}^0 = r_{j\gamma} - b^s_j{}^\epsilon (b^s_s{}^{-1})^\alpha \epsilon^s r_{i\alpha} \quad (76)$$

$$= r_{j\gamma} - s_j s^i r_{i\gamma} + \frac{1}{5} (\delta_{j\gamma} - s_j s^l \delta_{l\gamma}) s_k \delta^{k\alpha} s^i r_{i\alpha}, \quad (77)$$

$$\lambda_{i\gamma}^0 = 0, \quad (78)$$

$$\hat{u}_\gamma^\pm = \phi_\gamma \pm a^{s\alpha}{}_\gamma r_{i\alpha} \quad (79)$$

$$= \phi_\gamma \pm \left(s^i r_{i\gamma} + \frac{1}{3} s^\epsilon \delta_{\epsilon\gamma} \delta^{j\alpha} r_{j\alpha} \right), \quad (80)$$

$$\lambda_\gamma^\pm = \pm 1, \quad (81)$$

where the $r_{i\alpha}$ denote the reduction variables for the auxiliary fields ϕ_α .

At the domain boundary we want the solution to fall off like the Schwarzschild solution, i.e., $\psi = \frac{a}{r} + 1$ and $\alpha = \frac{b}{r} + 1$. This ansatz gives rise to the following Robin boundary conditions

$$\partial_s \psi|_{\partial\Omega} = \frac{1-\psi}{r}, \quad \partial_s \alpha|_{\partial\Omega} = \frac{1-\alpha}{r}. \quad (82)$$

For the shift we likewise impose a radial falloff by the Robin condition

$$\partial_s \beta^i|_{\partial\Omega} = \frac{\beta^i}{r}. \quad (83)$$

As an initial guess we always use the flat space solution, i.e., $\psi = 1$, $\alpha = 1$, $\beta^i = 0$. Of course an initial guess, that is

a good approximation to the solution is always the preferred start for the relaxation, since it will take less time to relax to the solution or might be necessary to relax at all. However we find our simple initial guess to work well and it demonstrates in a nice way the high robustness of the hyperbolic relaxation method exhibited in our experiments.

B. Scalar field

The energy-momentum tensor for a scalar field ϕ is given by

$$T_{ab} = \nabla_a \phi \nabla_b \phi - \frac{1}{2} g_{ab} (\nabla_c \phi \nabla^c \phi + m^2 \phi^2), \quad (84)$$

where ∇ denotes the covariant derivative compatible with g_{ab} . We consider conformally flat moment-of-time-symmetry initial data, i.e., $\tilde{\gamma}_{ij} = \delta_{ij}$, $n^a \nabla_a \phi = 0$, $\partial_i \tilde{\gamma}_{ij} = 0$ and maximal slicing $K = 0$, $\partial_t K = 0$. This yields for the matter quantities

$$\rho = \frac{1}{2} \gamma^{ij} (\partial_i \phi) (\partial_j \phi) + \frac{1}{2} m^2 \phi^2, \quad (85)$$

$$J^i = 0, \quad (86)$$

$$J = -\frac{1}{2} \gamma^{ij} (\partial_i \phi) (\partial_j \phi) - \frac{3}{2} m^2 \phi^2. \quad (87)$$

For a massless scalar field ($m = 0$) the solutions for lapse and shift are trivially given by $\alpha = 1$ and $\beta^i = 0$ and we only have to determine the conformal factor ψ by solving Eq. (65), now taking the form

$$0 = \delta^{ij} \partial_i \partial_j \psi + \pi \psi \delta^{ij} (\partial_i \phi) (\partial_j \phi). \quad (88)$$

For a massive scalar field ($m \neq 0$) one would have either have to solve additionally for the lapse or one gives up the requirement on $\partial_t K$. For the scalar field we choose radially symmetric, smooth initial data of the form

$$\phi(r) = p \left(\tanh \frac{r}{\sigma} - \tanh \frac{r}{\sigma} \right), \quad (89)$$

where σ and p are free parameters, that we choose for our test to be $\sigma = 1$ and $p = 0.1$.

The computational domain is divided into eight subpatches, where the inner five subpatches are of smaller extent to improve the resolution near the center. In Fig. 6 we show the numerical solution for the conformal factor and the absolute value of the residual of Eq. (65) for a resolution of 21 collocation points per subpatch. A general feature that can be observed in the residual are spikes at the boundary of subpatches, which are expected for quantities involving first and second derivatives in a discontinuous Galerkin approximation.

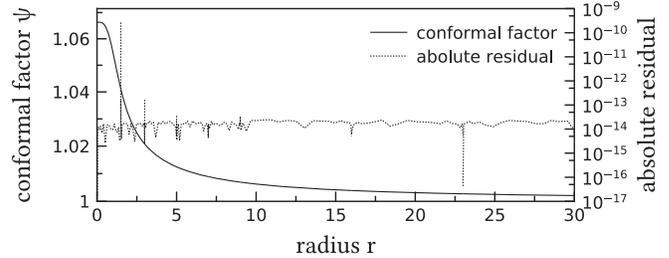


FIG. 6. Steady state at a resolution of 21 collocation points per subpatch for the initial data of the scalar field. Solid line: conformal factor. Dashed line: residual for conformal factor, as given by the right-hand side of Eq. (88).

We also investigate how the solution converges with increasing resolution. For this purpose we look at the Chebyshev coefficients and investigate their convergence against the coefficients of a high resolution solution. In Fig. 7 we present the convergence behavior of the lowest Chebyshev mode $C(0,0,0)$ against its value for a resolution of 31 collocation points. We observe an exponential convergence until we hit machine precision at around a resolution of 25 collocation points per subpatch.

C. Tolman-Oppenheimer-Volkoff star

The energy-momentum tensor of a perfect fluid is given by

$$T_{ab} = (\epsilon + p) u_a u_b + p g_{ab}, \quad (90)$$

where ϵ is the proper energy density, p the fluid pressure and u_a the fluid four-velocity. The Tolman-Oppenheimer-Volkoff (TOV) solution [29,31,32] is a static radially symmetric solution to general relativity, thus we have $u_a = (\alpha, 0)$, $\tilde{\gamma}_{ij} = \delta_{ij}$, $\partial_t \tilde{\gamma}_{ij} = 0$ and we consider again

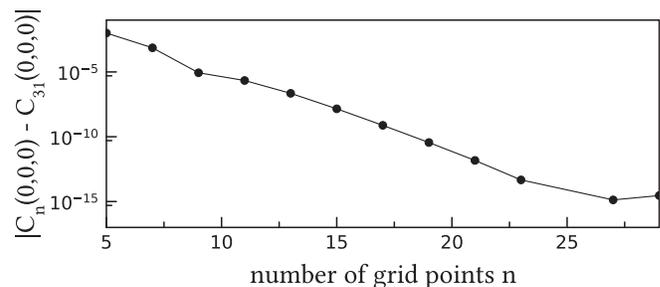


FIG. 7. Convergence of the lowest Chebyshev coefficient $C(0,0,0)$ at the innermost subpatch for the initial data of the scalar field. The system is relaxed for different numbers of collocation points. The plot shows the absolute value of the difference between the Chebyshev coefficient at the highest resolution (31 collocation points) and its value for n collocation points per subpatch. For $n = 25$ the difference is zero within machine precision and is therefore not displayed on the logarithmic axis.

maximal slicing $K = 0$, $\partial_t K = 0$. The matter quantities then become

$$\rho = \epsilon, \quad J^i = 0, \quad J = 3p. \quad (91)$$

We assume for our tests a polytropic equation of state and express the matter quantities in terms of the specific enthalpy h ,

$$\rho = \left(1 + \frac{n(h-1)}{1+n}\right) \left(\frac{h-1}{\kappa(1+n)}\right)^n, \quad (92)$$

$$J = 3\kappa \left(\frac{h-1}{\kappa(1+n)}\right)^{n+1}, \quad (93)$$

where κ is the adiabatic constant and n is the polytropic index. The Euler equation follows from energy-momentum conservation, i.e., for a fluid with temperature $T = 0$

$$u^a (\nabla_a (h u_b) - \nabla_b (h u_a)) = 0, \quad (94)$$

which has to be satisfied in addition to the XCTS equations. For our assumptions the Euler equation is satisfied for $\alpha h = \text{const}$ and thus specifying the specific enthalpy at the origin yields

$$h(r) = \begin{cases} \frac{h(0)\alpha(0)}{\alpha(r)} & \text{if } \alpha(r) < h(0)\alpha(0) \\ 1 & \text{else} \end{cases}. \quad (95)$$

We can immediately get the solution for the shift $\beta^i = 0$ and are left with solving Eqs. (65) and (68) of the XCTS system.

For our test we choose an adiabatic constant of $\kappa = 123.6489$ and a polytropic index of $n = 1$ and the enthalpy in the star's center is set to $h(0) = 1.2$. We present the solution for the conformal factor and the lapse in Fig. 8 and investigate the convergence of their lowest Chebyshev modes in Fig. 9. The residuals are large at the stellar surface, i.e., where $\alpha(r) = h(0)\alpha(0)$. This is caused by the fact that the matter terms are not smooth at this point, which can be seen in the kink in the specific enthalpy h . As for the nonsmooth right-hand sides discussed in Sec. IV B, we have to make sure that the kink lies at a subpatch boundary. By trial and error we find the stellar surface for the above parameters to be located at $r = 9.7098$ and we place a subpatch boundary at this position "by hand." A more sophisticated method is to fit coordinates automatically to the stellar surface [33]. This however is beyond the scope of this paper as we here want to focus on applications of the hyperbolic relaxation method. In contrast to what we observed in the nonsmooth case for the Poisson equation, here the Chebyshev coefficients converge exponentially despite the kink in the specific enthalpy. The convergence rate however is much smaller than that observed for the initial data of the scalar field.

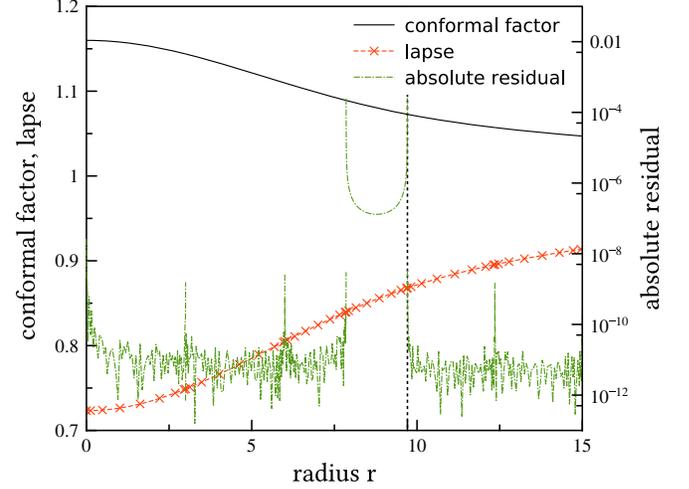


FIG. 8. Steady state at a resolution of 27 collocation points per subpatch for the initial data of the TOV star. Solid line: conformal factor. Orange dashed line with markers: lapse. Green dash-dotted line: absolute value of the residual for the conformal factor, as given by the right-hand side of Eq. (65). Vertical dashed line: position of the stellar surface.

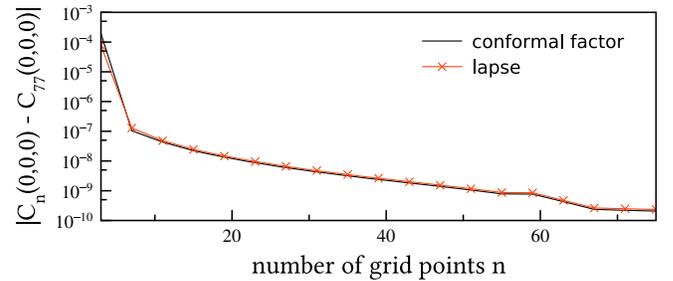


FIG. 9. Convergence of the lowest Chebyshev coefficient as in Fig. 7 for the initial data of the TOV star. The highest resolution used in these runs was 77 collocation points per subpatch.

D. Neutron star binaries

For the construction of neutron star binary initial data we follow the scheme of [32] using the constant three-velocity approximation. However, to solve the XCTS equations we do not rely on iterating the solution of the equations for the conformal factor, lapse and shift separately, but instead we solve for all variables simultaneously relaxing the complete XCTS system, which accelerates the solution process. This is a feature that most solvers for this type of initial data do not provide, and it could turn out to be an advantage of our method. Furthermore, we do not start with superposed (boosted) TOV solutions, but instead start with a flat metric $\psi = 1$, $\alpha = 1$, β^i , as discussed at the end of Sec. VA.

For our test we consider equal mass neutron stars with a specific enthalpy of $h = 1.01$ in each of their centers and a separation of 80. For the equation of state we choose again $\kappa = 123.6489$ and $n = 1$. The stars' centers are located at the z -axis and their velocities are parallel to the x -axis. We construct in initial data for irrotational stars on a

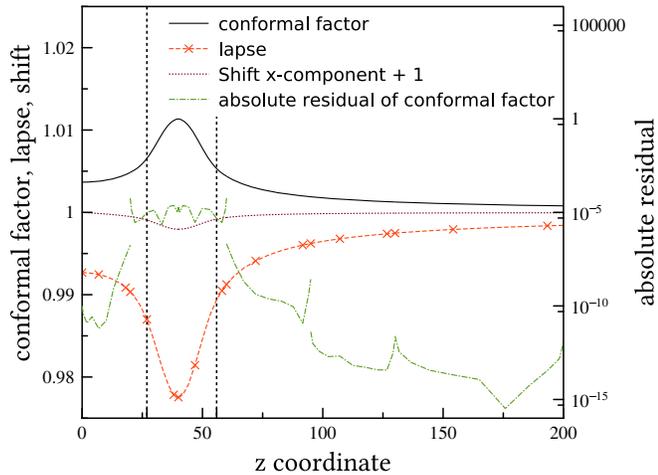


FIG. 10. Steady state at a resolution of 11 collocation points per subpatch for the initial data of a binary neutron star system. We show data along the positive z -axis. The values on the negative axis are symmetric (anti-symmetric for the shift component β^x). Solid line: conformal factor. Orange dashed line with markers: lapse. Purple dotted line: x -component of the shift, shown here with an offset of one for clarity. Green dash-dotted line: absolute value of the residual for the conformal factor, as given by the right-hand side of Eq. (65). Vertical dashed line: position of the stellar surface.

quasicircular orbit. In Fig. 10 we present results for the conformal factor, lapse, the x -component of the shift, the residual of the conformal factor equation. As for the TOV star initial data, the residuals are biggest on those subpatches which contain the stellar surfaces, where the matter fields are not smooth.

Because we are not using surface-fitted coordinates [33–35] we can not place the stellar surface at a subpatch boundary, and thus no high-order convergence in the norm of the residuals can be seen with increasing number of collocation points. Although this may be a disadvantage for studies of initial data *per se*, the situation changes if the goal is evolution of the data. Since in an actual evolution of this data surface-fitted coordinates are normally not retained, the high accuracy of initial data with surface-fitted coordinates will be lost relatively quickly anyway. On the other hand, methods like [33] require expensive iterations to determine the surface fitting coordinates as part of the solution process, so any method which works without special coordinates is more efficient in that part of the algorithm. In fact, part of the motivation for the multigrid method in [32] was to construct a solver which works on a general Cartesian grid without surface-fitting coordinates. The hyperbolic relaxation method achieves the same goal.

VI. CONCLUSIONS

The most common types of relaxation methods are based on the famous Gauss-Seidel method, which can be motivated by rewriting the problem as a parabolic diffusion

equation that relaxes from an initial guess to the solution of the elliptic PDE. In this paper we investigated a new class of relaxation methods, which are not based on parabolic, but rather on hyperbolic PDEs. In the literature hyperbolic relaxation is usually discussed from the point of view that a hyperbolic equation is given which may contain physical or numerical dissipation terms. In this work we assume that an elliptic equation is given, which is extended to a hyperbolic relaxation equation for the purpose of solving the elliptic equation. In some respect, hyperbolic relaxation might actually be as well suited for the solution of elliptic PDEs as parabolic relaxation.

We investigated how a hyperbolic relaxation method (HypRelax) can be constructed for a general class of second order, nonlinear elliptic PDEs and discussed its structure and properties. A discussion of the general hyperbolicity properties has been carried out and three specific choices for the relaxation were discussed. For the special case of the Laplace equation, a mode analysis revealed that there is a critical wave number at which the qualitative behavior of the modes is changing. It has also been seen that at low wave number the damping rate approaches that of the Jacobi method from above. It is an interesting topic for the future how the specific choice of the relaxation system (choice of $b^j_i \beta_\alpha$) affects the behavior of the modes, how that choice can be optimized, and what would be the modes for more general elliptic equations.

Furthermore, we have shown how the standard types of boundary conditions—Dirichlet, Neumann and Robin boundary conditions—can be implemented within the hyperbolic relaxation framework.

With regard to advantages, hyperbolic relaxation shares with several other methods the feature that it is “matrix free,” i.e., it is possible to avoid the construction of an explicit matrix form by applying the differential operator directly. Also note that as in Jacobi methods, nonlinear equations can be treated without linearization, avoiding the additional work of e.g., outer Newton-Raphson iterations.

The ease of implementation is a key feature of hyperbolic relaxation. In the past, we implemented a parallel geometric multigrid method for the BAM and CACTUS codes [32,36–38]. We also interfaced BAM to the HYPRE package [39,40] for access to a variety of elliptic solvers, in particular algebraic multigrid. To avoid complicated dependencies, some of the black hole initial data was later computed with a stand-alone Newton-Raphson method, or even with the direct solver in MATLAB for numerical stability [41]. Our current production runs for a wide range of binary neutron star initial data rely on the sophisticated spectral solver SGRID [33,42,43]. Unsurprisingly, compared to the various approaches just mentioned, hyperbolic relaxation is very straightforward to implement given a hyperbolic evolution code.

We have implemented the hyperbolic relaxation method in our spectral hyperbolic evolution code BAMPs [14,15] (and in a finite differencing test code). For numerical tests

we have applied it to Poisson's equation, and we also presented applications to numerical relativity initial data, where we have seen a high robustness with respect to the choice of an initial guess and regarding the simultaneous solution of the XCTS equations. Given the complexity of binary neutron star initial data and the correspondingly involved numerical methods that have been developed to solve these elliptic equations, e.g., [30] and references therein, it is a nontrivial result that the hyperbolic relaxation method results in a robust and quite efficient elliptic solver.

We have seen that the damping rate at low wave numbers is comparable to that of the Jacobi method and thus very low. In this study we were most interested in the basic properties of the hyperbolic relaxation method and thus we focussed on a simple scheme of successive refinement, but there exist more sophisticated accelerators, in particular multigrid methods, that could be implemented also for a hyperbolic relaxation scheme. Another possible extension of our hyperbolic relaxation implementation would be an adaptive mesh refinement scheme, for which we see two main advantages. First of all, there are the usual savings due to optimized local resolution, and second, since we want to use the solution of the elliptic equation as initial data, this will provide us with a grid that should already be well adapted to the evolution of the obtained data. Another idea

is to employ adaptive time stepping, for which however special care would be required to keep the scheme stable.

With regard to initial data for neutron stars, one of the potential problems is the lack of differentiability at the surface of the stars. We were prepared to evolve the neutron star data with a high-resolution shock-capturing method, but in fact this was not necessary given the strong damping of the equations. In applications where the wave propagation feature is important, shock handling may be a feature that comes at no extra cost assuming that there is an evolution code providing the appropriate methods.

The investigations of this paper can only serve as a first step in the exploration of this potentially promising branch of new relaxation methods. Hyperbolic relaxation might become with further study, and in particular in combination with acceleration methods, the basis of an alternative numerical solution method for elliptic equations.

ACKNOWLEDGMENTS

This work was supported in part by the DFG funded Graduiertenkolleg 1523/2, DFG grant BR 2176/5-1, and the FCT (Portugal) IF Program IF/00577/2015. We thank Niclas Moldenhauer, Tim Dietrich, and Wolfgang Tichy for fruitful discussions on the construction of neutron star initial data.

-
- [1] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. (SIAM Press, Philadelphia, USA, 2003).
 - [2] Y. Saad and H. van der Vorst, *J. Comput. Appl. Math.* **123**, 1 (2000).
 - [3] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3rd ed. (Cambridge University Press, Cambridge, England, 2007).
 - [4] L. Hsiao and T.-P. Liu, *Commun. Math. Phys.* **143**, 599 (1992).
 - [5] K. Nishihara, *J. Differ. Equations* **137**, 384 (1997).
 - [6] J. Wirth, *J. Math. Anal. Appl.* **414**, 666 (2014).
 - [7] R. Hirai, H. Nagakura, H. Okawa, and K. Fujisawa, *Phys. Rev. D* **93**, 083006 (2016).
 - [8] A. Ilyin, Y. Rykov, and S. Zelik, *Physica (Amsterdam)* **376–377D**, 171 (2018).
 - [9] M. Alcubierre, B. Brügmann, P. Diener, M. Koppitz, D. Pollney, E. Seidel, and R. Takahashi, *Phys. Rev. D* **67**, 084023 (2003).
 - [10] J. Balakrishna, G. Daues, E. Seidel, W.-M. Suen, M. Tobias, and E. Wang, *Classical and Quantum Gravity* **13**, L135 (1996).
 - [11] B. Gustafsson, H.-O. Kreiss, and J. Olinger, *Time Dependent Problems and Difference Methods* (Wiley, New York, 1995).
 - [12] O. Sarbach and M. Tiglio, *Living Rev. Relativity* **15**, 9 (2012).
 - [13] D. Hilditch, *Int. J. Mod. Phys. A* **28**, 1340015 (2013).
 - [14] B. Brügmann, *J. Comput. Phys.* **235**, 216 (2013).
 - [15] D. Hilditch, A. Weyhausen, and B. Brügmann, *Phys. Rev. D* **93**, 063006 (2016).
 - [16] S. Dain, *Lect. Notes Phys.* **692**, 117 (2006).
 - [17] H.-O. Kreiss, O. E. Ortiz, and O. A. Reula, *J. Differ. Equations* **142**, 78 (1998).
 - [18] H.-O. Kreiss and J. Lorenz, *Acta Numer.* **7**, 203 (1998).
 - [19] J. S. Hesthaven, *Applied Numerical Mathematics* **33**, 23 (2000).
 - [20] J. S. Hesthaven, S. Gottlieb, and D. Gottlieb, *Spectral Methods for Time-Dependent Problems* (Cambridge University Press, Cambridge, England, 2007).
 - [21] N. W. Taylor, L. E. Kidder, and S. A. Teukolsky, *Phys. Rev. D* **82**, 024037 (2010).
 - [22] C. Gundlach and J. M. Martín-García, *Classical Quantum Gravity* **23**, S387 (2006).
 - [23] C. Ronchi, R. Iacono, and P. Paolucci, *J. Comput. Phys.* **124**, 93 (1996).
 - [24] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, 2nd ed., revised (Dover Publications, New York, 2001).
 - [25] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. (Cambridge University Press, Cambridge, England, 2007).

- [26] R. Barrett, M. Berry, T. Chan, J. Dongarra, V. Eijkhout, C. Romine, and H. van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods* (SIAM, <http://www.netlib.org/templates/>, 1993).
- [27] X. I. Yang and R. Mittal, *J. Comput. Phys.* **274**, 695 (2014).
- [28] J. Adsuara, I. Cordero-Carrin, P. Cerd-Durn, and M. Aloy, *J. Comput. Phys.* **321**, 369 (2016).
- [29] T. W. Baumgarte and S. L. Shapiro, *Numerical Relativity: Solving Einstein's Equations on the Computer* (Cambridge University Press, Cambridge, England, 2010).
- [30] W. Tichy, *Rep. Prog. Phys.* **80**, 026901 (2017).
- [31] N. Moldenhauer, Master Thesis, University of Jena, 2012.
- [32] N. Moldenhauer, C. M. Markakis, N. K. Johnson-McDaniel, W. Tichy, and B. Brügmann, *Phys. Rev. D* **90**, 084043 (2014).
- [33] W. Tichy, *Classical Quantum Gravity* **26**, 175018 (2009).
- [34] S. Bonazzola, E.ourgoulhon, and J.-A. Marck, *Phys. Rev. D* **58**, 104020 (1998).
- [35] M. Ansorg, *Classical Quantum Gravity* **24**, S1 (2007).
- [36] S. Brandt and B. Brügmann, *Phys. Rev. Lett.* **78**, 3606 (1997).
- [37] M. Alcubierre, W. Bengert, B. Brügmann, G. Lanfermann, L. Neger, E. Seidel, and R. Takahashi, *Phys. Rev. Lett.* **87**, 271103 (2001).
- [38] B. Brügmann, J. A. González, M. Hannam, S. Husa, U. Sperhake, and W. Tichy, *Phys. Rev. D* **77**, 024027 (2008).
- [39] Hypre—High Performance Preconditioners: <http://www.llnl.gov/CASC/hypre/>.
- [40] M. Ansorg, B. Brügmann, and W. Tichy, *Phys. Rev. D* **70**, 064011 (2004).
- [41] T. Dietrich and B. Brügmann, *Phys. Rev. D* **89**, 024014 (2014).
- [42] W. Tichy, *Phys. Rev. D* **86**, 064024 (2012).
- [43] T. Dietrich, N. Moldenhauer, N. K. Johnson-McDaniel, S. Bernuzzi, C. M. Markakis, B. Brügmann, and W. Tichy, *Phys. Rev. D* **92**, 124007 (2015).