

Pseudospectral vs finite differences methods in Numerical Relativity

I. Avilés, G. Estrada, J.A. González and F.S. Guzmán

Laboratorio de Inteligencia Artificial y Supercómputo, Instituto de Física y Matemáticas, Universidad, Michoacana de San Nicolás de Hidalgo, Cd. Universitaria, Morelia, Michoacán, México.

Received 18 March 2016; accepted 3 August 2016

Inspired by the recent discovery of Gravitational Waves (GWs) by the LIGO array, we consider necessary to strengthen the formation of our students in Numerical Relativity (NR) in México. A key issue in the future GW astronomy is the efficiency at solving inverse problems, this means, reconstructing the physical parameters of the GW source. In the case of the Binary Black Holes (BBH) inspirals, we find the appropriate example where NR has shown its usefulness, because the waveforms predicted by NR simulations in BBH collisions are used to filter the data in the interferometer runs. The size of a catalog of such numerically generated waveforms is therefore of great importance because the more waveforms it has the easier is to reconstruct the intrinsic parameters of the BBH system. In this way, the purpose of this educative paper is to show the comparison, in terms of performance and accuracy, of the two different numerical methods used to build the catalogs, applied to the evolution of a training problem, namely the evolution of a single black hole. We present a comparison between a pseudospectral and a finite differences method in the solution of numerical general relativity equations. The system we choose to test the performance and accuracy is a spherically symmetric black hole in Eddington-Finkelstein coordinates. For the evolution we use the Einstein-Christoffel formulation of General Relativity. We compare accuracy in the violation of the Hamiltonian constraint and CPU time, both in terms of the spectral and finite differences resolution.

Keywords: Numerical relativity; black holes; spectral methods; finite differences.

PACS: 04.25.Dm; 04.25.dg

1. Introduction

There is no doubt that the recent discovery of Gravitational Waves (GWs) is of historic importance [1, 2], and that it will open a new window for astrophysics [3], mainly because it has confirmed the two main predictions of General Relativity (GR), namely, the existence of GWs and the existence of Black Holes on a strong gravitational field scenario, even though there is still room for other compact objects to source these GWs [4, 5].

What requires some patience and deep knowledge to consolidate this new branch is to acquire the know how on various ingredients of this discovery and future GWs astronomy work. On the one hand there is the knowledge of the mathematics of GR and its different implications on astrophysics, and on the other hand there is the data analysis behind the scene of GW observations.

Specifically, the analysis of the sources of GWs reduces to infer the properties of the source that generates a given observed signal. In the case of the discovery in [1], the system is an orbiting binary black hole system with masses of $29 M_{\odot}$ and $36 M_{\odot}$ within an uncertainty box. This information is obtained through the inversion of the problem, that is, given a signal one has to estimate the parameters of the BHs that better fit the observed GW signal. The parameters characterizing the system are many and of two types, on the one hand the intrinsic parameters associated to the masses and spins of the black holes and possibly the eccentricity of the orbit; on the other hand there are extrinsic parameters related to the location of the source on the sky, including luminosity distance, declination, inclination, polarization and orbital phase of the source [6, 7]. The inversion of the problem is specifically

the reconstruction of all the parameters of the astrophysical problem that generated a given GW signal.

The inversion would be easy if one could carry out as many simulations as desired of the collision of two black holes with any combination of parameters, and then decide which simulation better fits the data. Unfortunately, the simulation of the binary black hole system requires the solution of Einstein's equations, which is a system of PDEs on a chunk of space-time [8, 9], whose solution requires a considerable amount of computational resources. The good news are that after the first long term binary black hole evolutions [10–12], the simulations of a many orbits binary black hole collisions are doable, the bad news are that still the computer power required to explore such an amazingly big parameter space is currently unaffordable. A complete catalog of GW source signals should include many combinations of -at least- the intrinsic parameters of the binary system, namely masses of the black holes, their linear momenta and individual spins of the black holes. These quantities span an at least eight dimensional space [6]. The number of parameters obviously increases the number of simulations and processing time required to construct a complete catalog. For example, consider the parameter space of the masses and spins of the black holes, and for simplicity assume the spins are aligned and parallel to the orbital angular momentum, the number of dimensions of the parameter space would be two, the mass ratio (MR) between the black holes and the spin ratio (SR); considering the mass ratio ranges from 1 to 10 and the spin ratio between -1 and 1; exploring the mass ratio with $\Delta_{MR} = 0.1$ and $\Delta_{SR} = 0.1$ the number of simulations is $100 \times 21 = 2100$ runs. If each of these runs takes one

week using 128 powerful cores using high numerical resolution, it would take about 43 million cpu-hours assuming no technical problems happen and not taking into account the post-processing analysis required to extract the physical quantities. This must lead to imagine the scenario where the spins are not parallel.

A way out to this problem and all the computer power required, is to use waveforms generated by other means, including post-Newtonian expansions or effective one body approaches to the binary problem, avoiding in this way the need to solve directly the full Einstein's equations. These methods are very efficient however they lack of the information during the merger stage of the collision.

What has been done is to produce a standardized catalog [13] of NR simulations with representative combinations of intrinsic parameters that is used to match-filter the observed signals called the Numerical INjection Analysis (NINJA) Project, which was produced using different numerical codes that use various numerical methods and gauge conditions under severe accuracy restrictions [14]. The waveforms in the catalog are combined with and matched to orbital waveforms generated with post-Newtonian approximations [14]. These matched signals are used to filter potential waveforms in the data measured by the interferometers. Later on, these signals are used to approximately estimate parameters of a potential GW source [15]. Finally, as expected, the catalog is growing in size (e.g. [16, 17]).

The educative aspect we bring on, is the accuracy and efficiency of different numerical methods when solving Einstein's equations that later is reflected in the size of a given catalog. Following the standardized approach in the NINJA2 Project [14], we call the attention that among the codes used to submit waveforms, one of them, SpEC [18] uses a multidomain pseudospectral method that shows exponential convergence, whereas the rest of codes used in the collaboration (BAM [19], LazEv [20], LEAN [21], Llama [22], MayaKranc [23] and UIUC [24]) use finite differences methods.

Therefore we consider very educative a comparison between the two main numerical methods used to solve Einstein's equations, namely, pseudospectral and finite differences. We show how to construct and compare the accuracy and efficiency at solving a specific numerical relativity problem. For this we choose to evolve a black hole in spherical symmetry, without matter. The spherical symmetry is easy to implement from the scratch in a simple manner, so that we avoid to use a code from elsewhere that is a black box hiding many computational processes additional to the numerical methods. We do not involve matter because it is not necessary at this stage due to the small interaction between matter and GWs and because it would require the use of different numerical methods. This would be an obstacle to the comparison we want to develop.

The methods we compare are a flavor of Pseudospectral method (PS) used for instance by the SpEC code [18], and the Finite Differences method (FD), commonly used in many robust codes, for instance by the Einstein Toolkit package [25,26]. For the PS we do not use any fancy approach like calculation of the expansion coefficients using FFT, instead, we simply evaluate the derivatives of the functions using the Chebyshev collocation derivative matrix. For the FD method we use a uniform mesh with fourth order stencils.

A fair arena to compare both methods requires an appropriate formulation of General Relativity that allows the implementation of the two methods in a simple way, and that shows well posed boundary and initial conditions. For this we use the KST Einstein-Christoffel formulation of General Relativity, which has all these properties [27].

The paper is organized as follows. In Sec. 2 we present the evolution equations in the Einstein Christoffel (EC) formulation for spherically symmetric space-times and the initial data for a Schwarzschild black hole. In Sec. 3 we describe both, the FD method and the Pseudospectral method used for the comparison. In Sec. 4 we presents the results of our comparison and in Sec. 5 we mention some conclusions.

2. Evolution equations

2.1. Einstein-Christoffel system of equations in spherical symmetry

The idea is to solve the Einstein Field equations for a Schwarzschild Black Hole, which is a vacuum solution $G_{\mu\nu} = 0$, where $G_{\mu\nu}$ is the Einstein tensor. For this we assume the standard 3+1 decomposition metric in spherical symmetry (see for example [8, 9])

$$ds^2 = -\alpha^2 dt^2 + g_{rr}(dr + \beta^r dt)^2 + g_T r^2 (d\theta^2 + \sin^2 \theta d\phi^2) \quad (1)$$

where (t, r, θ, ϕ) are the time and spherical coordinates, with

$$g_T = \frac{g_{\theta\theta}}{r^2} = \frac{g_{\phi\phi}}{r^2 \sin^2 \theta}.$$

So far, the involved functions, the 3-metric of the spatial slices g_{ij} , the lapse function α and the shift β^r are assumed to depend on (t, r) only.

In the EC formulation, in spherically symmetric space-times, Einstein's equations in vacuum reduce to the following system of first order in time evolution equations, where a densitized lapse $\tilde{\alpha} = \alpha / (g_T \sqrt{g_{rr}})$ is used [27]

$$\begin{aligned} \partial_t g_{rr} &= \beta^r \partial_r g_{rr} - 2\alpha K_{rr} + 2g_{rr} \partial_r \beta^r, \\ \partial_t g_T &= \beta^r \partial_r g_T - 2\alpha K_T + 2 \frac{\beta^r}{r} g_T, \end{aligned}$$

$$\begin{aligned}
 \partial_t K_{rr} &= \beta^r \partial_r K_{rr} - \frac{\alpha}{g_{rr}} \partial_r f_{rrr} + \alpha \left[2 \frac{f_{rrr}}{g_{rr}} \left(\frac{f_{rrr}}{g_{rr}} + \frac{1}{r} - \frac{4f_{rT}}{g_T} \right) - \frac{6}{r^2} + K_{rr} \left(2 \frac{K_T}{g_T} - \frac{K_{rr}}{g_{rr}} \right) \right. \\
 &\quad \left. - 6 \left(\frac{f_{rT}}{g_T} \right)^2 - \partial_r^2 \ln \tilde{\alpha} - (\partial_r \ln \tilde{\alpha})^2 + \left(\frac{4}{r} - \frac{f_{rrr}}{g_{rr}} \right) \partial_r \ln \tilde{\alpha} \right] + 2K_{rr} \partial_r \beta^r, \\
 \partial_t K_T &= \beta^r \partial_r K_T - \frac{\alpha}{g_{rr}} \partial_r f_{rT} + \alpha \left(K_T \frac{K_{rr}}{g_{rr}} + \frac{1}{r^2} - \frac{2f_{rT}^2}{g_{rr}g_T} - \frac{f_{rT}}{g_{rr}} \partial_r \ln \tilde{\alpha} \right) + \frac{2\beta^r}{r} K_T, \\
 \partial_t f_{rrr} &= \beta^r \partial_r f_{rrr} - \alpha \partial_r K_{rr} + \alpha \left[4g_{rr} \frac{K_T}{g_T} \left(3 \frac{f_{rT}}{g_T} - \frac{f_{rrr}}{g_{rr}} + \frac{2}{r} - \partial_r \ln \tilde{\alpha} \right) \right. \\
 &\quad \left. - K_{rr} \left(10 \frac{f_{rT}}{g_T} + \frac{f_{rrr}}{g_{rr}} - \frac{2}{r} + \partial_r \ln \tilde{\alpha} \right) \right] + 3f_{rrr} \partial_r \beta^r + g_{rr} \partial_r^2 \beta^r, \\
 \partial_t f_{rT} &= \beta^r \partial_r f_{rT} - \alpha \partial_r K_T + \alpha \left[K_T \left(2 \frac{f_{rT}}{g_T} - \frac{f_{rrr}}{g_{rr}} - \partial_r \ln \tilde{\alpha} \right) \right] + \left(\partial_r \beta^r + \frac{2\beta^r}{r} \right) f_{rT}. \tag{2}
 \end{aligned}$$

This is a system that determines the evolution of the extrinsic curvature components of the spatial hypersurfaces K_{rr} and $K_T = K_{\theta\theta}/r^2 = K_{\phi\phi}/(r^2 \sin^2 \theta)$, which are the radial and transverse components of the extrinsic curvature, whereas f_{rrr} and $f_{rT} = f_{r\theta\theta}/r^2 = f_{r\phi\phi}/(r^2 \sin^2 \theta)$ are auxiliary variables defined by [27]

$$f_{kij} = \Gamma_{(ij)k} + g_{ki} g^{lm} \Gamma_{[lj]m} + g_{kj} g^{lm} \Gamma_{[li]m}.$$

This system of equations is attached to a set of four constraints, namely the usual Hamiltonian and Momentum Constraints in vacuum and two new constraints resulting from the definition of the auxiliary variables particular of the EC formulation

$$\begin{aligned}
 \mathcal{C} &:= \frac{\partial_r f_{rT}}{g_{rr}g_T} - \frac{1}{2r^2 g_T} + \frac{f_{rT}}{g_{rr}g_T} \left(\frac{2}{r} + \frac{7f_{rT}}{2g_T} - \frac{f_{rrr}}{g_{rr}} \right) \\
 &\quad - \frac{K_T}{g_T} \left(\frac{K_{rr}}{g_{rr}} + \frac{K_T}{2g_T} \right) = 0
 \end{aligned}$$

$$\mathcal{C}_r := \frac{\partial_r K_T}{g_T} + \frac{2K_T}{rg_T} - \frac{f_{rT}}{g_T} \left(\frac{K_{rr}}{g_{rr}} + \frac{K_T}{g_T} \right) = 0$$

$$\mathcal{C}_{rrr} := \partial_r g_{rr} + \frac{8g_{rr}f_{rT}}{g_T} - 2f_{rrr} = 0$$

$$\mathcal{C}_{rT} := \partial_r g_T + \frac{2g_T}{r} - 2f_{rT} = 0$$

all of which have to hold during the evolution. In the original ADM 3+1 formulation there are only the Hamiltonian and Momentum constraints, and the only variables evolving in time are the components of the induced 3-metric and the components of the extrinsic curvature of the spatial hypersurfaces. When new formulations appeared, like the BSSN and the EC, new variables were promoted as dynamical variables with their own evolution equations. The motivation of the new formulations was to improve the properties of the evolution system in terms of its characteristic structure, which is fundamental to define a well posed initial value problem [8,9]. Every time a new variable is defined a new

constraint appears in the system of evolution equations, and this is the meaning of the last two equations above.

These constraints are not being solved during the evolution and in practice the set of constraints of an evolution system is satisfied only up to numerical errors. Therefore, the way to know whether one is solving Einstein's equations is to monitor the violation of the constraints and check if it is small and converges to zero when improving errors of the numerical methods.

2.2. Initial data

We start the evolution of a Schwarzschild black hole using Eddington-Finkelstein coordinates. It is well known that standard Schwarzschild coordinates are singular at the event horizon, which makes all the variables divergent at that surface and makes impossible any numerical calculation there. The family of Eddington-Finkelstein foliations has the property that the spatial hypersurfaces of the space-time penetrate the horizon. In this description the equations are all regular except at the singularity located at the origin of coordinates. The use of this description allows one to solve the equations inside the event horizon as long as the singularity is not included in the numerical domain. This can be done using a technique called excision [28], which consists on defining the numerical domain for $r \in [r_{\min}, r_{\max}]$ with $0 < r_{\min} < r_{EH}$, where r_{\min} is an internal boundary, $r_{EH} = 2M$ is the radius of the event horizon and r_{\max} is a radius far away from the black hole. Explicitly, the metric functions, curvature components and f variables in these coordinates are

$$g_{rr} = 1 + \frac{2M}{r}$$

$$g_T = 1$$

$$\begin{aligned} \tilde{\alpha} &= \left(1 + \frac{2M}{r}\right)^{-1} \\ \beta^r &= \frac{2M}{r} \left(1 + \frac{2M}{r}\right)^{-1} \\ K_{rr} &= -\frac{2M}{r} \left(1 + \frac{M}{r}\right) \left(1 + \frac{2M}{r}\right)^{-1/2} \\ K_T &= \frac{2M}{r^2} \left(1 + \frac{2M}{r}\right)^{-1/2} \\ f_{rrr} &= \frac{1}{r} \left(4 + \frac{7M}{r}\right) \\ f_{rT} &= \frac{1}{r} \end{aligned}$$

where M is the mass of the black hole. These expressions satisfy identically the four constraints at initial time. With the evolution equations and initial conditions one only needs to start up the evolution of these initial data. Given these constraints are not being solved during the evolution, they start being violated due to the numerical errors of each method. It is a common practice that during the numerical evolution of a space-time the constraints have to be monitored in order to verify that the system remains near the constraint surface and converges in the continuum limit to no violation.

3. Numerical methods

In this section we explain the Pseudospectral and Finite Differences methods. In both cases we use a time and space discretization. The spatial discretization depends on the number of points used to describe the spatial coordinate. In both methods, the number of points used to discretize, defines a given resolution. As the number of points tends to infinity we say that we increase the resolution and expect the numerical solution to approach the solution of the continuum equations.

The time discretization is defined by a time resolution Δt . We use the method of lines to evolve the numerical solution along the discretized spatial domain. The method of lines is well described in [29] and here we describe the methods used for the spatial discretization.

3.1. Pseudospectral method

We use the pseudospectral Chebyshev method that we briefly describe here for clarity. This method assumes a smooth function $u(x)$ in the domain $x \in [-1, 1]$. The Chebyshev-Gauss-Lobatto (CGL) collocation points are set to the positions

$$x_i = \cos \frac{\pi i}{N} \quad i = 0, \dots, N,$$

which are the extrema of the N th order Chebyshev polynomial

$$T_N(x) = \cos(N \cos^{-1}(x)).$$

The function $u(x)$ is interpolated by a polynomial $P(x)$ of degree $\leq N$,

$$P(x) = \sum_{j=0}^N u_j L_j(x),$$

where $u_j = u(x_j)$ and L_j is the polynomial of degree N such that $L_j(x_k) = \delta_{jk}$. Therefore, in the CGL points we have

$$P_i = P(x_i) = \sum_{j=0}^N u_j L_j(x_i) = \sum_{j=0}^N u_j \delta_{ji} = u_i.$$

It can be shown (see [30]) that

$$L_j(x) = \frac{(-1)^{j+1}(1-x^2)T'_N(x)}{c_j N^2(x-x_j)}, \quad j = 0, \dots, N,$$

where T'_N is the derivative of a Chebyshev polynomial and

$$c_j = \begin{cases} 2, & j = 0, N, \\ 1, & j = 1, \dots, N-1. \end{cases}$$

The derivative of $u(x)$ at the CGL points can be approximated by

$$\frac{\partial u(x_i)}{\partial x} \approx \frac{\partial P_i}{\partial x} = \sum_{j=0}^N u_j L'_j(x_i) = \sum_{j=0}^N u_j (D_N)_{ij},$$

where D_N is the Chebyshev collocation derivative matrix with $(D_N)_{ij} = L'_j(x_i)$. The entries of D_N are

$$\begin{aligned} (D_N)_{ij} &= \frac{c_i(-1)^{i+j}}{c_j(x_i-x_j)}, \quad i \neq j, \\ (D_N)_{jj} &= \frac{-x_j}{2(1-x_j^2)} \quad j \neq 0, N, \\ (D_N)_{00} &= -(D_N)_{NN} = \frac{2N^2+1}{6}. \end{aligned}$$

In order to use Chebyshev polynomials, P_i is defined for $x \in [-1, 1]$. Hence for our radial coordinate on the arbitrary interval $r \in [r_{\min}, r_{\max}]$ we use the following application

$$\begin{aligned} r &: [-1, 1] \rightarrow [r_{\min}, r_{\max}] \\ r(x) &= \frac{r_{\max} - r_{\min}}{2}(x-1) + r_{\max}, \end{aligned}$$

that maps the collocation coordinate x onto the physical coordinate r . One can substitute this mapping into the differential operator and find for a given function u that spatial derivatives satisfy

$$\frac{\partial u(r)}{\partial r} = \frac{2}{r_{\max} - r_{\min}} \frac{\partial u(x)}{\partial x}.$$

This method is used to decompose each of the evolution variables in (2) and at each collocation point we integrate in time using the Method of Lines with a 4th order Runge-Kutta integrator.

3.2. Finite differences method

For the FD implementation we define a uniform grid with resolution $\Delta r = (r_{\max} - r_{\min})/N_r$ with grid points at $r_i = r_{\min} + i\Delta r$. The values of r_{\min} and r_{\max} indicate the location of the inner and outer boundaries of the numerical domain. An educative paper on finite differences can be found in [29]. The discretization tends to the continuum limit when the resolutions Δt and Δr tend to zero.

The system of evolution equations (2) is first order in time. The FD implementation uses standard 4th order accurate stencils, which requires both, centered and up and downwind election of neighboring grid points. For completeness we rewrite the first derivative of a given function u_i considering the possible combinations of neighboring points consistent with fourth order accuracy

$$\begin{aligned}
 \partial_r u_i &= C[3u_{i-4} - 16u_{i-3} + 36u_{i-2} \\
 &\quad - 48u_{i-1} + 25u_i] + \mathcal{O}(\Delta r^4) \\
 \partial_r u_i &= C[-u_{i-3} + 6u_{i-2} - 18u_{i-1} \\
 &\quad + 10u_i + 3u_{i+1}] + \mathcal{O}(\Delta r^4) \\
 \partial_r u_i &= C[u_{i-2} - 8u_{i-1} + 8u_{i+1} - u_{i+2}] + \mathcal{O}(\Delta r^4) \\
 \partial_r u_i &= C[-3u_{i-1} - 10u_i + 18u_{i+1} \\
 &\quad - 6u_{i+2} + u_{i+3}] + \mathcal{O}(\Delta r^4) \\
 \partial_r u_i &= C[-25u_i + 48u_{i+1} - 36u_{i+2} \\
 &\quad + 16u_{i+3} - 3u_{i+4}] + \mathcal{O}(\Delta r^4)
 \end{aligned} \tag{3}$$

where $C = 1/12\Delta r$. We have to be this explicit because the advection terms of the type $\beta_i \partial_r u$ in (2) may affect the causality of the evolution algorithm. In order to causally reconnect we use stencils with neighbors to the right of r_i when $\beta_i^r > 0$ and neighbors to the left when $\beta_i^r < 0$ for all the points r_i of the grid far from the boundaries. At the boundaries we simply use appropriately unbalanced stencils from (3) to calculate accurately the spatial derivatives.

As mentioned before, the time integration is performed using the Method of Lines on each grid point, using a 4th order accurate Runge-Kutta method like for the PS method.

3.3. Boundary conditions

One of the most important aspects of numerical relativity is that the final result is the construction of only a small piece of space-time, no matter whether is a system of two orbiting black holes or a supernova core-collapse. The result is limited both in space and time. It is limited in time because the simulations take a finite time and limited in space, because generally uses a finite spatial domain, defining an external artificial boundary. The binary black hole simulations are usually carried out in cubical domains in full 3D, and when excision is used there are two other internal excision boundaries, one inside each of the two black holes.

In the present problem with spherical symmetry, there are two boundaries, the external boundary is a sphere, which in our coordinates is located at the point $r = r_{\max}$ and the inner boundary, also a sphere located at $r = r_{\min}$.

We use the same boundary conditions for both methods. At the inner boundary there is no need to implement any boundary conditions because all the characteristic fields get off the domain through r_{\min} in the direction toward the singularity. This means that the light cones point toward the singularity at $r = 0$ and therefore, all the fields propagate with world-lines within the light cones.

Following the EC formulation of Einstein's equations in [27], for spherically symmetric space-times the radial characteristic modes can easily be identified for the variables in (2). Therefore, at the outer boundary in r_{\max} , the fields have, tangential and radial ingoing and outgoing modes. What is usually done in black hole evolutions is to apply outgoing boundary conditions and so we do here. There are four fields propagating at r_{\max} inwards, which eventually may contaminate the numerical solution. Two of these fields are $U_r^0 = g_{rr}$ and $U_t^0 = g_T$ moving with speed $-\beta^r$. Two other fields moving with speed $-\beta^r \pm \tilde{\alpha}g_T$ are

$$\begin{aligned}
 U_r^\pm &:= K_{rr} \pm \frac{f_{rrr}}{\sqrt{g_{rr}}}, \\
 U_T^\pm &:= K_T \pm \frac{f_{rT}}{\sqrt{g_{rr}}}.
 \end{aligned}$$

The boundary condition we impose are the freezing conditions, that is $\partial_t U = 0$ for the incoming part of these fields, namely U_r^0, U_t^0, U_r^- and U_T^- . These boundary conditions allow the evolution of the space-time for hundreds of crossing times.

4. The comparison

We now set parameters commonly used in real simulations in terms of domain, accuracy estimates and performance.

Spatial domain. We describe the black hole using horizon penetrating coordinates and use excision. We set the excision boundary at $r = 1.9 M$ near but inside the event horizon [28]. We run the simulations in two different domains a small one with $r \in [1.9, 11.9] M$ and a big one with $r \in [1.9, 101.9] M$, in order to estimate the accuracy in two different scenarios.

For the FD method the mesh is uniform in the whole domain, whereas the colocation points in the PS method are distributed in a non uniform way.

Numerical calculations always involve errors, in numerical relativity one monitors the violations of the constraints due to numerical errors. The most important source of numerical errors is due to numerical methods. In the case of FD it is the discretization error, which depends on the accuracy of the approximation and in the PS method it is the truncation error depending on the number of basis elements used to define derivatives.

Specifically the number of collocation points for the PS method and the resolution for FD are the following:

- For the PS method we use a different number of collocation points for each of the two spatial domains used in order to lie within the convergence regime. For the small domain $r \in [1.9, 11.9] M$ we use $N = 10, 14, 20, 24, 28, 32, 36, 40, 45, 48$. For the big domain $r \in [1.9, 101.9] M$ we use $N = 50, 60, 70, 80, 90, 100, 120, 140$.
- For the FD simulations we use the following resolutions $\Delta r_1 = 0.1$, $\Delta r_2 = \Delta r_1/2$, $\Delta r_3 = \Delta r_2/2$, $\Delta r_4 = \Delta r_3/2$, $\Delta r_5 = \Delta r_4/2$, $\Delta r_6 = \Delta r_5/2$ and $\Delta r_7 = \Delta r_6/2$. We apply these resolutions for the two different spatial domains $r \in [1.9, 11.9] M$ and $r \in [1.9, 101.9] M$.

Time stepping. For the PS method, given the collocation points are not equally spaced in the domain, we choose the time stepping guaranteeing stability of time integration, limited by the two closest collocation points separated by the distance $\sim \pi^2/N^2$. Thus we set the time step for the PS method to $\Delta t = 0.25\pi^2/N^2$.

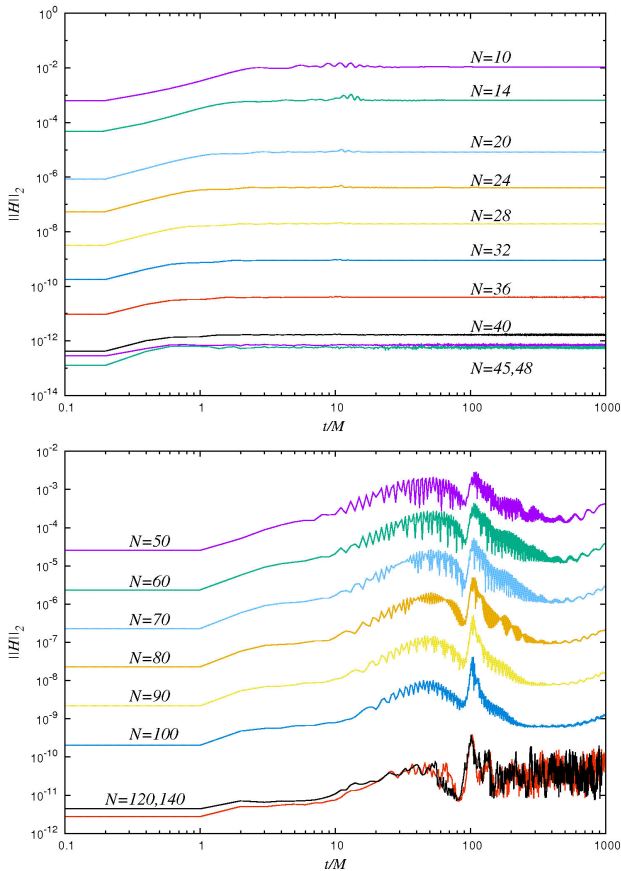


FIGURE 1. L_2 norm of the Hamiltonian constraint as a function of time using the PS method for a different number of collocation points N . On the top we show the results for the spatial domain $r \in [1.9, 11.9] M$ and on the bottom we show the results for $r \in [1.9, 101.9] M$.

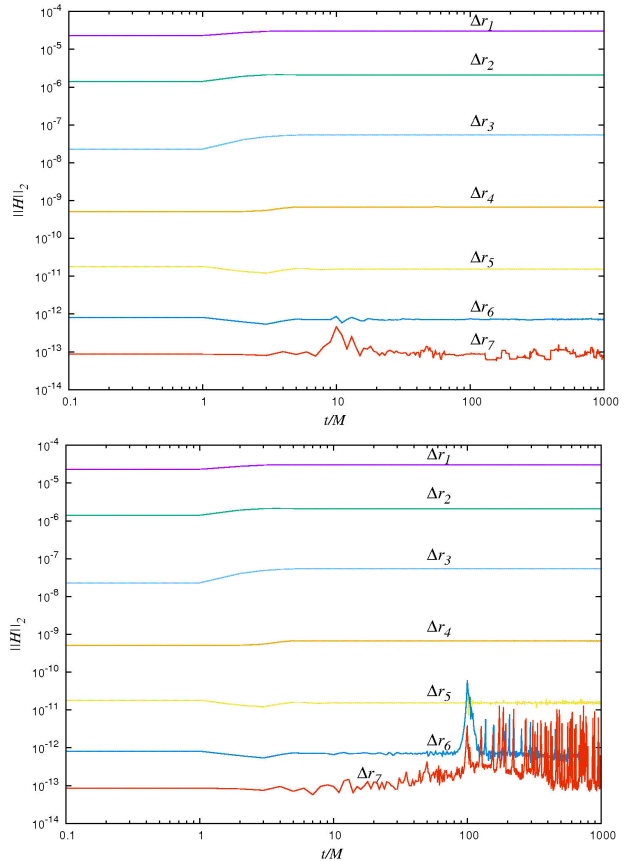


FIGURE 2. L_2 norm of the Hamiltonian constraint as a function of time using FD for seven different resolutions: $\Delta r_1, \dots, \Delta r_7$. The same values of Δr are used for both spatial domains $r \in [1.9, 11.9] M$ (top panel) and $r \in [1.9, 101.9] M$ (bottom panel).

In the FD method, given the mesh is uniform, we simply set the time step using the CFL relation $\Delta t = \lambda \Delta r$. We choose $\lambda = 0.25$ in all the runs.

Calculation of the Hamiltonian constraint. In order to verify that a numerical solution corresponds to the solution in the continuum we need to carry out a series of simulations with different resolutions and measure the resulting errors. For the sake of having meaningful simulations we choose resolutions within the convergent regime for both methods (for a detailed explanation of convergence see [29]). We check the convergence for various resolutions until we find a resolution for which the L_2 norm of the Hamiltonian constraint is of the order of round-off error precision and the convergence is lost.

The accuracy is compared using the violation of the Hamiltonian constraint and its convergence. In order to have a comparison between the two methods, the PS method calculates the L_2 norm of the Hamiltonian constraint on a FD mesh. We do this because it would be unfair to estimate the accuracy only measured at the collocation points.

In Figs. 1 and 2 we show the L_2 norm of the Hamiltonian constraint violation and convergence using the PS and FD methods respectively, for the two different spatial domains. Strictly this analysis has to be done for all the constraints,

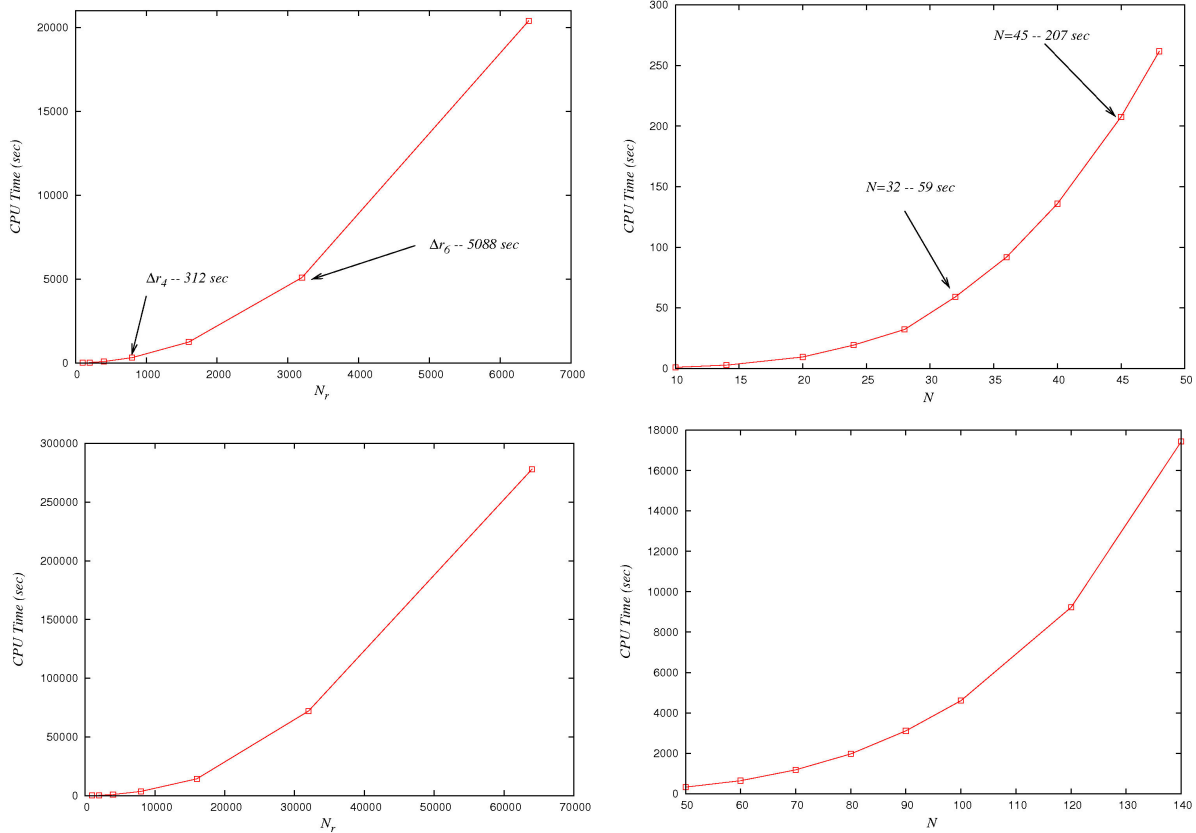


FIGURE 3. CPU time using the two numerical methods for a run on the domain $r \in [1.9, 11.9] M$ (top) and $[1.9, 101.9] M$ (bottom) during a time $t \in [0, 1000] M$. For the two methods we use the same resolutions used for the accuracy. We can appreciate the different scaling of the two methods. The plots on the left correspond to FD and the plots on the right correspond to PS.

nevertheless the common practice is to show only the Hamiltonian. The parameters used for the simulations lie within the convergence regime and when the resolution is increased, until the two methods achieve the round-off error accuracy and no longer converge.

In Fig. 1 for the small domain, the numerical solution is convergent for $N = 10$ until $N = 40$, and for $N = 45, 48$ it does not converge anymore. This is not a bad sign, on the contrary, this behavior of the error indicates that the method has achieved the best accuracy, only limited by the round-off error of the computer. The same effect can be seen in the case of the big domain for $N = 120, 140$.

In Fig. 2 we show the convergence for the FD method. Also the round-off error is achieved with resolution Δr_7 , which is the same for both the small and big domains. These two Figures show that both methods achieve the best possible accuracy for this problem.

Computing time. The advantage of programming our codes from the scratch is that it is easy to disable all the I/O and diagnostics routines during the evolution, something that cannot be done (at least straightforwardly) in a massive code like ETK or SpEC. With this advantage we thus switch off I/O and diagnostics so that we estimate the CPU time during the evolution.

The test for this comparison is done with the small domain with $r \in [1.9, 11.9]$. We run simulations during $t \in [0, 1000]$, which is equivalent to about 100 crossing times of signals moving at the speed of light. The two codes use the same compiler and optimization flags, and we make the comparison on the same computer. The results in calculation time are shown in Fig. 3 for the two methods. The behavior is non-linear in both cases. A fair comparison between the methods is to choose the resolution corresponding to similar accuracy and then compare the CPU time.

Case of the small domain. Let us compare one particular case for PS with $N = 32$ and FD with Δr_4 showing similar accuracy with L_2 norm of the violation of the constraint of the order of 10^{-9} as shown in Figs. 1 and 2. The CPU time using the PS method takes 59 seconds, whereas using FD with it takes 312 seconds, that is six times slower than the PS method.

Going to the first resolution achieving round-off error precision, another interesting particular case is that of PS with $N = 45$ and FD with Δr_6 . The CPU using PS is 207 seconds whereas using FD it takes 5088 seconds, which is 25 times slower than the PS method, see Fig. 3.

Case of the big domain. Unlike in the small domain, an accuracy of 10^{-9} is achieved with $N = 100$ which is

only about three times the number of collocation points for the small domain. On the other hand, using FD the accuracy is achieved with Δr_4 again, however this time, given the domain is ten times bigger, the number of points required to achieve this accuracy is ten times bigger as well.

5. Discussion and conclusions

We have compared the performance of two essentially different numerical methods, the PS and the FD methods to evolve the space-time of a black hole. The comparison has involved the accuracy and CPU time during the evolution of the space-time.

We did not introduce any type of matter in order to avoid ingredients requiring the mixture of different numerical methods. We also implemented one code for each method from the scratch in order to avoid any fancy extra complication that could blur the performance of any of the methods.

When using a small domain, the results indicate that a given accuracy with the PS method can be achieved and a 100 crossing times simulation would be faster than a simulation using FD, and the more accuracy required the faster PS is with respect to FD.

The accuracy becomes an issue for the PS methods when using a considerably big domain. Generally, gravitational wave extraction is measured by numerical detectors located far away from the source in a practically flat region of the space-time (see *e.g.* [9]), and therefore, the matter of accu-

racy may be an issue for PS methods. A way out to this problem with PD methods would be the use of domain decomposition techniques (*e.g.* see [18]). The way around with FD methods on the other hand, is the use of mesh refinement in the regions far from the source which also has an impact on the accuracy at the location of the detectors, however not in the region where the sources (*e.g.* black holes) lie.

The knowledge acquired is essential when deciding on the methods to be used in the construction of a code designed to solve Einstein's equations from the scratch. This must take into account that solving a problem in three spatial dimensions increases the number of equations, variables, constraints and therefore the computing performance is important. Under more complicated scenarios, for instance including magnetohydrodynamics, will also require the solution of the evolution equations for a plasma and not every numerical method will work.

Currently, in order to optimize the exploration of the parameter space, new methods involving the use of artificial intelligence techniques are being developed in our group at the moment for the binary black hole problem [31].

Acknowledgments

This research is supported by grants CIC-UMSNH 4.9 and 4.23, CONACyT 258726 (Fondo Sectorial de Investigación para la Educación). The simulations were carried out in the IFM Draco cluster funded by CONACyT 106466.

-
1. B.P. Abbott, *et al.*, *Phys. Rev. Lett* **116** (2016) 061102.
 2. B.P. Abbott, *et al.*, *Phys. Rev. Lett* **116** (2016) 241103.
 3. B.P. Abbott *et al.*, *ApJL* **818** (2016) L22.
 4. V. Cardoso, E. Franzin, P. Pani, *Phys. Rev. Lett.* **116** (2016) 171101.
 5. C. Chirenti and L. Rezzolla, arXiv: 1602.08759
 6. J. Veitch *et al.*, *Phys. Rev. D* **91** (2015) 042003.
 7. B.P. Abbott *et al.*, *Phys. Rev. Lett.* **116** (2016) 241102.
 8. M. Alcubierre, *Introduction to 3+1 Numerical Relativity*, (Oxford University press 2008).
 9. T.W. Baumgarte and S.L. Shapiro, *Numerical Relativity: solving Einstein's equations on the computer*. (Cambridge 2010).
 10. F. Pretorius, *Phys. Rev. Lett.* **95** (2005) 121101
 11. M. Campanelli, C.O. Lousto, P. Marronetti and Y. Zlochower, *Phys. Rev. Lett.* **96** (2006) 111101.
 12. J.G. Baker, J. Centrella, D.-I. Choi, M. Koppitz and J. van Meter, *Phys. Rev. Lett.* **96** (2006) 111102.
 13. <http://www.black-hole.org>
 14. P. Ajith *et al.* *Class. Quantum Grav.* **29** (2012) 124001.
 15. J. Aasi, *et al.* *Phys. Rev. D* **88** (2013) 062001.
 16. I. Hinder *et al.*, *Class. Quantum Grav.* **31** (2014) 025012.
 17. K. Jani *et al.*, arXiv:1605.03204
 18. M. Boyle, *et al.*, *Phys. Rev. D* **76** (2007) 124038. M.A. Scheel, *et al.*, *Phys. Rev. D* **74** (2006) 104006. M.A. Scheel, *et al.* *Phys. Rev. D* **79** (2009) 024003.
 19. B. Bruegmann *et al.*, *Phys. Rev. D* **77** (2008) 024027.
 20. M. Campanelli, C.O. Lousto, P. Marronetti and Y. Zlochower, *Phys. Rev. Lett.* **96** (2006) 111101.
 21. U. Sperhake, *Phys. Rev. D* **76** (2007) 104015.
 22. C. Reisswig, N.T. Bishop, D. Pollney and B. Szilagyi, *Class. Quantum Grav.* **27** (2010) 075014.
 23. I. Hinder, B. Vaishnav, F. Herrmann, D. Shoemaker and P. Laguna, *Phys. Rev. D* **77** (2008) 081502.
 24. Z.B. Etienne, Y.T. Liu, S.L. Shapiro and T.W. Baumgarte, *Phys. Rev. D* **79** (2009) 044024.
 25. Einstein toolkit, <http://www.einsteintoolkit.org>
 26. F. Löffler *et al.*, *Classical and Quantum Gravity* **29** (2012) 115001.
 27. L. Kidder, M. Scheel, S.A. Teukolsky, E.D. Carlson and G.B. Cook, *Phys. Rev. D* **62** (2000) 084032.
 28. E. Seidel, W-M Suen, *Phys. Rev. Lett.* **69** (1845) 2992.
 29. F.S. Guzmán, *Rev. Mex. Fis. E* **56** (2010) 51-68.
 30. J.P. Boyd, *Chebyshev and Fourier Spectral Methods*, Dover, 2nd edition, 2000.
 31. M. Carrillo, M. Gracia, J.A. González, F.S. Guzmán, *Gen. Rel. Grav.* submitted.