

PAPER

NRPyCritCol & SFcollapse1D: an open-source, user-friendly toolkit to study critical phenomena

To cite this article: Leonardo R Werneck *et al* 2021 *Class. Quantum Grav.* **38** 245005

View the [article online](#) for updates and enhancements.

You may also like

- [Critical phenomena of charged Einstein–Gauss–Bonnet black holes with charged scalar hair](#)
Yves Brihaye and Betti Hartmann
- [Fabrication and Characterization of TeO₂-ZnO Rib Waveguides](#)
Vanessa D. Del Cacho, Acacio L. Siarkowski, Nilton I. Morimoto *et al.*
- [Action of the gravitational field on the dynamical Casimir effect](#)
L C Céleri, F Pascoal and M H Y Moussa








IOP | ebooks™

Bringing together innovative digital publishing with leading authors from the global scientific community.

Start exploring the collection—download the first chapter of every title for free.

NRPyCritCol & SFcollapse1D: an open-source, user-friendly toolkit to study critical phenomena

Leonardo R Werneck^{1,2,3,*} , Zachariah B Etienne^{1,2,4} ,
Elcio Abdalla³ , Bertha Cuadros-Melgar⁵  and
CE Pellicer⁶ 

¹ Department of Physics and Astronomy, West Virginia University, Morgantown, WV 26506, United States of America

² Center for Gravitational Waves and Cosmology, West Virginia University, Chestnut Ridge Research Building, Morgantown, WV 26505, United States of America

³ Instituto de Física, Universidade de São Paulo, Caixa Postal 66318, CEP 05314-970, São Paulo, SP, Brazil

⁴ Department of Physics, University of Idaho, Moscow, ID 83843, United States of America

⁵ Escola de Engenharia de Lorena, Universidade de São Paulo, Estrada Municipal do Campinho S/N, CEP 12602-810, Lorena, SP, Brazil

⁶ Escola de Ciência e Tecnologia, Universidade Federal do Rio Grande do Norte, Campus Universitário Lagoa Nova, CEP 59078-970, Natal, RN, Brazil

E-mail: wernecklr@gmail.com

Received 15 June 2021, revised 1 October 2021

Accepted for publication 27 October 2021

Published 25 November 2021



CrossMark

Abstract

We present a new open-source, user-friendly toolkit of two codes—`SFcollapse1D` and `NRPyCritCol`—to study critical phenomena in the context of gravitational collapse. `SFcollapse1D` is a C/C++ tool designed to solve the problem of gravitational collapse of massless, spherically symmetric scalar fields with the ADM formalism in spherical-like coordinates. `NRPyCritCol` is a collection of Python modules that leverage the NRPy+ infrastructure to generate a highly optimized C-code for evolving massless scalar fields within a covariant BSSN formalism. The toolkit was developed with user-friendliness and code efficiency in mind, enabling the exploration of critical phenomena with consumer-grade computers. We present a study of critical phenomena from the collapse of massless scalar fields in spherical symmetry, using only these two codes and a laptop computer.

* Author to whom any correspondence should be addressed.

Keywords: critical phenomena, gravitational collapse, massless scalar field, numerical relativity

 Supplementary material for this article is available [online](#)

(Some figures may appear in colour only in the online journal)

1. Introduction

In 1993, Choptuik published a seminal paper announcing the discovery of critical phenomena in his studies of a massless scalar field propagating in a dynamical spacetime [1]. He found a one-parameter family of solutions of the Einstein–Klein–Gordon (EKG) equations defined by the parameter η , which for example could represent the amplitude of the initial scalar field. He demonstrated that there is a critical value for this parameter, η_* , such that solutions with $\eta > \eta_*$ were comprised of spacetimes in which the scalar field collapsed and formed a black hole, while solutions with $\eta < \eta_*$ were characterized by full dispersion of the scalar field, resulting in flat space as the end state.

By performing several *supercritical* runs (i.e. with $\eta > \eta_*$), Choptuik conjectured that the masses of the black holes, M_{BH} , which were formed when $\eta \gtrsim \eta_*$ followed the power law

$$M_{\text{BH}} \propto |\eta - \eta_*|^\gamma, \quad (1)$$

where the critical exponent γ is *universal*. The sense in which the critical exponent is said to be universal is the following: given a particular matter source, in this case a massless scalar field, the numerical value of the critical exponent is the same for all possible initial conditions. In his paper Choptuik performed runs with four different initial data families and they all resulted in the same numerical value $\gamma \approx 0.37$.

Furthermore, as we improve the fine-tuning of the parameter η and get closer to the critical solution, it is possible to observe a discrete self-similarity in the solution, which repeats itself at ever-decreasing spatiotemporal scales. Choptuik found that given a value of the logarithmic proper time ξ (to be defined below), the same profile of the scalar field is observed at $\xi + \Delta$, with $\Delta = 3.43$, but on a spatial scale which is $e^\Delta \approx 30$ times smaller than the first one. This phenomenon is known as (scale-)echoing and Δ is the echoing period, which is also universal.

After Choptuik's discovery, many other studies were performed considering, for example, different matter sources and spacetime dimensions (see e.g. [2, 3] for excellent reviews). In [4, 5], perturbation theory is used to show that the universal exponent is $\gamma \approx 0.374$, while the echoing period has been shown to be $\Delta = 3.445452402(3)$ using semi-analytical methods in [6]. Recently, the existence of the critical solution (often referred to as the Choptuik spacetime) was proven in [7], also using semi-analytical methods.

The power law behavior discovered by Choptuik has also been generalized in [4, 8] and now reads

$$\ln M_{\text{BH}} = C + \gamma x + f(x), \quad (2)$$

where C is a constant which depends on the particular initial data family and $f(x)$ is a function which is periodic in $x = \ln |\eta - \eta_*|$ with angular frequency

$$\omega = \frac{4\pi\gamma}{\Delta}. \quad (3)$$

Given the generality of the arguments used to show the behavior (2) for the black hole mass near criticality, a quantity \mathcal{E} with length dimension ℓ can be inferred to satisfy

$$\mathcal{E} \propto |\eta - \eta_*|^{\ell\gamma}, \quad (4)$$

or, more generally,

$$\ln \mathcal{E} = C + \ell\gamma \ln |\eta - \eta_*| + f(\ln |\eta - \eta_*|). \quad (5)$$

This allows us to determine the critical exponent γ using physical quantities other than the masses of the black holes formed in supercritical runs. In section 4 we show results that illustrate this, where we have performed a series of subcritical runs and analyzed the behavior of $\rho_{\text{central}}^{\text{max}}$, the maximum central energy density, as a function of x .

In his original study Choptuik solved the EKG equations using the Arnowitt–Deser–Misner (ADM) formalism [9]. More recently the Baumgarte–Shapiro–Shibata–Nakamura (BSSN) formalism [10–12] has been used to study critical phenomena as well [13–15], and, in particular, cases where the spherical symmetry condition is relaxed have been studied in [14, 15] (for similar studies that use the ADM formalism see [16, 17]).

As is the case with many research fields that heavily rely on numerical results, critical phenomena can be a daunting subject for those who have little to no experience with solving the EKG equations numerically. When using software developed by a particular group, one often encounters the problem that they were developed ‘by experts and for experts’, possessing a steep learning curve for new users. Considering the alternative track of developing one’s own software, usually months (or years!) are spent writing, testing, and debugging code, which presents its own enormous barrier to entry into the field. The work we present here aims to address these issues, presenting a new efficient, highly optimized, permissively licensed, easily extensible toolkit to study critical phenomena which has been specially designed with user-friendliness in mind.

`SFcollapse1D`⁷ [18] is a short (just over 1800 lines of code), well-documented and open-sourced [19] C++ code. Inspired by Choptuik’s original approach, `SFcollapse1D` has been specially developed to solve the EKG equations using the ADM formalism in spherical-like coordinates adopting the polar/radial gauge. This very restricted setup allowed us to greatly optimize `SFcollapse1D`, making it very efficient at performing the task for which it was designed. `SFcollapse1D` was not implemented with extensibility in mind, but adding support for different matter sources, coordinate systems, and gauge choices is both possible and the subject of ongoing work.

`NRPyCritCol`⁸ [18], on the other hand, is an open-source [20], highly extensible, state-of-the-art numerical relativity code which uses `NRPy+` [21, 22] to generate highly optimized C code kernels from Einstein-like expressions written in `Python` using the computer algebra package `SymPy` [23]. Because of this, `NRPyCritCol` inherits the ability of solving the BSSN equations in the multitude of curvilinear coordinate systems that `NRPy+` supports (e.g. spherical-like, cylindrical-like, and Cartesian-like, just to name a few). The entire code is documented using interactive Jupyter [24, 25] tutorial notebooks, making the code both well-documented and intuitive to new users.

⁷ Available for download at: <https://github.com/leowerneck/SFcollapse1D>

⁸ Available for download at: <https://github.com/zachetienne/nrpytutorial>

Both codes have been designed to optimize the use of memory while performing simulations, enabling the user to use consumer-grade desktop and laptop computers to study critical phenomena. As a proof of principle, all results presented in this paper have been obtained using a mid-2017 laptop computer with a 4-Core Intel Core i7 CPU.

This paper is organized as follows. In section 2 we present a brief review of the EKG equations using the ADM formalism in the polar/radial gauge, as well as the specific form of the BSSN equations implemented in our code. In section 3 we provide an overview of the numerical methods used by our codes, explaining e.g. our choice of coordinate system, boundary conditions, and initial data routines. In section 4 we study critical phenomena of a massless scalar field using both `SFCollapse1D` and `NRPyCritCol`. Finally, in section 5 we conclude and describe plans for future research projects.

2. Basic equations

Throughout this paper we adopt geometrized ($G = c = 1$) units. Greek indices denote the spacetime components of indexed expressions and range from 0 to 3, while Latin indices denote only the spatial components and range from 1 to 3. We also adopt Einstein's summation convention.

We are interested in solving Einstein's field equations,

$${}^{(4)}R_{\mu\nu} - \frac{1}{2}g_{\mu\nu} {}^{(4)}R = 8\pi T_{\mu\nu}, \quad (6)$$

where $g_{\mu\nu}$ is the spacetime metric, ${}^{(4)}R_{\mu\nu}$ is the four-dimensional Ricci tensor, ${}^{(4)}R = g^{\mu\nu} {}^{(4)}R_{\mu\nu}$ is the Ricci scalar, and $T_{\mu\nu}$ is the energy–momentum tensor of a massless scalar field, φ , and is given by

$$T_{\mu\nu} = (\nabla_\mu \varphi) (\nabla_\nu \varphi) - \frac{1}{2}g_{\mu\nu} (\nabla^\rho \varphi) (\nabla_\rho \varphi), \quad (7)$$

where ∇_μ is the covariant derivative compatible with $g_{\mu\nu}$. We cast Einstein's equations as an initial value problem by using the 3 + 1 formalism [9], in which the spacetime metric takes the form

$$ds^2 = g_{\mu\nu} dx^\mu dx^\nu = -\alpha^2 dt^2 + \gamma_{ij} (dx^i + \beta^i dt) (dx^j + \beta^j dt), \quad (8)$$

where α is the lapse function, β^i is the shift vector, and γ_{ij} is the spatial metric. The massless scalar field is evolved using the Klein–Gordon equation

$$\nabla^\mu \nabla_\mu \varphi = 0. \quad (9)$$

Next we present two formalisms for solving the EKG system of equations numerically. In section 2.1 we describe the fully constrained ADM formalism inspired by Choptuik's early work [1, 26–28], and in section 2.2 we outline our adopted covariant BSSN formalism of Brown [29].

2.1. EKG in the ADM formalism

When using the ADM equations, we adopt a fully constrained evolution scheme (see e.g. [30] for a pedagogical introduction) and adopt the polar/radial gauge, as well as spherical symmetry. The most general line element is then given by

$$ds^2 = -\alpha(t, r)^2 dt^2 + a(t, r)^2 dr^2 + r^2 (d\theta^2 + \sin^2 \theta d\varphi^2), \quad (10)$$

where $a \equiv \sqrt{\gamma_{rr}}$ is the radial metric function. Using (10) the Klein–Gordon equations can be written as

$$\partial_t \left(\frac{a}{\alpha} \partial_t \varphi \right) = \frac{1}{r^2} \partial_r \left(r^2 \frac{\alpha}{a} \partial_r \varphi \right). \quad (11)$$

Following [1], two auxiliary fields are introduced

$$\Phi(t, r) \equiv \partial_r \varphi(t, r); \quad \Pi(t, r) \equiv \frac{a(t, r)}{\alpha(t, r)} \partial_t \varphi(t, r), \quad (12)$$

enabling us to write equation (11) as the following system of first-order partial differential equations

$$\begin{aligned} \partial_t \Phi &= \partial_r \left(\frac{\alpha}{a} \Pi \right), \\ \partial_t \Pi &= \frac{1}{r^2} \partial_r \left(r^2 \frac{\alpha}{a} \Phi \right). \end{aligned} \quad (13)$$

The non-trivial metric quantities are not explicitly evolved in time, but instead determined using constraint equations. The Hamiltonian constraint in the ADM formalism is given by

$$\mathcal{H} = R + K^2 - K_{ij} K^{ij} - 16\pi\rho = 0, \quad (14)$$

where R is the Ricci scalar computed from γ_{ij} , K_{ij} is the extrinsic curvature, $K \equiv \gamma^{ij} K_{ij}$ is the mean curvature, and ρ is the energy density. In the polar slicing, we have $K = K_r^r$ and $K_\theta^\theta = 0 = K_\varphi^\varphi$. Using

$$\rho = \alpha^2 T^t{}_t = \frac{\Phi^2 + \Pi^2}{2a^2}, \quad (15)$$

we obtain a nonlinear first-order PDE that can be solved for the radial metric,

$$\frac{\partial_r a}{a} + \frac{a^2 - 1}{2r} = 2\pi r (\Phi^2 + \Pi^2). \quad (16)$$

Similarly, the polar slicing condition $\partial_t K_{\theta\theta} = 0$ yields a nonlinear first-order PDE, which we can solve to obtain the lapse,

$$\frac{\partial_r \alpha}{\alpha} - \frac{\partial_r a}{a} - \frac{a^2 - 1}{r} = 0. \quad (17)$$

In this coordinate system and with this gauge choice, K_r^r can be determined using the algebraic equation

$$K_r^r = -4\pi r \frac{\Phi \Pi}{\alpha}, \quad (18)$$

once we know (Φ, Π, α, a) and therefore does not play a role in the evolution.

2.2. EKG in the BSSN formalism

We now describe the covariant formulation of the Baumgarte–Shapiro–Shibata–Nakamura (BSSN) equations [10–12] proposed by Brown [29]. In the BSSN formalism, we perform a conformal decomposition of the spatial metric

$$\gamma_{ij} = e^{4\phi} \bar{\gamma}_{ij}, \quad (19)$$

where $\bar{\gamma}_{ij}$ is the conformal spatial metric, e^ϕ is the conformal factor, and ϕ is the conformal exponent. We also decompose the extrinsic curvature K_{ij} into its trace-free and full trace parts,

$$K_{ij} = A_{ij} + \frac{1}{3}\gamma_{ij}K. \quad (20)$$

We introduce the conformally rescaled extrinsic curvature,

$$\bar{A}_{ij} = e^{-4\phi}A_{ij}, \quad (21)$$

and evolve K and \bar{A}_{ij} instead of K_{ij} . Furthermore, we consider a background reference metric, $\hat{\gamma}_{ij}$, which is chosen to be the flat space metric in the coordinate system we are using, and define the tensor

$$\Delta\Gamma^i_{jk} \equiv \bar{\Gamma}^i_{jk} - \hat{\Gamma}^i_{jk}, \quad (22)$$

where $\bar{\Gamma}^i_{jk}$ and $\hat{\Gamma}^i_{jk}$ are the Christoffel symbols computed from $\bar{\gamma}_{ij}$ and $\hat{\gamma}_{ij}$, respectively. As a general rule, barred quantities are associated with $\bar{\gamma}_{ij}$ and hatted quantities are associated with $\hat{\gamma}_{ij}$. We evolve the spatial vector $\bar{\Lambda}^i$, which satisfies the constraints

$$\mathcal{C}^i \equiv \bar{\Lambda}^i - \Delta\Gamma^i = 0, \quad (23)$$

where

$$\Delta\Gamma^i \equiv \bar{\gamma}^{jk}\Delta\Gamma^i_{jk}. \quad (24)$$

We adopt Brown's 'Lagrangian' choice, in which the determinant of the conformal spatial metric is kept at its initial value throughout the numerical evolution, i.e.

$$\partial_t \bar{\gamma} = 0. \quad (25)$$

Defining

$$\partial_\perp \equiv \partial_t - \mathcal{L}_\beta, \quad (26)$$

where \mathcal{L}_β is the Lie derivative along the shift vector (see e.g. appendix A in [30] for an excellent review on Lie derivatives). We thus obtain the evolution equations [29]

$$\partial_\perp \bar{\gamma}_{ij} = \frac{2}{3}\bar{\gamma}_{ij}(\alpha\bar{A}^k_k - \bar{D}_k\beta^k) - 2\alpha\bar{A}_{ij}, \quad (27a)$$

$$\begin{aligned} \partial_\perp \bar{A}_{ij} = & -\frac{2}{3}\bar{A}_{ij}\bar{D}_k\beta^k - 2\alpha\bar{A}_{ik}\bar{A}^k_j + \alpha\bar{A}_{ij}K + e^{-4\phi}\{-2\alpha\bar{D}_i\bar{D}_j\phi + 4\alpha\bar{D}_i\phi\bar{D}_j\phi \\ & + 4\bar{D}_{(i}\alpha\bar{D}_{j)}\phi - \bar{D}_i\bar{D}_j\alpha + \alpha\bar{R}_{ij}\}^{\text{TF}}, \end{aligned} \quad (27b)$$

$$\partial_\perp \phi = \frac{1}{6}(\bar{D}_k\beta^k - \alpha K), \quad (27c)$$

$$\partial_\perp K = \frac{1}{3}\alpha K^2 + \alpha\bar{A}_{ij}\bar{A}^{ij} - e^{-4\phi}(\bar{D}_i\bar{D}^i\alpha + 2\bar{D}^i\alpha\bar{D}_i\phi), \quad (27d)$$

$$\begin{aligned} \partial_\perp \bar{\Lambda}^i = & \bar{\gamma}^{jk}\hat{D}_j\hat{D}_k\beta^i + \frac{2}{3}\Delta\Gamma^i\bar{D}_j\beta^j + \frac{1}{3}\bar{D}^i\bar{D}_j\beta^j \\ & - 2\bar{A}^{ij}(\partial_j\alpha - 6\partial_j\phi) + 2\alpha\bar{A}^{jk}\Delta\Gamma^i_{jk} - \frac{4}{3}\alpha\bar{\gamma}^{ij}\partial_jK. \end{aligned} \quad (27e)$$

where \bar{D}_i and \hat{D}_i are covariant derivatives compatible with $\bar{\gamma}_{ij}$ and $\hat{\gamma}_{ij}$, respectively. Note that, as discussed in [31], terms that include the Riemann tensor associated with the flat reference metric vanish and therefore have been omitted in the above set of equations. The TF superscript denotes the trace-free part of a tensor with respect to the conformal metric so that, given an arbitrary tensor M_{ij} , M_{ij}^{TF} is defined as

$$M_{ij}^{\text{TF}} \equiv M_{ij} - \frac{1}{3} \bar{\gamma}_{ij} (\bar{\gamma}^{mn} M_{mn}). \quad (28)$$

Further the source terms ρ , S_i , S_{ij} , and S are the energy density, momentum density, stress, and trace of the stress defined by a normal observer, respectively, and are given by

$$\rho \equiv n_\mu n_\nu T^{\mu\nu}, \quad (29)$$

$$S_i \equiv -\gamma_{i\mu} n_\nu T^{\mu\nu}, \quad (30)$$

$$S_{ij} \equiv \gamma_{i\mu} \gamma_{j\nu} T^{\mu\nu}, \quad (31)$$

$$S \equiv \gamma^{ij} S_{ij}, \quad (32)$$

where

$$\gamma_{\mu\nu} \equiv g_{\mu\nu} + n_\mu n_\nu, \quad (33)$$

is the metric induced by $g_{\mu\nu}$ onto the spatial hypersurface, and n^μ is the unit vector normal to the spatial hypersurface, such that

$$n^\mu = \alpha^{-1} (1, -\beta^i) \quad \text{and} \quad n_\mu = (-\alpha, 0, 0, 0). \quad (34)$$

The Hamiltonian constraint takes the form

$$\mathcal{H} = \frac{2}{3} K^2 - \bar{A}_{ij} \bar{A}^{ij} + e^{-4\phi} (\bar{R} - 8 \bar{D}^i \phi \bar{D}_i \phi - 8 \bar{D}^2 \phi) - 16\pi\rho = 0, \quad (35)$$

and the momentum constraints take the form

$$\mathcal{M}^i = e^{-4\phi} \left(\bar{D}_j \bar{A}^{ij} + 6 \bar{A}^{ij} \partial_j \phi - \frac{2}{3} \bar{\gamma}^{ij} \partial_j K \right) - 8\pi S^i = 0. \quad (36)$$

Unless otherwise stated, we adopt the moving puncture gauge [32, 33], in which the lapse is evolved using the ‘1 + log’ slicing condition [34],

$$\partial_t \alpha = \beta^i \partial_i \alpha - 2\alpha K, \quad (37)$$

and the shift is evolved using the second-order covariant Gamma-driver condition suggested by Brown [29],

$$\partial_t \beta^i = \beta^j \bar{D}_j \beta^i + \frac{3}{4} B^i, \quad (38)$$

$$\partial_t B^i = \beta^j \bar{D}_j B^i + (\partial_t \bar{\Lambda}^i - \beta^j \bar{D}_j \bar{\Lambda}^i) - \eta_S B^i. \quad (39)$$

When evolving the scalar field alongside the BSSN equations, we introduce the auxiliary variable

$$\Pi \equiv -n^\mu \nabla_\mu \varphi = -\frac{1}{\alpha} (\partial_t \varphi - \beta^i \partial_i \varphi), \quad (40)$$

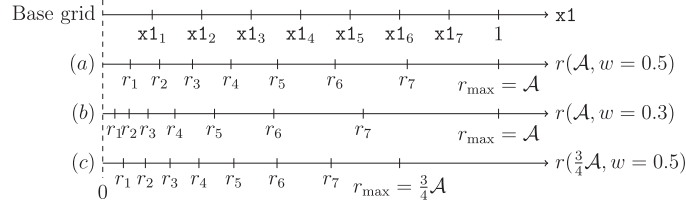


Figure 1. Illustration of the coordinate system we have adopted in our runs. The top axis displays the uniformly sampled $x1$ -coordinate, while the other three axes display, from top to bottom, the corresponding radial coordinate $r(r_{\max}, w)$ obtained using (43) for the cases $r(\mathcal{A}, 0.5)$, $r(\mathcal{A}, 0.3)$, and $r(\frac{3}{4}\mathcal{A}, 0.5)$, respectively.

so that (9) can be written in terms of the BSSN variables as

$$\partial_{\perp} \varphi = -\alpha \Pi, \quad (41)$$

$$\partial_{\perp} \Pi = \alpha K \Pi - e^{-4\phi} \bar{\gamma}^{ij} [\bar{D}_i (\alpha \bar{D}_j \varphi) + 2\alpha \bar{D}_i \varphi \bar{D}_j \phi]. \quad (42)$$

3. Numerical implementation

3.1. Solving the EKG equations: ADM formalism approach of *SFcollapse1D*

We use finite differences to approximate all spatial derivatives appearing in the evolution and constraint equations. When evolving the fully constrained ADM system, we adopt a second-order accurate leap frog scheme in order to march the initial hypersurface forward in time. A typical time step involves advancing Φ and Π using (13), then solving the Hamiltonian constraint (16) using a pointwise Newton–Raphson method in order to obtain the radial metric function a , and finally solving the polar slicing condition (17) to obtain the lapse function α . For the discretized versions of (13), (16), and (17), cf [28].

3.1.1. Coordinate system & numerical grids. We adopt a coordinate system described by the variables $x^i = (x1, x2, x3)$, which are uniformly sampled in the ranges $[0, 1]$, $[0, \pi]$, and $[-\pi, \pi]$, respectively. The relation between these coordinates and standard spherical coordinates (r, θ, ϑ) is given by

$$\begin{aligned} r &= \mathcal{A} \frac{\sinh(x1/w)}{\sinh(1/w)}, \\ \theta &= x2, \\ \vartheta &= x3, \end{aligned} \quad (43)$$

where $\mathcal{A} = r_{\max}$ corresponds to the domain size and w is a freely-specifiable parameter that controls how densely sampled the region near the origin is⁹. In figure 1 we illustrate the radial sampling of our coordinate system. We note that *SFcollapse1D* is a 1 + 1 code and therefore only the coordinate $x1$ in (43) is actually used in our implementation.

During our runs, we are mostly interested in the region near the origin. Nevertheless, the outer boundary cannot be placed too close to the center of the simulation, as spurious reflections

⁹ In NRPY+ this coordinate system is known as *SinhSpherical*.

of the outgoing waves may spoil our solution. This means that our numerical grids must be able to resolve a large range of length scales, while also having very high resolution near the origin, justifying our choice of the coordinate system (43). By adjusting the value of w , the resolution near the origin can be drastically increased while keeping the outer boundary a safe distance away from the strong-field region. Typical grid structures have 320 points in the radial direction, $\mathcal{A} = 16$, and $w = 0.08$, resulting in resolutions of $\Delta r_{\min} \sim 1 \times 10^{-6}$ near the origin, while having $\Delta r_{\max} \sim 1 \times 10^{-1}$ near the outer boundary. The ratio $\Delta r_{\max}/\Delta r_{\min} \sim 1 \times 10^5$ would require about 17 levels of traditional Berger–Olinger adaptive mesh refinement (AMR) [35] by factors of 2 to resolve the same range of length scales.

Having a very high resolution near the origin greatly constrains the time steps that we are allowed to take, since we must obey that Courant–Friedrichs–Lewy (CFL) condition

$$\Delta t = \Delta r_{\min} \mathcal{C}_0, \quad (44)$$

where \mathcal{C}_0 , the CFL factor, is a number that depends on the integration scheme being used. Nevertheless, when using the ADM formalism simply decreasing the value of w is enough to obtain production quality results from runs that complete in a reasonable amount of time. A typical run that uses `SFcollapse1D` with the grid resolution mentioned in the previous paragraph can be performed on consumer-grade laptop computers in about 45 min.

3.1.2. Boundary conditions. `SFcollapse1D` solves the EKG system in spherical-like coordinates imposing outgoing radiation boundary conditions of the form

$$\left[\partial_t(r\varphi) + \partial_r(r\varphi) \right] \Big|_{r=r_{\max}} = 0. \quad (45)$$

The boundary condition is applied to the scalar field φ itself, after which consistent values of the auxiliary fields Φ and Π are computed using (12). When applying boundary conditions, we use backwards finite differences whenever needed to ensure that both the spatial and time derivatives are all computed using valid data.

The PDEs that determine the radial metric and lapse functions are integrated from the origin outwards subject to the regularity conditions [26–28]

$$a(t, 0) = 1 \quad \text{and} \quad \partial_r \alpha(t, 0) = 0. \quad (46)$$

In order to be able to determine the lapse using the polar slicing condition (17), we use the regularity condition to fix the value of the lapse to a constant at the origin, which is arbitrarily set to unity. This is equivalent to the statement that central observers have coordinate time equal to proper time and is a choice made by some groups (see e.g. [30]). We choose to follow Choptuik’s prescription for rescaling the lapse [26], which involves computing the quantity

$$\kappa = \min \left[\frac{a}{\alpha} (1 - |\beta|) \right]. \quad (47)$$

Remember that for our gauge choice $\beta^i = 0$ and, thus, we get the simplified expression $\kappa = \min(a/\alpha)$. In general, the value of κ changes from one spatial hypersurface to the next. Since the polar slicing condition is homogeneous in α , multiplying it by a constant means that the condition is still satisfied. The particular form of κ is chosen so that no signals propagate faster than the speed of light on our numerical grids and so that the spacetime is asymptotically flat ($\lim_{r \rightarrow \infty} \alpha \rightarrow 1$). We refer the reader to chapter 4.4 of [26] for further details.

3.1.3. Initial data. When obtaining initial data, we specify the scalar field φ at the initial hypersurface and then analytically compute Φ from it. We assume that the initial hypersurface is at a ‘moment of time symmetry’, so that $\Pi = 0$ initially. We then solve the Hamiltonian constraint (16) in order to determine a and the polar slicing condition (17) in order to determine α , imposing the regularity conditions (46).

3.2. Solving the EKG equations: BSSN formalism approach of `NRPyCritCol`

The time-evolution of the BSSN equations (27a)–(27e), (37)–(39), (41), and (42) is more straightforward as they are written in a form that conforms to the basic pattern expected by the method of lines (MoL):

$$\partial_t \vec{f} = \vec{R}, \quad (48)$$

where the solution and right-hand sides vectors are denoted \vec{f} and \vec{R} , respectively. \vec{R} itself depends explicitly on the set of gridfunctions comprising \vec{f} and their spatial derivatives. Derivatives appearing within \vec{R} are evaluated adopting the usual centered/upwinded finite-difference approach for BSSN (outlined e.g. in [21]). Once \vec{R} is evaluated, the solution vector \vec{f} is stepped forward in time using a standard ODE numerical integration technique, applying boundary conditions as described in section 3.2.2.

In this paper we adopt the fourth-order accurate Runge-Kutta method (RK4) for evolving the BSSN equations forward in time via the MoL, and fourth-order accurate finite differences for approximating the spatial derivatives appearing in \vec{R} . Many other MoL ODE integration options are available in `NRPy+`, as well as the ability to finite difference spatial derivatives to arbitrary order. Since `NRPyCritCol` is based in `NRPy+`, it inherits these features automatically.

3.2.1. Coordinate system & numerical grids. `NRPyCritCol` adopts the same coordinate system as `SFcollapse1D`, which is described in section 3.1.1. There are several reasons, however, that make runs much more time consuming at very high resolutions when using `NRPyCritCol`. `NRPy+` is a 3 + 1 code that does not (as of writing) support spherical symmetry, thus, requiring a minimum number of points sampling the angular directions. This can negatively impact performance as boundary conditions in the angular directions are applied needlessly. Also, and less-avoidably, the BSSN formalism contains a much larger number of gridfunctions, and the right-hand sides of their evolution equations are far more complex than the simple ADM-based system evolved within `SFcollapse1D`. `NRPyCritCol` makes up for its inefficiencies with extreme flexibility and extensibility. For example, relaxing spherical symmetry, adjusting the MoL integration scheme, and increasing the finite difference order, can be performed by simply adjusting a few lines of code.

In order to dramatically speed up our runs, we have implemented a regridding infrastructure within `NRPyCritCol` that enables the grid resolution to be updated mid-run. Keeping the number of radial gridpoints fixed, the `SinhSpherical` (equation (43)) coordinate system offers two means for increasing the grid resolution near the origin: decreasing the value of \mathcal{A} or lowering the value of w . Once we change the value of \mathcal{A} or w , data from the original grid is interpolated onto a new grid, the time step is updated, and the run continues using the new grid. This technique has the advantage of adapting the resolution of the numerical grids to be well suited to the length scales associated with the solution.

We next provide a concrete example of our regridding algorithm. Let us assume that we have started a run with N_1 points sampling the x_1 coordinate, and with $r_{\max} = \mathcal{A}$ and $w = 0.5$. As shown in the top part of figure 1, `SinhSpherical` coordinates uniformly sample the coordinate x_1 in the interval $[0, 1]$, with grid (a) illustrating the corresponding radial coordinate

r obtained using (43). Let us now consider two different possible regridding options: changing the value of w so that $w_{\text{new}} = 0.3$ (case I) or changing the value of \mathcal{A} so that $\mathcal{A}_{\text{new}} = \frac{3}{4}\mathcal{A}$ (case II). By looking at figure 1, case I/II corresponds to changing grid (a) to grid (b)/(c), respectively. Our intention is to have the same base grid after regridding, with $x_1 \in [0, 1]$ and $N1_{\text{new}} = N1$. In order to do this, we compute $r_{\text{new}} = r(\mathcal{A}_{\text{new}}, w)$ or $r_{\text{new}} = r(\mathcal{A}, w_{\text{new}})$ and then compute $x1_{\text{new}} = x1(r_{\text{new}}, \mathcal{A}, w)$ by inverting (43). Armed with these values, we now have all the information needed to interpolate data onto the new grid ($x1_{\text{new}}$) from the original grid ($x1$). It is important to note that numerical relativity provides only approximate outer boundary conditions, so when \mathcal{A} is decreased, the physical domain becomes smaller. However, as the remaining duration of our simulation decreases with \mathcal{A} , spurious reflections arising from the approximate outer boundary conditions never come into causal contact with the physical region of interest (i.e. the central region of the numerical grid).

We now describe the regridding strategy we have used to obtain the results of section 4. We begin our runs with a numerical grid containing 320 points in the radial direction, $\mathcal{A} = 64$, and $w = 0.2$. Throughout the run, we perform a total of 19 regrids to resolve the critical solution. At each of the first 11 regrids, the adjustment $w \rightarrow w - 0.01$ is made to increase the radial sampling around the origin. The subsequent eight regrids gradually shrink the outer boundary using

$$\mathcal{A} \rightarrow \mathcal{A}_{\text{new}}, \quad \mathcal{A}_{\text{new}} = \{48, 32, 24, 16, 14, 12, 10, 8\}. \quad (49)$$

The total number of regrids was determined empirically by analysing the dynamics of the central values of the scalar field and the lapse function as we approached the critical solution, manually introducing additional regrids whenever the grid resolution was insufficient to resolve the features of the solution. To instill a deeper appreciation of this regridding approach, the ratio between the initial and final central resolutions for this particular case is $\Delta r_{\text{min}}^{\text{initial}} / \Delta r_{\text{min}}^{\text{final}} \approx 1.6 \times 10^3$. The final grid resolution ranges from $\Delta r_{\text{min}}^{\text{final}} \approx 8.3 \times 10^{-6}$ near the origin to $\Delta r_{\text{max}}^{\text{final}} \approx 2.8 \times 10^{-1}$ near the outer boundary, with ratio $\Delta r_{\text{max}}^{\text{final}} / \Delta r_{\text{min}}^{\text{final}} \approx 3.3 \times 10^4$. About 15 levels of traditional (factor-of-two) AMR levels of refinement would be necessary to resolve the same range of length scales.

Results presented here that made use of the regrid infrastructure were completed within 10–15 min on a laptop computer. We emphasize that the regridding infrastructure is not essential to obtain results of this quality, but simply a feature that allows the user to save significant computational resources. We also note that the regridding capacities of NRPyCritCol are not automated yet, requiring manual user input in order to work efficiently.

3.2.2. Boundary conditions. NRPyCritCol adopts cell-centered grids, and points lying near the boundaries of the grid interior depend upon finite-difference stencils that reach outside the interior. To address this, we extend our grids so that a certain number of points lie outside of the interior grid. These additional, exterior points are usually referred to as the *ghost zone points*.

Some of these ghost zone points exist beyond the outer boundary of our numerical domain ($r > r_{\text{max}}$), and must be filled through application of outer boundary conditions. While Sommerfeld boundary conditions are available in NRPy+, we consider in this paper only simple quadratic extrapolation boundary conditions, pushing the outer boundary out of causal contact with the center of the grid to avoid any contamination resulting from spurious reflections.

Other ghost zone points in the grid exterior map directly back to points in the grid *interior*. For example, in spherical coordinates, the value of a gridfunction $f(r, \theta, \vartheta)$ at a point in which $r < 0$ can be determined by simply identifying it to another point inside the grid, i.e.

$$f(-r, \theta, \vartheta) = f(r, \pi - \theta, \pi + \vartheta). \quad (50)$$

For a more detailed and general discussion of how these *inner* boundary ghost zones are filled in the NRPy+ infrastructure, see [21, 36].

3.2.3. Rescaling of tensorial quantities. We factor out coordinate singularities from tensorial quantities and evolve the remaining regular, ‘rescaled’ tensors following the procedure described in depth in [21] (see also [31, 37–39]). To briefly summarize, we write the reference metric as

$$\hat{\gamma}_{ij} = \delta_{kl} \mathbf{e}_i^{(k)} \mathbf{e}_j^{(l)}, \quad (51)$$

where $\mathbf{e}_j^{(i)}$ is the dual basis of the noncoordinate vectors $\mathbf{e}_{(j)}^i$, where (j) lists the individual basis vectors and i its components with respect to the coordinate basis. The Kronecker delta δ_{ij} is constant and equal to the identity in all coordinate systems, which defines the noncoordinate vector basis. By definition the set $\{\mathbf{e}_{(i)}\}$ is linearly independent and spans the tangent space at every point in the spatial hypersurface. Tensorial quantities are then rescaled using

$$T_{cd}^{ab} = \mathbf{e}_{(i)}^a \mathbf{e}_{(j)}^b \mathbf{e}_c^{(k)} \mathbf{e}_d^{(l)} t_{kl}^{ij}, \quad (52)$$

where the tensor t_{kl}^{ij} is well-defined on the entire hypersurface.

For example, consider the vector $\bar{\Lambda}^i$. Even if $\bar{\Lambda}^i$ is smooth in the Cartesian basis, Jacobians involved in the conversion to the spherical basis will generally contribute singular behavior at coordinate singularities. As the singular behavior is entirely encoded in the prescribed $\{\mathbf{e}_{(i)}\}$ for spherical coordinates, the tensor rescaling procedure effectively ‘factors out’ the singular part, so that the vector $\bar{\Lambda}^i$ in spherical coordinates is written in terms of the non-singular, rescaled vector $\bar{\lambda}^i$ as

$$\bar{\Lambda}^i = \mathbf{e}_{(j)}^i \bar{\lambda}^j = \begin{pmatrix} \bar{\lambda}^r \\ \bar{\lambda}^\theta / r \\ \bar{\lambda}^\varphi / (r \sin \theta) \end{pmatrix}. \quad (53)$$

Thus, if $\bar{\lambda}^i$ is smooth in the Cartesian basis (where $\bar{\lambda}^i = \bar{\Lambda}^i$), $\bar{\lambda}^i$ will be smooth in the spherical basis. Motivations behind this rescaling procedure become transparent especially when evaluating numerical spatial derivatives of $\bar{\Lambda}^i$ in a singular coordinate system; the numerical implementation provided by NRPy+ applies the product rule such that derivatives of $\bar{\lambda}^i$ (i.e. the smooth, rescaled variable) are entirely evaluated using numerical differentiation, and derivatives of the closed-form expressions $\{\mathbf{e}_{(i)}\}$ are evaluated exactly using SymPy.

The same regularization/rescaling procedure used to obtain $\bar{\lambda}^i$ is also applied to \bar{A}_{ij} , β^i , and B^i . The conformal metric is written as

$$\bar{\gamma}_{ij} = \hat{\gamma}_{ij} + \epsilon_{ij}, \quad (54)$$

where ϵ_{ij} —the quantity to which the regularization/rescaling is applied (instead of $\bar{\gamma}_{ij}$ directly)—stores the deviation from the flat space metric $\hat{\gamma}_{ij}$, and is *not* assumed to be small.

3.2.4. Numerical dissipation. When solving the BSSN equations we add Kreiss–Oliger dissipation [40] to the vector of right-hand sides \vec{R} . Defining R_f to be element f of \vec{R} , Kreiss–Oliger dissipation simply adds an operator Q acting on f :

$$R_f \rightarrow R_f + Qf. \quad (55)$$

The standard form of the Kreiss–Oliger dissipation operator Q is

$$Q \equiv \frac{\epsilon}{2^{2p}} h^{2p-1} (\mathcal{D}_+)^p (\mathcal{D}_-)^p, \quad (56)$$

where ϵ is the dissipation strength, $2(p-1)$ is the accuracy of the finite difference discretization, h is the step size, and \mathcal{D}_\pm are forward and backward finite difference operators. For a fourth-order accurate scheme in Cartesian coordinates, for which $p = 3$, one would have

$$Q = \frac{\epsilon}{64} \left(\Delta x^5 \frac{\partial^6}{\partial x^6} + \Delta y^5 \frac{\partial^6}{\partial y^6} + \Delta z^5 \frac{\partial^6}{\partial z^6} \right). \quad (57)$$

Extension to curvilinear coordinates involves normalizing each of the derivatives using $e^i_{(j)}$; in spherical coordinates, this becomes

$$Q = \frac{\epsilon}{64} \left(\Delta r^5 \frac{\partial^6}{\partial r^6} + \frac{\Delta \theta^5}{r} \frac{\partial^6}{\partial \theta^6} + \frac{\Delta \vartheta^5}{r \sin \theta} \frac{\partial^6}{\partial \vartheta^6} \right). \quad (58)$$

The addition of Qf acts to suppress high frequency oscillations on the grid, without altering the solution, as $Q \rightarrow 0$ in the continuum limit. The numerical solution should also not be greatly distorted, so long as ϵ is kept small. We generally choose $\epsilon = 0.1$, though values of up to 0.99 have been empirically demonstrated stable.

3.2.5. Initial data. When obtaining initial data in `NRPyCritCol`, we also assume that the initial hypersurface is at a ‘moment of time symmetry’, which means that

$$\bar{A}_{ij} = K = \Lambda^i = \beta^i = B^i = \Pi = 0. \quad (59)$$

Furthermore, we assume the conformal metric to be initially flat. With these choices, the momentum constraints (36) are automatically satisfied. The Hamiltonian constraint (35) can be written in the simplified form

$$\hat{D}^2 \psi = -2\pi \psi^5 \rho = -\pi (\hat{\gamma}^{ij} \partial_i \varphi \partial_j \varphi) \psi, \quad (60)$$

where $\psi = e^\phi$. One can then solve (60) to obtain the conformal factor ψ by imposing the boundary conditions

$$\partial_r \psi|_{r=0} = 0 \quad \text{and} \quad \lim_{r \rightarrow \infty} \psi = 1, \quad (61)$$

which correspond to regularity at the origin and asymptotic flatness, respectively. The only remaining quantity is then the lapse function, which can be freely specified. We choose the ‘pre-collapsed’ value $\alpha = \psi^{-2}$.

4. Results

We now present results obtained using both `SFcollapse1D` and `NRPyCritCol` in the study of critical phenomena of a massless scalar field in spherical symmetry. Although we will present more results obtained using `SFcollapse1D`, we emphasize that `NRPyCritCol` is by far the most versatile code. Fine-tuning of the critical solution with `NRPyCritCol` was performed with a single initial data sample, to validate against an `SFcollapse1D` evolution of the same initial data.

We begin our discussion by illustrating the critical solution. Consider the initial scalar field profile

$$\varphi_I = \eta \exp\left(-\frac{r^2}{\sigma^2}\right), \quad (62)$$

which corresponds to a spherically symmetric Gaussian centered at the origin, with η a dimensionless parameter indicating its amplitude. We choose $\sigma = 1$, so that all dimensionful quantities are given in units of σ .¹⁰ The subscript I indicates the initial data family, which we will need to distinguish later on.

For sufficiently small values of η the scalar field behaves as a spherical wave which bounces at the origin and then propagates outwards, resulting in flat space as the end state of the run. On the other hand, sufficiently large values of η lead to gravitational collapse of the scalar field with a black hole forming at the origin. We are then interested in obtaining the critical value η_* which separates subcritical runs (full dispersion) from supercritical runs (gravitational collapse).

The search for η_* starts by obtaining two values η_{weak} and η_{strong} , which correspond to subcritical and supercritical runs, respectively. It is clear that η_{weak} and η_{strong} bracket η_* . One can then systematically refine the interval containing η_* by using e.g. bisection. We define the fine-tuning parameter

$$\delta\eta \equiv \frac{\eta_{\text{strong}} - \eta_{\text{weak}}}{\eta_{\text{weak}}}, \quad (63)$$

which allows us to measure how close the amplitude η is to the critical value η_* . The better the fine-tuning, the longer the scalar field follows the critical solution before eventually dispersing or collapsing. All results below have a fine-tuning of $\delta\eta \sim 10^{-10}$ or better.

While the scalar field follows the critical solution, its profile repeats itself on ever smaller spatiotemporal scales, a phenomenon which is known as *echoing*. This echoing manifests itself in other quantities as well, for example the central value of the lapse function,

$$\alpha_{\text{central}} \equiv \alpha|_{r=0}. \quad (64)$$

In figures 2 and 3 we show α_{central} as a function of time for runs performed using the ADM and BSSN formalisms, respectively, after a high level of fine-tuning was obtained¹¹. Note that the difference in the overall behavior of α_{central} between the two figures is due to the different gauge choices we made when evolving the EKG equations using the different formalisms, as described in section 2.

The self-similar behavior of the central value of the scalar field is better illustrated in terms of the logarithmic proper time variable

$$\xi \equiv -\ln|\tau_* - \tau|, \quad (65)$$

¹⁰ Our problem is essentially scale-free, but setting $\sigma = 1$ can be thought of as making the problem completely dimensionless by choosing a particular mass scale, e.g. $M_\odot = 1$. One can recover physical units using powers of G , c , and M_\odot .

¹¹ Instructions for reproducing figures 2 and 4 can be found in [41] as well as in section 5.3.2 of the supplementary material (<https://stacks.iop.org/CQG/38/245005/mmedia>), while instructions for reproducing figure 3 and the corresponding BSSN version of figure 4 can be found in [42, 43].

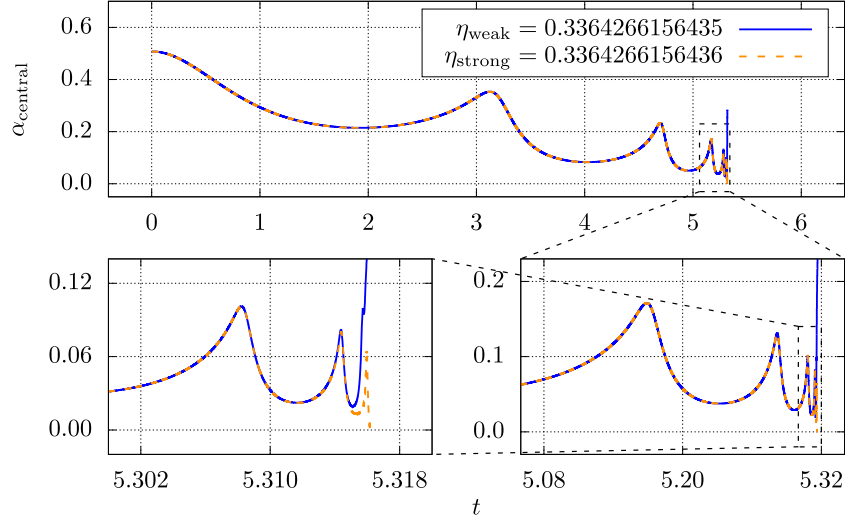


Figure 2. Central value of the lapse function as a function of coordinate time for a spherically symmetric run that adopts the ADM formalism. The top panel displays α_{central} throughout the entire run, while the bottom two panels are zooms of increasing magnitude toward the late behavior of the lapse; the bottom-right panel is of moderate zoom, and the bottom-left panel is at a more extreme zoom. We display a subcritical (blue, solid) solution with η_{weak} and a supercritical (orange, dashed) solution with η_{strong} , which bracket the critical solution for which $\eta = \eta_*$.

where

$$\tau(t) = \int_0^t \alpha(t') dt', \quad (66)$$

is the proper time and τ_* is the accumulation time, after which the solution no longer follows the critical one. As we approach the accumulation time, the oscillations associated with the critical solution ‘pile up’, as can be seen in figures 2 and 3. This can also be observed in the first two panels of figure 4, where we show φ_{central} from a subcritical run performed with `SFCollapse1D` as a function of t , τ , and ξ . The same behavior of φ_{central} is observed in runs using `NRPyCritCol`. We note that as we approach the critical solution, the oscillations of the scalar fields at the origin have a fixed amplitude of 0.61 (in our units and conventions), a behavior also observed by other groups (see e.g. [1, 13, 15]). The same amplitude is found for all initial data families studied in this paper and the value 0.61 is also independent of the particular value of η_* .

We follow [15] and consider two pairs of consecutive zero crossings of φ_{central} , (τ_n, τ_{n+1}) and (τ_m, τ_{m+1}) , and assume that the period of oscillation, Δ , in the logarithmic proper time is constant and equal to

$$\Delta = 2 \ln \left(\frac{\tau_* - \tau_n}{\tau_* - \tau_{n+1}} \right). \quad (67)$$

This allows us to estimate the accumulation time from the numerical data using

$$\tau_* = \frac{\tau_n \tau_{m+1} - \tau_m \tau_{n+1}}{\tau_n - \tau_{n+1} - \tau_m + \tau_{m+1}}. \quad (68)$$

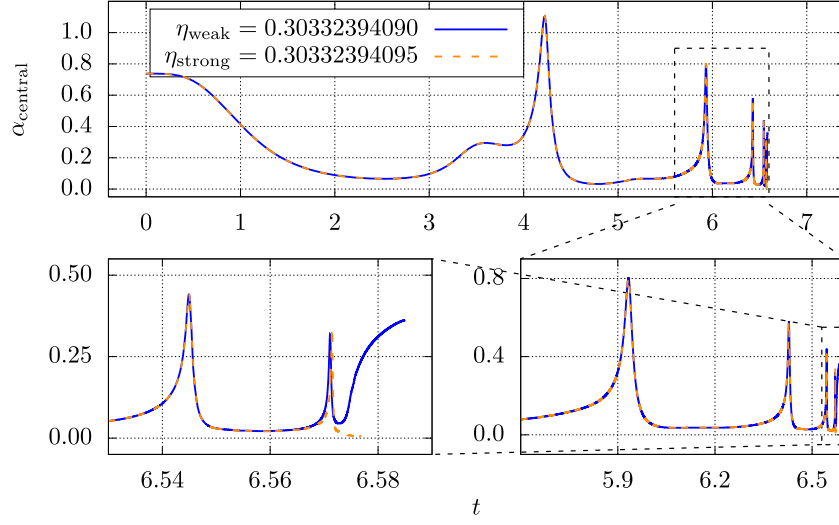


Figure 3. Same as figure 2 but for the BSSN formalism.

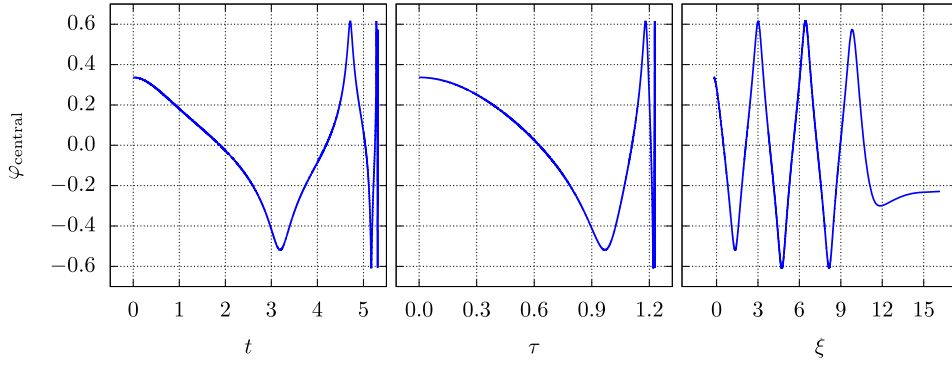


Figure 4. Central value of the scalar field near the critical solution as a function of coordinate time t (left), proper time τ (center), and logarithmic proper time $\xi \equiv -\ln|\tau_* - \tau|$ (right). The accumulation time, τ_* , was determined as the average value obtained from (68) after using all possible pairs of zero crossing proper times (τ_n, τ_{n+1}) for which the scalar field was already in the critical regime. We obtained $\Delta = 3.43(4)$ for the average oscillation period in logarithmic proper time, in agreement with the literature.

Next, we study the universality of the critical solution by repeating our analysis using `SFcollapse1D` and the additional initial data families

$$\varphi_{\text{II}} = \eta r^3 \exp \left[- \left(\frac{r - r_0}{\sigma} \right)^2 \right], \quad (69)$$

$$\varphi_{\text{III}} = \eta \left\{ 1 - \tanh \left[\left(\frac{r - r_0}{\sigma} \right)^2 \right] \right\}. \quad (70)$$

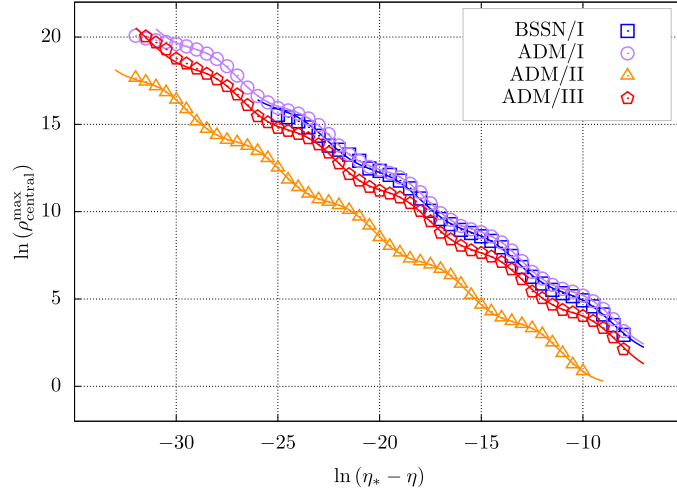


Figure 5. Maximum central density as a function of the scalar field initial amplitude η for several subcritical runs and for different initial data families. The points represent numerical results, while the solid lines are fits obtained using (71). The average critical exponent obtained from the fit is $\gamma = 0.375$, while the average echoing period is $\Delta = 3.475$. We refer the reader to table 1 for a detailed summary of the results.

We start by fine-tuning the critical parameter η_* for all different families. We then perform a series of *subcritical* runs (i.e. with $\eta < \eta_*$) and keep track of the maximum central energy density, $\rho_{\text{central}}^{\text{max}}$, which should behave as

$$\ln(\rho_{\text{central}}^{\text{max}}) = C - 2\gamma x + k \sin(\omega x + \phi_{\text{ph}}), \quad (71)$$

where γ is the universal exponent and

$$\omega = \frac{4\pi\gamma}{\Delta}, \quad (72)$$

is the frequency of the oscillation, with Δ the universal oscillation period. In the above expressions, C , k , and ϕ_{ph} are family-dependent constants and $x \equiv \ln|\eta_* - \eta|$.

For these runs, we adopt an initial value for the amplitude of the scalar field, η_0 , such that $x_0 \equiv \ln|\eta_* - \eta_0|$ is set to $x_0 = -32$ for runs that use `SFCollapse1D` and $x_0 = -25$ for runs that use `NRPYCriticalCol`, since the fine-tuning of the latter is smaller. We then choose a value $\Delta x = 0.5$ and perform a series of runs with

$$x_i = x_0 + i \cdot \Delta x, \quad i = 0, 1, 2, \dots, N, \quad (73)$$

typically stopping at $x_N = -8$. In figure 5 we show the results of these runs, displaying $\log(\rho_{\text{central}}^{\text{max}})$ as a function of x .

The universal exponent γ is determined by fitting the function (71) to the numerical data. The echoing period Δ is determined in two different ways. First, using the same fitting function used to determine γ , we find the oscillation frequency ω and compute Δ using (72). Second, we analyze the zero crossings of φ_{central} and estimate Δ using (67). In table 1 we summarize our results. The uncertainties in γ and Δ obtained using the fitting function correspond to the values for which the fit becomes visually worse to describe the numerical data. Using all

Table 1. Results for the critical exponent, echoing period, and accumulation time for different formalisms and initial data families.

Formalism	Family	r_0	σ	γ	Δ		τ_*
					(72)	(67)	
BSSN	I	—	1	0.374(3)	$3.51^{+0.01}_{-0.06}$	3.46(2)	1.56275073
ADM	I	—	1	0.375(1)	$3.47^{+0.02}_{-0.04}$	3.43(4)	1.22958674
ADM	II	2	2	0.376(1)	$3.45^{+0.02}_{-0.03}$	3.42(4)	4.82081707
ADM	III	0	2.5	0.375(1)	$3.47^{+0.03}_{-0.05}$	3.44(6)	2.59392679

possible zero crossing pairs in (67), we find the average value for Δ and its sample standard deviation, which are also presented in the table. The average value for the critical exponent is $\gamma = 0.375$, while the average value for the echoing period is $\Delta = 3.475$, both in excellent agreement with the literature.

5. Conclusions

In this paper we have presented two new open-source, user-friendly codes to study critical phenomena on consumer-grade computers: `SFcollapse1D` and `NRPyCritCol`. `SFcollapse1D` was designed to reproduce the original results of critical phenomena obtained by Choptuik, and serves as a good tool for new students and researchers to reproduce the seminal results in the field. `NRPyCritCol` is a state-of-the-art $3 + 1$ numerical relativity code for studying critical collapse, and is highly extensible and versatile.

Both codes have been designed to be fast and extremely memory efficient enabling the study of critical phenomena at the scale of consumer-grade desktop and laptop computers. Additionally we make use of highly optimized numerical grids, which both exploit symmetries and are adaptively adjusted to the lengthscales of the problem.

To demonstrate the capabilities of our codes, we have presented results of a study of critical phenomena of a massless scalar field in spherical symmetry. All of these results were obtained using a mid-2017 laptop computer with a 4-Core Intel Core i7 processor. With production-quality grids, runs with `SFcollapse1D` took about 45 min to complete, while runs with `NRPyCritCol` required as few as 10 min due to the implementation of manual regridding capabilities. Automation of this regridding infrastructure will be a focus of future work.

While in this paper we have focused our attention to the rather ‘canonical’ example of critical collapse in spherical symmetry, `NRPyCritCol` in particular is perfectly suited for more generic scenarios. Exploration of such scenarios will also be a focus of future work.

Acknowledgments

We thank T Baumgarte, M Choptuik, and I Ruchlin for useful discussions and suggestions. LRW was financially supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance Code 001. This work was also supported by NASA Grant No. TCAN-80NSSC18K1488. ZBE gratefully acknowledges the US NSF for financial support from Award PHY-1806596; and NASA for financial support from Award ISFM-80NSSC18K0538. EA wishes to thank FAPESP and CNPq (Brazil) for support.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

ORCID iDs

Leonardo R Werneck  <https://orcid.org/0000-0002-4541-8553>
 Zachariah B Etienne  <https://orcid.org/0000-0002-6838-9185>
 Elcio Abdalla  <https://orcid.org/0000-0002-9814-1690>
 Bertha Cuadros-Melgar  <https://orcid.org/0000-0001-8111-2431>
 CE Pellicer  <https://orcid.org/0000-0002-6347-7813>

References

- [1] Choptuik M W 1993 *Phys. Rev. Lett.* **70** 9
- [2] Gundlach C 2003 *Phys. Rep.* **376** 339–405
- [3] Gundlach C and Martín-García J M 2007 *Living Rev. Relativ.* **10** 5
- [4] Gundlach C 1997 *Phys. Rev. D* **55** 695
- [5] Martín-García J M and Gundlach C 1999 *Phys. Rev. D* **59** 064031
- [6] Martín-García J M and Gundlach C 2003 *Phys. Rev. D* **68** 024011
- [7] Reiterer M and Trubowitz E 2019 *Commun. Math. Phys.* **368** 143–86
- [8] Hod S and Piran T 1997 *Phys. Rev. D* **55** R440
- [9] Arnowitt R, Deser S and Misner C W 1959 *Phys. Rev.* **116** 1322
- [10] Nakamura T, Oohara K and Kojima Y 1987 *Prog. Theor. Phys. Suppl.* **90** 1–218
- [11] Shibata M and Nakamura T 1995 *Phys. Rev. D* **52** 5428
- [12] Baumgarte T W and Shapiro S L 1998 *Phys. Rev. D* **59** 024007
- [13] Akbarian A and Choptuik M W 2015 *Phys. Rev. D* **92** 084037
- [14] Clough K and Lim E A 2016 arXiv:1602.02568
- [15] Baumgarte T W 2018 *Phys. Rev. D* **98** 084012
- [16] Abrahams A M and Evans C R 1993 *Phys. Rev. Lett.* **70** 2980
- [17] Choptuik M W, Hirschmann E W, Liebling S L and Pretorius F 2003 *Phys. Rev. D* **68** 044007
- [18] Werneck L R 2020 *PhD Thesis* Universidade de São Paulo
- [19] The GNU public license version 3 <https://gnu.org/licenses/gpl-3.0.txt>
- [20] The 2-Clause BSD license <https://opensource.org/licenses/BSD-2-Clause>
- [21] Ruchlin I, Etienne Z B and Baumgarte T W 2018 *Phys. Rev. D* **97** 064036
- [22] NRPy+’s webpage <http://nrpyplus.net/>
- [23] Meurer A et al 2017 *PeerJ Comput. Sci.* **3** e103
- [24] Kluyver T et al 2016 *Jupyter Notebooks-A Publishing Format for Reproducible Computational Workflows* vol 2016 (Amsterdam : IOS Press BV)
- [25] Project Jupyter website <https://jupyter.org/>
- [26] Choptuik M W 1986 *PhD Thesis* University of British Columbia <http://hdl.handle.net/2429/27043>
- [27] Choptuik M 1992 *Approaches to Numerical Relativity* vol 202 (Cambridge: Cambridge University Press)
- [28] Choptuik M W 1994 *Deterministic Chaos in General Relativity* (Berlin: Springer) pp 155–75
- [29] Brown J D 2009 *Phys. Rev. D* **79** 104029
- [30] Baumgarte T W and Shapiro S L 2010 *Numerical Relativity: Solving Einstein’s Equations on the Computer* (Cambridge: Cambridge University Press)
- [31] Baumgarte T W, Montero P J, Cordero-Carrión I and Müller E 2013 *Phys. Rev. D* **87** 044026
- [32] Campanelli M, Lousto C O, Marronetti P and Zlochower Y 2006 *Phys. Rev. Lett.* **96** 111101
- [33] Baker J G, Centrella J, Choi D-I, Koppitz M and van Meter J 2006 *Phys. Rev. Lett.* **96** 111102
- [34] Bona C, Massó J, Seidel E and Stela J 1995 *Phys. Rev. Lett.* **75** 600
- [35] Berger M J and Oliger J 1984 *J. Comput. Phys.* **53** 484–512

- [36] NRPy+'s curvilinear boundary conditions tutorial notebook <https://tinyurl.com/4b3c3jdj>
- [37] Bonazzola S, Gourgoulhon E, Grandclément P and Novak J 2004 *Phys. Rev. D* **70** 104007
- [38] Shibata M, Urȳ K and Friedman J L 2004 *Phys. Rev. D* **70** 044044
- [39] Montero P J and Cordero-Carrión I 2012 *Phys. Rev. D* **85** 124037
- [40] Kreiss H, Kreiss H-O and Olinger J 1973 *Methods for the Approximate Solution of Time Dependent Problems* (International Council of Scientific Unions: World Meteorological Organization)
- [41] SFcollapse1D github repository <https://github.com/leowerneck/SFcollapse1D>
- [42] NRPy+'s scalar field collapse with regrids tutorial notebook <https://tinyurl.com/yd5vwraK>
- [43] NRPyCritCol original source code <https://github.com/leowerneck/NRPyCritCol-Ccodes>