PHYSICAL REVIEW D **95,** 084043 (2017)

# Scalar self-force for highly eccentric equatorial orbits in Kerr spacetime

Jonathan Thornburg[1,*] and Barry Wardell[2,3,†]

[1]*Department of Astronomy and Center for Spacetime Symmetries, Indiana University, Bloomington, Indiana 47405, USA*

[2]*School of Mathematics and Statistics and Institute for Discovery, University College Dublin, Belfield, Dublin 4, D04 V1W8, Ireland*

[3]*Department of Astronomy, Cornell University, Ithaca, New York 14853, USA*

(Received 28 October 2016; published 24 April 2017)

If a small "particle" of mass $\mu M$ (with $\mu \ll 1$) orbits a black hole of mass $M$, the leading-order radiation-reaction effect is an $\mathcal{O}(\mu^2)$ "self-force" acting on the particle, with a corresponding $\mathcal{O}(\mu)$ "self-acceleration" of the particle away from a geodesic. Such "extreme-mass-ratio inspiral" systems are likely to be important gravitational-wave sources for future space-based gravitational-wave detectors. Here we consider the "toy model" problem of computing the self-force for a scalar-field particle on a bound eccentric orbit in Kerr spacetime. We use the Barack-Golbourn-Vega-Detweiler effective-source regularization with a 4th-order puncture field, followed by an $e^{im\phi}$ ("$m$-mode") Fourier decomposition and a separate time-domain numerical evolution in $2+1$ dimensions for each $m$. We introduce a finite worldtube that surrounds the particle worldline and define our evolution equations in a piecewise manner so that the effective source is only used within the worldtube. Viewed as a spatial region, the worldtube moves to follow the particle's orbital motion. We use slices of constant Boyer-Lindquist time in the region of the particle's motion, deformed to be asymptotically hyperboloidal and compactified near the horizon and $\mathcal{J}^+$. Our numerical evolution uses Berger-Oliger mesh refinement with 4th-order finite differencing in space and time. Our computational scheme allows computation for highly eccentric orbits and should be generalizable to orbital evolution in the future. Our present implementation is restricted to equatorial geodesic orbits, but this restriction is not fundamental. We present numerical results for a number of test cases with orbital eccentricities as high as 0.98. In some cases we find large oscillations ("wiggles") in the self-force on the outgoing leg of the orbit shortly after periastron passage; these appear to be caused by the passage of the orbit through the strong-field region close to the background Kerr black hole.

DOI: 10.1103/PhysRevD.95.084043

## I. INTRODUCTION

Consider a small (compact) body of mass $\mu M$ (with $0 < \mu \ll 1$) moving freely in an asymptotically flat background spacetime (e.g., Kerr spacetime) of mass $M$. This system emits gravitational radiation, and there is a corresponding radiation-reaction influence on the small body's motion. Self-consistently calculating this motion and the emitted gravitational radiation (and in general, the perturbed spacetime) is a long-standing research question in general relativity.

There is also an astrophysical motivation for this calculation: If a neutron star or stellar-mass black hole of mass $\sim 1$–$100 M_\odot$ orbits a massive black hole of mass $\sim 10^5$–$10^7 M_\odot$,[1] the resulting "extreme-mass-ratio inspiral" (EMRI) system is expected to be a strong astrophysical gravitational-wave (GW) source detectable by the planned Laser Interferometer Space Array (LISA) space-based gravitational-wave detector.[2] LISA is expected to observe many such systems, some of them at quite high signal/noise ratios ([1–4]). The data analysis for, and indeed the detection of, such systems will generally require matched-filtering the detector data stream against appropriate precomputed GW templates. The problem of computing such templates provides the astrophysical motivation for our calculation.

We are particularly concerned with the case where the small body's orbit is highly relativistic, so post-Newtonian methods (see, for example, [5] Sec. 6.10; [6–9] and references therein) are not reliably accurate. Since the time scale for radiation reaction to shrink the orbit is very long ($\sim \mu^{-1} M$) while the required resolution near the small body is very high ($\sim \mu M$), a direct "numerical relativity" integration of the Einstein equations (see, for example, [10–14]

*jthorn@astro.indiana.edu

†barry.wardell@gmail.com

[1] $M_\odot$ denotes the solar mass.

[2] The LISA proposal has had various design and name changes during its lifetime. For a time it was known as the New Gravitational-Wave Observatory (NGO) or evolved LISA (eLISA), but recently it has returned to the original name, LISA.

 

and references therein) would be prohibitively expensive (and probably insufficiently accurate) for this problem.[3]

Instead, we use black-hole perturbation theory, treating the small body as an $\mathcal{O}(\mu)$ perturbation on the background spacetime. For this work we attempt to calculate leading-order radiation-reaction effects, i.e., $\mathcal{O}(\mu)$ field perturbations and $\mathcal{O}(\mu^2)$ radiation-reaction "self-forces" acting on the small body. Because of the technical difficulty of controlling gauge effects in gravitational perturbations, in this work we use a scalar-field "toy model" system with the expectation that the techniques developed and discoveries made in the scalar case will carry over to the gravitational case.

The obvious way to model the small body is as a small black hole. While conceptually elegant, this approach is technically somewhat complicated [21]. Instead, we model the small body as a point particle. Although one may be concerned about potential foundational issues with this approach,[4] in practice it works well and, importantly, it agrees with rigorous derivations that do not rely on the use of point particles.

The $\mathcal{O}(\mu)$ "MiSaTaQuWa" equations of motion for a gravitational point particle in a (strong-field) curved spacetime were first derived by Mino, Sasaki, and Tanaka [24] and Quinn and Wald [25] (also see Detweiler's analysis [26]) and have recently been rederived in a more rigorous manner by Gralla and Wald [27].[5] See [21,29–35] for general reviews of gravitational radiation-reaction dynamics.

The particle's motion may be modeled as either (i) nongeodesic motion in the background Schwarzschild/Kerr spacetime under the influence of a radiation-reaction "self-force," or (ii) geodesic motion in a perturbed spacetime. These two perspectives (which are in some ways analogous to Eulerian versus Lagrangian formulations of fluid dynamics) are equivalent [36]; in this work we use the formulation (i). The MiSaTaQuWa equations then give the self-force in terms of (the gradient of) the metric perturbation due to the particle, which must be computed using black-hole perturbation theory.

The computation of the field perturbation due to a point particle is particularly difficult because the "perturbation" is formally infinite at the particle and thus must be regularized. There are several different, but equivalent, regularization schemes known for this problem, notably the "mode-sum" or "$\ell$-mode" scheme developed by Barack and Ori [37–41], Detweiler, Messaritaki, and Whiting [42,43], and Haas and Poisson [44]; the Green-function approach [45–48]; and the "effective-source" scheme of Barack and Golbourn [49] and Vega and Detweiler [50].

For a detailed presentation of the different regularization/computation schemes and their advantages and disadvantages, see [51]. In the present context we observe that for a Kerr background the traditional mode-sum scheme becomes less desirable because the mode equations don't separate: all the (infinite set of) modes remain coupled. While the coupled modes can still be treated numerically (see, e.g., [52]), here we adopt a different approach, the effective-source regularization scheme.

As discussed in detail in Sec. II A, the effective-source scheme's basic concept is to analytically compute a "puncture field" which approximates the particle's Detweiler-Whiting singular field [42], then numerically solve for the difference between the actual field perturbation and the puncture field. We have previously described many of the details of the computation of the puncture field [53]; in this work we focus on the application of this scheme to a particular class of self-force computations.

Depending on how the partial differential equations (PDEs) are solved, there are two broad classes of self-force computations: frequency domain and time domain. Frequency-domain computations involve a Fourier transform of the PDEs in time, reducing the numerical computation to the solution of a set of ordinary differential equations (ODEs) (see, for example, [43]). The resulting computations are typically very efficient and accurate for circular or near-circular particle orbits,[6] but degrade rapidly in efficiency with increasing eccentricity of the particle's orbit, becoming impractical for highly eccentric orbits [58,59].[7] In contrast, time-domain computations involve a direct numerical time integration of the PDEs and are generally less efficient and accurate than frequency-domain computations. However, time-domain computations can

[3]A number of researchers have attempted direct numerical-relativity binary-black-hole simulations for systems with "intermediate" mass ratios up to 100:1 ($\mu = 0.01$), (see, for example, [15–20]). However, it has not (yet) been possible to extend these results to the extreme-mass-ratio case nor to accurately evolve even the 100:1 case for a radiation-reaction time scale.

[4]Geroch and Traschen [22] have shown that point particles in general relativity cannot consistently be described by metrics with $\delta$-function stress-energy tensors. More general Colombeau-algebra methods may be able to resolve this problem [23], but the precise meaning of the phrase "point particle" in general relativity remains a delicate question.

[5]Gralla, Harte, and Wald [28] have also recently obtained a rigorous derivation of the electromagnetic self-force in a curved spacetime.

[6]As notable examples of this accuracy, Blanchet *et al.* [54] and Shah *et al.* [55] have both recently computed the gravitational self-force for circular geodesic orbits in Schwarzschild spacetime to a relative accuracy of approximately one part in $10^{13}$, and Heffernan, Ottewill, and Wardell [56] (building on earlier work by Detweiler, Messaritaki, and Whiting [43]) have extended this to a few parts in $10^{17}$. Johnson-McDaniel, Shah, and Whiting [57] describe an "experimental mathematics" approach to computing post-Newtonian expansions of various invariants (again for circular geodesic orbits in Schwarzschild spacetime) by applying an integer-relation algorithm to numerical results calculated using up to 5000 decimal digits of precision.

[7]Barack, Ori, and Sago [60] have found an elegant solution for some other limitations which had previously affected frequency-domain calculations.

accommodate arbitrary particle orbits with only modest penalties in performance and accuracy [61], with some complications in the numerical schemes (see, for example, [62,63]).

In this work our goal is to consider highly eccentric orbits,[8] so we follow the time-domain approach. We use standard Berger-Oliger mesh-refinement techniques and compactified hyperboloidal slices for improved accuracy and efficiency.

The remainder of this paper is organized as follows:

Section I A summarizes our notation.

Section II gives a detailed description of our theoretical and computational formalism for self-force computations, with subsections on the effective-source regularization (II A), the $m$-mode Fourier decomposition (II B), the worldtube (II C), moving the worldtube (II D), hyperboloidal slices and compactification (II E), our reduction to a 1st-order-in-time system of evolution equations (II F), the computation of the puncture field and effective source (II G), the computation of the effective source close to the particle (II H), boundary conditions (II I), initial data (II J), how the self-force is computed from our evolved field variables (II K), the large-$m$ "tail series" (II L), selecting the time interval for analysis within an evolution (II M), selecting a "low-noise" subset of times within an evolution (II N), how we split the self-force into dissipative and conservative parts (II O), and a summary of our computation and data analysis (II P).

Section III presents our numerical results and compares them to values obtained by other authors, with subsections on our test configurations and parameters (III A), an example of our data analysis (III B), the convergence of our results with numerical resolution (III C), a numerical verification that our results are independent of the choice of worldtube and other numerical parameters (III D), comparison of our results with those of other researchers (III E), an overview of our computed self-force for each configuration (III F), our results for highly eccentric orbits (III G), our results for zoom-whirl orbits (III H), and strong oscillations ("wiggles") in the self-force shortly after periastron (III I).

Section IV presents a general discussion of this work, the conclusions to be drawn from it, and some directions for future research.

Appendix A describes the transformation between $\tilde{\phi}$ and $\phi$ derivatives, where $\tilde{\phi}$ is the "untwisted" azimuthal coordinate defined by (2.8).

Appendix B describes our computational scheme in more detail, with subsections on the numerical computation of $r(r_*)$ (B 1), the numerical integration of equatorial eccentric Kerr geodesics (B 2), gradual turn-on of the effective source (B 3), our algorithm for moving the worldtube (B 4), constraints on moving the worldtube early in the time evolution (B 5), finite differencing across the worldtube boundary (B 6), computing the set of grid points where adjusted finite differencing is needed (B 7), computing the set of grid points where the puncture field is needed (B 8), the numerical time evolution using Berger-Oliger mesh refinement (B 9), finite differencing near the particle (B 10), and implicit-explicit (IMEX) evolution schemes (B 11).

[8]Hopman and Alexander [64] find that LISA EMRIs are likely to have eccentricities up to $e \sim 0.8$. *Intermediate*-mass-ratio inspirals (where the small body has a mass $100M_\odot \lesssim \mu M \lesssim 10^4 M_\odot$) are likely to have very high eccentricities $0.995 \lesssim e \lesssim 0.998$; these systems are likely much rarer than EMRIs, but are also much stronger GW sources.

### A. Notation

We generally follow the sign and notation conventions of Wald [65], with $G = c = 1$ units and a $(-,+,+,+)$ metric signature. We use the Penrose abstract-index notation, with indices $abcd$ running over spacetime coordinates, $ijk$ running over the spatial coordinates, $\ell$ running over only the $m$-mode coordinates $(t, r, \theta)$, and $s$ running over only the spatial $m$-mode coordinates $(r, \theta)$ [in both of the latter cases, the coordinates are defined by (1.1) below]. $\nabla_a$ is the (spacetime) covariant derivative operator. $X \coloneqq Y$ means that $X$ is defined to be $Y$. $\Box \coloneqq \nabla_a \nabla^a$ is the 4-dimensional (scalar) wave operator [66,67]. conj$[z]$ is the complex conjugate of the complex number $z$. $\partial S$ is the boundary of the set $S$. $(a)_n$ denotes the Pochhammer symbol $\Pi_{k=a}^{a+n-1} k$.

We use Boyer-Lindquist coordinates $(t, r, \theta, \phi)$ on Kerr spacetime, defined by the line element

$$\begin{aligned} ds^2 = &-\left(1 - \frac{2Mr}{\Sigma}\right) dt^2 - 4M^2 \tilde{a} \frac{r \sin^2\theta}{\Sigma} dt\, d\phi \\ &+ \frac{\Sigma}{\Delta} dr^2 + \Sigma d\theta^2 \\ &+ \left(r^2 + M^2 \tilde{a}^2 + 2M^3 \tilde{a}^2 \frac{r \sin^2\theta}{\Sigma}\right) \sin^2\theta\, d\phi^2, \end{aligned} \quad (1.1)$$

where $M$ is the spacetime mass, $\tilde{a} = J/M^2$ is the dimensionless spin of the black hole (limited to $|\tilde{a}| < 1$), $\Sigma = r^2 + M^2 \tilde{a}^2 \cos^2\theta$, and $\Delta = r^2 - 2Mr + M^2 \tilde{a}^2$. In Boyer-Lindquist coordinates the event horizon is the coordinate sphere $r = r_h = r_+ = M(1 + \sqrt{1 - \tilde{a}^2})$ and the inner horizon is the coordinate sphere $r = r_- = M(1 - \sqrt{1 - \tilde{a}^2})$.

We take the particle to orbit in the equatorial plane in the $d\phi/dt > 0$ direction, with $\tilde{a} > 0$ for prograde orbits and $\tilde{a} < 0$ for retrograde orbits. We parametrize the particle's (bound equatorial geodesic) orbit by the usual dimensionless semilatus rectum $p$ and eccentricity $e$; these are defined in detail in Appendix B 2. We refer to the combination of a spacetime and a particle orbit as a "configuration," and parametrize it with the triplet $(\tilde{a}, p, e)$. We define $T_r$ to be the coordinate-time period of the particle's radial motion;

we usually refer to $T_r$ as the particle's "orbital period." We define the "modulo time" to be the coordinate time modulo $T_r$.

To aid in assessing the accuracy of our computed self-forces, we define a positive-definite pointwise norm on covariant or contravariant 4-vectors,

$$\|v_a\|_+ := (|v_t v^t| + |v_i v^i|)^{1/2} \tag{1.2a}$$

$$\|v^a\|_+ := (|v_t v^t| + |v_i v^i|)^{1/2}, \tag{1.2b}$$

where all indices are raised and lowered with the Boyer-Lindquist 4-metric.

We use $x^a_{\text{particle}}(t)$ to denote the particle's worldline, which we consider to be known in advance, i.e., we do *not* consider changes to the particle's worldline induced by the self-force. $\mathcal{E}$ and $\mathcal{L}$ are the particle's specific energy and specific angular momentum (i.e., the particle's energy and angular momentum per unit mass).

When referring to finite-difference molecules (stencils) we use `i` and `j` as generic integer grid coordinates in the radial ($R_*$) (where $R_*$ is the compactified tortise coordinate defined by (2.16), (2.27), and (2.28)) and angular ($\theta$) directions, respectively. Considering a finite-difference molecule evaluated at the grid point $(\mathtt{i},\mathtt{j})$, we define the molecule's "radius" in a given direction ($\mathtt{i}+$, $\mathtt{i}-$, $\mathtt{j}+$, or $\mathtt{j}-$) as the maximum integer $\delta \geq 0$ such that the molecule has a nonzero coefficient at $\mathtt{i} \pm \delta$ or $\mathtt{j} \pm \delta$, respectively, and we refer to these as $R_{\mathtt{i}+}$, $R_{\mathtt{i}-}$, $R_{\mathtt{j}+}$, and $R_{\mathtt{j}-}$ respectively. For example, the usual 3-point centered 2nd-order molecule approximating the radial partial derivative $\partial_{R_*}$ has $R_{\mathtt{i}+} = R_{\mathtt{i}-} = 1$ and $R_{\mathtt{j}+} = R_{\mathtt{j}-} = 0$.

We use a pseudocode notation to describe algorithms: Lines are numbered for reference, but the line numbers are not used in the algorithm itself. # marks comment lines, while keywords are typeset in **bold font**. Procedures are marked with the keyword **procedure** and have bodies delimited by "{" and "}". Code layout and indentation are solely for clarity and (unlike Python) do not have any explicit semantics. Procedure names are typeset in `typewriter font`. Value-returning procedures (functions) have an explicitly declared return type (e.g., "**boolean procedure**") and return a value with a **return** statement. When referring to a procedure as a noun in a figure caption or in the main text of this paper, the procedure name is suffixed with "`()`", as in "`foo()`".

Variable names are either mathematical expressions, such as "$R_{i+}$", or are typeset in `typewriter font`. "`var` $\leftarrow X$" means that the variable `var` is assigned the value of the expression $X$. Variables are always declared before use. The declaration of a variable explicitly states the variable's type (**integer**, **floating_point**, **interval**, or **region**, the last of these being a rectangular region in the integer plane $\mathbb{Z} \times \mathbb{Z}$) and may also be combined with the assignment of an initial value, as in "**region** $W \leftarrow$ worldtube region". Conditional expressions have *C*-style syntax and semantics, *condition*? *expression-if-true*: *expression-if-false*, while conditional statements have explicit **if**, **then**, and **else** keywords.

In Appendix B 11 we use lower-case sans-serif letters $\mathsf{u}$, $\mathsf{k}$, and $\tilde{\mathsf{k}}$ for state vectors, and upper-case sans-serif letters $\mathsf{F}$ and $\mathsf{G}$ for state-vector-valued functions.

## II. THEORETICAL FORMALISM

Ignoring questions of divergence and regularization near the particle, in general the (4-vector) radiation-reaction self-force on a scalar particle moving in an arbitrary (specified) background spacetime is given by

$$F_a = q(\nabla_a \Phi)|_{\text{particle}}, \tag{2.1}$$

where the particle's scalar charge is $q$ (which may vary along the particle's worldline), and the (real) scalar field $\Phi$ satisfies the wave equation

$$\Box\Phi = q\delta(x^a - x^a_{\text{particle}}(t)), \tag{2.2}$$

where $\Box$ is the curved-space wave operator in the background spacetime [66].

Because of the $\delta$-function source in (2.2), $\Phi$ diverges on the particle's worldline, so that some type of regularization is essential in order to obtain a finite self-force.

### A. Effective-source regularization

We use the "effective-source" or "puncture-field" regularization scheme introduced by Barack and Golbourn [49] and Vega and Detweiler [50] (see [68] for a recent review). This regularization is based on the Detweiler-Whiting decomposition [42] of $\Phi$ into the sum of a "singular" and a "regular" field, $\Phi = \Phi_{\text{singular}} + \Phi_{\text{regular}}$, with the following properties:

(i) The singular field is divergent on the particle's worldline but is (in a suitable sense) spherically symmetric at the particle and hence exerts no self-force.

(ii) The regular field is finite—in fact $C^\infty$—at the particle and exerts the entire self-force. That is, the correct self-force may be obtained by applying (2.1) to the regular field,

$$F_a = q(\nabla_a \Phi_{\text{regular}})|_{\text{particle}}. \tag{2.3}$$

Unfortunately, it is very difficult to compute the exact Detweiler-Whiting singular or regular fields in Schwarzschild or Kerr spacetime. The basic concept of the effective-source regularization is to instead compute a "puncture-field" approximation $\Phi_{\text{puncture}} \approx \Phi_{\text{singular}}$, chosen (in a manner to be described in detail below) so that the "residual field" $\Phi_{\text{residual}} := \Phi - \Phi_{\text{puncture}}$ is finite and "somewhat differentiable" (in our case $C^2$) in a neighborhood of the particle. We then have

$$\begin{aligned}\Box\Phi_{\rm residual} &= \Box\Phi - \Box\Phi_{\rm puncture}\\ &= q\delta(x - x_{\rm particle}(t)) - \Box\Phi_{\rm puncture}\\ &= \begin{cases} 0 & \text{on the particle worldline}\\ & \text{(our choice of } \Phi_{\rm puncture}\\ & \text{will ensure this)}\\ -\Box\Phi_{\rm puncture} & \text{elsewhere}\end{cases} \qquad (2.4)\\ &:= S_{\rm effective}, \qquad (2.5)\end{aligned}$$

where we define the "effective source" $S_{\rm effective}$ to be the right-hand side of (2.4).

In more detail, we choose $\Phi_{\rm puncture}$ so that for some chosen integer $n \geq 3$,

$$\Phi_{\rm puncture} - \Phi_{\rm singular} = \mathcal{O}(\|x - x_{\rm particle}(t)\|^{n-1}) \qquad (2.6)$$

in a neighborhood of the particle. (This is equivalent to choosing $\Phi_{\rm puncture}$ so that its Laurent series about the particle position matches the first $n$ terms of $\Phi_{\rm singular}$'s Laurent series; both series begin with $\|x - x_{\rm particle}(t)\|^{-1}$ terms.) Since $\Phi_{\rm regular}$ is $C^\infty$ at the particle and $\Phi_{\rm residual} = \Phi_{\rm regular} + (\Phi_{\rm singular} - \Phi_{\rm puncture}) = \Phi_{\rm regular} + \mathcal{O}(\|x - x_{\rm particle}(t)\|^{n-1})$ in a neighborhood of the particle, we have $(\nabla\Phi_{\rm residual})|_{\rm particle} = (\nabla\Phi_{\rm regular})|_{\rm particle}$. By virtue of (2.3) the radiation-reaction self-force is thus given by

$$F_a = q(\nabla_a \Phi_{\rm residual})|_{\rm particle}. \qquad (2.7)$$

In this work we choose $n = 4$, so that $\Phi_{\rm residual}$ is $C^2$ at the particle and $S_{\rm effective}$ is $C^0$ at the particle. Note, however, that the criterion (2.6) still leaves considerable freedom in the choice (definition) of $\Phi_{\rm puncture}$. We describe our choice in detail in Sec. II G.

### B. *m*-mode Fourier decomposition

Given the basic effective-source formalism, some authors (e.g., [50,68–71]) choose to solve (2.5) via a direct numerical integration in $3+1$ dimensions. However, following [49,72–75], we prefer to instead exploit the axisymmetry of the background (Kerr) spacetime and introduce an $m$-mode (Fourier) decomposition.

To avoid infinite twisting of the Boyer-Lindquist $\phi$ coordinate at the event horizon, we follow [76] by introducing an "untwisted" azimuthal coordinate

$$\tilde{\phi} = \phi + f(r) \qquad (2.8)$$

with the function $f$ chosen such that

$$d\tilde{\phi} = d\phi + \frac{M\tilde{a}}{\Delta}dr. \qquad (2.9)$$

It is straightforward to integrate this to give

$$f(r) = \frac{\tilde{a}}{2\sqrt{1-\tilde{a}^2}}\ln\left|\frac{r - r_+}{r - r_-}\right| + \text{constant}. \qquad (2.10)$$

Using the $\tilde{\phi}$-derivative transformations derived in Appendix A, $\Box\Phi$ can be written in $(t, r, \theta, \tilde{\phi})$ coordinates[9] as

$$\begin{aligned}\Sigma\Box\Phi = &-\left[\frac{(r^2 + M^2\tilde{a}^2)^2}{\Delta} - M^2\tilde{a}^2\sin^2\theta\right]\partial_{tt}\Phi\\ &- \frac{4M^2\tilde{a}r}{\Delta}\partial_{t\tilde{\phi}}\Phi + \partial_r(\Delta\partial_r\Phi) + 2M\tilde{a}\partial_{r\tilde{\phi}}\Phi\\ &+ \partial_{\theta\theta}\Phi + \cot\theta\partial_\theta\Phi + \frac{1}{\sin^2\theta}\partial_{\tilde{\phi}\tilde{\phi}}\Phi. \qquad (2.11)\end{aligned}$$

We Fourier decompose the field in $e^{im\tilde{\phi}}$ modes, writing

$$\Phi(t, r, \theta, \phi) = \sum_{m=-\infty}^{\infty} e^{im\tilde{\phi}}\Psi_m(t, r, \theta) \qquad (2.12)$$

and analogously for the other fields $\Phi_{\rm puncture}$, $\Phi_{\rm residual}$, and $S_{\rm effective}$. For each integer $m$, the (complex) $m$-mode fields are given by

$$\Psi_m(t, r, \theta) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\Phi(t, r, \theta, \phi)e^{-im\tilde{\phi}}d\tilde{\phi} \qquad (2.13)$$

and analogously for the other fields $\Psi_{{\rm puncture},m}$, $\Psi_{{\rm residual},m}$, and $S_{{\rm effective},m}$. We then introduce the (complex) radial-factored field

$$\varphi_m = r\Psi_m \qquad (2.14)$$

(and analogously for $\varphi_{{\rm puncture},m}$ and $\varphi_{{\rm residual},m}$) so that the far-field falloffs around an asymptotically flat system are $\varphi_m = \mathcal{O}(1)$ when $\Psi_m = \mathcal{O}(1/r)$.

Following [77], we introduce the tortoise coordinate $r_*$ defined (up to an arbitrary additive constant) by

$$\frac{dr_*}{dr} = \frac{r^2 + M^2\tilde{a}^2}{\Delta}. \qquad (2.15)$$

Again following [77], we fix the additive constant by choosing

[9] In an early version of our theoretical formalism we wrote the equations using $\eta = \cos\theta$ as an angular variable. Provided that $\Phi$ is a nonsingular function of $\eta$ near the $z$ axis, this automatically enforces the boundary condition $\partial_\theta\Phi = 0$ there (cf. Sec. II I). However, $\partial_{\theta\theta}\Phi = \sin^2\theta\partial_{\eta\eta}\Phi - \cos\theta\partial_\eta\Phi$, so that on the $z$ axis $\partial_{\theta\theta}\Phi = -\partial_\eta\Phi$. This means that specifying $\partial_\eta\Phi$ on the $z$ axis (which should *a priori* be a reasonable boundary condition) would implicitly also specify $\partial_{\theta\theta}\Phi$ there, which should actually be determined by the field (evolution) equations. In other words, such a "boundary condition" would in fact overconstrain the evolution system. To avoid the possibility of such an overconstraint, we abandoned the $\eta = \cos\theta$ scheme.

$$r_* = r + 2M\frac{r_+}{r_+ - r_-}\ln\left(\frac{r-r_+}{2M}\right) - 2M\frac{r_-}{r_+ - r_-}\ln\left(\frac{r-r_-}{2M}\right). \tag{2.16}$$

We describe the numerical computation of $r(r_*)$ in Appendix B 1. For any scalar quantity $Q$ we have (using the chain rule)

$$\frac{\partial Q}{\partial r} = \frac{r^2 + M^2\tilde{a}^2}{\Delta}\frac{\partial Q}{\partial r_*}. \tag{2.17}$$

The scalar wave operator $\Box\Phi$ then becomes

$$\Box\Phi = \sum_{m=-\infty}^{\infty}\frac{e^{im\tilde{\phi}}}{r}\Box_m\varphi_m \tag{2.18}$$

and each $m$-mode of the residual field satisfies

$$\Box_m\varphi_{\text{residual},m} = S_{\text{effective},m}, \tag{2.19}$$

where

$$\begin{aligned}\Box_m\varphi_= &- \frac{1}{r\Sigma}\left[\frac{(r^2+M^2\tilde{a}^2)^2}{\Delta} - M^2\tilde{a}^2\sin^2\theta\right]\partial_{tt}\varphi \\ &- 4im\frac{M^2\tilde{a}}{\Delta\Sigma}\partial_t\varphi + \frac{(r^2+M^2\tilde{a}^2)^2}{r\Delta\Sigma}\partial_{r_*r_*}\varphi \\ &+ \left[-2\frac{M^2\tilde{a}^2}{r^2\Sigma} + 2imM\tilde{a}\frac{r^2+M^2\tilde{a}^2}{r\Delta\Sigma}\right]\partial_{r_*}\varphi \\ &+ \frac{1}{r\Sigma}\partial_{\theta\theta}\varphi + \frac{\cot\theta}{r\Sigma}\partial_\theta\varphi \\ &- \left[\frac{2}{r^2\Sigma}\left(M - \frac{M^2\tilde{a}^2}{r}\right) + \frac{m^2}{r\Sigma\sin^2\theta} + 2im\frac{M\tilde{a}}{r^2\Sigma}\right]\varphi.\end{aligned} \tag{2.20}$$

### C. The worldtube

Our construction of the puncture field and effective source ([53] and Sec. II G) is only valid in an finite $(r,\theta)$ neighborhood of the particle. Moreover, it is not clear what far-field boundary conditions the residual field should satisfy. Therefore, rather than solving (2.19) directly, for each $m$ we introduce a finite worldtube $W_m$ chosen so that its interior contains the particle worldline, and the puncture field and effective source are defined everywhere in the worldtube. [Notice that $W_m$ logically "lives" in the $m$-mode $(t,r,\theta)$ space, *not* in spacetime.]

For each $m$ we define the piecewise "numerical field"

$$\varphi_{\text{num},m} = \begin{cases}\varphi_{\text{residual},m} & \text{in the worldtube} \\ \varphi_m & \text{outside the worldtube}\end{cases}. \tag{2.21}$$

This field has a jump discontinuity across the worldtube boundary,

$$\lim_{\substack{x^\ell\to b^\ell \\ x^\ell\in W_m}}\varphi_{\text{num},m}(x^\ell) = \left[\lim_{\substack{x^\ell\to b^\ell \\ x^\ell\notin W_m}}\varphi_{\text{num},m}(x^\ell)\right] - \varphi_{\text{puncture},m}(b^\ell) \tag{2.22}$$

for any worldtube-boundary point $b^\ell\in\partial W_m$, and it also satisfies

$$\Box_m\varphi_{\text{num},m} = \begin{cases}S_{\text{effective},m} & \text{inside the worldtube} \\ 0 & \text{outside the worldtube.}\end{cases} \tag{2.23}$$

We numerically solve (2.23) via a separate Cauchy time evolution for each $m$. The form of (2.23) ensures that the effective source only needs to be computed inside the worldtube, and (as discussed in detail in Sec. II D and Appendices B 6 and B 8) the puncture field only needs to be computed within a small neighborhood of the worldtube boundary.

The precise choice of the worldtube may be made for computational convenience; by construction, the computed self-force is independent of this choice (see Sec. III D for a numerical verification of this independence). The worldtube's size should reflect a tradeoff between numerical cost and accuracy:

(i) A larger worldtube requires computing $S_{\text{effective},m}$ (which is expensive) at a larger set of events.
(ii) A smaller worldtube (more precisely, one whose complement includes points closer to the particle) requires numerically computing—and hence finite differencing—$\varphi_m$ closer to its singularity at the particle, leading to larger numerical errors.

For a given worldtube shape and size, the best accuracy is generally obtained by choosing the worldtube to be approximately centered on the particle.

In practice we typically choose a worldtube which is a rectangle in $(r_*,\theta)$ of half-width $5M$ in $r_*$ and approximately $\pi/8$ in $\theta$.

Since we use Berger-Oliger mesh refinement (Appendix B 9), the question arises of how the worldtube should interact with the mesh refinement. In particular, should the worldtube differ from one refinement level to another? For simplicity we have chosen a computational scheme where this is *not* the case—in our scheme the worldtube is the same at all refinement levels. This means that the Berger-Oliger mesh-refinement algorithm does *not* need to make the adjustment (2.24) when copying or interpolating data between different refinement levels. The worldtube boundary is effectively quantized to the coarsest (base) grid, but we do not find this to be a problem in practice.

### D. Moving the worldtube

If the particle's orbit has a sufficiently small eccentricity then a reasonably sized time-independent worldtube in $(r_*,\theta)$ can encompass the particle's entire orbital motion.

However, our main interest is in the case where the particle's orbit is highly eccentric. This requires the worldtube to be time dependent in order to enclose the particle throughout the particle's entire orbital motion. In our computational scheme we move the worldtube in $(r_*,\theta)$ in discontinuous jumps so as to always keep the worldtube's coordinate center within a small distance (typically $\sim 0.5M$) of the particle position. (More precisely, this is the case after the startup phase of the computation; we discuss this in detail in Sec. B 5.)

When the worldtube moves, those "transition" grid points which were formerly inside the worldtube and are now outside, or vice versa, essentially have the computation of $\Box\varphi_{\text{puncture}}$ switched between being done analytically versus via finite differencing. In the continuum limit these two computations agree, but at finite resolutions they differ slightly. Therefore, moving the worldtube introduces numerical noise into the evolved field $\varphi_{\text{num},m}$.

Our actual worldtube-moving algorithm (described in detail in Appendix B 4) incorporates a number of refinements to help mitigate this numerical noise and achieve the most accurate numerical evolutions possible:

(i) Basically, the algorithm moves the worldtube any time the particle position is "too far" from the worldtube center.
(ii) When moving the worldtube, the algorithm places the new worldtube center somewhat ahead of the particle in the direction of the particle's motion. The algorithm includes a small amount of hysteresis so as to avoid unnecessary back-and-forth worldtube moves.
(iii) The algorithm limits the maximum distance the worldtube can be moved at any one time.
(iv) The algorithm imposes a minimum time interval between worldtube moves.

Because $\varphi_{\text{num},m}$ has the jump discontinuity (2.22) across the worldtube boundary, each time the worldtube is moved the evolved fields $\varphi_{\text{num},m}$ and $\Pi_{\text{num},m}$ must be adjusted at transition grid points:

$$\varphi_{\text{num},m} \leftarrow \varphi_{\text{num},m} \pm \varphi_{\text{puncture},m} \tag{2.24a}$$

$$\Pi_{\text{num},m} \leftarrow \Pi_{\text{num},m} \pm \partial_t \varphi_{\text{puncture},m}, \tag{2.24b}$$

where the "+" applies to grid points which were formerly inside the worldtube and are now outside it, and the "−" applies to grid points which were formerly outside the worldtube and are now inside it.

### E. Hyperboloidal slices and compactification

Conceptually, (2.23) should be solved on the entire spacetime, with outflow boundary conditions on the event horizon and null infinity ($\mathcal{J}^+$). To accomplish this computationally, we use a hyperboloidal compactification scheme developed by Zenginoğlu [78–85]. This scheme has a number of desirable properties, including:

(1) The hyperboloidal slices reach the event horizon and $\mathcal{J}^+$, allowing pure-outflow boundary conditions to be posed there.
(2) The transformed evolution equations do *not* suffer the "infinite blue-shifting" problem (cf. the discussion of [80]) in the compactification region—they have finite and nonzero propagation speeds throughout the computational domain, and outgoing waves suffer at most $\mathcal{O}(1)$ compression (blue shifting) or expansion (red shifting) as they propagate from the region of the particle to the event horizon and to $\mathcal{J}^+$.
(3) The transformed evolution equations can be formulated to be nonsingular everywhere, with all coefficients having finite limiting values near to and on both the event horizon and $\mathcal{J}^+$.
(4) The (time-independent) compactification transformation can be chosen to be the identity transformation throughout a neighborhood of the entire range of the particle's orbital motion. This means that the computation of the effective source and puncture field, the various adjustments to the computations when crossing the worldtube boundary or when moving the worldtube, and the computation of the self-force from the evolved field $\varphi_m$, are all unaffected by the compactification.
(5) The scheme is easy to implement, requiring only relatively modest modifications to our previous (non-compactified) numerical code.

We primarily follow the version of Zenginoğlu's compactification scheme described in [81], although with slightly different notation to more conveniently allow a unified treatment of compactification near the event horizon and near $\mathcal{J}^+$.

For purposes of compactification, it is convenient to rewrite the evolution equation (2.23) and (2.20) in the generic form

$$\begin{aligned} &C[\partial_{tt}\varphi]\partial_{tt}\varphi + C[\partial_{tr_*}\varphi]\partial_{tr_*}\varphi + C[\partial_{r_*r_*}\varphi]\partial_{r_*r_*}\varphi \\ &\quad + C[\partial_t\varphi]\partial_t\varphi + C[\partial_{r_*}\varphi]\partial_{r_*}\varphi + C[\partial_{\theta\theta}\varphi]\partial_{\theta\theta}\varphi + C[\partial_\theta\varphi]\partial_\theta\varphi \\ &\quad + C[\varphi]\varphi + C[1] = 0, \end{aligned} \tag{2.25}$$

where we have dropped the subscript on $\varphi_m$, and where the $C[\cdot]$ coefficients can be read off from the evolution equations. ($C[\partial_{tr_*}\varphi] = 0$ for our evolution equations, but is included for generality.)

To make the equations nonsingular near to and on the event horizon, we multiply (2.23) through by a factor of $r\Sigma\Delta$. It is also useful for the coefficients to be finite near to and at $\mathcal{J}^+$, so we further multiply through by a factor of $(r^2 + M^2\tilde{a}^2)^{-2}$. The resulting coefficients are

$$C[\partial_{tt}\varphi] = \frac{M^2\tilde{a}^2\Delta\sin^2\theta}{(r^2 + M^2\tilde{a}^2)^2} - 1, \tag{2.26a}$$

$$C[\partial_t\varphi] = -i\frac{4mM^2\tilde{a}r}{(r^2+M^2\tilde{a}^2)^2}, \tag{2.26b}$$

$$C[\partial_{r_*r_*}\varphi] = 1, \tag{2.26c}$$

$$C[\partial_{r_*}\varphi] = -\frac{2M^2\tilde{a}^2\Delta}{r(r^2+M^2\tilde{a}^2)^2} + i\frac{2mM\tilde{a}}{r^2+M^2\tilde{a}^2}, \tag{2.26d}$$

$$C[\partial_{\theta\theta}\varphi] = \frac{\Delta}{(r^2+M^2\tilde{a}^2)^2}, \tag{2.26e}$$

$$C[\partial_\theta\varphi] = \frac{\Delta\cot\theta}{(r^2+M^2\tilde{a}^2)^2}, \tag{2.26f}$$

$$C[\varphi] = -\frac{2\Delta}{r(r^2+M^2\tilde{a}^2)^2}\left(M-\frac{M^2\tilde{a}^2}{r}\right) - \frac{m^2\Delta}{(r^2+M^2\tilde{a}^2)^2\sin^2\theta} - i\frac{2mM\tilde{a}\Delta}{r(r^2+M^2\tilde{a}^2)^2}, \tag{2.26g}$$

$$C[1] = \begin{cases} -\dfrac{r\Sigma\Delta}{(r^2+M^2\tilde{a}^2)^2}S_{\text{effective},m} & \text{inside} \\ 0 & \text{outside.} \end{cases} \tag{2.26h}$$

We define the compactified radial coordinate $R_*$ by

$$r_* = \frac{R_*}{\Omega(R_*)}, \tag{2.27}$$

where we choose the (time-independent) conformal factor $\Omega$ so that the event horizon and $\mathcal{J}^+$ are at the (finite) $R_*$ coordinates $R_*^h$ and $R_*^{\mathcal{J}^+}$ respectively. More precisely, we introduce the four parameters $R_*^h < R_*^- < 0 < R_*^+ < R_*^{\mathcal{J}^+}$, chosen such that the particle and worldtube always lie within the region $R_*^- < R_* < R_*^+$ (where we will choose the compactification transformation to be the identity transformation). We define

$$\Omega(R_*) = \begin{cases} 1-\left(\dfrac{R_*^- - R_*}{R_*^- - R_*^h}\right)^4 & \text{if } R_* < R_*^- \\ 1 & \text{if } R_*^- \le R_* \le R_*^+ \\ 1-\left(\dfrac{R_* - R_*^+}{R_*^{\mathcal{J}^+} - R_*^+}\right)^4 & \text{if } R_* > R_*^+ \end{cases} \tag{2.28}$$

so that the compactification transformation is indeed the identity transformation ($\Omega = 1$ and $r_* = R_*$) throughout the region $R_*^- < R_* < R_*^+$. We refer to $R_*^-$ and $R_*^+$ as the inner and outer compactification radii, respectively. Our numerical grid spans the full range $R_*^h \le R_* \le R_*^{\mathcal{J}^+}$.

To ensure the absence of infinite blue shifting ("desirable property" 2), the time coordinate must also be transformed. We define the transformed time coordinate $T$ by

$$T = t - h(R_*), \tag{2.29}$$

where the "height" function $h$ is given by

$$h(R_*) = \begin{cases} R_* - r_* & \text{if } R_* < R_*^- \\ 0 & \text{if } R_*^- \le R_* \le R_*^+ \\ r_* - R_* & \text{if } R_* > R_*^+ \end{cases} = \begin{cases} R_*\left(1-\dfrac{1}{\Omega}\right) & \text{if } R_* < R_*^- \\ 0 & \text{if } R_*^- \le R_* \le R_*^+ \\ R_*\left(\dfrac{1}{\Omega}-1\right) & \text{if } R_* > R_*^+ \end{cases} \tag{2.30}$$

In order to express the equations in a simple form, it is convenient to define the "generalized boost" function

$$\bar{H} = \frac{dR_*}{dr_*} = \frac{\Omega^2}{\Omega - R_*\Omega'}, \tag{2.31}$$

where $X' := dX/dR_*$ for any quantity $X$, so that

$$\bar{H}' = \frac{2\Omega\Omega'}{\Omega - R_*\Omega'} + \frac{R_*\Omega^2\Omega''}{(\Omega - R_*\Omega')^2}. \tag{2.32}$$

We define the "boost" function $H$ by

$$H = \frac{dh}{dr_*} = \begin{cases} \bar{H} - 1 & \text{if } R_* < R_*^- \\ 0 & \text{if } R_*^- \le R_* \le R_*^+ \\ 1 - \bar{H} & \text{if } R_* > R_*^+ \end{cases} \tag{2.33}$$

so that

$$H' = \begin{cases} \bar{H}' & \text{if } R_* < R_*^- \\ 0 & \text{if } R_*^- \le R_* \le R_*^+ \\ -\bar{H}' & \text{if } R_* > R_*^+ \end{cases}. \tag{2.34}$$

Figure 1 shows an example of these quantities and the resultant compactification.

Transforming the generic evolution equations (2.25) from $(t, r_*, \theta, \phi)$ coordinates to $(T, R_*, \theta, \phi)$ coordinates, we see immediately that the coefficients $C[\partial_{\theta\theta}\varphi]$, $C[\partial_\theta\varphi]$, $C[\varphi]$, and $C[1]$ are all unchanged by the transformation.

At all points other than the event horizon or $\mathcal{J}^+$, the nontrivially transformed coefficients are

$$C[\partial_{TT}\varphi] = \frac{1}{\bar{H}}C[\partial_{tt}\varphi] - \frac{H}{\bar{H}}C[\partial_{tr_*}\varphi] + \frac{H^2}{\bar{H}}C[\partial_{r_*r_*}\varphi], \tag{2.35a}$$

$$C[\partial_{TR_*}\varphi] = C[\partial_{tr_*}\varphi] - 2HC[\partial_{r_*r_*}\varphi], \tag{2.35b}$$

$$C[\partial_{R_*R_*}\varphi] = \bar{H}C[\partial_{r_*r_*}\varphi], \tag{2.35c}$$

$$C[\partial_T\varphi] = -H'C[\partial_{r_*r_*}\varphi] + \frac{1}{\bar{H}}C[\partial_t\varphi] - \frac{H}{\bar{H}}C[\partial_{r_*}\varphi], \tag{2.35d}$$

$$C[\partial_{R_*}\varphi] = \bar{H}'C[\partial_{r_*r_*}\varphi] + C[\partial_{r_*}\varphi]. \tag{2.35e}$$

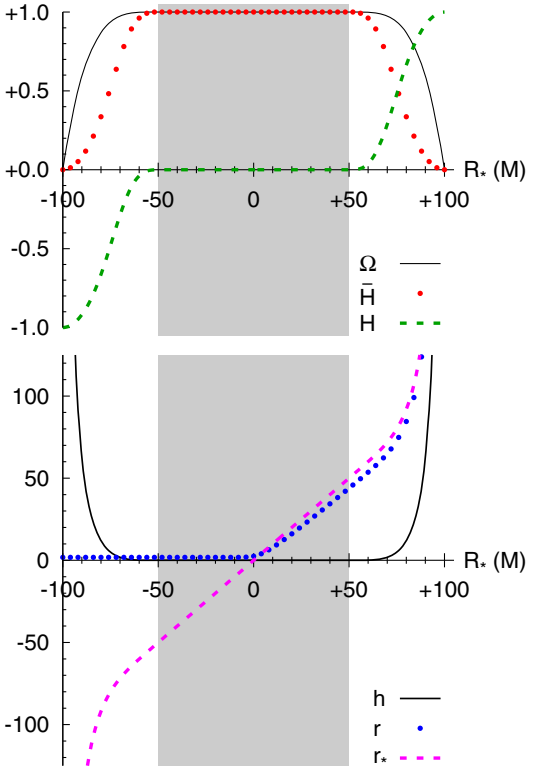


FIG. 1. This figure shows an example of the compactification for a Kerr spacetime with dimensionless spin $\tilde{a} = 0.6$. The compactification parameters (here chosen for visual clarity rather than optimum computational efficiency/accuracy) are $(R_*^h, R_*^-, R_*^+, R_*^{\mathcal{J}^+}) = (-100, -50, +50, +100)M$. The upper subfigure shows $\Omega$, $\bar{H}$, and $H$, and the lower subfigure shows $h$, $r$, and $r_*$, all as functions of $R_*$. The compactification transformation is only nontrivial outside the shaded region $R_*^- \le R_* \le R_*^+$; the transformation is the identity transformation ($\Omega = \bar{H} = 1$, $H = h = 0$, and $r_* = R_*$) in the shaded region. For $R_* \ll 0$, $r \to r_h$ $(= 1.8M)$.

On the event horizon the limiting values of these (transformed) coefficients are

$$C[\partial_{TT}\varphi] = -2, \tag{2.36a}$$

$$C[\partial_{TR_*}\varphi] = +2 \tag{2.36b}$$

$$C[\partial_{R_*R_*}\varphi] = 0, \tag{2.36c}$$

$$C[\partial_T\varphi] = -i\frac{m\tilde{a}}{r_h}, \tag{2.36d}$$

$$C[\partial_{R_*}\varphi] = i\frac{2mM\tilde{a}}{r_h^2 + M^2\tilde{a}^2}, \tag{2.36e}$$

$$C[\partial_{\theta\theta}\varphi] = C[\partial_\theta\varphi] = 0, \tag{2.36f}$$

$$C[\varphi] = C[1] = 0, \tag{2.36g}$$

while at $\mathcal{J}^+$ the limiting values are

$$C[\partial_{TT}\varphi] = 2\frac{M^2\tilde{a}^2}{\bar{H}''^{\mathcal{J}^+}}\left(\frac{\Omega'^{\mathcal{J}^+}}{R_*^{\mathcal{J}^+}}\right)^2 \sin^2\theta - 2, \tag{2.37a}$$

$$C[\partial_{TR_*}\varphi] = -2, \tag{2.37b}$$

$$C[\partial_{R_*R_*}\varphi] = 0, \tag{2.37c}$$

$$C[\partial_T\varphi] = -i\frac{4mM\tilde{a}}{\bar{H}''^{\mathcal{J}^+}}\left(\frac{\Omega'^{\mathcal{J}^+}}{R_*^{\mathcal{J}^+}}\right)^2, \tag{2.37d}$$

$$C[\partial_{R_*}\varphi] = 0, \tag{2.37e}$$

$$C[\partial_{\theta\theta}\varphi] = \frac{2}{\bar{H}''^{\mathcal{J}^+}}\left(\frac{\Omega'^{\mathcal{J}^+}}{R_*^{\mathcal{J}^+}}\right)^2, \tag{2.37f}$$

$$C[\partial_\theta\varphi] = \frac{2}{\bar{H}''^{\mathcal{J}^+}}\left(\frac{\Omega'^{\mathcal{J}^+}}{R_*^{\mathcal{J}^+}}\right)^2 \frac{1}{\tan\theta}, \tag{2.37g}$$

$$C[\varphi] = -\frac{2m^2}{\bar{H}''^{\mathcal{J}^+}}\left(\frac{\Omega'^{\mathcal{J}^+}}{R_*^{\mathcal{J}^+}}\right)^2 \frac{1}{\sin^2\theta}, \tag{2.37h}$$

$$C[1] = 0, \tag{2.37i}$$

where

$$\Omega'^{\mathcal{J}^+} := \lim_{R_*\to R_*^{\mathcal{J}^+}} \Omega' = \frac{4}{R_*^{\mathcal{J}^+} - R_*^+}, \tag{2.38a}$$

$$\bar{H}''^{\mathcal{J}^+} := \lim_{R_*\to R_*^{\mathcal{J}^+}} \bar{H}'' = \frac{2\Omega'^{\mathcal{J}^+}}{R_*^{\mathcal{J}^+}}. \tag{2.38b}$$

While conceptually straightforward, the calculation of $\bar{H}''^{\mathcal{J}^+}$ is somewhat lengthy; we used the Maple symbolic-algebra system (Version 18 for x86-64 Linux, [86]) to obtain the result given here.

### F. 1st-order-in-time equations

To numerically solve the evolution equation (2.23) it is convenient to introduce the auxiliary variable

$$\Pi_{\text{num},m} = \partial_t\varphi_{\text{num},m} \tag{2.39}$$

so as to obtain a 1st-order-in-time evolution system. The compactified evolution equation then becomes

$$\begin{aligned} &C[\partial_{TT}\varphi]\partial_T\Pi + C[\partial_{TR_*}\varphi]\partial_{R_*}\Pi + C[\partial_{R_*R_*}\varphi]\partial_{R_*R_*}\varphi \\ &+ C[\partial_T\varphi]\Pi + C[\partial_{R_*}\varphi]\partial_{R_*}\varphi + C[\partial_{\theta\theta}\varphi]\partial_{\theta\theta}\varphi + C[\partial_\theta\varphi]\partial_\theta\varphi \\ &+ C[\varphi]\varphi + C[1] = 0, \end{aligned} \tag{2.40}$$

where we have dropped the subscripts on $\varphi_{\text{num},m}$ and $\Pi_{\text{num},m}$.

Our final evolution system comprises (2.39) and (2.40) using the coefficients (2.35), (2.36), and (2.37), modified by applying L'Hopital's rule on the $z$ axis, applying boundary conditions (Sec. II I), the gradual turn-on of the effective source (Appendix B 3), the adjustment of $\varphi$

and $\Pi$ when the worldtube is moved (Sec. II D), and the addition of numerical dissipation (Appendix B 9).

### G. Computing the puncture field and effective source

There is considerable freedom in the particular choice of puncture field used to construct an effective source. As mentioned in Sec. II A, we work with a puncture field which agrees with the Detweiler-Whiting singular field in the first four orders in its expansion about the worldline. This ensures that the computed self-force is finite and uniquely determined, and that the numerical methods used to compute it converge reasonably well. Other than that, we shall exploit the freedom to modify the higher-order terms in the expansion to adapt it to the $m$-mode scheme.

We begin with a coordinate series approximation for the Detweiler-Whiting singular field of a scalar charge on an eccentric equatorial geodesic of the Kerr spacetime, as can be obtained using, e.g., the methods of [56,87]. Our starting point is thus a coordinate series expansion of the form

$$\Phi^{[n]}_{\text{singular}}(x;x_{\text{particle}}) = \sum_{i=1}^{n} \frac{B^{a(3i-3)}}{\rho^{2i-1}}\epsilon^{i-2} + \mathcal{O}(\epsilon^{n-1}), \quad (2.41)$$

where

$$B^{a(k)} \equiv b^{a}_{a_1 a_2 \cdots a_k}(x_{\text{particle}})\Delta x^{a_1}\Delta x^{a_2}\cdots\Delta x^{a_k}, \quad (2.42)$$

$$\rho^2 = (g_{ab} + u_a u_b)\Delta x^a \Delta x^b, \quad (2.43)$$

and $g_{ab}$ and $u^a$ are evaluated on $x_{\text{particle}}$. Here, we introduce $\epsilon := 1$ as a formal power-counting parameter used to keep track of powers of distance from the particle; this amounts to inserting a factor of $\epsilon$ for each power of $\Delta x^a = [0, r - r_{\text{particle}}(t), \theta - \pi/2, \phi - \phi_{\text{particle}}(t)]^a$ appearing either explicitly or implicitly (through powers of $\rho$). Since we are choosing to include the first four orders in the expansion of the Detweiler-Whiting singular field, we take $n = 4$ and our approximation neglects terms of order $\epsilon^3$ and higher.

We next make two crucial modifications that make the puncture more amenable to analytic $m$-mode decomposition. To motivate these modifications, consider the general form of the function $\rho$ in the case of equatorial orbits in Kerr spacetime, which in Boyer-Lindquist coordinates is given by

$$\rho^2 = (g_{rr} + u_r u_r)\Delta r^2 + g_{\theta\theta}\Delta\theta^2 + (g_{\phi\phi} + u_\phi u_\phi)\Delta\phi^2 + u_r u_\phi \Delta r \Delta\phi. \quad (2.44)$$

Now, the integration involved in the $m$-mode decomposition of the $m=0$ mode of the leading-order $1/\rho$ term in the expansion of the singular field almost has the form of a complete elliptic integral of the first kind, $\mathcal{K}(k) \equiv \int_0^{\pi/2} \times (1 - k\sin^2\phi)^{-1/2} d\phi$, where the argument $k$ is a function of $x^a_{\text{particle}}$, $u^a$, $\Delta r$ and $\Delta\theta$. It would be desirable to have it in the exact form of an elliptic integral, as then it can be efficiently evaluated without having to resort to numerical quadrature. Fortunately, the only modifications required to turn it into elliptic-integral form are to rewrite $\Delta\phi^2$ in terms of $\sin^2\Delta\phi$ (or equivalently $\sin^2\frac{\Delta\phi}{2}$ up to an overall factor of 2 in the resulting integral), and to eliminate the $\Delta r\Delta\phi$ cross term. Both of these can be done using methods previously used in self-force calculations; the former can be achieved using the "Q-R" scheme described in [88], and the latter by combining this with a radially dependent change of variable, $\Delta\phi \to \Delta\hat{\phi} - c\Delta r$, where

$$c = \frac{L r_0^3 u^r}{[a^2 + r_0(r_0 - 2M)][a^2(2M + r_0) + r_0(L^2 + r_0^2)]} \quad (2.45)$$

is chosen such that the cross term vanishes. This second trick was first used by Mino, Nakano and Sasaki [89] and later also employed by Haas and Poisson [44].

Given these two modifications to $\rho$, we are then left with an expression for $\hat{\rho} = \rho + \mathcal{O}(\epsilon^2)$ that is of the form

$$\hat{\rho}^2 = A(r_0, u^a, \Delta r, \Delta\theta) + B(r_0, u^a)\sin^2(\Delta\hat{\phi}/2), \quad (2.46)$$

where $A(r_0, u^a, \Delta r, \Delta\theta)$ is a quadratic polynomial in $\Delta r$ and $\Delta\theta$. Note that our manipulations introduce an additional $r$ and $t$ dependence hidden inside the definition of $\Delta\hat{\phi}$; it is important to take this into account when computing derivatives of the puncture field, and also when evaluating it for $\Delta r \neq 0$. The advantage of working with $\hat{\rho}$ instead of $\rho$ is that the $m = 0$ mode of $1/\hat{\rho}$ is analytically given by a complete elliptic integral of the first kind,

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \hat{\rho}^{-1} d\hat{\phi} = \frac{2}{\pi\sqrt{A + B}}\mathcal{K}\left(\frac{B}{A + B}\right), \quad (2.47)$$

and similarly the $m = 0$ mode of $\hat{\rho}$ is analytically given by a complete elliptic integral of the second kind,

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \hat{\rho}\, d\hat{\phi} = \frac{2\sqrt{A + B}}{\pi}\mathcal{E}\left(\frac{B}{A + B}\right). \quad (2.48)$$

Returning to the problem of obtaining an $m$-mode decomposed puncture field, we have to generalize this in three ways: (i) We need to handle other integer powers of $\hat{\rho}$; (ii) We need to handle the additional dependence of $\Phi_{\text{singular}}$ on $\Delta\hat{\phi}$ other than that appearing in $\hat{\rho}$; (iii) We need to handle all $m \geq 0$ modes (the fact that the full 4-dimensional scalar field is real means that the $m < 0$ modes are trivially related to the $m > 0$ modes). To make things explicit, we use the two previously described modifications to rewrite our approximation to the singular field, (2.41), in the form

$$\Phi_{\text{singular}}(x;x_{\text{particle}}) = \frac{1}{\hat{\rho}^{2n-1}}\left[\sum_{\substack{i=0\\ i\,\text{even}}}^{3n-3} C_{n,i}\sin^i(\Delta\hat{\phi}/2) + 2\sum_{\substack{i=0\\ i\,\text{odd}}}^{3n-3} C_{n,i}\sin^i(\Delta\hat{\phi}/2)\cos(\Delta\hat{\phi}/2)\right] + \mathcal{O}(\epsilon^{n-1}), \quad (2.49)$$

where the coefficients $C_{n,i}$ are functions of $r_0$, $u^a$, $\Delta r$ and $\Delta\theta$, and where we have replaced $R = \sin\Delta\hat{\phi}$ with the equivalent expression $2\sin(\Delta\hat{\phi}/2)\cos(\Delta\hat{\phi}/2)$. To define our puncture field, we truncate this expansion at order $n = 4$ and decompose into $m$-modes,

$$\Psi_{\text{puncture},m} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \Phi^{[4]}_{\text{singular}} e^{-im\hat{\phi}} d\hat{\phi}. \tag{2.50}$$

Writing

$$e^{-im\hat{\phi}} = e^{-im\hat{\phi}_0}\sum_{k=0}^{2m}\binom{2m}{k}(-1)^{k/2}\cos^{2m-k}(\Delta\hat{\phi}/2) \times \sin^k(\Delta\hat{\phi}/2), \tag{2.51}$$

and inspecting the form of the integrals, we see that (apart from a trivial phase factor) the real part of the puncture is determined purely by the first term in (2.49), while the imaginary part is determined purely by the second term. Furthermore, in all cases we are left with integrals involving only even powers of $\sin(\Delta\hat{\phi}/2)$ and $\cos(\Delta\hat{\phi}/2)$. Then, the three generalizations listed previously can be handled through the application of two sets of identities,

$$\int_{-\pi}^{\pi}\sin^{2i}(\Delta\hat{\phi}/2)\cos^{2j}(\Delta\hat{\phi}/2)\hat{\rho}^k d\hat{\phi} = \int_{-\pi}^{\pi}\left[\frac{\hat{\rho}^2 - A}{B}\right]^i\left[\frac{A + B - \hat{\rho}^2}{B}\right]^j\hat{\rho}^k d\hat{\phi} \tag{2.52}$$

and

$$\int_{-\pi}^{\pi}\hat{\rho}^k d\hat{\phi} = \int_{-\pi}^{\pi}\frac{1}{A(k+2)(A+B)}[(k+3)(2A+B)\hat{\rho}^{k+2} - (k+4)\hat{\rho}^{k+4}]d\hat{\phi} \quad \text{for } k < -1, \tag{2.53a}$$

$$\int_{-\pi}^{\pi}\hat{\rho}^k d\hat{\phi} = \int_{-\pi}^{\pi}\frac{1}{k}[A(2-k)(A+B)\hat{\rho}^{k-4} + (k-1) \times (2A+B)\hat{\rho}^{k-2}]d\hat{\phi} \quad \text{for } k > 1. \tag{2.53b}$$

The first of these is a direct consequence of the definition of $\hat{\rho}$, while the second pair can be obtained from, e.g., Eq. (1) of ([90], Sec. 1.5.27). The first identity eliminates all powers of $\sin(\Delta\hat{\phi}/2)$ and $\cos(\Delta\hat{\phi}/2)$ not appearing inside $\hat{\rho}$, while the second pair of identities may be recursively applied to rewrite arbitrary (odd integer) powers of $\hat{\rho}$ in terms of $\hat{\rho}^{-1}$ and $\hat{\rho}$. Thus we can reduce all cases to elliptic-integral form and obtain analytic expressions for the puncture-field modes in terms of these easily evaluated elliptic integrals. In practice, the expressions take the form of an $m$-dependent polynomial in $\frac{A}{B}$ multiplied by $\mathcal{K}$ plus a second polynomial in $\frac{A}{B}$ multiplied by $\mathcal{E}$.

Given the puncture field $\Phi_{\text{puncture}}$, we compute the effective source $S_{\text{effective}}$ via (2.5) and then the $m$-mode effective source $S_{\text{effective},m}$ via the Fourier integral (2.13). Note that the $\Box$ operator (2.11) must be applied *analytically* to the series expansion for the puncture field in order to correctly cancel all divergent terms; a numerical calculation of the $\Box$ operator would be insufficiently accurate. The entire computation of $\Phi_{\text{puncture},m}$ and $S_{\text{effective},m}$ takes approximately 500 lines of Mathematica code. The Mathematica notebook is included in the Supplemental Material [91].

Our final expressions for $\Phi_{\text{puncture},m}$ and $S_{\text{effective},m}$ involve multivariate polynomials in $\Delta r$ and $\Delta\theta$, the $\mathcal{E}$ and $\mathcal{K}$ elliptic integrals (and their derivatives for $S_{\text{effective},m}$) and trigonometric polynomials. The coefficients in these expressions are functions (only) of the particle position and 4-velocity, so that at each distinct time at which $\Phi_{\text{puncture},m}$ and/or $S_{\text{effective},m}$ need to be computed, we first precompute these coefficients. This precomputation is done using $C$ code and numerical coefficients which are machine generated (once) by the Mathematica program. The machine-generated $C$ code is large (∼10 megabytes) and involves very lengthy arithmetic expressions (it contains $\sim 1.5\times 10^6$ arithmetic operations); compiling it is slow and requires large amounts of memory. Fortunately, the execution of the code (to actually precompute the coefficients) uses only a small fraction of our code's total CPU time, so this (machine-generated) code may be compiled without optimization.

The actual evaluation of $\Phi_{\text{puncture},m}$ and $S_{\text{effective},m}$ at each grid point is done using hand-written $C$ code. In total (i.e., summed over all grid points and RHS-evaluation times where this evaluation is needed) this evaluation uses the majority of our code's total CPU time; the finite differencing and numerical time integration are relatively minor contributors.

### H. Computing the effective source close to the particle

As we have noted previously ([88] Sec. III.C.3), our series expressions for the effective source suffer from severe cancellations when evaluated close to the particle. Because of the Fourier integral (2.13), $S_{\text{effective},m}$ need not—and typically does not—vanish at the particle, so the "interpolate along a ray" scheme we described in [88] (Sec. III.C.3) is not valid here.

Instead, we use the following scheme. We define a minimum-distance parameter $D_{\min}$ (typically set to $0.01M$), and if $(\Delta r)^2 + (r\Delta\theta)^2 < D_{\min}^2$, then we interpolate $S_{\text{effective},m}$ at $(\Delta r, \Delta\theta)$ using a 4th-order Lagrange interpolating polynomial defined by the values of $S_{\text{effective},m}$ at the 5 points $(-2D_{\min}, \Delta\theta)$, $(-D_{\min}, \Delta\theta)$, $(+D_{\min}, \Delta\theta)$, $(+2D_{\min}, \Delta\theta)$, and $(+3D_{\min}, \Delta\theta)$. As shown in Fig. 2, with this scheme the source is never evaluated closer than a Euclidean distance $D_{\min}$ from the particle. The interpolation is only needed at at most a few points per slice, so the computational cost is negligible.

While this scheme has proved adequate for our purposes, it does have the weakness that if the evaluation point lies in (or very close to) the equatorial plane $\theta = \pi/2$, then the interpolation molecule crosses (or almost crosses) the particle position, leading to reduced accuracy because $\varphi_{\text{num},m}$ is only $C^2$ there.

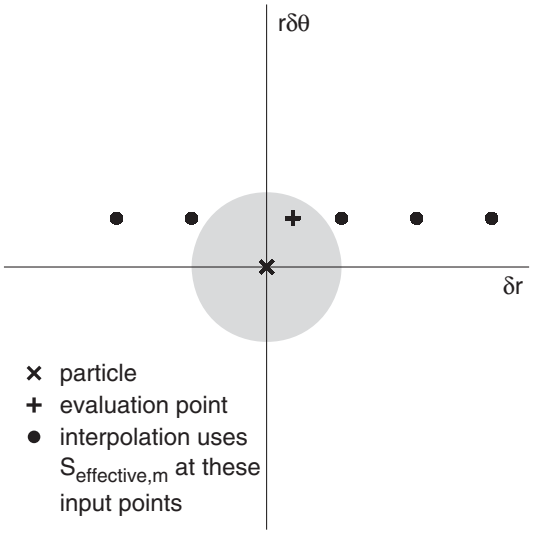


FIG. 2. This figure shows our interpolation scheme for computing the 2-dimensional effective source $S_{\text{effective},m}$ near the particle. We never evaluate $S_{\text{effective},m}$ at a Euclidean distance $< D_{\min}$ from the particle, i.e., within the region shown as the shaded disk. Instead, for an evaluation point within this region we interpolate $S_{\text{effective},m}$ using $S_{\text{effective},m}$ values computed at more distant points.

### I. Boundary conditions

We implement boundary conditions using finite-differencing ghost zones which lie immediately adjacent to, but outside, the nominal problem domain. At each RHS-evaluation time we first use the boundary conditions to compute $\varphi_{\text{num},m}$ and $\Pi_{\text{num},m}$ at all ghost-zone grid points. We then evaluate the RHS (and use this to time-integrate the evolution equations) at all grid points in the nominal problem domain.

#### *1. Physical boundary conditions*

We use pure-outflow boundary conditions at the event horizon and $\mathcal{J}^+$, i.e., we apply the interior evolution equations at these grid points, using (conceptually) 1-sided finite-difference molecules for radial derivatives.[10]

#### *2. z axis symmetry boundary conditions*

As discussed by ([49] Sec. IV.C), the $z$ axis symmetry boundary conditions for $\varphi_{\text{num},m}$ (and hence also $\Pi_{\text{num},m}$) depend on $m$.

**$m = 0$** In this case $\varphi_{\text{num},m}$ is even across the $z$ axis, i.e., $\partial_\theta \varphi_{\text{num},m} = \partial_\theta \Pi_{\text{num},m} = 0$. The $m^2/\sin^2\theta$ term in (2.40) vanishes identically because $m = 0$, and L'Hopital's rule gives the other singular term as $\lim_{\theta\to 0} \cot\theta \partial_\theta \varphi_{\text{num},m} = \partial_{\theta\theta}\varphi_{\text{num},m}$.

**$m \neq 0$** In this case $\varphi_{\text{num},m}$ is odd across the $z$ axis so that $\varphi_{\text{num},m} = \Pi_{\text{num},m} = 0$ there. To implement this we specify zero initial data on the $z$ axis and replace our evolution equations by $\partial_t \varphi_{\text{num},m} = \partial_t \Pi_{\text{num},m} = 0$ there.

#### *3. Equatorial-reflection symmetry boundary conditions*

If the particle orbit is equatorial (as is the case for all the numerical computations discussed here), then the entire physical system has equatorial-reflection symmetry, i.e., all fields must be even across the equator ($\theta = \pi/2$).

### J. Initial data

The correct initial data for (2.23) are unknown (they would represent the equilibrium field configuration around the particle, which is what we are trying to compute). Instead, we follow the usual practice in time-domain self-force computations (e.g., [73]) and specify arbitrary (zero) initial data $\varphi_{\text{num},m} = \Pi_{\text{num},m} = 0$ on our initial slice. This initial data is not a solution of the sourced evolution equation (2.23), but we find that the "junk" [the deviation of the field configuration from (2.23)] quickly radiates away towards the inner and outer boundaries, so that after sufficient time $\varphi_{\text{num},m}$ relaxes to a solution of (2.23) throughout an (expanding) neighborhood of the worldtube. We see no sign of the persistent (nonradiative) "Jost junk solutions" described by [92,93]. This is to be expected for at least two reasons: (i) the source for our field equations does not contain the derivative of a Dirac delta function, and (ii) we are using a second-order-in-space, rather than first-order-in-space formulation of the field equations.

### K. Computing the self-force from the evolved fields

Because the physical scalar fields $\Phi$, $\Phi_{\text{puncture}}$, and $\Phi_{\text{residual}}$ are real, the Fourier inversion (2.13) implies that $\varphi_{-m} = \text{conj}[\varphi_m]$, and similarly for the other $m$-mode fields. Hence we only need to (numerically) compute the $m$-modes $m \geq 0$.

We thus have

$$\Phi_{\text{residual}}(t,r,\theta,\phi) = \sum_{m=0}^{\infty} \Upsilon^{(\Phi)}_{\text{residual},m}(t,r,\theta,\phi), \qquad (2.54)$$

where the (real) field $\Upsilon^{(\Phi)}_{\text{residual},m}$ is given in a neighborhood of the particle by

$$\Upsilon^{(\Phi)}_{\text{residual},m}(t,r,\theta,\phi) := \begin{cases} 2\text{Re}\left[\dfrac{e^{im\tilde{\phi}}}{r}\varphi_{\text{num},m}(t,r,\theta)\right] & \text{if } m \neq 0 \\ \dfrac{1}{r}\varphi_{\text{num},m}(t,r,\theta) & \text{if } m = 0. \end{cases} \qquad (2.55)$$

[10]For ease of implementation and code organization, we actually implement this by first extrapolating $\varphi$ and $\Pi$ into the radial ghost zones using 5th-order Lagrange polynomial extrapolation, then applying the interior evolution equations using our usual centered finite-difference scheme.

We compute the self-force by substituting (2.54) into (2.7) and differentiating at the particle position. A straightforward calculation gives

$$F_a = q\sum_{m=0}^{\infty}\left(\Upsilon^{(\partial_a\Phi)}_{\text{residual},m}\right)\Big|_{\text{particle}}, \tag{2.56}$$

where the "self-force modes" $\Upsilon^{(\partial_a\Phi)}_{\text{residual},m} = \Upsilon^{(\partial_a\Phi)}_{\text{residual},m}(t,r,\theta)$ are defined in a neighborhood of the particle by

$$\Upsilon^{(\partial_t\Phi)}_{\text{residual},m} = \begin{cases} 2\text{Re}\left[\dfrac{e^{im\tilde{\phi}}}{r}\Pi_{\text{num},m}\right] & \text{if } m\neq 0 \\ \dfrac{1}{r}\Pi_{\text{num},m} & \text{if } m=0, \end{cases} \tag{2.57a}$$

$$\Upsilon^{(\partial_s\Phi)}_{\text{residual},m} = \partial_s\Upsilon^{(\Phi)}_{\text{residual},m}, \tag{2.57b}$$

$$\Upsilon^{(\partial_\phi\Phi)}_{\text{residual},m} = \begin{cases} -2m\text{Im}\left[\dfrac{e^{im\tilde{\phi}}}{r}\varphi_{\text{num},m}\right] & \text{if } m\neq 0 \\ 0 & \text{if } m=0. \end{cases} \tag{2.57c}$$

We compute each self-force mode at the particle by first computing it in a finite-difference-molecule-sized region about the particle, then interpolating it to the particle position using the "C2" interpolating function described in Appendix B 10. [For $\Upsilon^{(\partial_s\Phi)}_{\text{residual},m}$, an alternative would be to apply a "differentiating interpolator"[11] directly to $\Upsilon^{(\Phi)}_{\text{residual},m}$. This would be more elegant and efficient than interpolating a molecule-sized $\Upsilon^{(\partial_s\Phi)}_{\text{residual},m}$ grid function. However, the cost of even the interpolate-a-molecule-size-grid-function scheme is still only a minute fraction of the overall self-force computation, so we did not bother with the additional software complexity of the differentiating interpolator.]

### L. The tail series

In practice we can only numerically compute a finite number of $m$-modes $0 \le m \le m_{\text{num,max}}$. We thus partition each of the infinite sums in (2.54) and (2.56) into a finite "numerical sum" plus an infinite "tail sum,"

$$\sum_{m=0}^{\infty} = \sum_{m=0}^{m_{\text{num,max}}} + \sum_{m=m_{\text{num,max}}+1}^{\infty}, \tag{2.58}$$

and account for the tail sum in much the same way as is done in the mode-sum regularization scheme.

To estimate the tail sum for the self-force computation (2.56),[12] we use the fact that the modes have a known power-law behavior that can be attributed to the nonsmoothness of the residual field. Explicitly, the behavior of the modes of the residual field is given by

$$\left(\Upsilon^{(\Phi)}_{\text{residual},m}\right)\Big|_{\text{particle}} = \sum_{\substack{\alpha\ge n \\ \alpha\,\text{even}}} k^{(\Phi)}_{\alpha} f^{(\Phi)}_{\alpha,m} + \left(\Upsilon^{(\Phi)}_{\text{regular},m}\right)\Big|_{\text{particle}}, \tag{2.59}$$

where $\Upsilon^{(\Phi)}_{\text{regular},m}$ comes from the $C^\infty$ regular field and falls off faster than any power of $m$; it can therefore be ignored for $m_{\text{num,max}}$ sufficiently large. The remaining piece of the tail sum is effectively an even power series in $1/m$, starting at an order, $m^{-n}$, that is determined by the order of the puncture field. In our case ($n=4$), the basis functions $f$ for the $m$ dependence are given by

$$\begin{aligned} f^{(\Phi)}_{4,m} &= \frac{1}{(m-\frac{3}{2})(m-\frac{1}{2})(m+\frac{1}{2})(m+\frac{3}{2})} \\ f^{(\Phi)}_{6,m} &= \frac{1}{(m-\frac{5}{2})(m-\frac{3}{2})(m-\frac{1}{2})(m+\frac{1}{2})(m+\frac{3}{2})(m+\frac{5}{2})} \\ f^{(\Phi)}_{8,m} &= \frac{1}{(m-\frac{7}{2})(m-\frac{5}{2})(m-\frac{3}{2})(m-\frac{1}{2})(m+\frac{1}{2})(m+\frac{3}{2})(m+\frac{5}{2})(m+\frac{7}{2})} \\ &\vdots \\ f^{(\Phi)}_{\alpha,m} &= \frac{1}{(m-\frac{\alpha-1}{2})_\alpha}, \end{aligned} \tag{2.60}$$

and the coefficient functions, $k$, are given by the $m$-mode decomposition of higher-order terms (i.e., those that have *not* been included in the definition of the puncture field) in the series expansion of the Detweiler-Whiting singular field [87].

[11] An interpolator generally works by (conceptually) locally fitting a fitting function [in our case the C2 interpolant (B14)] to the data points in a neighborhood of the interpolation point, then evaluating the fitting function at the interpolation point. A differentiating interpolator instead evaluates a *derivative* of the fitting function at the interpolation point. This has the effect of interpolating the corresponding derivative of the input data to the interpolation point without ever needing to form a grid function of that derivative.

[12] The physical scalar field $\Phi$ at the particle can also be computed by applying similar techniques to the infinite sum (2.54).

Derivatives of the field behave in a similar manner, so that in addition to using this approach for $\Upsilon^{(\Phi)}_{\text{residual},m}$, we may also use it for the fields $\Upsilon^{(X)}_{\text{residual},m}$, where $X$ is one of $\partial_s\Phi$, $\partial_t\Phi$ or $\partial_\phi\Phi$. The only caveat is that the $m$ dependence is slightly modified: the $\phi$ derivative introduces a factor of $m^2$, so $f^{(\partial_\phi\Phi)}_{\alpha,m} = m^2 f^{(\Phi)}_{\alpha,m}$. The $t$ derivative of the Detweiler-Whiting singular field can be written in terms of $r$ and $\phi$ derivatives, so $f^{(\partial_t\Phi)}_{\alpha,m}$ has both kinds of terms present.

For any given $X$, $\alpha$ and $m_{\text{num,max}}$, the infinite sum

$$S^{(X)}_{\alpha,m_{\text{num,max}}+1} := \sum_{m=m_{\text{num,max}}+1}^{\infty} f^{(X)}_{\alpha,m} \tag{2.61}$$

can be computed exactly. Using the facts that

$$\sum_{m=-\infty}^{\infty} \left(m - \frac{\alpha-1}{2}\right)_\alpha = 0 \quad \text{for even } \alpha \geq 2, \tag{2.62}$$

$$\sum_{m=-\infty}^{\infty} m^2 \left(m - \frac{\alpha-1}{2}\right)_\alpha = 0 \quad \text{for even } \alpha \geq 4, \tag{2.63}$$

we obtain

$$S^{(\Phi)}_{\alpha,m_{\text{num,max}}+1} = \frac{1}{(\alpha-1)(m_{\text{num,max}} - \frac{\alpha-3}{2})_{\alpha-1}}, \tag{2.64}$$

$$S^{(\partial_\phi\Phi)}_{\alpha,m_{\text{num,max}}+1} = \frac{m_{\text{num,max}}(m_{\text{num,max}}+1)}{(\alpha-3)(m_{\text{num,max}} - \frac{\alpha-3}{2})_{\alpha-1}}. \tag{2.65}$$

Analytical expressions for the $k^{(X)}_4$ coefficients (in this context known as "$m$-mode regularization parameters") compatible with our choice of puncture field were given in [87]. As they are extremely lengthy we will not repeat them here; a Mathematica notebook for computing them is included in the Supplemental Material [91].

While the higher-order coefficients could be analytically determined in a similar manner, we choose instead an alternative approach. To estimate some finite set $\alpha \in \alpha_{\text{tail-set}}$ of the remaining $k^{(X)}_\alpha$ coefficients, we first truncate the series (2.59) to only the terms $\alpha = 4$ and $\alpha \in \alpha_{\text{tail-set}}$,

$$(\Upsilon^{(X)}_{\text{residual},m})|_{\text{particle}} \approx k^{(X)}_4 f^{(X)}_{4,m} + \sum_{\substack{\alpha \text{ even} \\ \alpha\in\alpha_{\text{tail-set}}}} k^{(X)}_\alpha f^{(X)}_{\alpha,m}. \tag{2.66}$$

For a specified particle position, we then estimate the corresponding set of $k^{(X)}_\alpha$ by least-squares fitting the numerically computed $(\Upsilon^{(X)}_{\text{residual},m})|_{\text{particle}}$ with $m_{\text{fit,min}} \leq m \leq m_{\text{fit,max}}$ to the truncated series (2.66).[13] For all analyses reported in this paper we take $\alpha_{\text{tail-set}}$ to be either empty (no tail fit) or $\{6, 8\}$. Table III gives $m_{\text{fit,min}}$ and $m_{\text{fit,max}}$ for each of our configurations where a tail fit is done.

Finally, we compute (estimate) each self-force component $F_a$ at each of these times by substituting (2.58), (2.59), and (2.61) into (2.56), giving

$$\frac{F_a}{q} = \sum_{m=0}^{m_{\text{num,max}}} (\Upsilon^{(X)}_{\text{residual},m})|_{\text{particle}} + k^{(X)}_4 S^{(X)}_{4,m_{\text{num,max}}+1} + \sum_{\substack{\alpha \text{ even} \\ \alpha\in\alpha_{\text{tail-set}}}} k^{(X)}_\alpha S^{(X)}_{\alpha,m_{\text{num,max}}+1}. \tag{2.67}$$

### M. Selecting the time interval for analysis within an evolution

Our discussion in Secs. II K and II L assumed that a time series of the self-force modes $\Upsilon^{(\partial_a\Phi)}_{\text{residual},m}$ is available at a suitable set of points around the orbit for each $m = 0, 1, 2, \ldots, m_{\text{num,max}}$. However, as described in Sec. II J, the initial part of each such time series is contaminated by "junk" radiation. Here we describe how we determine when this junk radiation has decayed to a negligible level (below our numerical noise level).

The key fact which underlies our algorithm for making this determination is that since the particle orbit is periodic,[14] the self-force modes should also be periodic with the orbital period $T_r$.

Given a time series of some numerically computed self-force mode $\Upsilon^{(\partial_a\Phi)}_{\text{residual},m}$, we define its "orbit-difference" time series by

$$\Delta[\Upsilon^{(\partial_a\Phi)}_{\text{residual},m}](t) := |\Upsilon^{(\partial_a\Phi)}_{\text{residual},m}(t+T_r) - \Upsilon^{(\partial_a\Phi)}_{\text{residual},m}(t)|. \tag{2.68}$$

The orbit-difference time series is one orbit shorter in duration than the original time series.

Because of the initial junk radiation, the orbit difference is initially large. As the junk radiation radiates away from the particle and worldtube, the orbit-difference decays until it eventually becomes roughly constant (at a nonzero value due to finite differencing and other numerical errors) or, in some cases, varying with the orbital period (since the numerical errors are similarly periodic). (This behavior can be seen in Fig. 3.)

It is thus quite easy to determine the time when the junk radiation has decayed to a negligible level by visually inspecting a graph of the orbit difference as a function of time. Although this process could probably be automated by searching backwards in the orbit-difference time series for a sustained rise (in fact, we implemented such an algorithm), we find that the visual inspection is valuable for detecting a variety of other numerical problems that might

[13]For each $\alpha$, we normalize $f^{(X)}_{\alpha,m}$ to have unit magnitude at the mean $m$ in $\alpha_{\text{tail-set}}$. This reduces to a tolerable level what would otherwise be severe numerical ill conditioning in the least-squares fit [94].

[14]More precisely, the particle orbit is periodic modulo an overall rotation in $\phi$, which is ignorable because Kerr spacetime is axisymmetric.

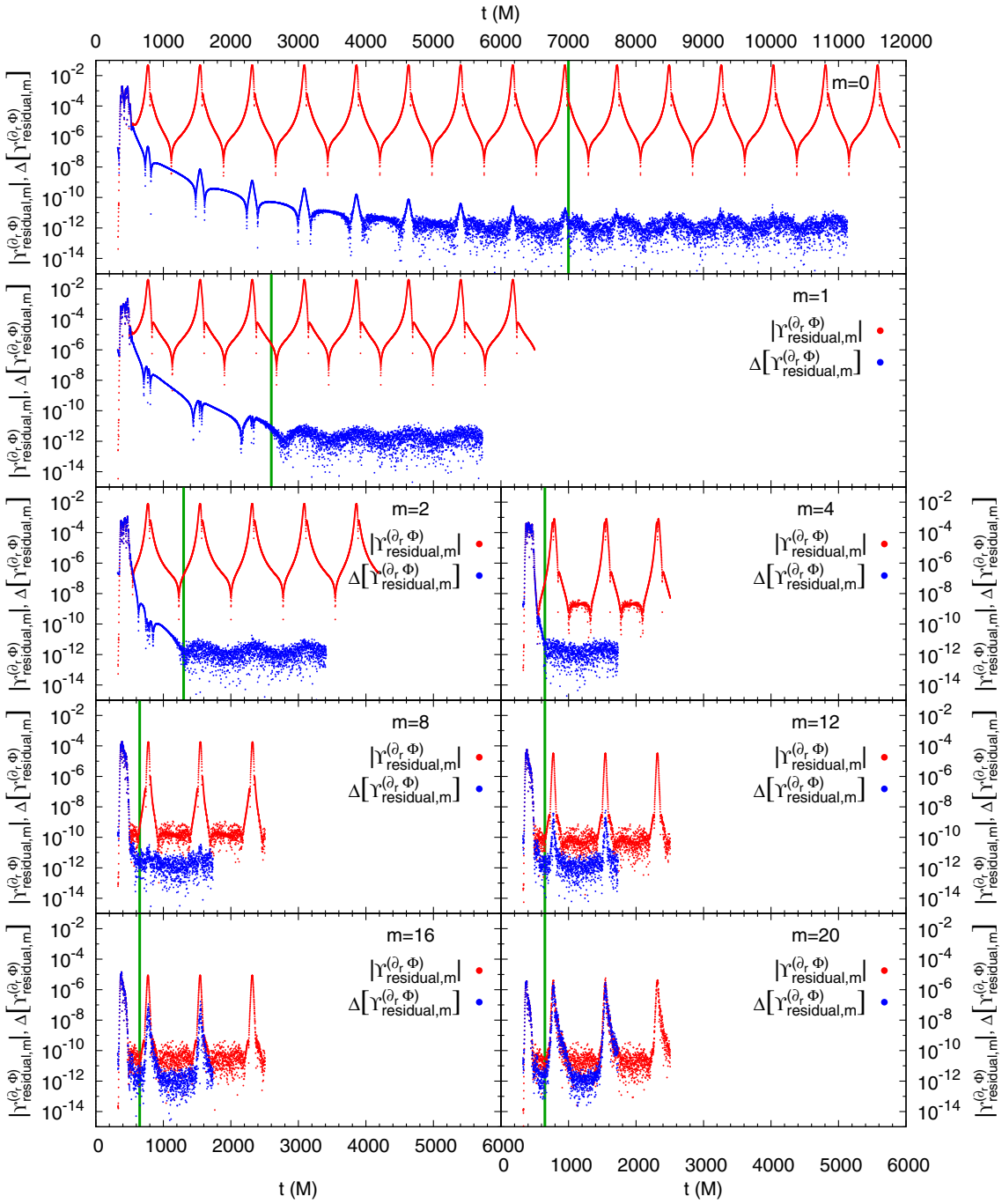


FIG. 3. This figure shows some of the self-force modes $\Upsilon^{(\partial_r\Phi)}_{\text{residual},m}$ and their orbit differences $\Delta[\Upsilon^{(\partial_r\Phi)}_{\text{residual},m}]$ for the e8 configuration, which has $(\tilde{a}, p, e) = (0.6, 8, 0.8)$. In each subplot the green vertical line marks the self-force computation starting time (when the orbit differences have decayed to the numerical noise level). For each $m$ the orbit differences are only defined for a time interval that is one orbital period shorter than the self-force mode. As $m$ increases the initial junk decays faster, so the self-force computation starting time can be earlier in the evolution. Correspondingly, we choose shorter numerical evolutions for larger $m$.

occur, so we have chosen not to routinely use an automated algorithm here.

### N. Selecting a low-noise subset of times within an evolution

Because of the interaction between finite differencing and the limited differentiability of $\varphi_{\text{num}}$ at the particle, as well as other numerical errors, there is numerical noise in the self-force modes $\Upsilon^{(\partial_a\Phi)}_{\text{residual},m}$. For highly eccentric orbits, we find that the higher-$m$ modes may be completely dominated by numerical noise in the outer parts of the orbit. (This can be seen in, for example, Figs. 4 and 5.)

Including these modes in the self-force sum (2.67) would add significant numerical noise to the computed self-force

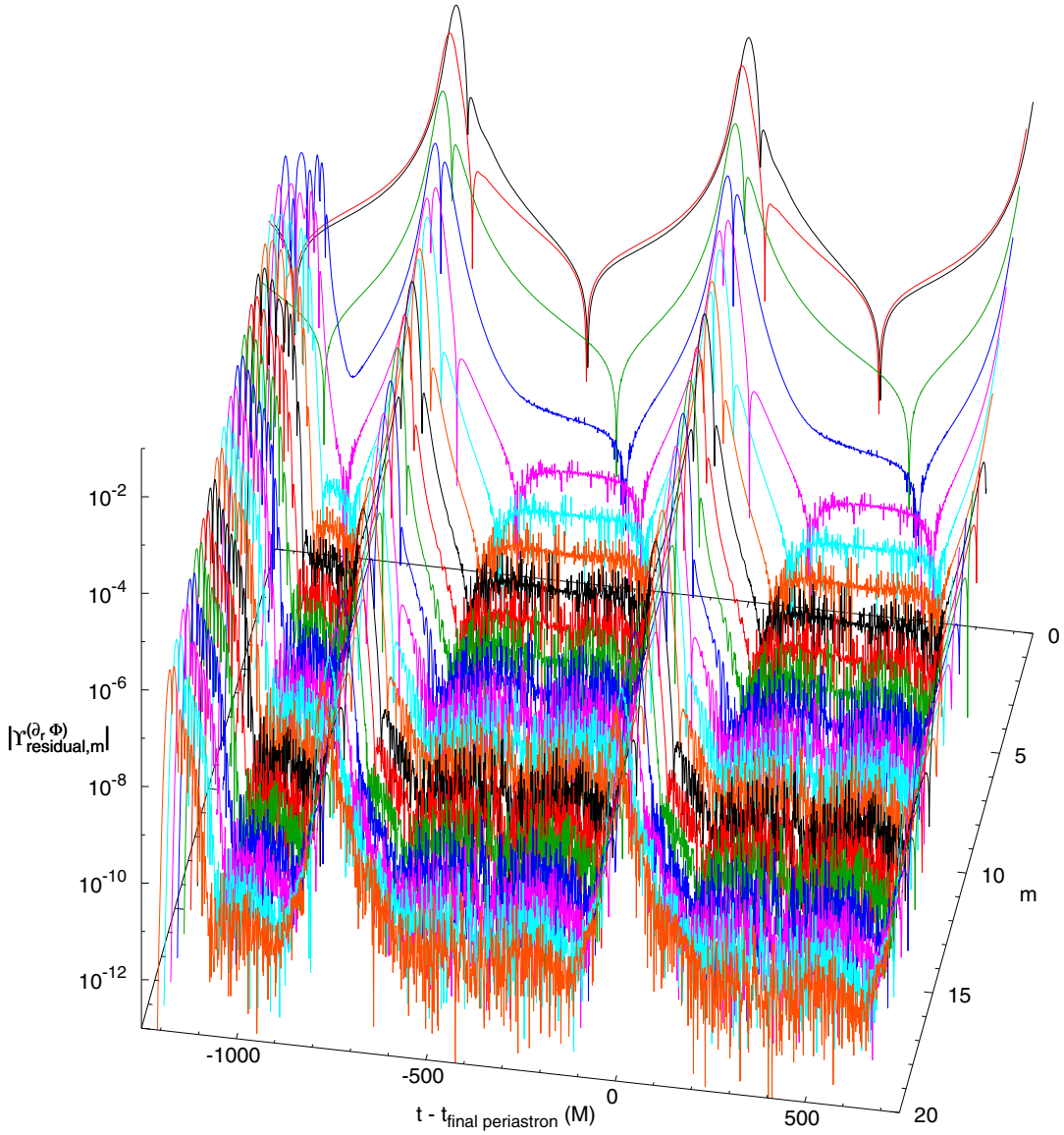


FIG. 4. This figure shows all of our numerically computed self-force modes $\Upsilon^{(\partial_r\Phi)}_{\text{residual},m}$ for the last 2.6 orbital periods for each $m \in [0, 20]$ for the e8 configuration, which has $(\tilde{a}, p, e) = (0.6, 8, 0.8)$. Compare these with the low-noise subset of modes shown in Fig. 6.

while (in many cases) not adding any significant "signal." Therefore, it is useful (again, in many although not all cases) to omit the noisy modes from the self-force sum (2.67), effectively treating these modes/times as missing data.

To estimate the noise level at any point in an $\Upsilon^{(\partial_a\Phi)}_{\text{residual},m}$ time series, we first define a smoothed time series $\mathcal{S}[\Upsilon^{(\partial_a\Phi)}_{\text{residual},m}]$ using Savitzky-Golay moving-window smoothing [95] ([96], Sec. 14.8). For all analyses reported in this paper we use a 6th-degree polynomial over a current position $\pm$ 10-sample moving window in the time series.

We then define the (absolute) noise time series as

$$\text{noise}\,[\Upsilon^{(\partial_a\Phi)}_{\text{residual},m}](t) \coloneqq \underset{\text{SG window}(t)}{\text{RMS}}\{\Upsilon^{(\partial_a\Phi)}_{\text{residual},m} - \mathcal{S}[\Upsilon^{(\partial_a\Phi)}_{\text{residual},m}]\} \tag{2.69}$$

and the "relative noise" time series as

$$\text{relative\_noise}\,[\Upsilon^{(\partial_a\Phi)}_{\text{residual},m}](t) \coloneqq \text{noise}\,[\Upsilon^{(\partial_a\Phi)}_{\text{residual},m}](t) \Big/ \underset{\text{SG window}(t)}{\text{RMS}}\{\Upsilon^{(\partial_a\Phi)}_{\text{residual},m}\}, \tag{2.70}$$

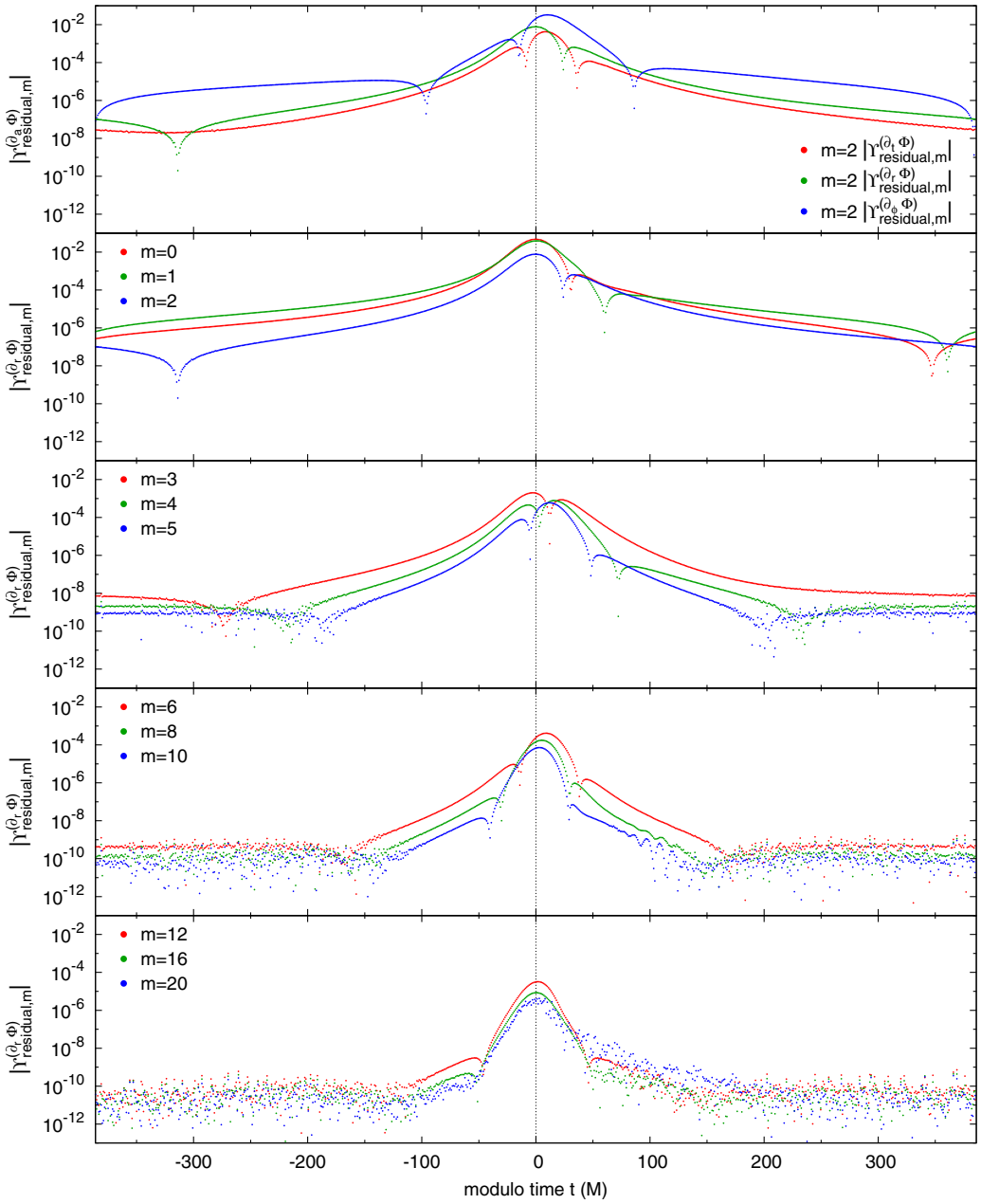


FIG. 5. This figure shows some of the self-force modes $\Upsilon^{(\partial_a\Phi)}_{\text{residual},m}$ for the e8 configuration, which has $(\tilde{a}, p, e) = (0.6, 8, 0.8)$. The figure shows only data for the final orbit simulated for each $m$. Compare these modes with the low-noise subset of modes shown in Fig. 7.

where $\text{RMS}_{\text{SG window}(t)}\{\cdots\}$ is the root-mean-square value over the Savitzky-Golay smoothing window.

Using these definitions we select a low-noise subset of the time samples by omitting those samples from the $\Upsilon^{(\partial_a\Phi)}_{\text{residual},m}$ time series which have $m \geq m_{\text{noise,min}}$ and relative_noise $[\Upsilon^{(\partial_a\Phi)}_{\text{residual},m}] > \varepsilon_{\text{relative,max}}$, where $m_{\text{noise,min}}$ is a parameter chosen so that time intervals immediately around zero crossings in lower-$m$ modes are not falsely excluded, and where $\varepsilon_{\text{relative,max}}$ is a parameter chosen to tune the tolerable level of numerical noise. Table III gives $m_{\text{noise,min}}$ and $\varepsilon_{\text{relative,max}}$ for each of our configurations where smoothing is done.

### O. Dissipative and conservative parts of the self-force

As well as calculating the overall self-force, it is useful to split the self-force into dissipative and conservative contributions [71,97–100]: the dissipative part affects the $\mathcal{O}(\mu)$ orbital evolution while the conservative part only affects the orbital evolution at $\mathcal{O}(\mu^2)$. As discussed by [30] (Sec. 8.1),

TABLE I. This table summarizes the main physical parameters for the configurations presented in this paper. $(\tilde{a}, p, e)$ uniquely characterize the spacetime and the particle orbit. $\mathcal{E}$ and $\mathcal{L}$ are the particle's specific energy and angular momentum, respectively. "min $r$" and "max $r$" are the particle's periastron and apoastron coordinate radii, respectively. The orbital period is given in three forms: the coordinate time $T_r$ and proper time along the particle orbit $\tau_r$ of the radial motion, and the long-term mean coordinate-time period $T_\phi$ of the azimuthal ($\phi$) motion (i.e., the mean coordinate time $t$ during which $\phi$ advances by $2\pi$). "$\delta\phi$ per orbit" denotes the advance in $\phi$ (in units of $2\pi$) during one period of the orbit's radial motion (i.e., during a coordinate time $T_r$); this is given by $T_r/T_\phi$ and is $\gg 1$ orbit for a zoom-whirl orbit.

| | | | | | | | | Orbital Period | | | $\delta\phi$ per |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | min $r$ | max $r$ | radial | | azimuthal | orbit |
| Name | $\tilde{a}$ | $p$ | $e$ | $\mathcal{E}$ $(M)$ | $\mathcal{L}$ $(M^2)$ | $(M)$ | $(M)$ | $T_r$ $(M)$ | $\tau_r$ $(M)$ | $T_\phi$ $(M)$ | (orbits) |
| ns5 | 0.0 | 7.2 | 0.5 | 0.956 876 | 3.622 713 | 4.8 | 14.4 | 405.662 | 317.366 | 134.285 | 3.021 |
| n-55 | −0.5 | 10.0 | 0.5 | 0.967 896 | 4.100 631 | 6.667 | 20.0 | 505.428 | 434.465 | 249.488 | 2.026 |
| n95 | 0.9 | 10.0 | 0.5 | 0.963 778 | 3.489 553 | 6.667 | 20.0 | 378.408 | 333.027 | 293.070 | 1.291 |
| e8 | 0.6 | 8.0 | 0.8 | 0.978 270 | 3.405 897 | 4.444 | 40.000 | 771.968 | 709.796 | 502.435 | 1.536 |
| e8b | 0.8 | 8.0 | 0.8 | 0.978 056 | 3.292 113 | 4.444 | 40.000 | 756.641 | 697.570 | 527.812 | 1.434 |
| e9 | 0.99 | 7.0 | 0.9 | 0.986 565 | 3.052 860 | 3.684 | 70.000 | 1513.855 | 1442.724 | 1060.526 | 1.427 |
| e95 | 0.99 | 5.0 | 0.95 | 0.990 315 | 2.699 644 | 2.564 | 100.000 | 2436.050 | 2349.870 | 1445.400 | 1.685 |
| ze4 | 0.2 | 6.15 | 0.4 | 0.945 536 | 3.366 468 | 4.393 | 10.25 | 354.628 | 255.966 | 95.799 | 3.702 |
| ze9 | 0.0 | 7.800 1 | 0.9 | 0.988 333 | 3.904 885 | 4.105 | 78.001 | 2112.079 | 1913.402 | 339.855 | 6.215 |
| zze9 | 0.0 | 7.800 001 | 0.9 | 0.988 332 | 3.904 884 | 4.105 | 78.000 | 2224.815 | 1971.883 | 265.734 | 8.372 |
| ze98 | 0.99 | 2.4 | 0.98 | 0.991 798 | 2.180 959 | 1.212 | 120.000 | 3304.620 | 3021.480 | 215.851 | 15.310 |
| circ-ze4 | 0.2 | 4.392 857 | 0.0 | 0.943 384 | 3.346 263 | 4.392 857 | 4.392 857 | | | 59.106 | |
| circ-ze9 | 0.0 | 4.105 316 | 0.0 | 0.988 327 | 3.904 841 | 4.105 316 | 4.105 316 | | | 52.264 | |
| circ-zze9 | 0.0 | 4.105 264 | 0.0 | 0.988 332 | 3.904 884 | 4.105 264 | 4.105 264 | | | 52.263 | |
| circ-ze98 | 0.99 | 1.212 121 | 0.0 | 0.984 732 | 2.164 538 | 1.212 121 | 1.212 121 | | | 14.605 | |

for equatorial orbits these can be computed from the even-in-time and odd-in-time parts of the self-force,[15]

$$F_{\mathrm{diss},t} = F_{\mathrm{even},t} \qquad F_{\mathrm{cons},t} = F_{\mathrm{odd},t} \tag{2.71a}$$

$$F_{\mathrm{diss},r} = F_{\mathrm{odd},r} \qquad F_{\mathrm{cons},r} = F_{\mathrm{even},r} \tag{2.71b}$$

$$F_{\mathrm{diss},\phi} = F_{\mathrm{even},\phi} \qquad F_{\mathrm{cons},\phi} = F_{\mathrm{odd},\phi} \tag{2.71c}$$

where

$$F_{\mathrm{even},a}(t) = \frac{1}{2}[F_a(t) + F_a(T_r - t)] \tag{2.72a}$$

$$F_{\mathrm{odd},a}(t) = \frac{1}{2}[F_a(t) - F_a(T_r - t)] \tag{2.72b}$$

with $t$ being the modulo time.

To allow this computation without requiring time interpolation, we always choose our self-force computation times to be uniformly spaced in coordinate time $t$, with a spacing $\Delta t_{\mathrm{sample}}$ which integrally divides the orbital period $T_r$.

[15]It would be possible to similarly compute the dissipative and conservative parts of each individual self-force mode in the sums (2.56). This would have the advantage that the dissipative part of the self-force could be computed very accurately (its tail sums should converge exponentially fast), with only the conservative part requiring the full tail-sum computation described in Sec. II L. However, for historical reasons we have not taken this approach.

### P. Summary of computation and data analysis

To summarize, our overall computational and data-analysis scheme involves a sequence of operations:

(i) For each $m$, we perform a numerical evolution of the 1st-order-in-time evolution system described in Sec. II F. Our evolution code writes out time series of each self-force mode $\Upsilon^{(\partial_a\Phi)}_{\mathrm{residual},m}$, sampled at uniform coordinate-time intervals. We always choose the sampling time $\Delta t_{\mathrm{sample}}$ to be the same for each $m$ and (as noted in Sec. II O) to integrally divide the period $T_r$ of the particle's radial motion.

(ii) For each $m$, we use the orbit-differences algorithm described in Sec. II M to select a point in each of the self-force modes' time series when the initial junk radiation has decayed to a level below our numerical noise level. For all our further data analysis we use only the modes from times $\geq$ this "self-force computation start time" for each $m$.

(iii) For most configurations, for each $m$ we use the noise-estimation and low-noise-selection algorithms described in Sec. II N to select a subset of the self-force mode $\Upsilon^{(\partial_a\Phi)}_{\mathrm{residual},m}$ time series which has relatively low numerical noise.

TABLE II. For each configuration, this table gives the number of self-force output samples per orbit (more precisely, per radial orbital period for the eccentric-orbit configurations, and per azimuthal orbital period for the circular-orbit configurations), the sampling interval, the maximum $m$ of the numerically computed modes in the self-force sum (2.67), the time at which the self-force computation begins (after the initial transients have decayed), and the time at which the numerical evolution ends (or at which the self-force computation ends, if this is earlier). All times are coordinate times in units of $M$ and (except for $\Delta t_{\text{sample}}$) are rounded to the nearest integer. For some configurations (footnoted), some large-$m$ evolutions use earlier starting and/or ending times (chosen to select low-numerical-noise sections of data and/or limited by machine failures or queue-time limits). For the eccentric-orbit configurations there is always at least one orbital period between the starting and ending times; for the circular-orbit configurations the self-force is time independent so there is no need for an extended self-force computation interval. The circ-ze98 configuration is omitted because we were unable to obtain stable evolutions for it for $m \geq 6$.

| | | | | | Self-Force Computation Start Time | | | | | Evolution End Time | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | $N_{\text{sample}}$ | $\Delta t_{\text{sample}}$ $(M)$ | $m_{\text{num,max}}$ | $t_{\text{initial}}$ $(M)$ | $m=0$ $(M)$ | $m=1$ $(M)$ | $m=2$ $(M)$ | $m=3$ $(M)$ | $m\geq 4$ $(M)$ | $m=0$ $(M)$ | $m=1$ $(M)$ | $m=2$ $(M)$ | $m=3$ $(M)$ | $m\geq 4$ $(M)$ |
| ns5 | 406 | 0.999 | 20 | 143 | 8 000 | 2 000 | 700 | 365 | 310 | 11 990 | 6 994 | 3 997 | 1 349 | 1 349[a] |
| n-55 | 506 | 0.999 | 20 | 183 | 7 200 | 2 700 | 900 | 480 | 420 | 10 291 | 5 237 | 3 721 | 1 348 | 1 348 |
| n95 | 378 | 1.001 | 20 | 116 | 5 115 | 2 850 | 890 | 400 | 350 | 12 013 | 8 009 | 4 004 | 1 251 | 1 251 |
| e8 | 770 | 1.003 | 20 | 324 | 7 000 | 2 600 | 1 300 | 800 | 650 | 11 903 | 6 500 | 4 184 | 2 506 | 2 506 |
| e8b | 760 | 0.996 | 20 | 289 | 8 700 | 4 200 | 1 900 | 1 025 | 820 | 11 728 | 7 945 | 5 675 | 3 405 | 3 405 |
| e9 | 1 514 | 1.000 | 20 | 680 | 10 000 | 5 300 | 2 450 | 1 300 | 1 180[b] | 15 818 | 8 249 | 5 221 | 3 300 | 3 300 |
| e95 | 2 436 | 1.000 | 20 | 1 090 | 7 000 | 4 700 | 2 550 | 2 200 | 1 950 | 13 270 | 8 398 | 5 962 | 5 400 | 5 400 |
| ze4 | 360 | 0.985 | 20 | 111 | 8 000 | 2 000 | 550 | 400 | 360 | 11 821 | 6 896 | 3 940 | 1 182 | 1 182 |
| ze9 | 2 112 | 1.000 | 20 | 920 | 12 888 | 7 888 | 2 500 | 1 600 | 1 600 | 15 001 | 10 000 | 8 000 | 4 800 | 4 800[a] |
| zze9 | 2 224 | 1.000 | 20 | 1 005 | 10 000 | 5 550 | 3 350 | 1 700 | 1 575 | 22 361 | 14 461 | 12 236 | 7 787 | 7 787[a] |
| ze98 | 13 216 | 0.250 | 12 | 1 448 | 11 600 | 8 250 | 4 950 | 4 950 | 4 950[b] | 18 093 | 14 871 | 14 871 | 13 218 | 10 623[a] |
| circ-ze4 | 60 | 0.982 | 20 | 0 | 6 000 | 2 000 | 450 | 315 | 300[b] | 6 154 | 4 000 | 940 | 700 | 600[a] |
| circ-ze9 | 52 | 1.005 | 20 | 0 | 10 000 | 1 630 | 410 | 325 | 335[b] | 10 453 | 1 980 | 950 | 685 | 425[a] |
| circ-zze9 | 52 | 1.005 | 20 | 0 | 10 000 | 1 630 | 410 | 325 | 335[b] | 10 453 | 1 980 | 950 | 685 | 425[a] |

[a]Some large-$m$ evolutions end earlier.
[b]Some large-$m$ evolutions start the self-force computation earlier.

TABLE III. This table shows the low-noise-selection and tail-fit parameters used for computing the self-force for each configuration presented in this paper. The circ-ze98 configuration is omitted because we were unable to obtain stable evolutions for it for $m \geq 6$.

| | Low-Noise-Selection Parameters | | Tail-Fit Parameters | |
|---|---|---|---|---|
| Name | $m_{\text{noise,min}}$ | $\epsilon_{\text{relative,max}}$ | $m_{\text{fit,min}}$ | $m_{\text{fit,max}}$ |
| ns5 | 10 | 0.05 | 9 | 18 |
| n-55 | 10 | 0.05 | 9 | 18 |
| n95 | 10 | 0.05 | 9 | 18 |
| e8 | 4 | 0.3 | 12 | 20 |
| e8b | 4 | 0.3 | 12 | 20 |
| e9 | 3 | 0.3 | 12 | 20 |
| e95 | 2 | 0.3 | 12 | 20 |
| ze4 | 10 | 0.05 | 8 | 18 |
| ze9 | 2 | 0.3 | 12 | 20 |
| zze9 | 2 | 0.3 | 12 | 20 |
| ze98 | no low-noise selection | | —no tail fit— | |
| circ-ze4 | no low-noise selection | | 12 | 20 |
| circ-ze9 | no low-noise selection | | 12 | 20 |
| circ-zze9 | no low-noise selection | | 12 | 20 |

TABLE IV. This table shows which numerical grids were used in simulating the configurations presented in this paper. See Tables V and VI for details of these grids.

| | Numerical Grid | | | | |
|---|---|---|---|---|---|
| Name | dro4-32 normal | dro6-48 normal | dro6-48 variant | dro8-64 normal | dro10-80 normal |
| ns5 | ✓ | ✓ | | ✓ | |
| n-55 | ✓ | ✓ | | | |
| n95 | ✓ | ✓ | | ✓ | |
| e8 | ✓ | ✓ | | ✓ | |
| e8b | ✓ | ✓ | | | |
| e9 | ✓ | ✓ | ✓ | ✓ | ✓[a] |
| e95 | ✓ | ✓ | | ✓[b] | |
| ze4 | ✓ | ✓ | | | |
| ze9 | ✓ | ✓ | | | |
| zze9 | ✓ | ✓ | | ✓ | |
| ze98 | ✓ | ✓ | | ✓ | ✓ |
| circ-ze4 | ✓ | | | ✓ | |
| circ-ze9 | ✓ | | | ✓ | |
| circ-zze9 | ✓ | | | ✓ | |
| circ-ze98 | ✓ | | | ✓ | |

[a]$m \leq 15$ only.
[b]$m = 0$, 1, and 2 only.

TABLE V. This table shows the range of grid resolutions used for each of our standard grid structures. Each grid structure has a base grid and 3 refined grids, with a 2:1 refinement ratio between adjacent refinement levels. See Table VI for the sizes and shapes of each refinement level.

| | | Base Grid | | Finest Grid | |
|---|---|---|---|---|---|
| | | $R_*$ ($M$) | $\theta$ (radians) | $R_*$ ($M$) | $\theta$ (radians) |
| dro4-32 | normal | 1/4 | $\pi/72$ | 1/32 | $\pi/576$ |
| dro6-48 | normal | 1/6 | $\pi/108$ | 1/48 | $\pi/864$ |
| dro6-48 | variant | 1/6 | $\pi/96$ | 1/48 | $\pi/768$ |
| dro8-64 | normal | 1/8 | $\pi/144$ | 1/64 | $\pi/1152$ |
| dro10-80 | normal | 1/10 | $\pi/180$ | 1/80 | $\pi/1440$ |

(iv) For each modulo time for which we have self-force modes (at times $\geq$ the self-force computation start time, and with sufficiently low estimated noise), we compute the $t$, $r$, and $\phi$ components of the self-force using the mode summation and tail-fitting algorithms described in Secs. II K and II L.

## III. NUMERICAL RESULTS

### A. Configurations and parameters

Tables I–VIII summarize the main physical and computational parameters for the configurations

TABLE VI. This table shows the size and shape of each refinement level in our numerical grids. `WT_center` is the $R_*$ coordinate of the worldtube center. See Table V for the grid resolutions.

| | | | $R_*$ | | $\theta$ | |
|---|---|---|---|---|---|---|
| Grid Type | Refinement Level | Moves with Worldtube? | Min | Max | Min (radians) | Max (radians) |
| Normal | 0 | no | $R_*^h$ | $R_*^{\mathcal{J}^+}$ | 0 | $\pi/2$ |
| | 1 | yes | `WT_center` $- 30M$ | `WT_center` $+ 30M$ | 0 | $\pi/2$ |
| | 2 | yes | `WT_center` $- 15M$ | `WT_center` $+ 15M$ | $\pi/4$ | $\pi/2$ |
| | 3 | yes | `WT_center` $- 8M$ | `WT_center` $+ 8M$ | $\pi/3$ | $\pi/2$ |
| Variant | 0 | no | $R_*^h$ | $R_*^{\mathcal{J}^+}$ | 0 | $\pi/2$ |
| | 1 | yes | `WT_center` $- 35M$ | `WT_center` $+ 40M$ | 0 | $\pi/2$ |
| | 2 | yes | `WT_center` $- 18M$ | `WT_center` $+ 18M$ | $5\pi/24$ | $\pi/2$ |
| | 3 | yes | `WT_center` $- 6.5M$ | `WT_center` $+ 7.5M$ | $5\pi/16$ | $\pi/2$ |

TABLE VII. This table summarizes miscellaneous computational parameters for the e8 runs.

| | |
|---|---|
| Initial startup | |
| initial time ($t_{\text{initial}}$) | 323.825 $M$ |
| particle $R_*$ at initial time | 45.016 $M$ |
| particle apoastron time | 385.984 $M$ |
| particle $R_*$ at apoastron | 45.889 $M$ |
| time of first worldtube move ($m = 2$) | 448.706 $M$ |
| particle $R_*$ at time of first worldtube move | 45.000 $M$ |
| time interval from initial time to first worldtube move ($m = 2$) | 124.881 $M$ |
| Worldtube | |
| $R_*$ (radial) radius (`WT_radius`) | 5.0 $M$ |
| $\theta$ (angular) radius | $\pi/8$ radians |
| initial value of worldtube center $R_*$ (`WT_center`) | 45.5 $M$ |
| worldtube center $\theta$ | $\pi/2$ radians |
| move worldtube if $\lvert$particle $R_* -$ `WT_center`$\rvert >$ $f_{\text{move}} \times$ `WT_radius`, where $f_{\text{move}} = \ldots$ | |
| initial startup | 0.10 |
| main evolution | 0.05 |
| when moving worldtube, place new worldtube center ahead of particle $R_*$ | |
| (where "ahead" is defined based on sign of particle $R_*$ 3-velocity) | |
| by $f_{\text{ahead}} \times f_{\text{move}} \times$ `WT_radius`, where $f_{\text{ahead}} = \ldots$ | 0.9 |
| maximum $R_*$ distance to move worldtube at any one time $= f_{\text{max-move}} \times$ `WT_radius`, where $f_{\text{max-move}} = \ldots$ | 0.1 |
| minimum time interval between worldtube moves | 1.0 $M$ |
| Overall evolution | |
| number of worldtube moves per orbit | 164 |

TABLE VIII. This table summarizes the compactification parameters for the configurations presented in this paper.

| | Particle Motion | | Compactification | | | |
|---|---|---|---|---|---|---|
| Name | min $R_*$ ($M$) | max $R_*$ ($M$) | $R_*^h$ ($M$) | $R_*^-$ ($M$) | $R_*^+$ ($M$) | $R_*^{\mathcal{J}^+}$ ($M$) |
| ns5 | 4.8 | 14.4 | −70 | −45 | +70 | +95 |
| n-55 | 8.370 | 24.395 | −70 | −45 | +75 | +100 |
| n95 | 8.390 | 24.397 | −70 | −45 | +75 | +100 |
| e8 | 4.884 | 45.889 | −75 | −50 | +125 | +150 |
| e8b | 4.884 | 45.889 | −75 | −50 | +125 | +150 |
| e9 | 3.524 | 77.053 | −75 | −50 | +160 | +185 |
| e95 | 0.782 | 107.784 | −75 | −50 | +190 | +215 |
| ze4 | 4.756 | 13.085 | −70 | −45 | +65 | +90 |
| ze9 | 4.208 | 85.276 | −75 | −50 | +135 | +160 |
| zze9 | 4.208 | 85.275 | −75 | −50 | +135 | +160 |
| ze98 | −15.227 | 120.000 | −90 | −65 | +180 | +205 |
| circ-ze4 | 4.756 | 4.756 | −70 | −45 | +55 | +80 |
| circ-ze9 | 4.208 | 4.208 | −70 | −45 | +55 | +80 |
| circ-zze9 | 4.208 | 4.208 | −70 | −45 | +55 | +80 |
| circ-ze98 | −15.227 | −15.227 | −90 | −65 | +50 | +75 |

presented here.[16,17] These configurations fall into four (overlapping) families:

(i) The ns5, n-55, n95, ze4, and e8b configurations are ones which have also been calculated by other researchers, allowing us to validate our code against their results (both published and unpublished).
(ii) The e8, e8b, e9, and e95 configurations are (non-zoom-whirl) highly eccentric orbits.
(iii) The ze4, ze9, zze9, and ze98 configurations are zoom-whirl orbits; of these the ze4 configuration is of moderate eccentricity while the ze9, zze9, and ze98 configurations are highly eccentric.
(iv) The circ-ze4, circ-ze9, circ-zze9, and circ-ze98 configurations are circular-orbit configurations with orbital radii matching the periastrons of the corresponding zoom-whirl configurations.

### B. Example of data analysis

Here we give an example of the data-analysis "pipeline" described in Sec. II P, for the e8 configuration, which has $(\tilde{a}, p, e) = (0.6, 8, 0.8)$.

Figure 3 shows a selection of the modes $\Upsilon^{(\partial_r\Phi)}_{\text{residual},m}$ and their orbit differences $\Delta[\Upsilon^{(\partial_r\Phi)}_{\text{residual},m}]$ for the entire time span of each $m$'s evolution. Figure 4 shows all of the $\Upsilon^{(\partial_r\Phi)}_{\text{residual},m}$ for the last 2.85 orbital periods for each $m \in [0, 20]$ for the e8 configuration. Figure 5 shows a selection of the modes $\Upsilon^{(\partial_a\Phi)}_{\text{residual},m}$ in more detail as a function of modulo time.

After applying the low-noise-selection criteria described in Sec. II N, Fig. 6 shows the resulting low-noise subset of the $\Upsilon^{(\partial_r\Phi)}_{\text{residual},m}$ for the last 2.6 orbital periods for each $m \in [0, 20]$ for the e8 configuration, and Fig. 7 shows a selection of the low-noise modes $\Upsilon^{(\partial_a\Phi)}_{\text{residual},m}$ in more detail as a function of modulo time. We use these modes to compute the self-force using the mode summation and tail-fitting algorithms described in Secs. II K and II L.

Figure 8 shows some example tail fits of the low-noise modes $\Upsilon^{(\partial_r\Phi)}_{\text{residual},m}$ to the tail series (2.67) for the n95 and e8 configurations.

### C. Convergence of results with numerical resolution

When numerically solving partial differential equations, the results should (must) converge to a continuum limit. More precisely (for finite-difference computations), as the grid is refined, at each event the results should in general be convergent with the correct convergence order for the finite-differencing scheme [101]. However, our numerical scheme is an exception: as the particle moves through the grid, the limited differentiability of our numerical fields at the particle position introduces finite-differencing errors which fluctuate in a "bump function" manner ([102], Appendix F) from one particle position to another. Moreover, these fluctuations are typically not coherent between different-resolution evolutions. Correspondingly, we expect the convergence of our numerical results to fluctuate from one modulo-time (orbital position) sample to the next.

Figure 9 illustrates this fluctuating convergence for the n95 and e9 configurations. As expected, the self-force difference norms $\|F_a^{(\text{low})} - F_a^{(\text{high})}\|_+$ and the convergence ratio $\|F_a^{(\text{low})} - F_a^{(\text{medium})}\|_+ / \|F_a^{(\text{medium})} - F_a^{(\text{high})}\|_+$ fluctuate strongly (typically by an order of magnitude or more) from one sample to the next. This makes it difficult to accurately estimate an overall order of convergence. However, several conclusions can be drawn:

(i) For both configurations there is no systematic difference in the convergence ratio between the ingoing and outgoing legs of the orbit at any given radius $r$.
(ii) For the n95 configuration the convergence order is roughly similar everywhere in the orbit, averaging somewhat better than 2nd order.
(iii) For the e9 configuration the convergence averages much better than 4th order for $r \lesssim 10M$, somewhat worse than 2nd order for $10M \lesssim r \lesssim 20M$, and roughly 4th order for $r \gtrsim 25M$.

[16]The input parameter files and data-analysis scripts for the highest-resolution evolutions for each configuration, as well as for the variant-grid dro6-48 evolutions for the e9 configuration, are included in the Supplemental Material [91].

[17]These simulations all used the Karst cluster at Indiana University.

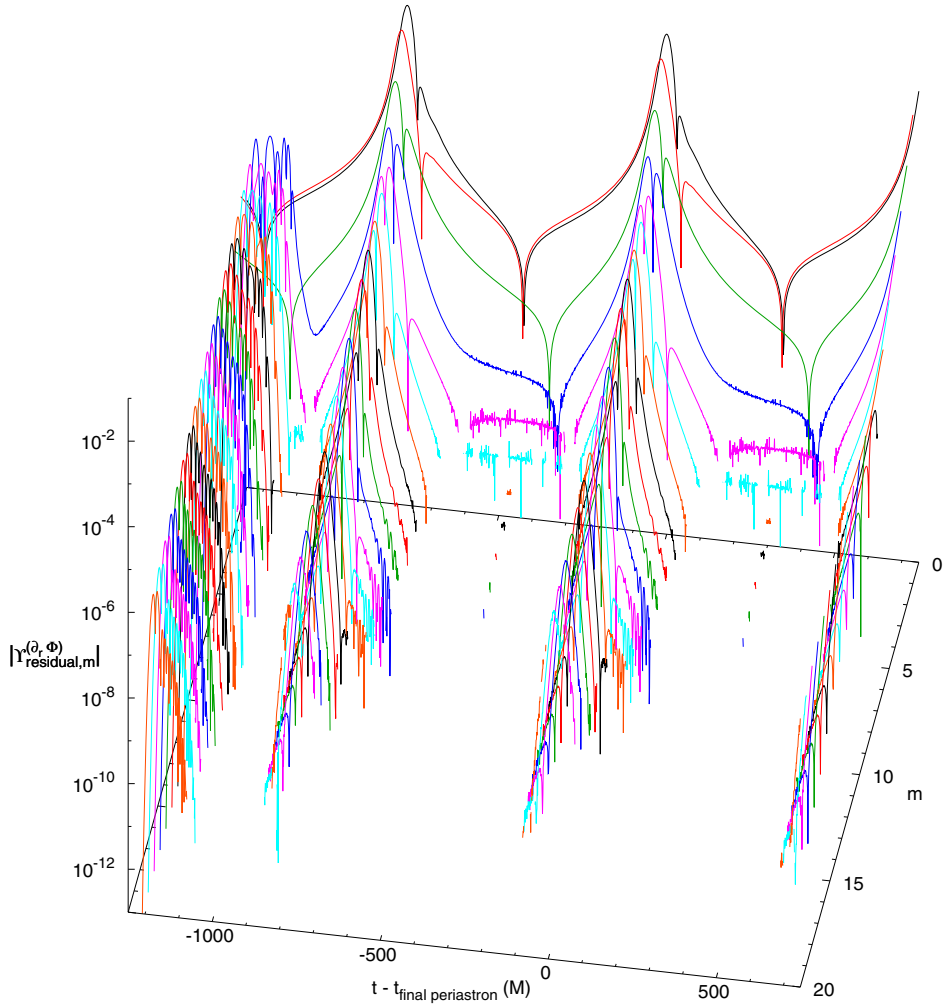


FIG. 6. This figure shows the low-noise numerically computed self-force modes $\Upsilon^{(\partial_r\Phi)}_{\text{residual},m}$ for the last 2.6 orbital periods for each $m \in [0, 20]$ for the e8 configuration, which has $(\tilde{a}, p, e) = (0.6, 8, 0.8)$. Compare these with the full set of modes shown in Fig. 4.

We have not yet been able to determine the reason for this somewhat peculiar convergence behavior. However, since our overall finite-differencing scheme is 4th order accurate (in both space and time) in the bulk, achieving an average convergence *higher* than this implies that one or more of the (e9) evolutions must have insufficient resolution to be in the asymptotic-convergence regime.

Our grid structure for these evolutions (Table VI) moves the finest 3 refinement levels with the worldtube, which in turn moves so that its center is always very close to the particle. Thus, if the particle is at a sufficiently large radius the strong-field region close to the black hole will *not* be covered by the finest grid. For example, if the particle is at $r = R = 10M$ $(R_* = 12.8M)$ then the finest grid extends inward only as far as $r = R = 4.4M$ $(R_* = 4.8M)$. For phenomena nearer to the black hole than this, the local grid resolution is lower. As we discuss in Sec. IV B 3, an adaptive-mesh-refinement scheme might well provide improved accuracy—and convergence—in this situation.

On a more qualitative level, Fig. 23 shows visually that the difference between our highest and 2nd-highest-resolution results is very small for the near-periastron parts of the ze4, ze9, and zze9 orbits.

### D. Verification that results are independent of the choice of worldtube and other numerical parameters

As discussed in Sec. II C, our numerically computed self-force should be independent of the choice of the worldtube. To test this independence numerically, we compare results for the e9 configuration computed using the normal and variant dro6-48 numerical grids (these are described in detail in Tables V and VI). As well as varying the sizes and positions of each refined grid, these computations also use different $\Delta R_*/\Delta\theta$ grid aspect ratios

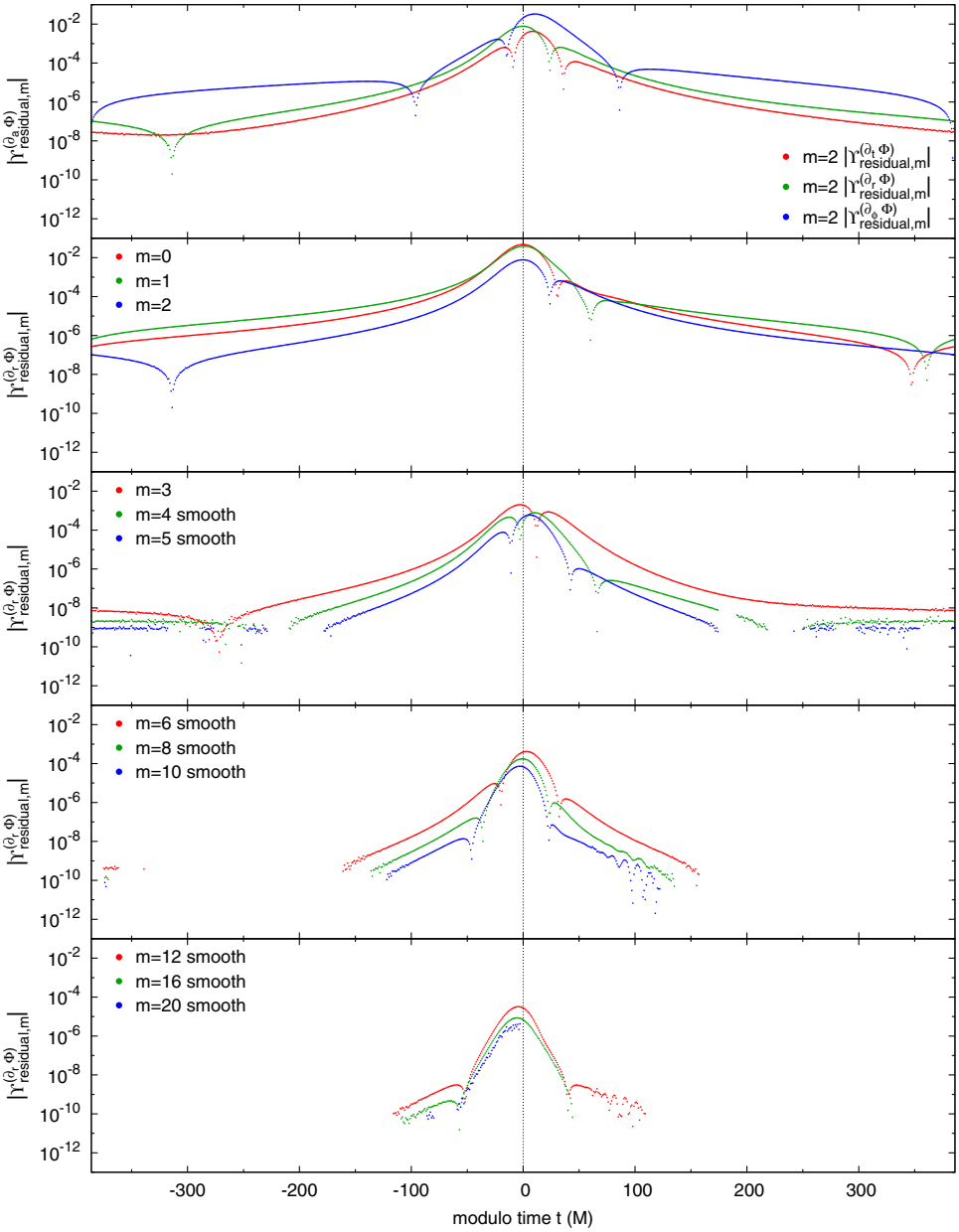


FIG. 7. This figure shows some of the low-noise self-force modes $\Upsilon^{(\partial_a \Phi)}_{\text{residual},m}$ for the final orbit for each $m$ for the e8 configuration, which has $(\tilde{a}, p, e) = (0.6, 8, 0.8)$. Compare these modes with the full set of modes shown in Fig. 5.

(Table V), different worldtube sizes, and different worldtube-moving parameters $f_{\text{move}}$ and `max_move_distance` (these parameters are defined in Fig. 26). Figure 10 shows a numerical comparison of the self-force between these computations. It is apparent that changing these parameters changes the computed self-force by only a very small amount (similar in size to the change induced by a factor-of-1.5 change in numerical resolution).

### E. Comparison with other researchers' results

As an external check on the accuracy of our results, we compare these against results computed using Warburton and Barack's frequency-domain code [52]. Figure 11 shows this comparison for the ns5, n-55, n95, ze4, and e8b configurations. These span a considerable range of black-hole spins and particle orbits, including both prograde and retrograde orbits, eccentricities ranging up to $e = 0.8$ (the e8b configuration), a zoom-whirl orbit (the ze4 configuration), and an occurrence of "wiggles" (the e8b configuration).

For all but the e8b configuration, the two codes agree everywhere around the orbit to within approximately one part in $10^5$ (dissipative part) or one part in $10^4$ (conservative part). The e8b configuration has a

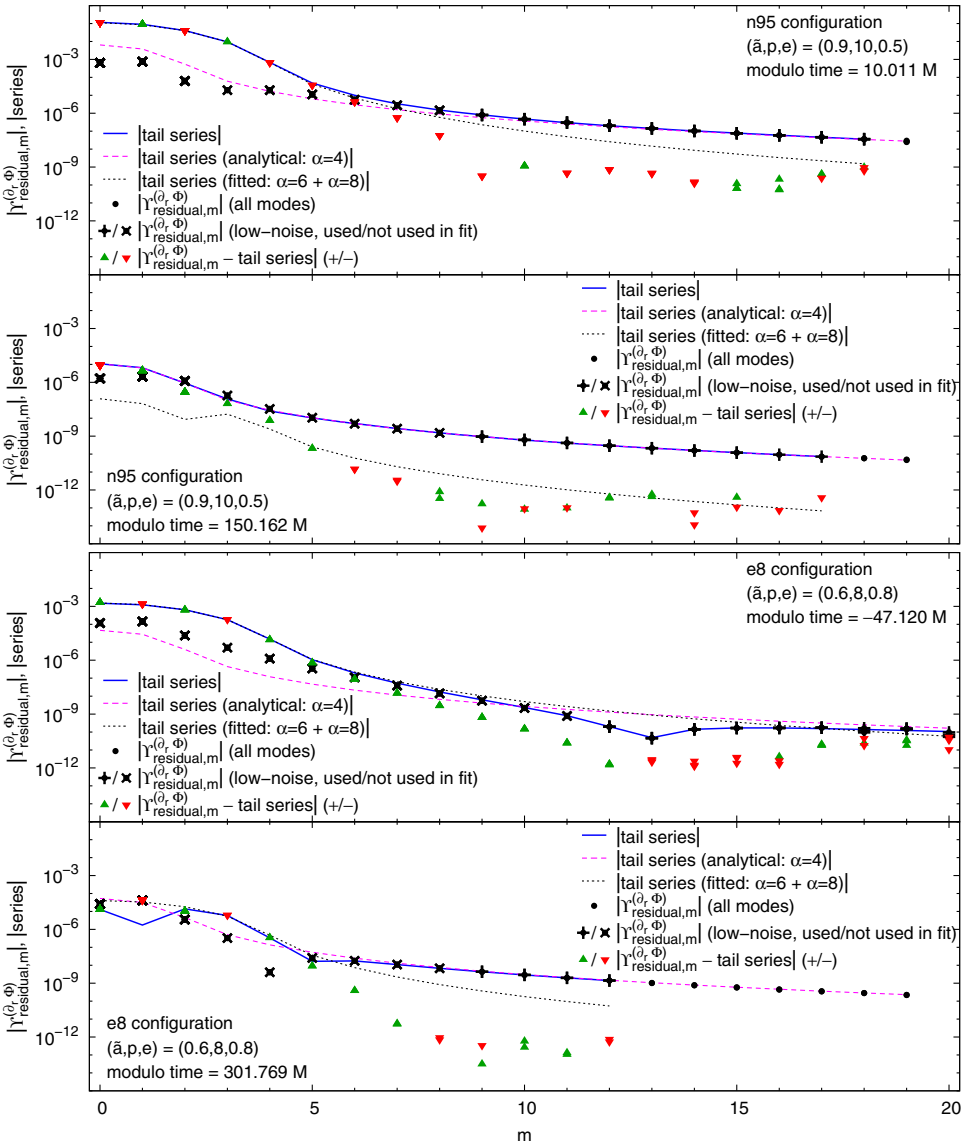


FIG. 8. This figure shows sample fits of the numerically computed $\Upsilon^{(\partial_r\Phi)}_{\text{residual},m}$ to the tail series (2.67) for selected times in the n95 and e8 configurations. The tail series, fitted tail series, and tail-series difference $|\Upsilon^{(\partial_a\Phi)}_{\text{residual},m} - \text{tail series}|$ are only shown for the low-noise modes. For this difference, ▲ (green) or ▼ (red) means $\Upsilon^{(\partial_a\Phi)}_{\text{residual},m} - \text{tail series} > 0$ or $< 0$, respectively.

highly eccentric orbit ($e = 0.8$) that is difficult for the frequency-domain code to compute accurately, so the somewhat lower accuracy is expected. The strong peaks in the e8b difference norms in the region $8M \lesssim r \lesssim 15M$, and also the similar but less prominent peaks in the ns5 and ze4 configurations near $r = 9M$ and $7M \lesssim r \lesssim 8M$ respectively, are probably due to the frequency-domain code switching between "inner" and "outer" approximants [103].

Overall, the agreement between the two codes is excellent, particularly given that they use different regularizations (effective source versus mode sum), different evolution formulations (time domain versus frequency domain), and were/are independently programmed by disjoint sets of researchers. This agreement gives quite high confidence that both codes are in fact computing correct solutions to the $\mathcal{O}(\mu)$-perturbed scalar-field equations.

### F. Overview of self-forces

Figures 12–22 give an overview of the computed self-forces for all our configurations. To facilitate comparison between the different configurations, these figures all use a common format (with one exception noted below):

(i) The top row of each figure shows auxiliary information; the lower three rows show (respectively) $F_t$, $F_r$, and $F_\phi$.

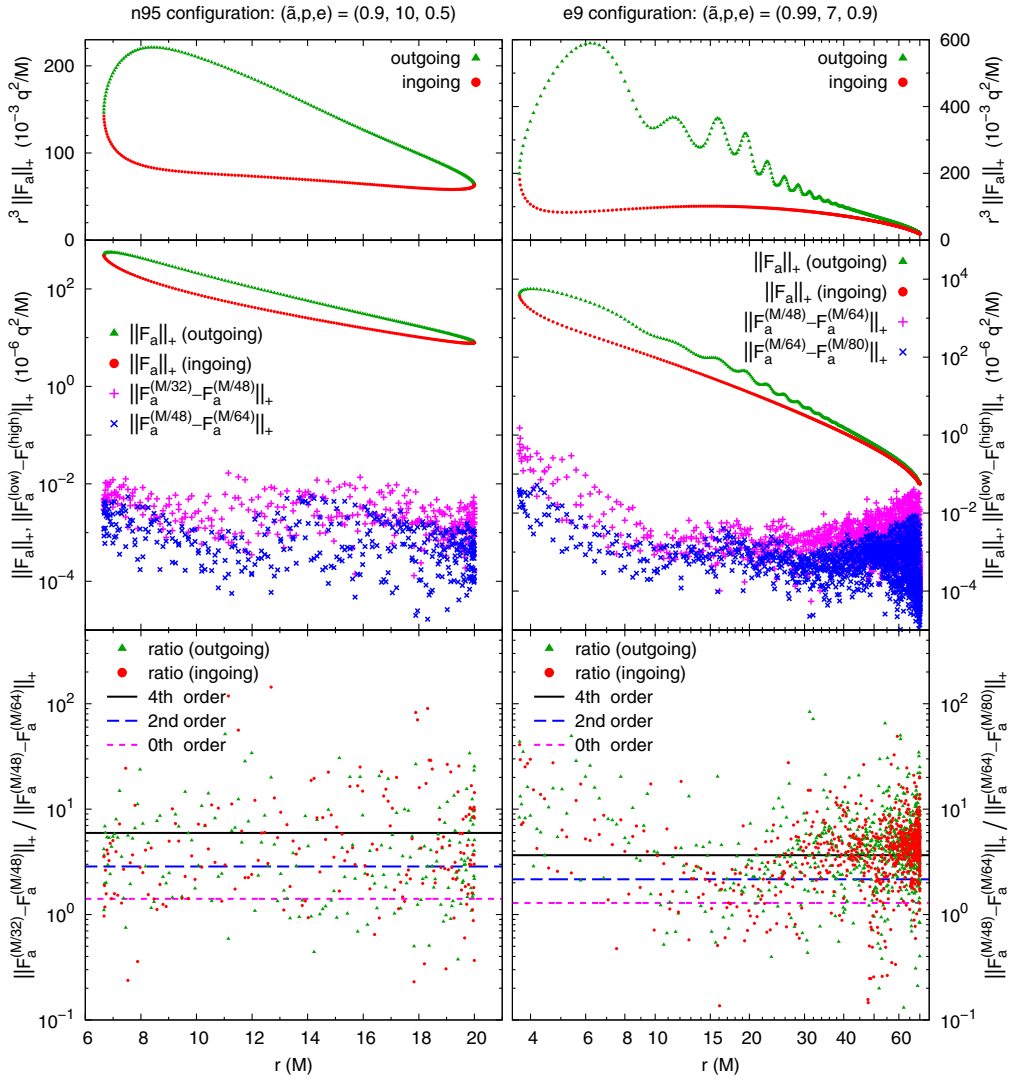


FIG. 9. This figure shows the convergence of the self-force with numerical resolution for the n95 and e9 configurations. For the n95 configuration the convergence is calculated using the dro4-32, dro6-48, and dro8-64 numerical grids (labeled by their finest $\Delta R_*$ of $M/32$, $M/48$, and $M/64$ respectively), while for the e9 configuration the convergence is calculated using the dro6-48, dro8-64, and dro10-80 numerical grids (labeled by their finest $\Delta R_*$ of $M/48$, $M/64$, and $M/80$ respectively). The top subplots in each column show the configuration's self-force loop for the positive-definite pointwise norm of the self-force, $\|F_a\|_+$. The middle subplots show the difference norms $\|F_a^{(\mathrm{low})} - F_a^{(\mathrm{medium})}\|_+$ and $\|F_a^{(\mathrm{medium})} - F_a^{(\mathrm{high})}\|_+$. The bottom subplots show the convergence ratios $\|F_a^{(\mathrm{low})} - F_a^{(\mathrm{medium})}\|_+ / \|F_a^{(\mathrm{medium})} - F_a^{(\mathrm{high})}\|_+$ (plotted separately for the ingoing and outgoing legs of the orbit), along with the theoretical values of this ratio for 0th-, 2nd-, and 4th-order convergence. Notice that there is no systematic difference in the convergence ratios between the ingoing (red symbols) and outgoing (green symbols) legs of the orbit.

(ii) In the top row, the left plot shows $r$ and $\phi$ as functions of the coordinate time $t$, while the right plot shows a plan view of the orbit, i.e., a parametric plot with $x = r(t)\cos(\phi(t))$ and $y = r(t)\sin(\phi(t))$.

(iii) The coordinate-time scale always runs from $-\frac{1}{2}T_r$ to $+\frac{1}{2}T_r$, with $t = 0$ corresponding to periastron. (That is, this "coordinate time" is in fact identical to the modulo time.)

(iv) In the lower three rows of each figure, the left column of plots shows each $F_i$ (in units of $10^{-6}q^2/M$) as a function of coordinate time $t$. For the ze4, ze9, and zze9 zoom-whirl configurations, these plots also show the self-force for the circular-orbit configurations (circ-ze4, circ-ze9, and circ-zze9, respectively) with orbital radius equal to the zoom-whirl configurations' periastron radius.

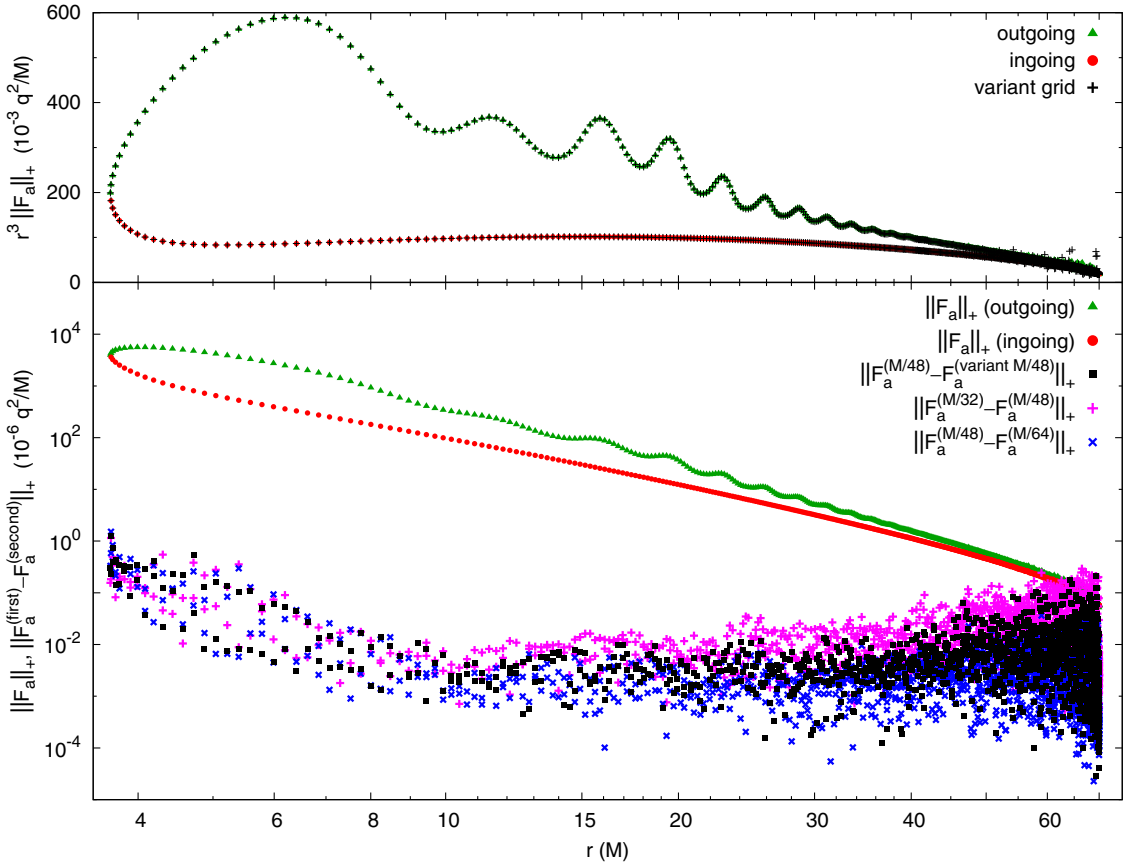


FIG. 10. This figure shows a numerical verification that our computed self-force is (approximately) independent of the choice of worldtube and other numerical parameters, for the e9 configuration [which has $(\tilde{a}, p, e) = (0.99, 7, 0.9)$]. The top subplot shows the self-force loop for the positive-definite pointwise norm of the self-force, $\|F_a\|_+$, with computations using the dro6-48 and variant dro6-48 worldtube/grids overplotted. The points all coincide visually to high accuracy. The bottom subplot shows a quantitative assessment, the norm of the change in the self-force between the dro6-48 and variant dro6-48 worldtube/grid computations, $\|F_a^{(M/48)} - F_a^{(\text{variant } M/48)}\|_+$, together with the change-in-resolution difference norms $\|F_a^{(M/32)} - F_a^{(M/48)}\|_+$ and $\|F_a^{(M/48)} - F_a^{(M/64)}\|_+$ for comparison. Notice that the variant-grid change in the self-force is very small, similar in magnitude to the change-in-resolutions change in the self-force.

(v) In the lower three rows of each figure, the center and right columns of plots each show the scaled self-force $(r/M)^3 F_i$ (in units of $10^{-3} q^2/M$). The center column of plots show $(r/M)^3 F_i$ as a a function of coordinate time $t$. The right column of plots show $(r/M)^3 F_i$ as a function of $r$, forming self-force "loops" plots of the type introduced by [71].

(vi) In each self-force plot (except the ze98 $r^3 F_i$ plots) the total self-force is shown in black and labeled "total", the dissipative part of the self-force is shown in red and labeled "diss," and the conservative part of the self-force is shown in green and labeled "cons." The dissipative and conservative parts are omitted in the ze98 $r^3 F_i$ plots to reduce clutter.

(vii) In each self-force plot the outgoing half of the orbit ($t \geq 0$) is shown in fully saturated color (black, red, or green), while the ingoing half of the orbit ($t \leq 0$) is shown in partially saturated color (grey, red, or green).

(viii) In the self-force loop plots (the right column) the loops are labeled with arrows to show the particle's direction of motion. The dissipative part of $F_t$, the conservative part of $F_r$, and the dissipative part of $F_\phi$ are each independent of the direction of motion. The conservative part of $F_t$, the dissipative part of $F_r$, and the conservative part of $F_\phi$ typically differ between ingoing (pre-periastron, $t < 0$) and outgoing (post-periastron, $t > 0$) motion, forming visible loops.

### G. High-eccentricity orbits

Figures 15–18 show our computed self-force for the e8, e8b, e9, and e95 high-eccentricity configurations, respectively.

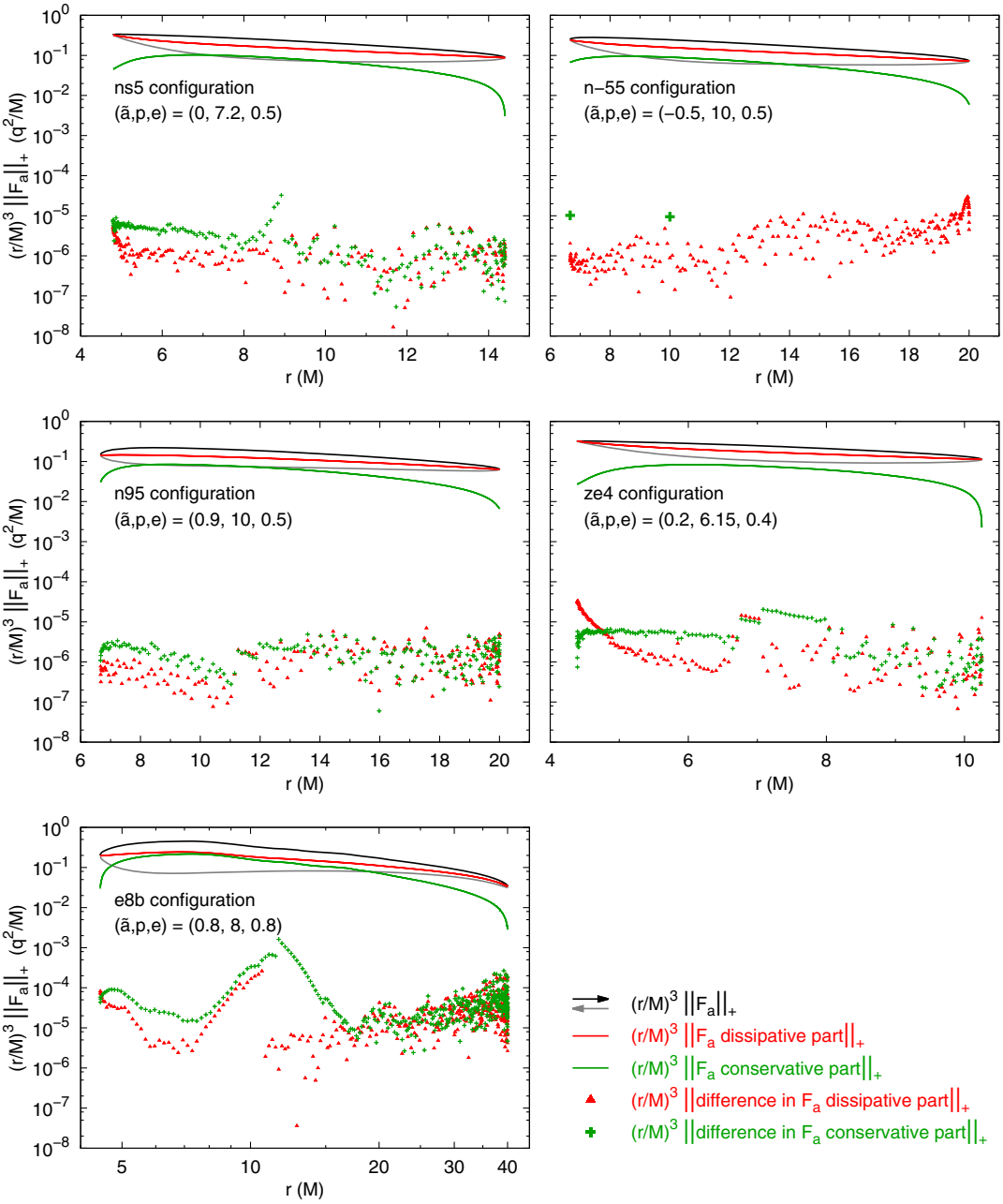


FIG. 11. This figure shows the pointwise norms $\|\cdot\|_+$ of the differences between our computed self-force dissipative and conservative parts and values computed using Warburton and Barack's frequency-domain code [52]. For the n-55 configuration the conservative part is compared with data for $\chi = 0$ and $\chi = \pi/2$ from [52] (Table II); all other comparisons are with unpublished results kindly provided by Warburton.

For these configurations the self-force is strongly localized around the periastron passage. Even though the particle spends most of its time at large radii, the $\sim r^{-3}$ far-field scaling of the self-force with radius implies that the orbital evolution will also be dominated by the periastron passage.

These configurations also show strong oscillations ("wiggles") in the self-force shortly after the periastron passage; we discuss these in Sec. III I.

### H. Zoom-whirl orbits

Figures 19–22 give an overview of our computed self-force for the ze4, ze9, zze9, and ze98 zoom-whirl configurations, respectively. Figures 23 and 24 show the self-force during the whirl phase in more detail for these configurations.

Although the self-force is strictly speaking nonlocal, influenced by the particle's entire past trajectory, in

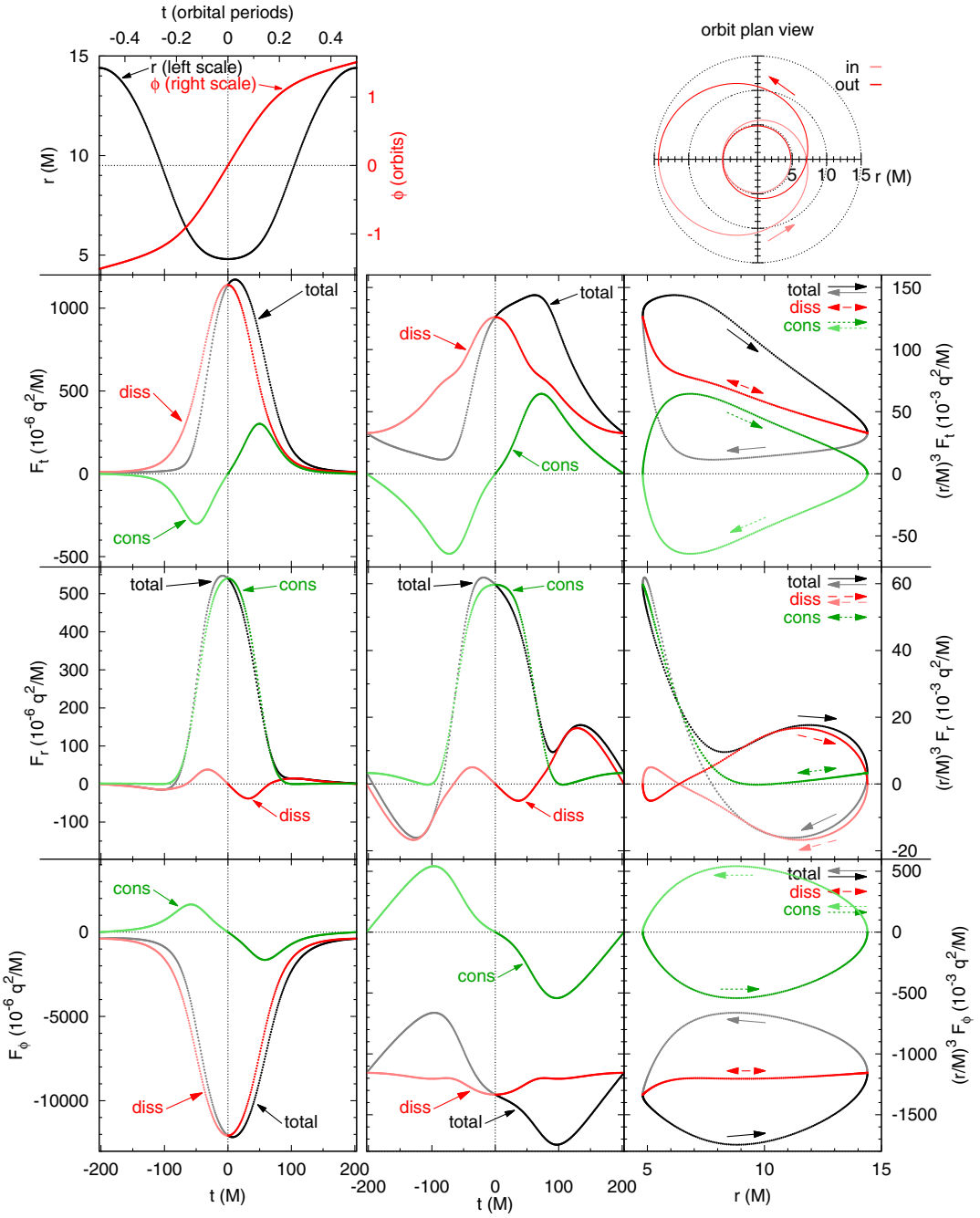


FIG. 12. This figure shows the self-force for the ns5 configuration, which has $(\tilde{a}, p, e) = (0, 7.2, 0.5)$.

practice the influence of distant times is usually small, i.e., the self-force is usually dominated by the effects of the particle's immediate past. We thus expect that if the whirl phase of a zoom-whirl orbit is sufficiently long, the self-force should be very close to that of a circular orbit at the same radius. Figure 23 shows a numerical test of this hypothesis for the ze4, ze9, and zze9 configurations, comparing their whirl-phase self-forces to those of the corresponding circ-ze4, circ-ze9, and circ-zze9 circular-orbit configurations, respectively.[18] For the ze4 configuration the agreement is only modest, presumably because of the relatively short whirl phase. For the ze9 and zze9 configurations the agreement is excellent.

[18]We were unable to calculate the self-force for the circ-ze98 configuration due to numerical instabilities in our evolution code for $m \geq 6$.

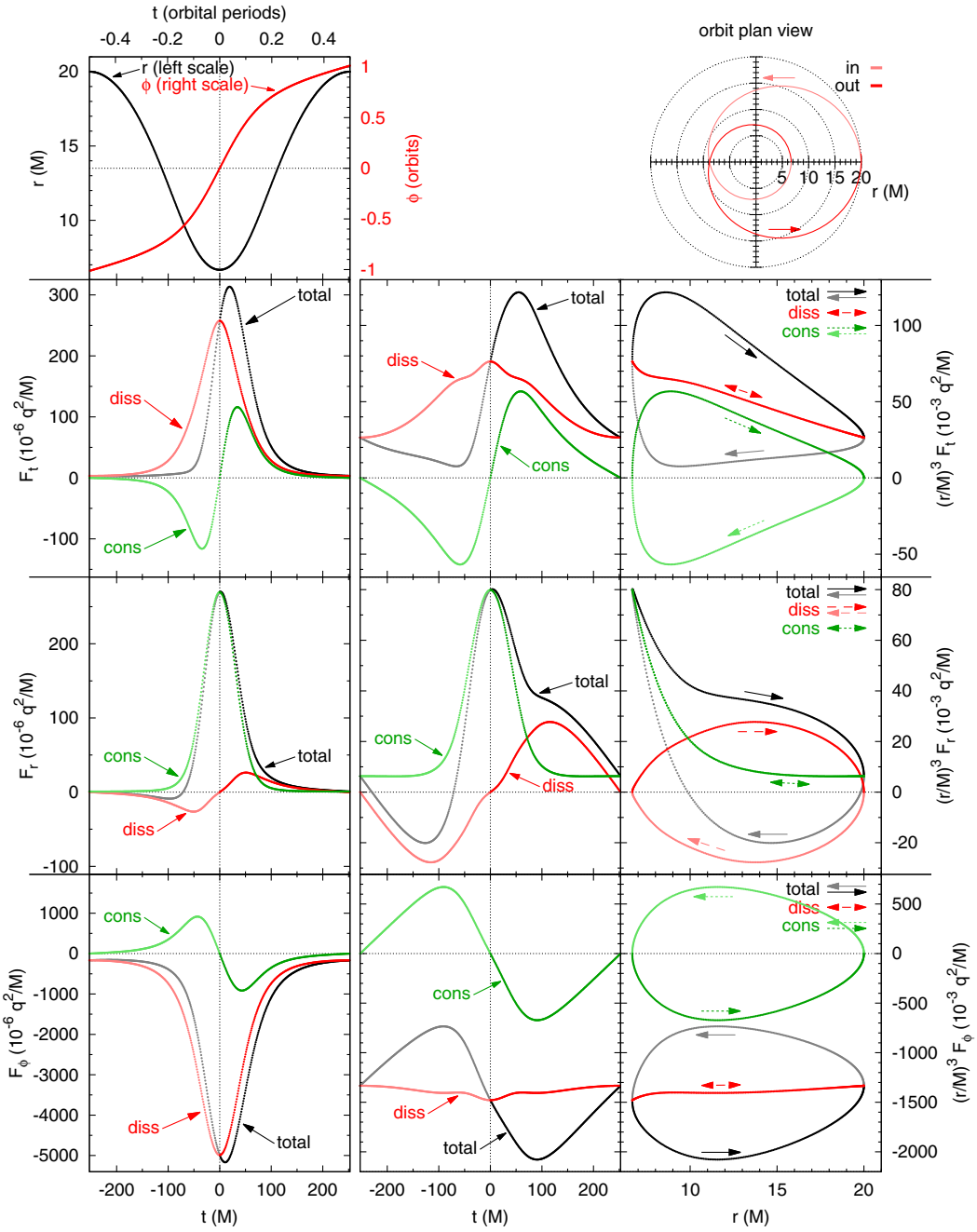


FIG. 13. This figure shows the self-force for the n-55 configuration, which has $(\tilde{a}, p, e) = (-0.5, 10, 0.5)$.

A close examination of Figs. 20 and 21 shows small "spikes" in $F_r$ at the entry/exit to the ze9 and zze9 configurations' whirl phases. These can be seen at an expanded scale in Fig. 23. At the whirl-phase entry, these configurations' $F_r$ first becomes slightly negative, then rises to slightly overshoot its whirl-phase value (this is the "spike" visible in Figs. 20 and 21), then decreases slightly to reach the whirl-phase value. At the zoom-whirl exit, $F_r$ decreases smoothly to a slightly negative value, then rises slightly to its post-whirl (near-zero) value.[19] Haas ([62], Fig. 17) has calculated the self-force for our ze9 configuration and finds similar overshooting behavior. Barack [104] suggests that the underlying cause of this behavior is the particle's strong radial acceleration when entering/leaving the whirl

[19]The visual appearance of these $F_r$ curves in Fig. 23 somewhat resembles a step function passed through a low-pass filter, although we make no claim that this is in any way the actual mechanism involved.

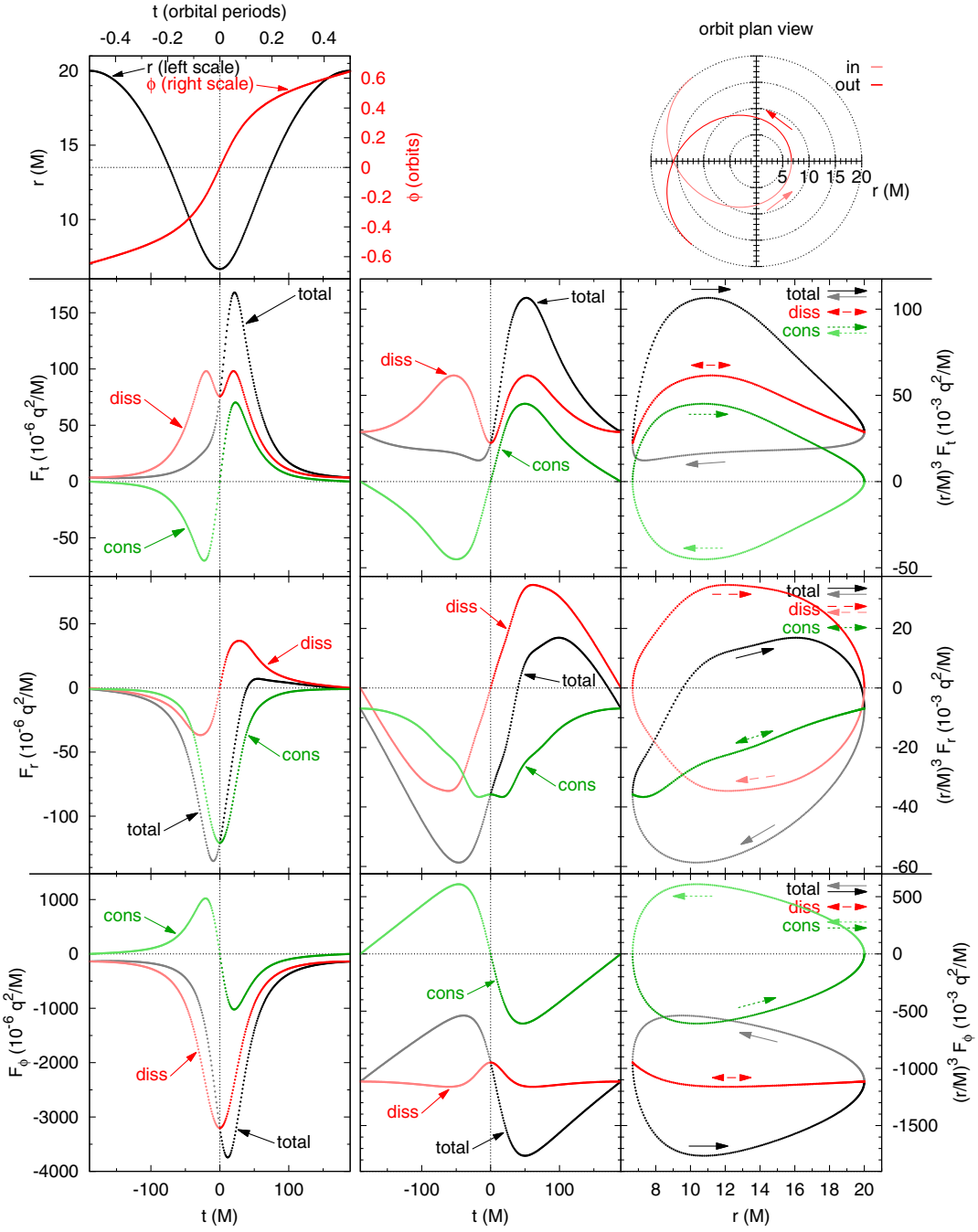


FIG. 14. This figure shows the self-force for the n95 configuration, which has $(\tilde{a}, p, e) = (0.9, 10, 0.5)$.

phase, but so far as we know no quantitative explanation is known.

For the ze98 configuration (an extreme zoom-whirl orbit), Fig. 24 shows quite complicated phenomenology.

(i) At the entrance to the whirl phase (times $-110M \lesssim t \lesssim -40M$), $F_r$ shows small high-frequency oscillations superimposed on a larger lower-frequency oscillation; these oscillations last for approximately $60M$ (about $1/4$ of the entire whirl phase's duration). $F_t$ and $F_\phi$ show small overshoots of their whirl-phase values, but no visible high-frequency oscillations.

(ii) Well before the exit from the whirl phase (times $30M \lesssim t \lesssim 75M$), while the particle is still very close to a circular orbit, $F_r$ increases in amplitude by $\sim 5\%$ (becoming more negative). Unfortunately, while our highest and 2nd-highest-resolution results agree on the overall sign of this change, they differ by roughly a factor of 2 in its magnitude. (This is the only time at which these results differ significantly.) This suggests that higher-resolution data is needed to reliably quantify this feature.

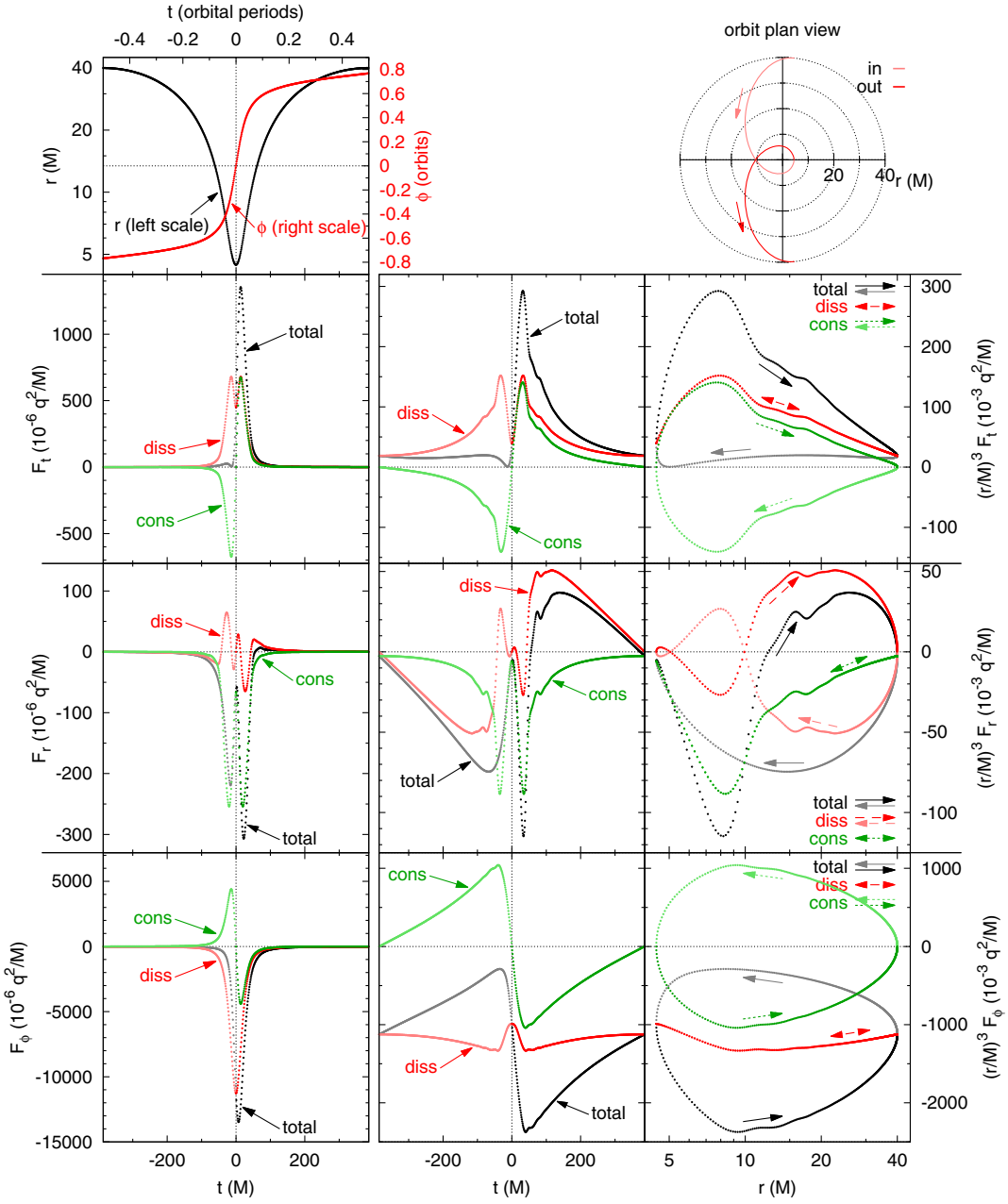


FIG. 15. This figure shows the self-force for the e8 configuration, which has $(\tilde{a}, p, e) = (0.6, 8, 0.8)$. The self-force loops (right column) are plotted using a logarithmic radial scale. Notice the wiggle in the self-force on the outgoing leg of the orbit, near $t = 100M$ past periastron, at $r \approx 16M$; we discuss this in Sec. III I. Because the dissipative-conservative decomposition (2.71) and (2.72) is nonlocal, the dissipative and conservative parts of the self-force also show wiggles before periastron.

(iii) In this same time period (times $30M \lesssim t \lesssim 75M$) $F_t$ and $F_\phi$ both decrease in amplitude.
(iv) Shortly before the exit from the whirl phase (times $75M \lesssim t \lesssim 110M$), when the particle is significantly departing from a near-circular orbit, all components of $F_a$ decrease in magnitude towards their post-whirl (small) values. None of the components shows any visible overshoot.
(v) All components of $F_a$ are significantly time asymmetric about the periastron passage.

This phenomenology is generally consistent between the dro10-80 and dro8-64 numerical resolutions. However, this configuration is a very difficult one for our numerical evolution scheme[20] and it remains possible that some of these features are numerical artifacts. We

[20]At lower resolutions we see numerical instabilities in the ze98 evolutions at times close to periastron. Our numerical evolutions are unstable for $m \geq 6$ for the circ-ze98 configuration (a circular orbit at the ze98 configuration's periastron radius).

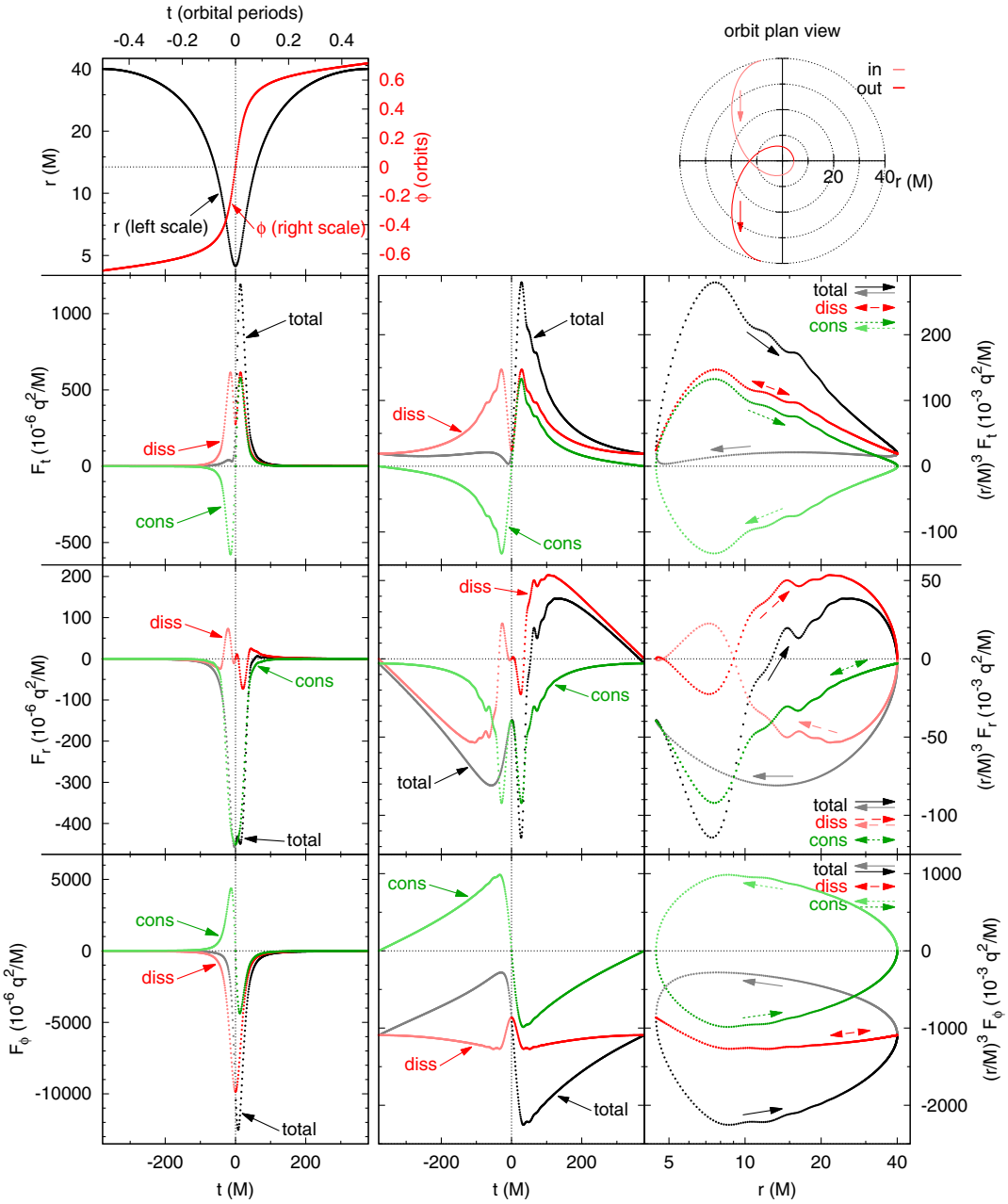


FIG. 16. This figure shows the self-force for the e8b configuration, which has $(\tilde{a}, p, e) = (0.8, 8, 0.8)$. The self-force loops (right column) are plotted using a logarithmic radial scale. Notice the wiggles in the self-force on the outgoing leg of the orbit, between $t \approx 50M$ and $100M$ past periastron, at $r \approx 15M$; we discuss this in Sec. III I. Because the dissipative-conservative decomposition (2.71) and (2.72) is nonlocal, the dissipative and conservative parts of the self-force also show wiggles before periastron.

will need to obtain higher-resolution data to resolve this question.

## I. Wiggles

In the configurations which combine a highly spinning black hole and a prograde high-eccentricity orbit (the e9, e95, and ze98 configurations, shown in Figs. 17, 18, and 22 respectively), there are prominent and rapid oscillations ("wiggles") in $r^3 F_a$ shortly after periastron. These oscillations are also visible to a lesser extent in the configurations with moderate black-hole spins and prograde moderate-eccentricity orbits, the e8 and e8b configurations (shown in Figs. 15 and 16 respectively). Figure 25 shows the wiggles for the e9, e95, and ze98 configurations at an expanded scale.

Notice that (except for the ze98 configuration, discussed in Sec. III H) the self-force varies relatively smoothly prior

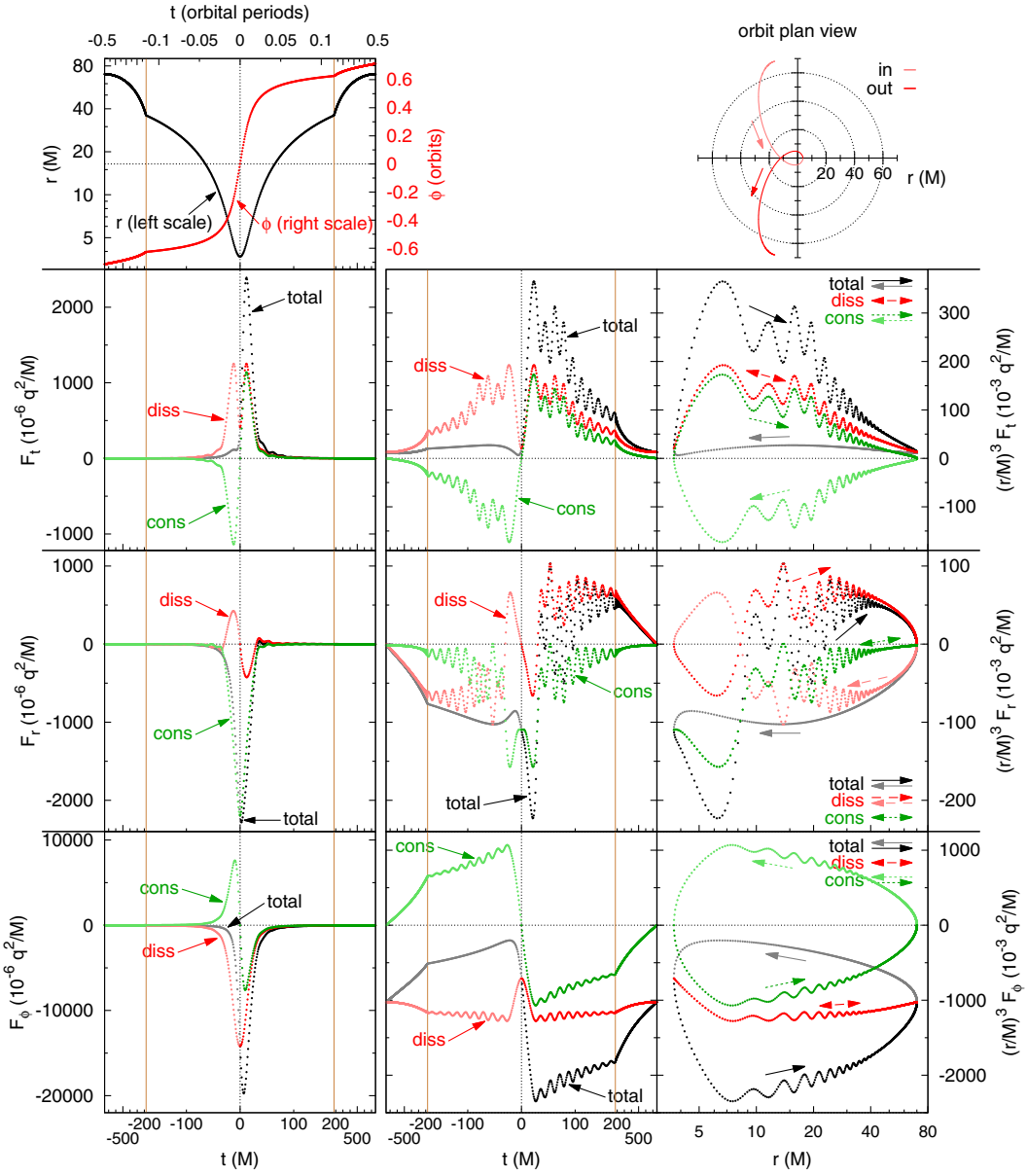


FIG. 17. This figure shows the self-force for the e9 configuration, which has $(\tilde{a}, p, e) = (0.99, 7, 0.9)$. In the time-domain plots (left and center columns) the central $|t| \leq 175M$ around periastron (marked by the vertical lines) is plotted at an expanded horizontal scale. The self-force loops (right column) are plotted using a logarithmic radial scale. Notice the many wiggles in the self-force on the outgoing leg of the orbit; we discuss these in Sec. III I. Because the dissipative-conservative decomposition (2.71) and (2.72) is nonlocal, the dissipative and conservative parts of the self-force also show wiggles before periastron.

to periastron—wiggles occur only *after* the particle's periastron passage ($t = 0$). This suggests that the wiggles are in some way *caused* by the particle's close passage by the large black hole. We will discuss wiggles' phenomenology and causal mechanisms in a following publication.

## IV. DISCUSSION

### A. Overall assessment

Our computational scheme combines a number of ingredients:

(i) the initial formulation of the scalar-field toy model for the $\mathcal{O}(\mu)$-perturbed scalar-field equations, using a point-particle source,
(ii) the Barack-Golbourn-Vega-Detweiler effective-source regularization,
(iii) our specific choice for the puncture field,
(iv) the $m$-mode Fourier decomposition, and the corresponding formulation of the puncture field and effective source in terms of elliptic integrals,
(v) the introduction of a worldtube, which moves in $(r, \theta)$ to follow the particle's motion around the orbit,

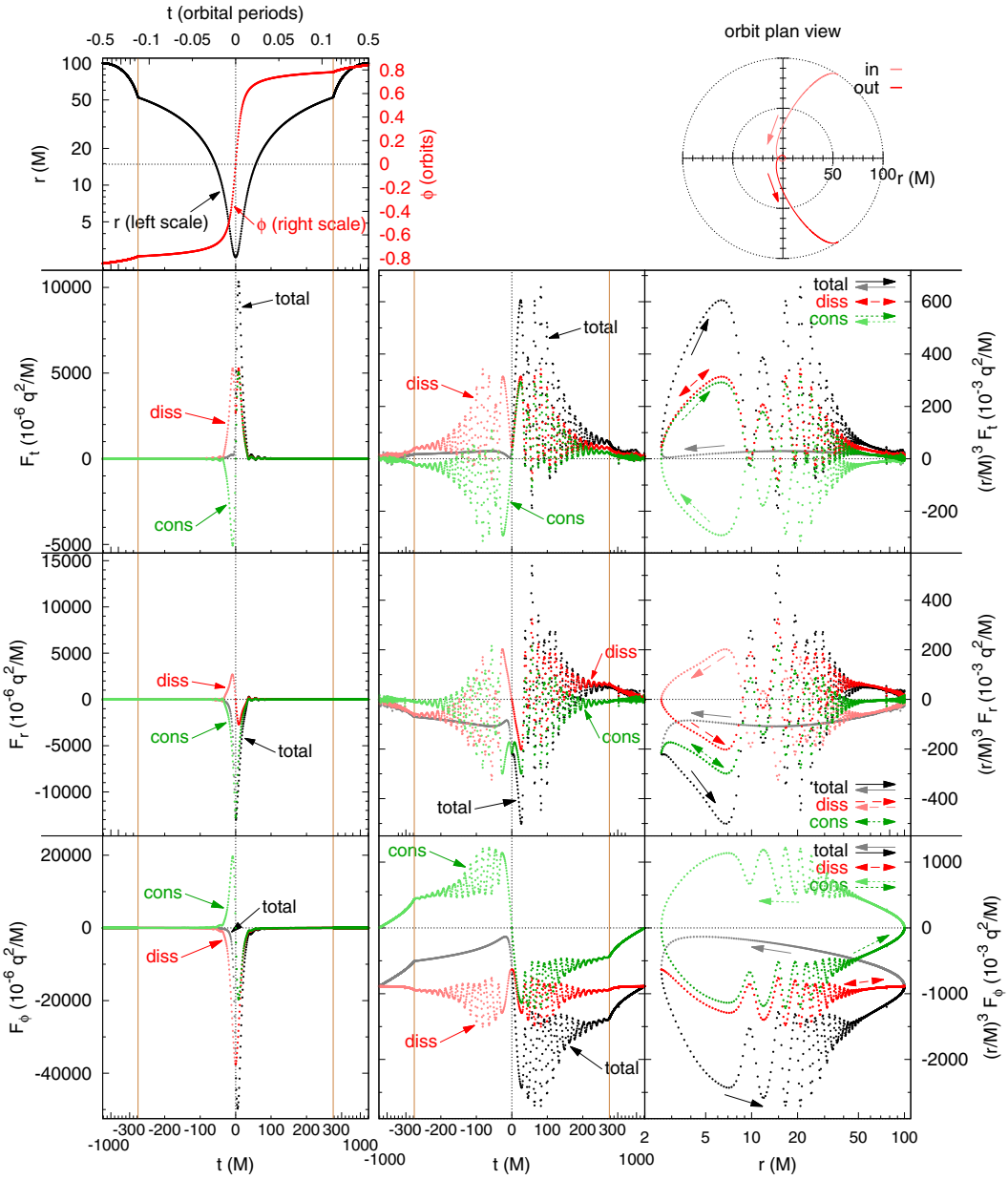


FIG. 18. This figure shows the self-force for the e95 configuration, which has $(\tilde{a}, p, e) = (0.99, 5, 0.95)$. In the time-domain plots (left and center columns) the central $|t| \leq 275M$ (marked by the vertical lines) is plotted at an expanded horizontal scale. The self-force loops (right column) are plotted using a logarithmic radial scale. Notice the many wiggles in the self-force on the outgoing leg of the orbit; we discuss these in Sec. III I. Because the dissipative-conservative decomposition (2.71) and (2.72) is nonlocal, the dissipative and conservative parts of the self-force also show wiggles before periastron.

(vi) the Zenginoğlu compactification and hyperboloidal slices, and

(vii) a finite-difference numerical evolution using Berger-Oliger mesh refinement and OpenMP-based parallelization.

The initial $\mathcal{O}(\mu)$ perturbation formulation with a point-particle source is clearly a reasonable starting point for the scalar-self-force problem. We discuss possible extensions to this in Sec. IV B 4.

The Barack-Golbourn-Vega-Detweiler effective-source regularization scheme works well. It involves no approximations [a solution of the regularized equation (2.5) is an exact solution of the $\mathcal{O}(\mu)$ field equations], the analytical computation of the singular field and effective source can be done with symbolic-algebra software, and the resulting regularized equation is computationally tractable.

In this work, we use a 4th-order puncture for equatorial orbits in Kerr spacetime. While higher-order,

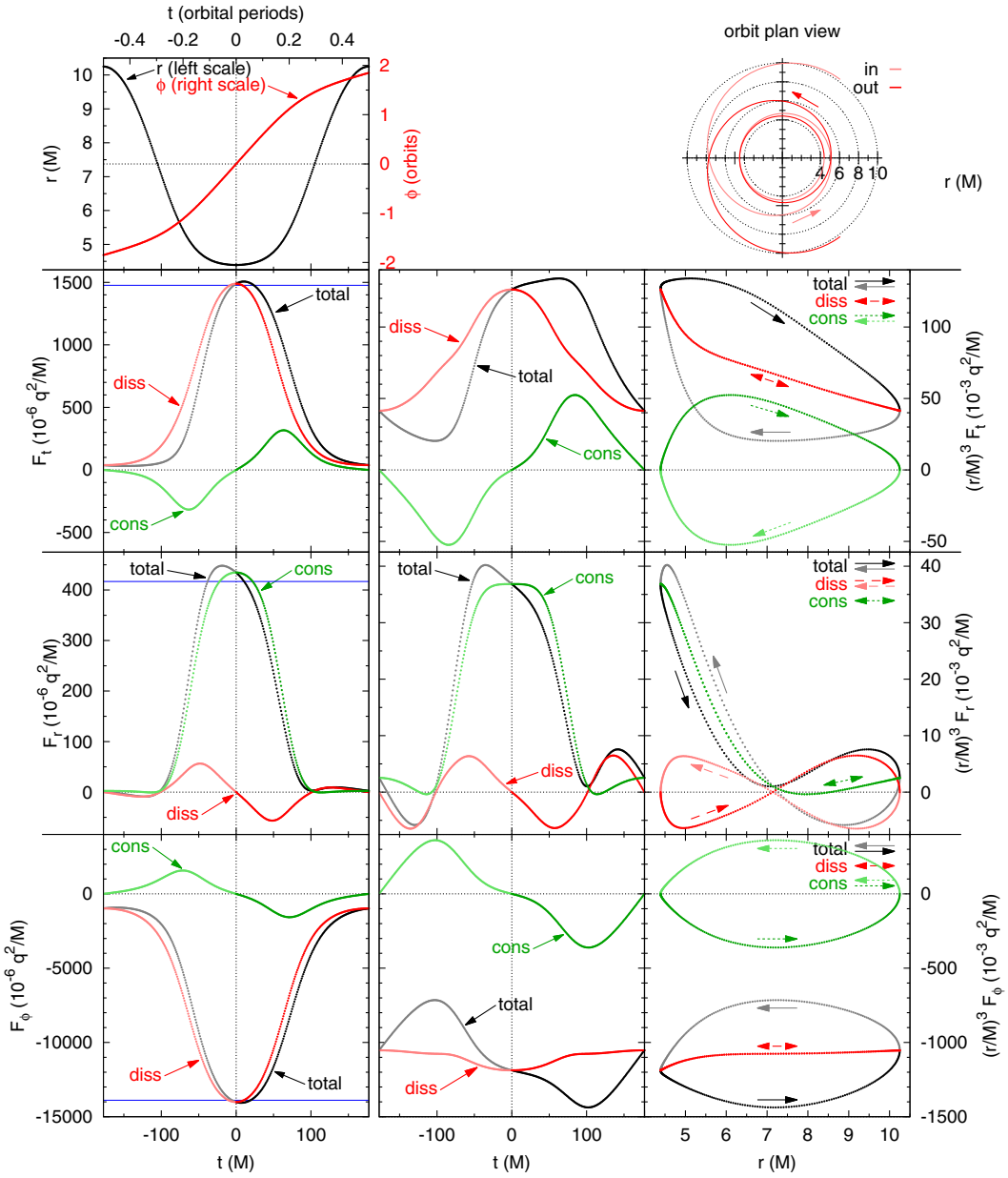


FIG. 19. This figure shows the self-force for the ze4 configuration, which has $(\tilde{a}, p, e) = (0.2, 6.15, 0.4)$. This is a mild zoom-whirl orbit; the particle completes about 2 orbits at $r \approx 4.5M$ during the approximately $125M$ of the whirl phase. In the left column, the horizontal blue line in each self-force subplot shows the self-force for the circ-ze4 circular-orbit configuration; this configuration has the same orbital radius as the ze4 configuration's periastron radius. The self-force near to and during the whirl phase is shown at an expanded scale in Fig 23.

smoother punctures are available [56,87], we (like other researchers [69,71,74,88]) find that 4th order represents a good "sweet spot" compromise between a high-order puncture—which enables high numerical accuracy and fast convergence at the cost of a having a complicated and expensive-to-evaluate source—and a low-order puncture, which is simple and fast to evaluate, but yields poor convergence and numerical accuracy. However, the computation of the effective source is still computationally expensive. Further optimization of this computation would be very useful.

The $m$-mode Fourier decomposition works very well: it provides some parallelism "for free" (each $m$-mode evolution can be performed independently), it reduces the dimensionality and hence the maximum CPU and memory usage of each individual evolution, and—perhaps most importantly—it allows different numerical techniques and/or parameters to be used for different modes' evolutions.

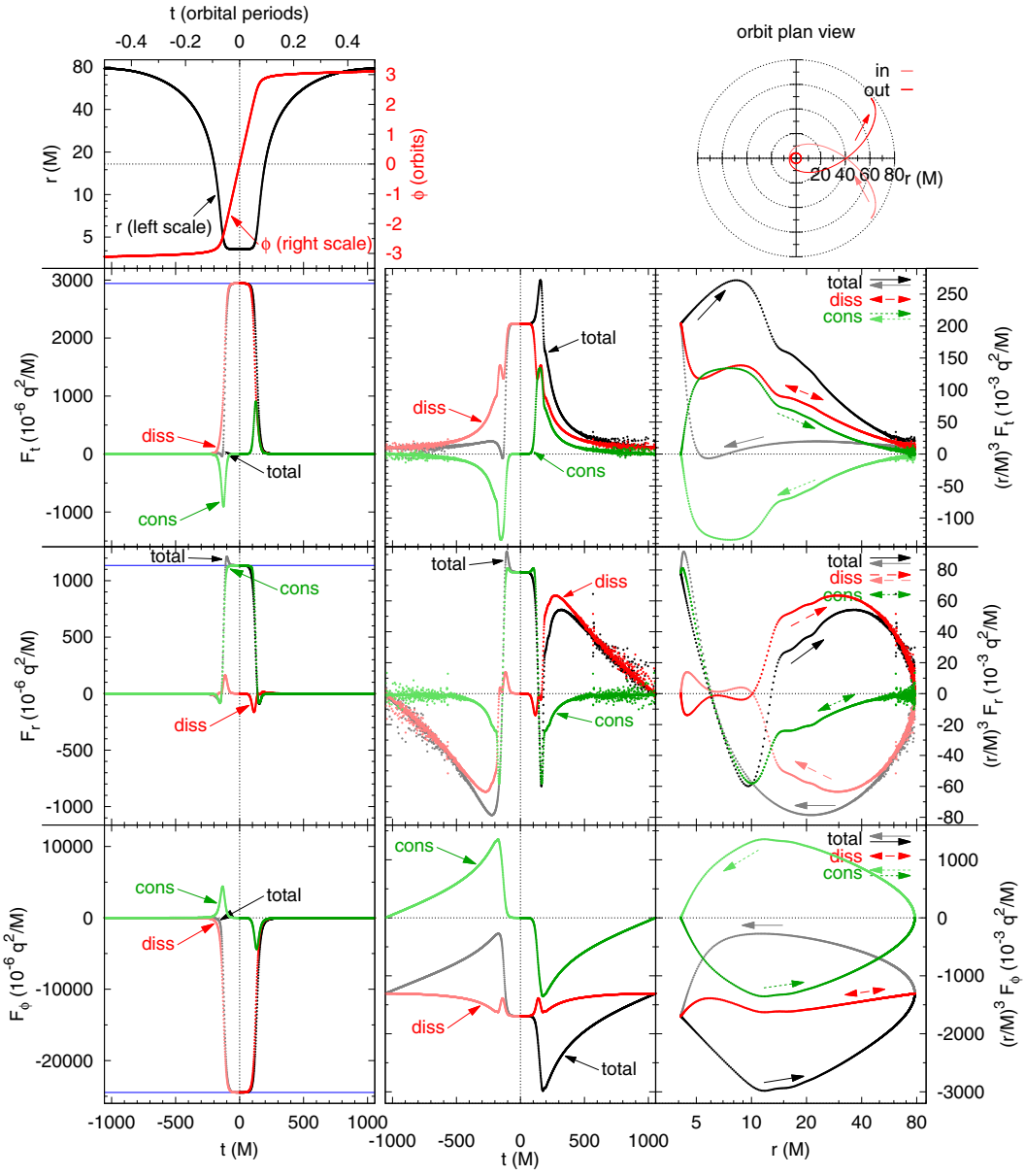


FIG. 20. This figure shows the self-force for the ze9 configuration, which has $(\tilde{a}, p, e) = (0.0, 7.8001, 0.9)$. This is a strong zoom-whirl orbit; the particle completes about $5\frac{1}{3}$ orbits at $r \approx 4.1M$ during the approximately $300M$ of the whirl phase. During the whirl phase the self-force is large and nearly constant; there are also "spikes" in $F_r$ at this phase's entry and exit. In the left column, the horizontal blue line in each self-force subplot shows the self-force for the circ-ze9 circular-orbit configuration; this configuration has the same orbital radius as the ze9 configuration's periastron radius. The self-force loops (right column) are plotted using a logarithmic radial scale. The self-force near to and during the whirl phase is shown at an expanded scale in Fig. 23.

This last advantage may be of great importance in extending our work to the gravitational case, where Dolan and Barack [75] found that the $m = 0$ and $m = 1$ modes suffer from gauge instabilities (they were able to control the $m = 0$ gauge modes, but not the $m = 1$ modes), while the $m \geq 2$ modes are stable.

The moving-worldtube scheme works well, allowing highly eccentric orbits to be simulated while only requiring the (expensive) effective-source computation in a relatively small region of spacetime. We found the implementation of the worldtube at a finite-differencing level to be straightforward (cf. Appendix B 6) once the Boolean predicates for where to use adjusted finite differencing and where to (pre)compute the puncture field were defined correctly (cf. Appendices B 7 and B 8). For orbits of low to moderate eccentricity, the alternative of using a smooth blending "window" function [68,71] is also known to work well. However,

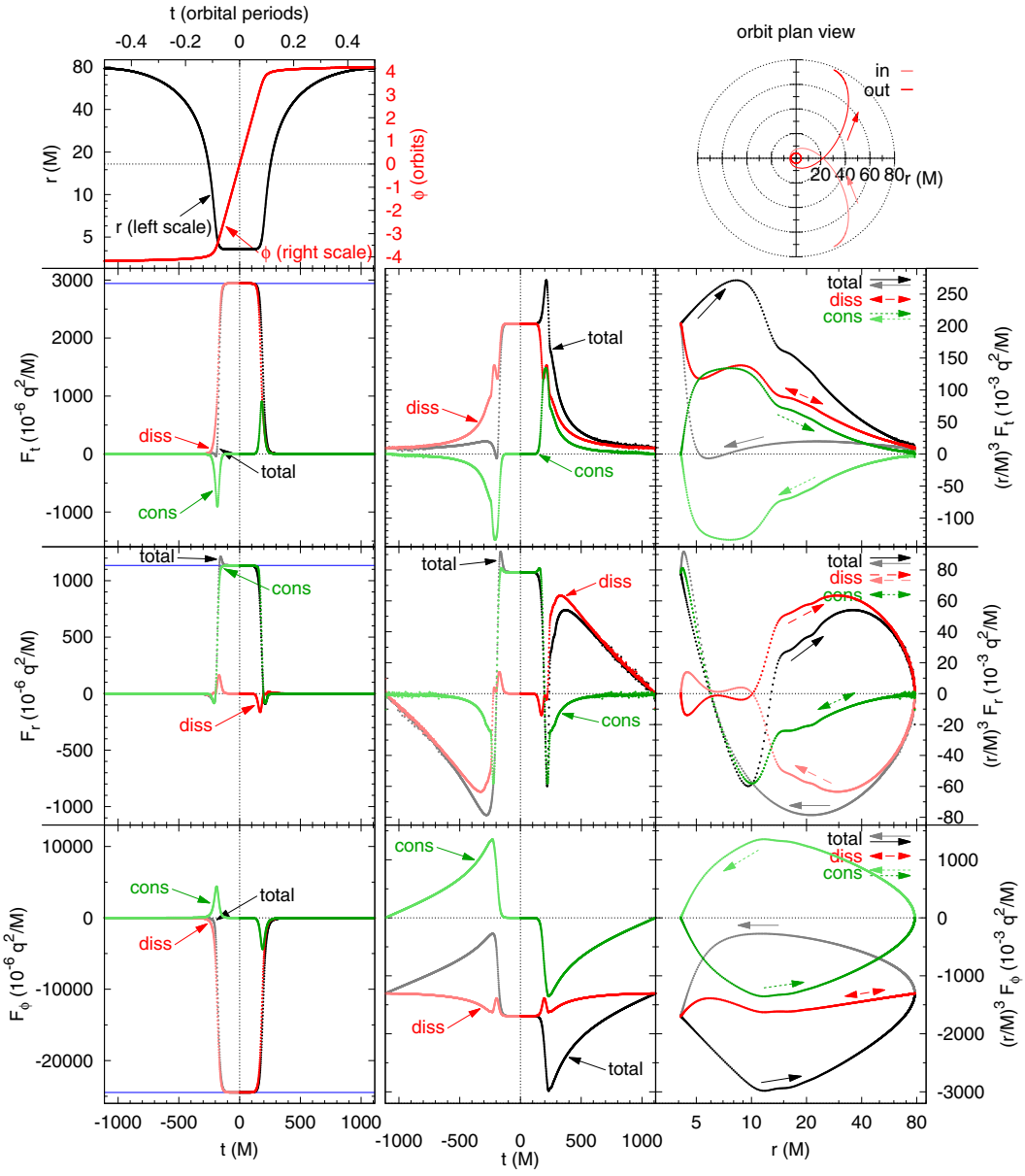


FIG. 21. This figure shows the self-force for the zze9 configuration, which has $(\tilde{a}, p, e) = (0.0, 7.800\,001, 0.9)$. This is a very strong zoom-whirl orbit; the particle completes about $7\frac{3}{4}$ orbits at $r \approx 4.1M$ during the approximately $450M$ of the whirl phase. During the whirl phase the self-force is large and nearly constant; there are also "spikes" in $F_r$ at this phase's entry and exit. In the left column, the horizontal blue line in each self-force subplot shows the self-force for the circ-zze9 circular-orbit configuration; this configuration has the same orbital radius as the zze9 configuration's periastron radius. The self-force loops (right column) are plotted using a logarithmic radial scale. The self-force near to and during the whirl phase is shown at an expanded scale in Fig. 23.

extending this to highly eccentric orbits may require making the window function time dependent, which would introduce additional terms into the evolution equations.

Like other researchers (e.g., [68]), we find the Zenginoğlu compactification and hyperboloidal slices to work very well. They are easy to implement and provide slices which span the entire spacetime outside the event horizon, allowing stable and highly accurate horizon and $\mathcal{J}^+$ outgoing boundary conditions. Slices which reach $\mathcal{J}^+$ also allow a direct computation of the emitted radiation reaching $\mathcal{J}^+$, although for simplicity we have not done so here.

Our numerical evolution uses finite differencing and Berger-Oliger mesh-refinement techniques which are now standard in numerical relativity. However, there are three main complications which combine to make the use of standard adaptive-mesh-refinement frameworks such as

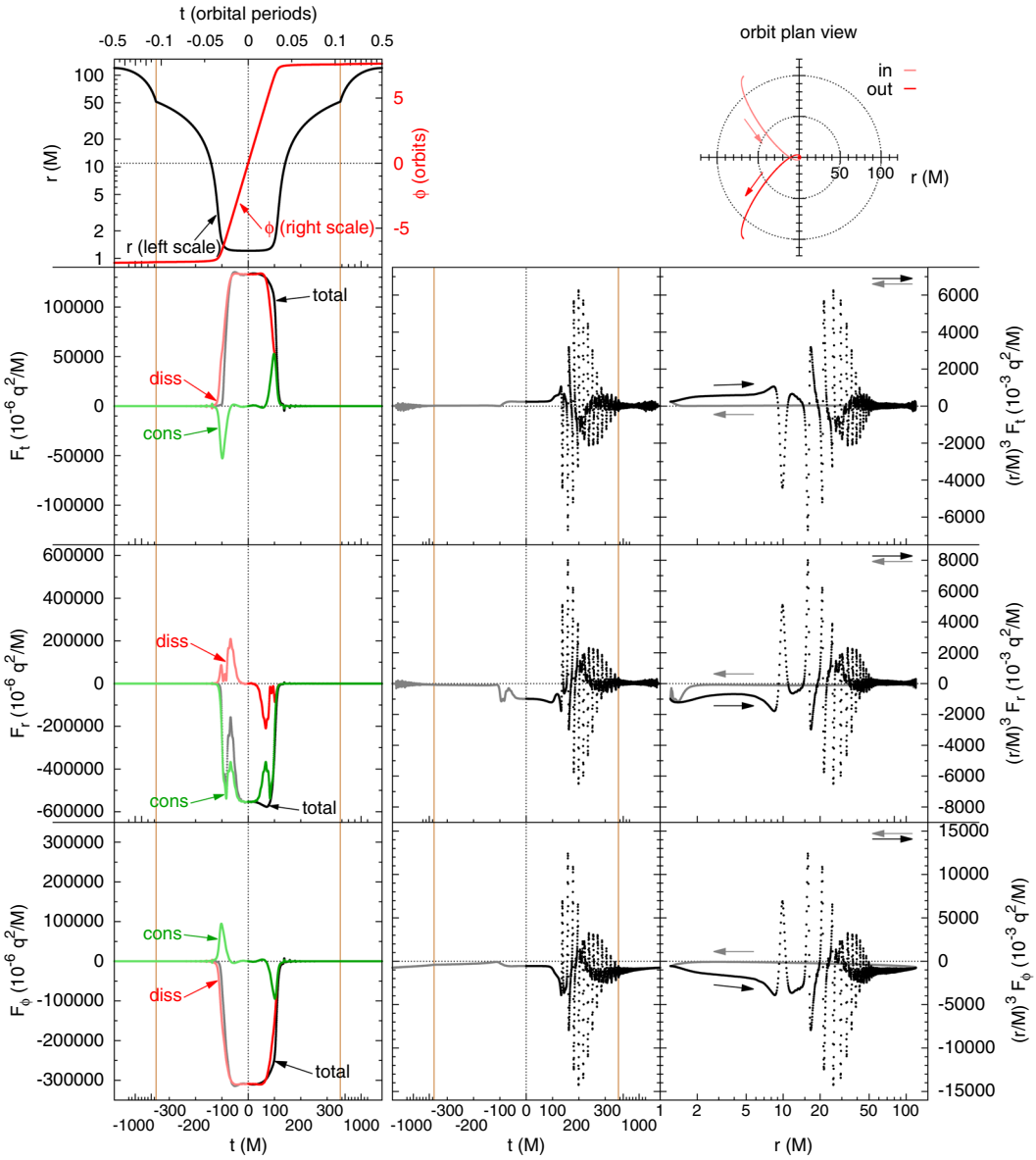


FIG. 22. This figure shows the self-force for the ze98 configuration, which has $(\tilde{a}, p, e) = (0.99, 2.4, 0.98)$. This is an extreme zoom-whirl orbit; the particle completes about 15 orbits at $r \approx 1.2M$ during the approximately $220M$ of the whirl phase. The self-force loops (right column) are plotted using a logarithmic radial scale. During the whirl phase the self-force is very large (more than 40 times the peak self-force of any other configuration in this study) and shows a variety of complicated phenomenology; we discuss this in Sec. III H. Notice the many wiggles in the self-force on the outgoing leg of the orbit; we discuss these in Sec. III I. The self-force near to and during the whirl phase is shown at an expanded scale in Fig. 24.

`Cactus` [105–107][21] more difficult and less advantageous than would be the case in many other numerical-relativity calculations:

(i) Our use of a worldtube, and the associated (time-dependent) jump discontinuity in the evolved field $\varphi_m$, means that interpolation and restriction operators must "adjust" the field variables when crossing the worldtube boundary (cf. Appendix B 6). This means that standard mesh-refinement software requires modification to accommodate the worldtube scheme.

(ii) The effective source is expensive to compute, but is only needed inside the worldtube, so the overall cost of integrating our equations at a single grid point is much larger inside the worldtube than outside. The default domain-decomposition parallelization heuristics used by `Cactus` and many other adaptive-mesh-refinement toolkits assume a roughly uniform level of computational cost per grid point across the

[21]See [108] for a survey of other such frameworks.

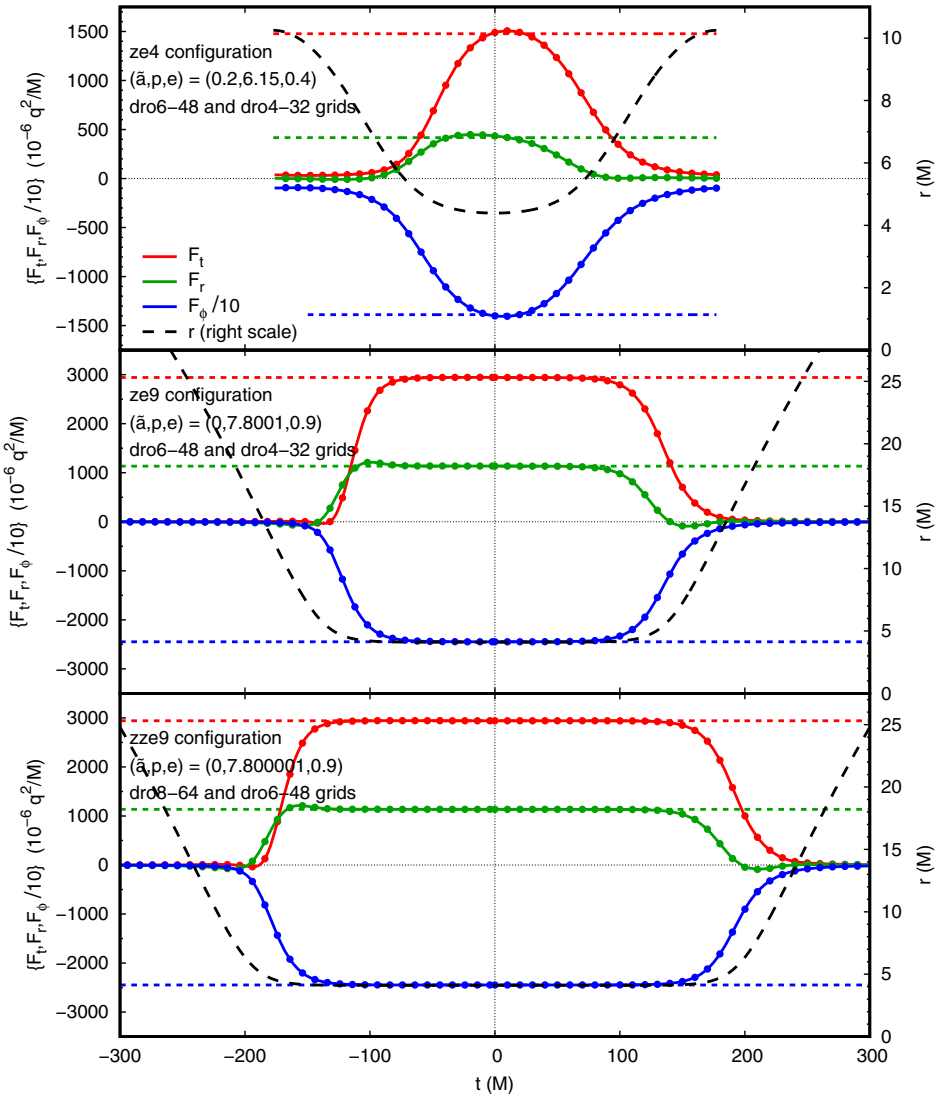


FIG. 23. This figure shows the self-force during the whirl phase for the ze4, ze9, and zze9 zoom-whirl configurations. For each configuration the solid lines show the highest-resolution data, while the dots show the lower-resolution data (sampled approximately every $10M$); these are visually identical. The horizontal short-dashed lines show the self-force for the corresponding circular-orbit configurations (circ-ze4, circ-ze9, and circ-zze9, respectively); these have the same orbit radii as the zoom-whirl configurations' periastrons.

problem domain, and thus would give relatively poor parallel performance on our computation.

(iii) The non-smoothness of the evolved field $\varphi_m$ at the puncture (particle) position limits the finite-differencing order of accuracy attainable there. For our 4th-order puncture, the accuracy is limited to at best $\mathcal{O}((\Delta R_*)^2, (\Delta\theta)^2)$ because our evolution equation (2.23) is 2nd order in space. This reduces the benefits gained from high-order finite-differencing schemes (which are now provided by many mesh-refinement software libraries).

## B. Possible improvements

There are a number of ways in which our results might plausibly be improved. While there is an accuracy/performance tradeoff in almost any finite-difference computation, computational improvements can still usefully be categorized into those which would improve the accuracy of the self-force computation for a given finest-grid resolution, versus those which would improve the efficiency of computing results using essentially the same numerical scheme, versus those which would improve both accuracy and efficiency.

### 1. Computational improvements: Accuracy

There are several ways in which our computational scheme might be improved so as to provide more accurate results for the same finest-grid resolution.

As noted in Sec. II H, our interpolation scheme for computing the effective source close to the particle uses an

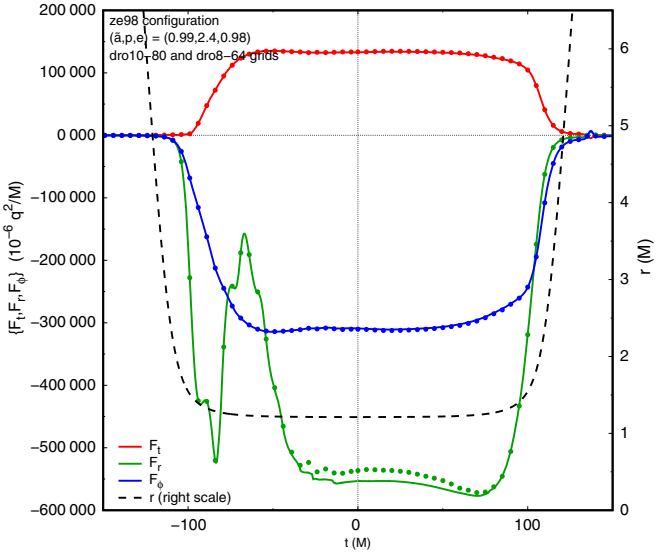


FIG. 24. This figure shows the self-force during the whirl phase for the ze98 zoom-whirl configuration. The solid lines show the highest-resolution data (dro10-80 grids), while the dots show the lower-resolution data (dro8-64 grids, sampled approximately every $5M$); these are visually almost identical except for a $\lesssim 3.5\%$ difference in $F_r$ at times $-35M \lesssim t \lesssim 80M$.

interpolation molecule which crosses the particle position in some cases, reducing the interpolation accuracy. An improved interpolation scheme might improve the overall accuracy of the computation.

As the particle moves through the grid, the limited differentiability of $\varphi_m$ at the puncture effectively introduces noise into the evolution and prevents us from obtaining proper (in our case 4th order) finite-differencing convergence of our results with grid resolution [101]. One way to eliminate this noise and obtain proper finite-differencing convergence would be to use finite-difference operators which specifically "know" the actual functional form of $\varphi_m$ near the puncture. We have experimented with several finite-differencing schemes of this type, but so far with only limited success. At present our code uses the "C2" scheme described in Appendix B 10. We find that this lowers the noise level in the computed self-force by roughly a factor of 3, but our results remain quite noisy and their overall convergence order with respect to grid resolution is still much lower than we would like. Further research on finite-difference operators which incorporate more of the puncture's actual singularity structure would be useful. (We mention one possible finite-differencing scheme of this type in Appendix B 10, but we were not able to obtain stable evolutions with this scheme.)

As noted in Sec. III C, for the e9 configuration we find poor convergence at small radii ($r \lesssim 10M$). We do not yet know the cause of this poor convergence, but fixing it would obviously be highly desirable.

Another possible route to more accurate finite differencing near the puncture might be to use many mesh-refinement levels of small grids in the puncture's immediate neighborhood, so as to obtain very high resolutions at the puncture. Given a Berger-Oliger-style mesh-refinement infrastructure, this is not difficult. However, the interpolations of the fine-grid boundary values from the coarser grids might limit the accuracy improvement, even if buffer zones [109] are used. Further experimentation with this type of grid structure would be useful.

Raising the order of the puncture would improve the smoothness of $\varphi_m$ at the puncture, improving the finite-differencing accuracy there. However, a higher-order puncture would also yield a much more complicated and expensive-to-compute effective source. Our current choice of a 4th-order puncture seems to be a good compromise between smoothness and computational expense.

### 2. Computational improvements: Efficiency

There are a number of ways in which our computational scheme might be made more efficient.

At present our code computes the puncture field and effective source anew at each right-hand-side evaluation whose time coordinate differs from that of the previous evaluation (this happens 50% of the time for the classical 4th-order Runge-Kutta time-integration scheme we currently use). For periodic orbits (including all equatorial orbits) the puncture field and effective source are the same (at a given time-past-periastron) from one orbit to the next, so a much more efficient choice would be to cache the effective source in memory, reusing cached values for all of the evolution after the first orbit. However, such a cache would use a very large amount of memory and would give no benefit for nonperiodic orbits (including almost all nonequatorial orbits).

Simulations of this type are computationally expensive. Our code is currently only partially parallelized, using OpenMP to spread the computation of the singular field, effective source, and evolution-equation right-hand-side across multiple cores of a single processor. This is easy to implement and typically gives a wall-clock speedup of a factor of 12 to 13 using 16 cores. Grid-based parallelism (ultimately based on message passing) is an obvious and widely used way of achieving higher parallelism, and is now well supported by numerical-relativity adaptive-mesh-refinement toolkits such as `Cactus` [105–107] and the Einstein toolkit [110,111]. However, Cactus and many other adaptive-mesh-refinement toolkits generally assume that the cost of computing a grid point is roughly constant across the problem domain. Our worldtube scheme strongly violates that assumption: points inside the worldtube require computing the effective source and thus cost much more than points outside the worldtube. This means that without significant changes to the domain-decomposition heuristics, standard toolkits would give only limited parallel speedup for our worldtube scheme. One possible way to sidestep this issue is to use a domain decomposition for the calculation of the effective source that is independent of the normal domain

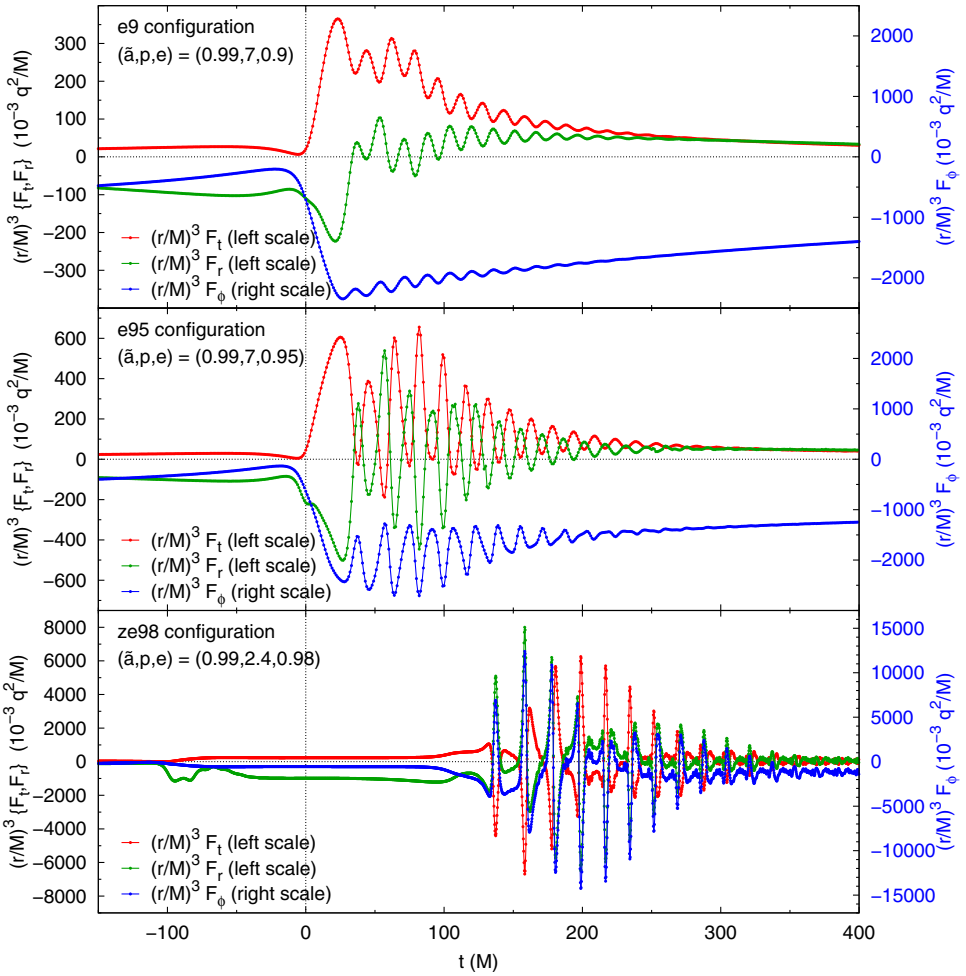


FIG. 25. This figure shows the "wiggle" oscillations in $(r/M)^3 F_a$ for the e9, e95, and ze98 configurations. Notice that wiggles are present only *after* the particle's periastron passage ($t = 0$). Notice also that for the e95 and ze98 configurations the individual wiggles are often nonsinusoidal, with shapes differing between different wiggles (this is particularly evident in the e95 $(r/M)^3 F_\phi$ wiggles). The wiggles' shapes also differ between different $F_a$ components.

decomposition of the full computational grid; such a method was used (without mesh refinement), for example, in [71]. Similarly, it might be that other parallelization techniques such as the task-based model used by the SpECTRE code [112] would yield better parallel speedup.

### 3. Computational improvements: Accuracy and efficiency

At present our computational scheme uses finite differencing with Berger-Oliger mesh refinement. A discontinuous Galerkin method [113] might give spectral (i.e., much better) accuracy/efficiency even with the limited differentiability of $\varphi_{\rm num}$ at the particle. These methods have been used successfully in other numerical-relativity and self-force computations by a number of researchers, e.g., [114–116], as well as in other areas of computational physics involving nonsmooth solutions, e.g., [112,117].

Within the general framework of finite differencing and Berger-Oliger mesh refinement, there are a number of ways in which our computational scheme might be enhanced to better adjust the computations to the solution dynamics, yielding both improved accuracy (higher effective grid resolution) and efficiency (fewer high-resolution grid points "wasted" on regions of spacetime where $\varphi_m$ is relatively slowly varying):

(i) At present our mesh-refinement scheme moves the finer grids with the worldtube but does not otherwise adapt to the solution's dynamics. For an orbit with substantial eccentricity, the field dynamics near the particle are quite different between the particle's periastron and apoastron. It seems likely that an adaptive-mesh-refinement scheme (of the type now widely used in fully nonlinear binary-black-hole simulations) for varying the grid structure around the orbit would substantially improve the computation's overall accuracy/efficiency.

(ii) At present our computational scheme keeps the worldtube size and shape fixed throughout the evolution. An adaptive scheme to adjust (optimize)

these around the orbit could significantly improve the code's accuracy and efficiency. However, unlike the case for adaptive mesh refinement, there are no existing algorithms for making this adjustment. Further research in this area would be valuable.

(iii) For a highly eccentric orbit, many of the higher-$m$ self-force modes are below our code's noise level during much of the orbit. (This can be seen, for example, in Figs. 4 and 5.) The overall efficiency of the computation could be greatly improved by not computing these modes at times when they are essentially purely noise. This would require some means of estimating the time intervals in question, and changes to our initial-worldtube-setup scheme (described in Sec. B 5) to accommodate (re)starting the computation of these modes at a time when the particle is moving much faster than near apoastron.

#### 4. Extensions to more general physical systems

In this work we focus on computing the instantaneous scalar self-force acting on the small body. One straightforward extension to this would be to also compute the scalar field at the particle, using the method suggested in footnote 12. Another straightforward extension would be to also compute the scalar field radiated to infinity ($\mathcal{J}^+$). Given our use of asymptotically hyperboloidal slices which reach $\mathcal{J}^+$, this information is readily available. We have preliminary implementations of both of these extensions; we will discuss their results in a following publication.

Our present results are limited to (bound, geodesic) equatorial particle orbits. Apart from the computational complexity of computing the effective source (which is probably manageable with some reorganization of the Mathematica-generated $C$ code),[22] there appears to be no fundamental obstacle to allowing nonequatorial orbits, and this would be a very useful extension. In particular, this would allow direct exploration of transient $\theta$-$\phi$ resonances [118,119].

Our present results are limited to the "toy model" of a scalar-field particle. Extending these results to a point mass and its gravitational field perturbations would be very interesting but also challenging. While the basic effective-source regularization scheme is already known to be valid for the gravitational case, Dolan and Barack [75] found that the $m = 1$ evolutions suffered from linearly-growing-in-time Lorenz gauge modes which they were not able to control. Stabilizing these modes, and more generally achieving long-time-stable evolutions for all $m$, is an important area for further research.

Our present results are also limited to $\mathcal{O}(\mu)$ perturbations of the (Kerr) background spacetime. LISA could benefit from EMRI waveform templates with $\sim 10^{-8}$ or better fractional orbital-phase accuracy ([120], Sec. 4), which would require the inclusion of both $\mathcal{O}(\mu^2)$ terms and "extended-body" effects caused by the finite size and (in general) nonzero spin of the small body (see, for example, [121] and references therein).

In the longer term, it will also be essential to extend self-force calculations to include orbital evolution. This is conceptually straightforward (though computationally demanding) if the osculating-geodesic approximation is retained (as was done by Warburton *et al.* [122] in their pioneering calculation of gravitational inspiral in Schwarzschild spacetime over a time span of more than 75000 orbits). However, going beyond the osculating-geodesic approximation is more difficult. Diener *et al.* [70] have demonstrated that this can be done for a scalar-field particle in Schwarzschild spacetime, but they were only able to attain relatively modest accuracies and integration time spans ($\sim$20 orbits). Extending their work to higher accuracies and longer integrations is an important area for further research.

[22] Our preliminary experiments with generalizing our current singular field and effective source to nonequatorial orbits suggest that the complexity of the effective-source coefficients increases by a factor of $\sim$40, with a corresponding increase in the size of the machine-generated $C$ code for computing the coefficients. This computation would need to be reorganized in order for it to be practical to compile the Mathematica-generated $C$ code. We believe this is possible, but have not yet done so.

## ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant Number 1417132. B. W. was supported by Science Foundation Ireland under Grant No. 10/RFP/PHY2847, by the John Templeton Foundation New Frontiers Program under Grant No. 37426 (University of Chicago) - FP050136-B (Cornell University), and by the Irish Research Council, which is funded under the National Development Plan for Ireland. This material is based upon work supported by the U.S. National Science Foundation (NSF) under Grant No. CNS-0521433. This research was supported in part by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute, and in part by the Indiana Metabolomics and Cytomics (METACyt) Initiative. The Indiana METACyt Initiative at IU is also supported in part by Lilly Endowment, Inc. We are grateful to Niels Warburton for providing unpublished orbit and self-force results from the code described in [52]. We thank Ian Hinder for performing an eigenvalue analysis of our evolution system, and for helpful discussions on the origin of our evolution scheme's very restrictive stability limit for large $m$. We thank Leor Barack, Sam Dolan, and the other participants of the Capra meetings on Radiation Reaction for many illuminating conversations. J. T. thanks Eric Ost for valuable assistance with a computer cluster used for numerical calculations with early versions of our code, and Virginia J. Vitzthum for comments on this manuscript. J. T.

developed the numerical evolution and tail-fitting/mode-sum codes, performed the numerical evolutions, and did the main data analysis. B. W. developed the algorithms and symbolic-algebra code for computing the puncture field and effective source, and for machine generating the $C$ code for this computation. Both authors contributed to the preparation of this manuscript.

## APPENDIX A: $\tilde{\phi}$ DERIVATIVES

Clearly $d\tilde{\phi} = d\phi$ if $dr = 0$, so for any scalar quantity $Q$ we have

$$\left.\frac{\partial Q}{\partial \phi}\right|_r = \left.\frac{\partial Q}{\partial \tilde{\phi}}\right|_r, \tag{A1}$$

i.e., (since $Q$ is arbitrary),

$$\left.\frac{\partial}{\partial \phi}\right|_r = \left.\frac{\partial}{\partial \tilde{\phi}}\right|_r. \tag{A2}$$

To relate $\partial/\partial_r|_\phi$ and $\partial/\partial_r|_{\tilde{\phi}}$, consider two infinitesimally separated events $X$ and $Y$, with coordinates

$$\begin{aligned} X&\colon\ r = r_X, \qquad \phi = \phi_X, \qquad \tilde{\phi} = \tilde{\phi}_X; \\ Y&\colon\ r = r_X + dr, \qquad \phi = \phi_X. \end{aligned} \tag{A3}$$

Since $\phi$ is the same for events $X$ and $Y$, the definition (2.9) of $\tilde{\phi}$ implies that $\tilde{\phi}_Y = \tilde{\phi}_X + d\tilde{\phi}$ with

$$d\tilde{\phi} = \frac{M\tilde{a}}{\Delta} dr. \tag{A4}$$

Thus for any scalar quantity $Q$ we have (using the chain rule in $(r, \phi)$ coordinates)

$$Q_Y - Q_X = dr \cdot \left.\frac{\partial Q}{\partial r}\right|_\phi \quad \text{since } d\phi = 0. \tag{A5}$$

Using the chain rule in $(r, \tilde{\phi})$ coordinates, we also have

$$Q_Y - Q_X = dr \cdot \left.\frac{\partial Q}{\partial r}\right|_{\tilde{\phi}} + d\tilde{\phi} \cdot \left.\frac{\partial Q}{\partial \tilde{\phi}}\right|_r \tag{A6}$$

$$= dr \cdot \left.\frac{\partial Q}{\partial r}\right|_{\tilde{\phi}} + \frac{M\tilde{a}}{\Delta} dr \cdot \left.\frac{\partial Q}{\partial \tilde{\phi}}\right|_r \quad \text{via (A4)}, \tag{A7}$$

so that [comparing (A5) and (A7)] we have

$$\left.\frac{\partial Q}{\partial r}\right|_\phi = \left.\frac{\partial Q}{\partial r}\right|_{\tilde{\phi}} + \frac{M\tilde{a}}{\Delta} \left.\frac{\partial Q}{\partial \tilde{\phi}}\right|_r, \tag{A8}$$

i.e., (since $Q$ is arbitrary),

$$\left.\frac{\partial}{\partial r}\right|_\phi = \left.\frac{\partial}{\partial r}\right|_{\tilde{\phi}} + \frac{M\tilde{a}}{\Delta} \left.\frac{\partial}{\partial \tilde{\phi}}\right|_r. \tag{A9}$$

## APPENDIX B: DETAILS OF OUR COMPUTATIONAL SCHEME

### 1. Computing $r(r_*)$

Our computational scheme uses grids which are locally uniform in $(R_*, \theta)$. However, the coefficients in many of our equations are given as explicit functions of $r$, so the code needs to know the $r$ coordinate of each grid point. Since $r_*(R)$ is given explicitly by the compactification (2.27) and (2.28), it only remains to compute $r(r_*)$.

Given an input value $r_*^{(\text{input})}$, the corresponding $r^{(\text{input})}$ could be found by using Newton's method to solve the equation $r_*(r) = r_*^{(\text{input})}$ using the definition (2.16). However, for positions just outside the event horizon ($r_* \ll 0$) the near cancellation in computing $r - r_+$ would make this algorithm numerically inaccurate.

Instead, we define a new radial coordinate $y$ by

$$y = \ln\left(\frac{r - r_+}{2M}\right) \tag{B1}$$

so that

$$r = r_+ + 2Me^y. \tag{B2}$$

The definition (2.16) can then be rewritten as

$$r_* = r + 2M\frac{r_+}{r_+ - r_-} y - 2M\frac{r_-}{r_+ - r_-} \ln\left(\frac{r_+ - r_-}{2M} + e^y\right). \tag{B3}$$

Given an input value $r_*^{(\text{input})}$, we first find the corresponding $y^{(\text{input})}$ by using Newton's method to solve the equation

$$r_*(y) = r_*^{(\text{input})} \tag{B4}$$

for $y = y^{(\text{input})}$, then computing $r^{(\text{input})}$ via (B2).

Newton's method requires an initial guess $y^{(\text{initial})}$. If $r_*^{(\text{input})} > r_+$ we guess $r^{(\text{initial})} = r_*$ and use (B1) to compute $y^{(\text{initial})}$. Otherwise, we approximate the right-hand side of (B3) by its first two terms only, so that

$$y^{(\text{initial})} = \frac{r_+ - r_-}{2M}\left(\frac{r_*^{(\text{input})}}{r_+} - 1\right). \tag{B5}$$

The Newton's-method solution is moderately expensive for a computation which (logically) is needed at each grid point: it typically requires 3–10 iterations, with each iteration needing an exp() and a log() computation as well as ~10 floating-point arithmetic operations. Our code therefore precomputes and caches $r$ for each radial grid point.

### 2. Integrating Kerr geodesics

We use the Glampedakis-Kennefick formulation [58] to integrate the Kerr geodesic equations.[23] This parametrizes the radial motion as

$$r = \frac{pM}{1 + e\cos\chi}, \tag{B6}$$

where $p$ is the dimensionless semilatus rectum and $e$ the eccentricity. To solve for the particle position we numerically integrate the ODEs

$$\frac{dt}{d\chi} = \mathrm{RHS\_t}(\chi) \tag{B7a}$$

$$\frac{d\phi}{d\chi} = \mathrm{RHS\_}\phi(\chi) \tag{B7b}$$

$$\frac{d\tau}{d\chi} = \frac{(dr/d\chi)/(e\sin\chi)}{(dr/d\tau)/(e\sin\chi)} \tag{B7c}$$

using $\chi$ is the independent variable. The right-hand-side functions RHS_t and RHS_$\phi$ are given by Glampedakis and Kennefick's equations (17) and (16) respectively, while the right-hand side of (B7c) is computed using

$$\frac{dr/d\chi}{e\sin\chi} = \frac{pM}{(1 + e\cos\chi)^2} \tag{B8a}$$

$$\frac{dr/d\tau}{e\sin\chi} = \frac{1}{p}\sqrt{\tilde{V}_r(\chi)} \tag{B8b}$$

with $\tilde{V}_r(\chi)$ given by Glampedakis and Kennefick's equation (18). With this formalism the equations are nonsingular at the radial turning points, and all square roots have their principal values (i.e., there are no $\pm$ sign ambiguities). However, integrating to a specified coordinate time $t$ requires either an explicit root-finding loop around the ODE integration or using an ODE integrator with built-in root-finding capabilities.

We use the ODEPACK ODE integrator [123,124], whose DLSODAR subroutine provides ODE integration with built-in root finding. We typically set both the DLSODAR relative and absolute error tolerances to $100\varepsilon$, where $\varepsilon$ is the floating-point "machine epsilon."[24] We set the DLSODAR `MXSTEP` parameter (the maximum number of internal integration steps per DLSODAR call) to $10^5$. This allows DLSODAR to integrate a full orbit (and hence determine the orbital period) of an extreme zoom-whirl orbit like our ze98 configuration in a single call.

The ODEPACK library is written in Fortran 77, which makes its use somewhat awkward in our context. Notably, ODEPACK keeps internal state in static storage arrays and Fortran common blocks. In the context of Berger-Oliger mesh refinement it is natural to use a separate (concurrent) integration for each refinement level; in our code this requires explicitly saving and restoring the integrator state to multiplex the multiple concurrent integrations onto the single-threaded ODEPACK.

The next-generation version of ODEPACK, now known as SUNDIALS [126],[25] is written in *C* and (along with other algorithmic and computational improvements) directly supports multiple concurrent integrations. This should make it easier to use than the Fortran version.

### 3. Gradual turn-on of the effective source

Because of the jump discontinuity in the right-hand side of (2.23), the process of radiating away the initial junk generates high-spatial-frequency noise in $\varphi_{\mathrm{num},m}$ in and near to the worldtube, leading to high noise levels in the computed self-force time series. Therefore, we use a gradual turn-on of the effective source, replacing (2.23) with

$$\Box_m \varphi_{\mathrm{num},m} = \begin{cases} f(t) S_{\mathrm{effective},m} & \text{inside the worldtube} \\ 0 & \text{outside the worldtube} \end{cases}, \tag{B9}$$

where $f$ is a smooth function which is very small (ideally 0) at the initial time of an evolution and increases to asymptote to 1 at late times. We use

$$f(x) = \frac{1}{2}(1 + \mathrm{erf}(x)), \tag{B10a}$$

where the scaled time coordinate $x$ is defined by

$$x(t) = A + \frac{t - t_{\mathrm{initial}}}{B}, \tag{B10b}$$

where $t_{\mathrm{initial}}$ is the initial time of the time evolution, and $A = -5$, and $B = 10M$.[26] This gives $f(t_{\mathrm{initial}}) \approx 8 \times 10^{-13}$ (sufficiently small that the noise due to $f$ being nonzero is below our code's overall numerical noise level from other sources) and $f > 0.999999$ for $t > t_{\mathrm{initial}} + 83.6M$ [so that our evolution equation (B9) approximates (2.23) to within one part per million for all later times].

[23] Note that we differ slightly from [58] in that we use a dimensionless definition for $p$.

[24] $\varepsilon$ is the difference between 1.0 and the next larger floating-point number, approximately $1.1 \times 10^{-16}$ for IEEE-standard double-precision floating-point arithmetic [125].

[25] SUNDIALS is available at no cost from https://computation.llnl.gov/casc/sundials/main.html.

[26] Note that the expression (B10a) suffers from severe numerical cancellations for $x \ll 0$ (i.e., early in the evolution). Instead, we use the equivalent expression

$$f(t) = \begin{cases} 1 - \frac{1}{2}\mathrm{erfc}(x) & \text{if } x \geq 0 \\ \frac{1}{2}\mathrm{erfc}(-x) & \text{if } x < 0 \end{cases} \tag{B10c}$$

which is almost entirely free of numerical cancellation.

```
# overall algorithm parameters
floating_point f_move ← maximum fraction of worldtube radius that particle may be off-centered
                        (typically 0.05 to 0.1)
floating_point f_max-move ← maximum fraction of worldtube radius that worldtube may move
                            (typically 0.1)
floating_point f_ahead ← hysteresis factor when moving worldtube (typically 0.9)

# should the worldtube be moved at the current time?
# procedure arguments:
#    particle_Rstar = current particle R_* position
#    WT_center = current worldtube center R_*
#    WT_radius = worldtube radius in the R_* direction
#    previous_WT_move_time = most recent previous coordinate time that the worldtube was moved,
#                            or −∞ if the worldtube hasn't (yet) been moved in this evolution
#    t = current coordinate time
boolean procedure should_worldtube_be_moved (floating_point particle_Rstar,
                                             floating_point WT_center,
                                             floating_point WT_radius,
                                             floating_point previous_WT_move_time,
                                             floating_point t)
{
floating_point Δt_min ← minimum time interval between worldtube moves

if (t − previous_WT_move_time < Δt_min)
  then return false                          # don't move the worldtube
if (|particle_Rstar − WT_center| > f_move × WT_radius)
  then return true                           # move the worldtube
return false                                 # don't move the worldtube
}

# if the worldtube is being moved, this procedure returns the new worldtube center R_*
# procedure arguments:
#    old_WT_center = worldtube center R_* before the move
#    WT_radius = worldtube radius in the R_* direction
#    particle_Rstar = current particle R_* position
#    particle_direction = direction of the particle's 3-velocity:
#                         −1 if particle dR_*/dt < 0
#                          0 if particle dR_*/dt = 0
#                         +1 if particle dR_*/dt > 0
floating_point procedure choose_worldtube_center(floating_point old_WT_center,
                                                 floating_point WT_radius,
                                                 floating_point particle_Rstar,
                                                 floating_point particle_direction)
{
floating_point trial_offset ← particle_direction × f_ahead × f_move × WT_radius
floating_point trial_position ← particle_Rstar + trial_offset

floating_point max_move_distance ← f_max-move × WT_radius
interval I ← [old_WT_center − max_move_distance, old_WT_center + max_move_distance]
floating_point clamped_position ← (trial_position ∈ I)
                                  ? trial_position
                                  : whichever endpoint of I is closer to trial_position

return closest base-grid grid point to clamped_position
}
```

FIG. 26. This figure shows our algorithm for moving the worldtube. The procedure `should_worldtube_be_moved()` is run at each base-grid time step to determine whether or not the worldtube should be moved at the current time, and if so, the procedure `choose_worldtube_center()` determines the new worldtube position.

Using the gradual turn-on of the effective source, we find that $\varphi_{\text{num},m}$ is smooth throughout a neighborhood of the worldtube [apart from being only $C^2$ at the particle and having the jump discontinuity (2.22) across the worldtube boundary] once the gradual turn-on is complete and the field configuration has had time to adjust. In practice this initial startup phase has a duration of $\delta t_{\text{startup}} \sim 100M$ to $150M$.

### 4. Moving the worldtube

Figure 26 gives our worldtube-moving algorithm in detail. The algorithm is run at each base-grid time step, and has two parts: determining whether or not the worldtube should be moved at the current time and, if it should be moved, determining the new worldtube position. Table VII gives the parameters for this algorithm (among others). In practice,

we find that our computed results are quite insensitive to the precise values of these parameters (cf. Sec. III D).

### 5. Constraints on moving the worldtube early in the evolution

When moving the worldtube, the grid-function adjustments (2.24) implicitly assume that $\varphi_{\rm num}$ has the jump discontinuity (2.22) across the worldtube boundary. While this is true once the field is in its equilibrium configuration, it is *not* true for our initial data ($\varphi_{{\rm num},m} = \Pi_{{\rm num},m} = 0$, cf. Sec. II J). When the evolution begins, it takes some time (in practice ∼100 to 150$M$) for the gradual turn-on of the effective source (Sec. B 3) to be essentially complete and for the field to relax to an equilibrium configuration where the worldtube-boundary jump condition (2.22) is satisfied.

During this initial "startup" phase of the computation we do not know the actual jump conditions satisfied by $\varphi_{\rm num}$, so the worldtube cannot be moved. This in turn means that the initial worldtube must encompass the entire range of motion of the particle in $(R_*, \theta)$ during the startup phase. We use the following strategy to ensure this (for a equatorial geodesic or near-geodesic particle orbit) without requiring an excessively large worldtube:

(i) We first choose a particle apoastron time $t_{\rm apoastron}$. Notice that the particle position $R_* = R_*(t)$ is locally symmetric about an (any) apoastron time.

(ii) We then choose the startup time interval to be symmetric about $t_{\rm apoastron}$. That is, given an estimate for the startup time interval's duration $\delta t_{\rm startup}$ (typically 100$M$ to 150$M$), we begin the numerical evolution at $t = t_{\rm initial} = t_{\rm apoastron} - \frac{1}{2}\delta t_{\rm startup}$, so that the startup phase lasts until $t = t_{\rm startup\to main} = t_{\rm apoastron} + \frac{1}{2}\delta t_{\rm startup}$. During the startup time interval the particle first moves outwards, then moves back inwards, reaching its initial radius again at $t_{\rm startup\to main}$.

(iii) We initially center the worldtube in $(R_*, \theta)$ at (the base-grid point nearest) the average of $x^i_{\rm particle}(t_{\rm initial})$ and $x^i_{\rm particle}(t_{\rm apoastron})$, and choose the worldtube-moving parameters so that the worldtube will not be moved during the startup phase.

(iv) At the end of the startup phase at $t_{\rm startup\to main}$ (when the particle returns to its initial position, now moving inwards), we change the worldtube-moving parameters to values which keep the worldtube's coordinate center within approximately half a coarse-grid spacing of the particle for the remainder of the evolution. The first worldtube move generally occurs immediately after the new parameters take effect.

### 6. Finite differencing across the worldtube boundary

We numerically implement the jump condition (2.22) on the worldtube boundary in the same manner as Barack and Golbourn [49] and Dolan and Barack [73]. That is, suppose we are finite differencing the equations at an "evaluation" grid point which is inside (outside) the worldtube, using a finite-difference molecule which has a nonempty set $S$ of input grid points which are outside (inside) the worldtube. Then instead of applying the finite-difference molecule to the $\varphi_{{\rm num},m}$ grid function in the usual manner, we instead copy $\varphi_{{\rm num},m}$ at all the molecule input points to a (molecule-sized) temporary grid function $\varphi^{\rm temp}_{{\rm num},m}$, then adjust the values of that temporary grid function to have the same inside/outside-the-worldtube semantics as the evaluation point via

$$\varphi^{\rm temp}_{{\rm num},m} \leftarrow \varphi^{\rm temp}_{{\rm num},m} \mp \varphi_{{\rm puncture},m} \tag{B11}$$

at each grid point in the set $S$, then finally apply the usual finite-difference molecule to the adjusted values.

Notice that this "adjusted finite differencing" need only be used for (roughly) those grid points which are within a finite-difference molecule radius of the worldtube boundary. (We discuss the precise choice of those grid points in the following section.) Because these comprise only a tiny fraction of all grid points, the adjusted finite differencing does not itself significantly slow the code. Rather, its main computational cost is the test—at each spatial grid point at each time the evolution equations are evaluated by the time integrator[27]—for whether or not adjusted finite differencing should be used. As discussed further in the following section, this test costs only ∼10 arithmetic and logical operations, which is easily tolerable.

### 7. Computing the set of grid points where adjusted finite differencing is needed

In developing our numerical code we found that it was (is) much more difficult than might be expected to compute the precise set of grid points where adjusted finite differencing should be done. As noted in the previous section, this is approximately the set of all grid points within a finite-difference molecule radius of the worldtube boundary. However, in the presence of equatorial-reflection symmetry (cf. Sec. II I 3) this set is not quite correct: there are certain grid points near the intersection of the worldtube boundary with the $\theta = \pi/2$ equatorial-reflection symmetry plane which *are* within a finite-difference molecule radius of the worldtube boundary, but where adjusted finite differencing should (must) *not* be used. Figure 27 shows an example of this.

The technique we eventually adopted involves two parts:

(1) We build up the "should this finite-difference operator be adjusted via (B11) at this grid point?" predicate in stages via Boolean and set operations on sets of grid points. Figure 28 shows the resulting algorithm. With this approach, the semantics of each individual function are very clear, which allowed us

[27]This evaluation typically happens several times per time step; we discuss our time-evolution algorithms in detail in Sec. B 9.

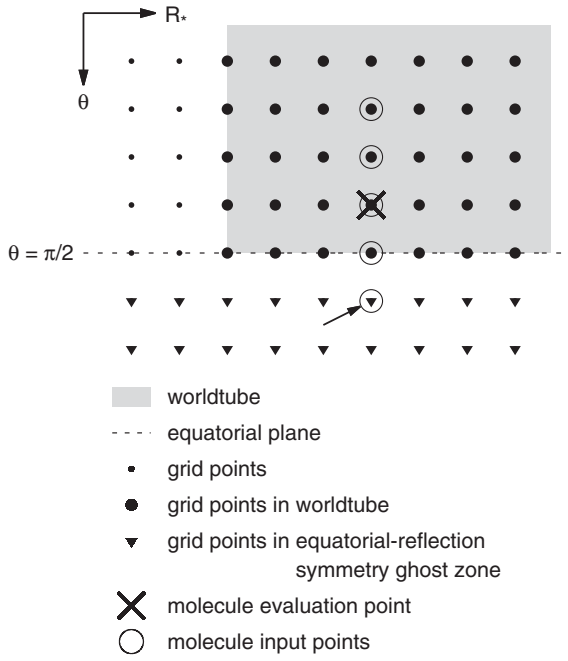


FIG. 27. This figure shows an example where the naive algorithm "use adjusted finite differencing at all grid points within a finite-difference molecule radius of the worldtube boundary" would give incorrect results. The worldtube is shown by the shaded region; the worldtube boundary of interest is the "equator" $\theta = \pi/2$ (shown by the dashed line). The molecules being considered are 5-point centered molecules in the $\theta$ direction, as would be used to approximate $\partial_\theta$ or $\partial_{\theta\theta}$; these molecules have radius 2 in the $\pm\theta$ directions. The molecule evaluation point shown as $\times$ is only 1 grid point away from the worldtube boundary, so the naive algorithm would say that adjusted finite differencing should be used for this molecule. However, the arrowed point is actually within the equatorial-reflection symmetry ghost zone's "reflection" of the worldtube, so in terms of the adjustment (B11) this point has *inside*-the-worldtube semantics, and hence adjusted finite differencing should *not* be used for this molecule.

to develop an extensive test suite to help validate the algorithm.

(2) When using equatorial-reflection symmetry, we use a numerical grid which spans only the "northern hemisphere" $0 \le \theta \le \pi/2$ radians, *but* we still consider the worldtube to be the full region that it would have occupied in the absence of equatorial-reflection symmetry, i.e., (assuming an equatorial particle orbit) we take the worldtube to be symmetric about the equatorial plane.

### 8. Computing the set of grid points where the puncture field is needed

Given that the adjustment (B11) is to be applied, there remains the problem of computing $\varphi_{\text{puncture},m}$ at each finite-difference molecule input point.

In our evolution scheme there are (for 5-point centered molecules) typically 9 molecule input points per evaluation point,[28] so there is a significant performance boost from computing $\varphi_{\text{puncture},m}$ only once at each grid point where it is needed, rather than 9 times if it were (re)computed each time it is used at a molecule input point.

[Notice that—even apart from any performance cost— we cannot simply compute $\varphi_{\text{puncture},m}$ at *all* spatial grid points (at each evaluation time), because (a) $\varphi_{\text{puncture},m}$ diverges at the particle, and (b) our series expansions for $\varphi_{\text{puncture},m}$ may be ill-behaved (e.g., they may involve division by zero) sufficiently far from the particle (outside the worldtube).]

There are two plausible ways of ensuring that $\varphi_{\text{puncture},m}$ is computed at the desired set of grid points without trying to compute it at any point where the computation would blow up:

(i) $\varphi_{\text{puncture},m}$ could be stored as a "smart grid function." comprising a standard grid function of complex numbers ($\varphi_{\text{puncture},m}$ values) together with an auxiliary grid function of Boolean "valid" flags recording whether or not $\varphi_{\text{puncture},m}$ has already been computed at the corresponding grid point at the current time. On each access to the grid function, the Boolean flag would be checked, and if $\varphi_{\text{puncture},m}$ had not already been computed at that grid point at the current time, it would be computed, stored (cached) in the corresponding grid function, and the corresponding Boolean flag would be set to record that this grid-function value was now valid, so that future access could use the cached value.

(ii) Alternatively, we could use a standard complex grid function to store $\varphi_{\text{puncture},m}$ and *precompute* [i.e., compute before starting to compute the adjustment (B11)] $\varphi_{\text{puncture},m}$ at all the grid points where it will be needed, storing it in the grid function. (At grid points where $\varphi_{\text{puncture},m}$ will not be needed, the grid function can either be left uninitialized or be set to dummy values—these will not affect the result of any finite-differencing operation.) The adjustment (B11) can then use the stored $\varphi_{\text{puncture},m}$ values with no further validity checking needed.

We have chosen the second option as likely being simpler and more efficient.

The precomputation algorithm does not actually require an exact computation of the "is $\varphi_{\text{puncture},m}$ needed at this grid point?" predicate: no harm is done if we precompute $\varphi_{\text{puncture},m}$ at some points where it will not actually be used, so the predicate need only return true at a (possibly proper) *superset* of the actual set of grid points where $\varphi_{\text{puncture},m}$ is needed. This suggests that the naive algorithm of precomputing $\varphi_{\text{puncture},m}$ at every grid point that is within

[28]There are no $\partial_{R_*\theta}$ terms in our evolution equations; if there were, then (again assuming 5-point centered molecules) there would be 25 molecule input points per evaluation point.

```
1  # does adjusted finite differencing need to be used for a radial molecule
2  # (with radia (R_i-, R_i+)) evaluated at (i,j)?
3  boolean procedure adjusted_FD_used_for_radial_molecule (integer R_i-, integer R_i+,
4                                                          integer i, integer j)
5  {
6  region W ← worldtube region
7  return ((i, j) ∈ W) ? radial_molecule_has_input_outside_region (W, R_i-, R_i+, i, j)
8                      : radial_molecule_has_input_in_region       (W, R_i-, R_i+, i, j)
9  }
10
11 # does a radial molecule (with radia (R_i-, R_i+)) evaluated at (i, j)
12 # have any input points which are outside the region input_R?
13 boolean procedure radial_molecule_has_input_outside_region(region input_R,
14                                                            integer R_i-, integer R_i+,
15                                                            integer i, integer j)
16 {
17 return not radial_molecule_has_all_inputs_in_region(input_R, R_i-, R_i+, i, j)
18 }
19
20 # does a radial molecule (with radia (R_i-, R_i+)) evaluated at (i, j)
21 # have all of its input points in the region input_R?
22 boolean procedure radial_molecule_has_all_inputs_in_region(region input_R,
23                                                            integer R_i-, integer R_i+,
24                                                            integer i, integer j)
25 {
26 region molecule_input_region ← [i − R_i-, i + R_i+] × {j}
27 return molecule_input_region ⊆ input_R
28 }
29
30 # does a radial molecule (with radia (R_i-, R_i+)) evaluated at (i, j)
31 # have any input points in the region input_R?
32 boolean procedure radial_molecule_has_input_in_region(region input_R,
33                                                       integer R_i-, integer R_i+,
34                                                       integer i, integer j)
35 {
36 region molecule_input_region ← [i − R_i-, i + R_i+] × {j}
37 return molecule_input_region ∩ input_R ≠ ∅
38 }
```

FIG. 28. This figure shows our algorithm for computing the "should this finite-difference operator be adjusted via (B11) at this grid point?" predicate. Only the procedures for radial finite-difference molecules are shown; those for angular molecules are analogous.

a finite-difference molecule radius of the worldtube boundary might well yield correct results.

However, for consistency and to maximize our confidence that equatorial-reflection symmetry cases like the one shown in Fig. 27 are handled correctly, we choose instead to build up the "is $\varphi_{\text{puncture},m}$ needed at this grid point?" predicate in stages using Boolean and set operations on sets of grid points, in a manner very similar to our construction of the "should adjusted finite differencing be used at this grid point?" algorithm (Fig. 28). Figure 29 shows the resulting algorithm for determining where $\varphi_{\text{puncture},m}$ is needed. Like the adjusted-finite-differencing algorithm, this algorithm has very clear semantics for each individual function, which allowed us to develop an extensive test suite to help validate the algorithm.

### 9. Numerical time evolution

We numerically solve the evolution system (2.39) and (2.40) using the method of lines, with locally uniform spatial grids in $(R_*, \theta)$. We discretize all spatial derivatives using (5-point) 4th-order centered finite differencing, except that within a few grid points of the particle we use the "C2" finite-differencing scheme described in Appendix B 10. For all results reported here, we use the classical 4th-order Runge-Kutta method for the time evolution.

We use Berger-Oliger mesh refinement ([127–130]) with a 2:1 refinement ratio, full subcycling in time, and buffer zones [109]. We use 5th-order (6-point) Lagrange polynomial interpolation in space and time for the coarse-to-fine Berger-Oliger interpolations. (This requires keeping 6 time levels for all but the finest refinement level; the latter needs only a single time level.) For the results reported here we use 4 refinement levels with the finest 3 refinement levels moved to follow the worldtube (Sec. II D and Table VI). In the terminology of Berger-Oliger mesh refinement our grid placement is "nonadaptive," in that it does not depend on the values of the field variables.

While our evolution scheme is stable on moderate time scales, we find that long-time evolutions can be made much less noisy by adding 6th-order Kreiss-Oliger dissipation in the form

```
1  # does the puncture function need to be computed at the position (i,j)?
2  boolean procedure puncture_fn_used_at_ij(integer i, integer j)
3  {
4  integer R^max_i− ← maximum radius of any molecule in the i− direction
5  integer R^max_i+ ← maximum radius of any molecule in the i+ direction
6  integer R^max_j− ← maximum radius of any molecule in the j− direction
7  integer R^max_j+ ← maximum radius of any molecule in the j+ direction
8  return     puncture_fn_used_by_radial_molecule  (R^max_i−, R^max_i+, i,j)
9         or puncture_fn_used_by_angular_molecule(R^max_j−, R^max_j+, i,j)
10 }
11
12 # is the puncture function used by a radial molecule
13 # (with radia (R_i−, R_i+)) with the input point (i,j)?
14 boolean procedure puncture_fn_used_by_radial_molecule(integer R_i−, integer R_i+,
15                                                       integer i, integer j)
16 {
17 region W ← worldtube region
18 return ((i, j) ∈ W) ? radial_molecule_outside_region_has_input_ij(W, R_i−, R_i+, i, j)
19                     : radial_molecule_in_region_has_input_ij      (W, R_i−, R_i+, i, j)
20 }
21
22 # does any radial molecule (with radia (R_i−, R_i+)) evaluated at a point
23 # that's outside the region eval_R have the input point (i, j)?
24 boolean procedure radial_molecule_outside_region_has_input_ij(region eval_R,
25                                                               integer R_i−, integer R_i+,
26                                                               integer i, integer j)
27 {
28 # region of evaluation points where a molecule would have the input point (i, j)
29 region eval_region_with_input_ij ← [i − R_i+, i + R_i−] × {j}
30
31 boolean all_eval_points_in_eval_R ← eval_region_with_input_ij ⊆ eval_R
32 return not all_eval_points_in_eval_R
33 }
34
35 # does any radial molecule (with radia (R_i−, R_i+)) evaluated at a point
36 # in the region eval_R have the input point (i, j)?
37 boolean procedure radial_molecule_in_region_has_input_ij(region eval_R,
38                                                          integer R_i−, integer R_i+,
39                                                          integer i, integer j)
40 {
41 # region of evaluation points where a molecule would have the input point (i, j)
42 region eval_region_with_input_ij ← [i − R_i+, i + R_i−] × {j}
43
44 return eval_region_with_input_ij ∩ eval_R ≠ ∅
45 }
```

FIG. 29. This figure shows our algorithm for computing the "is $\varphi_{\text{puncture},m}$ needed at this grid point?" predicate. Apart from `puncture_fn_used_at_ij()`, only the procedures for radial finite-difference molecules are shown; those for angular molecules are analogous.

$$\partial_t \varphi_{\text{num},m} \to \partial_t \varphi_{\text{num},m} + \varepsilon(\mathsf{D}_{R_*}(\varphi_{\text{num},m}) + \mathsf{D}_\theta(\varphi_{\text{num},m})), \tag{B12a}$$

$$\partial_t \Pi_{\text{num},m} \to \partial_t \Pi_{\text{num},m} + \varepsilon(\mathsf{D}_{R_*}(\Pi_{\text{num},m}) + \mathsf{D}_\theta(\Pi_{\text{num},m})), \tag{B12b}$$

where

$$\begin{aligned}(\mathsf{D}(g))_i &= \frac{1}{64}(\Delta x)^5 (D_+^3 D_-^3 g)_i \\ &= \frac{1}{64\Delta x}(g_{i-3} - 6g_{i-2} + 15g_{i-1} - 20g_i + 15g_{i+1} \\ &\quad - 6g_{i+2} + g_{i+3}).\end{aligned} \tag{B13}$$

To obtain stable evolutions we found it crucial to add dissipation *only* at those grid points where the following 3 conditions are satisfied:

(i) The dissipation molecule does not cross the particle, i.e., the closest grid point to the particle is not one of the points $i-3$ through $i+3$ inclusive in the expression (B13).

(ii) The dissipation molecule does not cross the worldtube boundary, i.e., it does not have input points both inside and outside the worldtube.

(iii) The dissipation molecule does not have any input points outside the union of the nominal grid and any symmetry ghost zones. In practice this prevents dissipation from being added close to mesh-refinement

TABLE IX. For each $m \in [0, 20]$, this table shows the largest Courant number $\nu = \Delta t/\Delta R_*$ for which we obtain a stable evolution. For these stability tests we use a dissipation coefficient of $\varepsilon = 0.01$ and a 2-refinement-level grid with base resolution $\Delta R_* = M/4$; the stability limit depends only weakly on these parameters.

| $m$ | $\nu_{\max}$ |
|---|---|
| 0 | 0.63 |
| 1 | 0.65 |
| 2 | 0.59 |
| 3 | 0.48 |
| 4 | 0.39 |
| 5 | 0.33 |
| 6 | 0.283 |
| 7 | 0.246 |
| 8 | 0.217 |
| 9 | 0.195 |
| 10 | 0.176 |
| 11 | 0.160 |
| 12 | 0.147 |
| 13 | 0.136 |
| 14 | 0.127 |
| 15 | 0.118 |
| 16 | 0.111 |
| 17 | 0.105 |
| 18 | 0.099 |
| 19 | 0.094 |
| 20 | 0.089 |

boundaries or close to the horizon or $\mathcal{J}^+$ grid boundaries.

We use $\varepsilon = 0.1$ for the evolutions reported here.

Table IX shows the empirically determined Courant-Friedrichs-Lewy (CFL) stability limit $\nu_{\max}$ [131] of our evolution scheme as a function of $m$. Our code chooses the base-grid time step $\Delta t$ by first computing $\Delta t^{(\text{preliminary})} = \kappa \nu_{\max} \Delta R_*$ (where $\kappa = 0.9$ is a safety-factor parameter), then choosing $\Delta t$ to be the largest value $\le \Delta t^{(\text{preliminary})}$ which integrally divides the output sampling time. The time steps for all refined grids are defined by the Berger-Oliger mesh-refinement scheme.

It is clear from Table IX that at large $m$ our evolution scheme has a very restrictive CFL stability limit (small $\nu_{\max}$ and hence small $\Delta t$), making the evolution quite inefficient. As discussed in Appendix B 11, we have experimented with an IMEX time-evolution scheme in an attempt to alleviate the large-$m$ CFL restriction, but thus far these experiments have not yielded larger stable Courant numbers. This remains a topic for further research.

### 10. Finite differencing near the particle

Because $\varphi_m$ is only $C^2$ at the particle, standard finite-difference molecules do not attain their full order of accuracy near (within roughly a molecule radius of) the particle. One way to view this problem is to conceptualize a finite-difference molecule as being derived by fitting a local (sliding-window) Lagrange interpolating polynomial to the operand grid function, then differentiating that interpolating polynomial. This suggests that one way to obtain more accurate finite differencing near the particle might be to use a more general interpolating function that better represents the actual behavior of $\varphi_m$ near the particle.

To this end, without loss of generality, we consider the finite differencing of a (real or complex) function $g$ which is defined on a suitable neighborhood of the origin on the real line, using a 1-dimensional numerical grid with grid points at integer coordinates. Without loss of generality, we assume the particle to be at the (known) position $p \in [0, \frac{1}{2}]$. We consider the piecewise-polynomial interpolating function

$$I(x) = a_0 + a_1(x-p) + a_2(x-p)^2 + \begin{cases} b_3(x-p)^3 + b_4(x-p)^4 + b_5(x-p)^5 & \text{if } x \le 0 \\ c_3(x-p)^3 + c_4(x-p)^4 + c_5(x-p)^5 & \text{if } x > 0. \end{cases} \tag{B14}$$

The 9 coefficients $\{a_0, a_1, a_2, b_3, b_4, b_5, c_3, c_4, c_5\}$ can be uniquely determined (as functions of the parameter $p$) by requiring $I$ to match the specified function $g$ at the 9 adjacent grid points in the range $-4 \le x \le 4$. $I$, $dI/dx$, and $d^2I/dx^2$ can then be evaluated at any desired position to obtain finite-difference approximations to $g$, $dg/dx$, and $d^2g/dx^2$ respectively. Using a symbolic-algebra system, these finite-difference operators can be written as linear combinations of the values of $g$ at the grid points, with coefficients depending only on $p$ and the evaluation position.

Figure 30 shows how we use these "C2" finite-difference operators at various grid points near the particle. In the present work the particle is always in the background Kerr spacetime's equatorial plane, and we always place a $\theta =$ constant row of grid points on the equatorial plane. Considering the numerical grid in 2 dimensions $(R_*, \theta)$, with corresponding integer grid coordinates $(\mathsf{i}, \mathsf{j})$, suppose that the closest grid point to the particle is at $(\mathsf{i}_{\text{particle}}, \mathsf{j}_{\text{equator}})$. Then we use the C2 scheme for $\partial_{R_*}$ and $\partial_{R_*R_*}$ derivatives evaluated at grid points on the equator (i.e., for grid points with $\mathsf{j} = \mathsf{j}_{\text{equator}}$) and $\mathsf{i}$ near $\mathsf{i}_{\text{particle}}$, in the manner shown in the figure. We also use this scheme in the $\mathsf{j}$ direction for $\partial_\theta$ and $\partial_{\theta\theta}$ derivatives evaluated at grid points with $\mathsf{i} = \mathsf{i}_{\text{particle}}$ and $\mathsf{j}$ near the equator. We use standard (5-point) centered 4th-order molecules at all other grid points.

As discussed in Sec. II K, we also use the interpolating function $I$ directly in computing the self-force.

Overall, we find that switching from using 4th-order centered spatial finite differencing everywhere to using the C2 finite-difference operators near the particle reduces the noise level in the computed self-force by about a factor of 2 to 3.

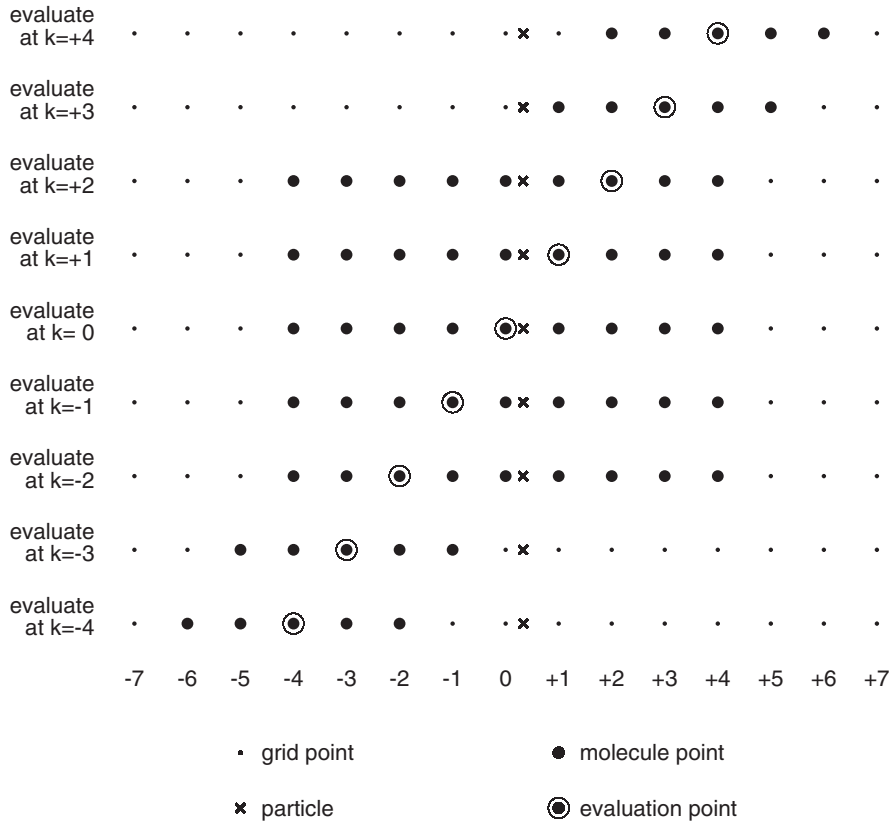


FIG. 30. This figure shows our "C2" spatial finite-differencing scheme for use near the particle. Each row of the diagram shows the finite-difference molecule used for a different evaluation point. k refers to the integer grid coordinate in the $x$ direction, with the origin set so that $\mathsf{k} = 0$ is the grid point closest to the particle. We use molecules based on the piecewise-polynomial interpolating function (B14) for evaluations points $-2 \le \mathsf{k} \le +2$ and standard (5-point) centered 4th-order molecules at all other evaluation points.

We also experimented with a more general interpolating function

$$I(x) = a_0 + a_1(x-p) + a_2(x-p)^2 + \begin{cases} b_3(x-p)^3 + b_{3\ell}(x-p)^3\log(K(x-p)^2) \\ +b_4(x-p)^4 + b_{4\ell}(x-p)^4\log(K(x-p)^2) \\ +b_5(x-p)^5 + b_{5\ell}(x-p)^5\log(K(x-p)^2) & \text{if } x \le 0 \\ c_3(x-p)^3 + c_{3\ell}(x-p)^3\log(K(x-p)^2) \\ +c_4(x-p)^4 + c_{4\ell}(x-p)^4\log(K(x-p)^2) \\ +c_5(x-p)^5 + c_{5\ell}(x-p)^5\log(K(x-p)^2) & \text{if } x > 0, \end{cases} \tag{B15}$$

where $K$ is a scaling constant and the 15 coefficients $\{a_i, b_i, b_{i\ell}, c_i, c_{i\ell}\}$ are determined by solving a system of 15 linear equations using the values of $g$ at 15 adjacent grid points. (A purely symbolic solution to this linear system proved impractical, but it is easy to solve numerically. This needs to be done once for each choice of the grid spacing and $K$.) However, we were not able to obtain stable evolutions with this scheme.

### 11. IMEX evolution schemes

As discussed in Appendix B 9, when using an explicit (Runge-Kutta) time-evolution scheme we find that the CFL stability limit [131] is very restrictive for large $m$, with the largest stable Courant number being approximately proportional to $1/m$. Examination of the derivative structure of our evolution system, together with an eigenvalue analysis kindly performed by I. Hinder, suggests that $m^2$ coefficient in the $\varphi$ term in the scalar wave operator (2.20) may be a major contributor to the large-$m$ time-step restriction.

We thus consider the use of an implicit time-evolution scheme. More precisely, we consider the use of an IMEX time-evolution scheme. There is a large literature on these schemes; see, for example, [132–138]. The basic concept of an IMEX scheme is to partition the right-hand-side function into explicit and implicit parts,

$$\dot{\mathsf{u}} = \mathsf{F}(\mathsf{u}, t) + \mathsf{G}(\mathsf{u}, t), \tag{B16}$$

where $\mathsf{u}$ is the state vector, then treat $\mathsf{F}$ explicitly and $\mathsf{G}$ implicitly. For our application, we will place all the spatial derivatives into the explicit term, thus avoiding having to solve an elliptic system at each time step.

We have chosen the scheme proposed by Boscarino [137] [in particular, his BHR(5,5,3) scheme, variant 2] as being efficient, relatively easy to implement, and having good accuracy (3rd-order overall) without the "order reduction" problems of many other schemes.[29]

We write a generic implicit-explicit Runge-Kutta scheme for the ODE (B16) as[30]

$$\mathsf{u}^{(n+1)} = \mathsf{u}^{(n)} + h\sum_{i=1}^{s} b_i \mathsf{k}_i + h\sum_{i=1}^{\tilde{s}} \tilde{b}_i \tilde{\mathsf{k}}_i, \tag{B18}$$

where $h$ is the time step, superscripts $(n)$ and $(n+1)$ refer to time levels, subscripts refer to Runge-Kutta stages numbered $1, \ldots, s$, and the Runge-Kutta stages are given by

$$\mathsf{k}_i = F(\mathsf{u}_i, t^{(n)} + hc_i), \tag{B19a}$$

$$\tilde{\mathsf{k}}_i = G(\mathsf{u}_i, t^{(n)} + hc_i), \tag{B19b}$$

$$\mathsf{u}_i = \mathsf{u}^{(n)} + h\sum_{j<i} d_{ij}\mathsf{k}_j + h\sum_{j\le i} \tilde{d}_{ij}\tilde{\mathsf{k}}_j, \tag{B19c}$$

with the coefficients $\{b_i\}$, $\{\tilde{b}_i\}$, $\{c_i\}$, $\{d_{ij}\}$, and $\{\tilde{d}_{ij}\}$.

For example (eliding the evaluation times for clarity), the first few stages are

$$\mathsf{u}_1 = \mathsf{u}^{(n)} + h\tilde{d}_{11}\mathsf{G}(\mathsf{u}_1), \tag{B20a}$$

$$\mathsf{u}_2 = \mathsf{u}^{(n)} + hd_{21}\mathsf{F}(\mathsf{u}_1) + h\tilde{d}_{21}\mathsf{G}(\mathsf{u}_1) + h\tilde{d}_{22}\mathsf{G}(\mathsf{u}_2), \tag{B20b}$$

$$\mathsf{u}_3 = \mathsf{u}^{(n)} + hd_{31}\mathsf{F}(\mathsf{u}_1) + hd_{32}\mathsf{F}(\mathsf{u}_2) + h\tilde{d}_{31}\mathsf{G}(\mathsf{u}_1) + h\tilde{d}_{32}\mathsf{G}(\mathsf{u}_2) + h\tilde{d}_{33}\mathsf{G}(\mathsf{u}_3). \tag{B20c}$$

To solve the implicit equations (B19), we observe that our state vector $\mathsf{u}$ is of the form

$$\mathsf{u} = \begin{pmatrix} \varphi \\ \Pi \end{pmatrix}, \tag{B21}$$

so we can write

$$\mathsf{F}\begin{pmatrix} \varphi \\ \Pi \end{pmatrix} = \begin{pmatrix} y(\varphi, \Pi) \\ z(\varphi, \Pi) \end{pmatrix} \tag{B22}$$

and

$$\mathsf{G}\begin{pmatrix} \varphi \\ \Pi \end{pmatrix} = \begin{pmatrix} 0 \\ \tilde{z}(\varphi, \Pi) \end{pmatrix} = \begin{pmatrix} 0 \\ \alpha\varphi + \beta\Pi \end{pmatrix} \tag{B23}$$

with known coefficients $\alpha$ and $\beta$. (This use of $\alpha$ is unrelated to its use as a tail-series exponent in Sec. II L.)

The 2-component $\mathsf{F}$ function (B22) includes all the main evolution equations (2.39), (2.40), and (B12), as well as all spatial boundary conditions. Evaluating $F$ requires computing (or retrieving from a cache) the 2D puncture field and effective source.[31]

We have considered a number of possible choices for precisely which terms from the main evolution equation (2.40) should be treated implicitly (i.e., put into $G$). Conceptually, we have $\alpha = \alpha_1 + \alpha_2 + \alpha_3$, where

$$\alpha_1 = 0 \quad \text{or} \quad -\frac{2\Delta}{r(r^2 + M^2\tilde{a}^2)^2}\left(M - \frac{M^2\tilde{a}^2}{r}\right), \tag{B24a}$$

$$\alpha_2 = 0 \quad \text{or} \quad -\frac{m^2\Delta}{(r^2 + M^2\tilde{a}^2)^2\sin^2\theta}, \tag{B24b}$$

$$\alpha_3 = 0 \quad \text{or} \quad -i\frac{2mM\tilde{a}\Delta}{r(r^2 + M^2\tilde{a}^2)^2}, \tag{B24c}$$

and

$$\beta = 0 \quad \text{or} \quad -i\frac{4mM^2\tilde{a}r}{(r^2 + M^2\tilde{a}^2)^2} \tag{B24d}$$

[29] We warn the reader of the following typographical errors in [137]:

(i) Equation (4) should read

$$U^i = U_n + h\sum_{j=1}^{i-1} \tilde{a}_{ij} F(t_n + \tilde{c}_j h, U^j) + h\sum_{j=1}^{i} a_{ij}\frac{1}{\varepsilon} G(t_n + c_j h, U^j) \tag{B17}$$

(ii) In the Appendix, in the left (explicit) Butcher tableau, the $b$ coefficients should read $[b_1 \quad 0 \quad b_3 \quad b_4 \quad \gamma]$.

(iii) In the Appendix, in the right (implicit) Butcher tableau, the coefficients for the last stage (the 5th row of the matrix) should be identical to the $b$ coefficients, i.e., they should read $[b_1 \quad 0 \quad b_3 \quad b_4 \quad \gamma]$.

[30] Our notation in the remainder of this appendix is somewhat different from Boscarino's; notably, we swap the tilde and nontilde coefficients.

[31] Unfortunately, in all IMEX schemes of which we are aware it is *not* the case that there are repeated evaluations of $\mathsf{F}$ with different state vectors at the same time coordinate, so there is no reuse possible of the puncture field and effective source from one evaluation to the next. In contrast (as noted in Sec. IV B 2), with the classical RK4 scheme 50% of evaluations are repeated in this way, so—since the effective-source computation dominates the code's overall running time—there is an easy factor-of-two saving in computational cost by caching and reusing the effective source from one evaluation to the next if the evaluation time is unchanged.

modified by the compactification transformation (2.35), together with the spatial boundary conditions. This gives 16 possible variant schemes, depending on which subset of $\{\alpha_1, \alpha_2, \alpha_3, \beta\}$ is nonzero (treated implicitly). For each of these variants,

(i) $\mathsf{G}$ is linear in $\varphi$ and $\Pi$ at each grid point,
(ii) $\mathsf{G}$ may be evaluated independently at each grid point, and
(iii) this evaluation does *not* require computing the 2D puncture field or effective source.

Together, these properties make the scheme efficient and relatively easy to implement.

Substituting the 2-component $\mathsf{u}$, $\mathsf{F}$, and $\mathsf{G}$ functions (B21), (B22), and (B23) into the implicit Runge-Kutta equations (B19), we have

$$\mathsf{u}_i = \begin{pmatrix} \varphi_i \\ \Pi_i \end{pmatrix} = \begin{pmatrix} \varphi^{(n)} \\ \Pi^{(n)} \end{pmatrix} + h\sum_{j<i} d_{ij} \begin{pmatrix} y_j \\ z_j \end{pmatrix} + h\sum_{j<i} \tilde{d}_{ij} \begin{pmatrix} 0 \\ \tilde{z}_j \end{pmatrix} + h\tilde{d}_{ii} \begin{pmatrix} 0 \\ \alpha\varphi_i + \beta\Pi_i \end{pmatrix}. \tag{B25}$$

We solve this equation at each Runge-Kutta stage by first computing

$$\varphi_i = \varphi^{(n)} + h\sum_{j<i} d_{ij} y_j \tag{B26a}$$

and then computing

$$\Pi_i = \frac{\Pi^{(n)} + h\sum_{j<i} d_{ij} z_j + h\sum_{j<i} \tilde{d}_{ij} \tilde{z}_j + h\tilde{d}_{ii}\alpha\varphi_i}{1 - h\tilde{d}_{ii}\beta}. \tag{B26b}$$

We have implemented these 16 variant schemes, but unfortunately we find that all of them have CFL stability limits which are (to within the ∼1% accuracy of our trial-and-error estimation of the stability limit on test problems) identical to those of the classical RK4 scheme (Table IX). Since the RK4 scheme is simpler and offers a factor-of-two overall speedup by caching and reusing the effective source at repeated evaluation times, we use it for all the computations presented in this paper. We hope to further investigate different partitionings of the right-hand-side function between $\mathsf{F}$ and $\mathsf{G}$ in the future in the hopes of alleviating the large-$m$ time-step restriction.


[1] J. R. Gair, L. Barack, T. Creighton, C. Cutler, S. L. Larson, E. S. Phinney, and M. Vallisneri, Classical Quantum Gravity **21**, S1595 (2004).
[2] L. Barack and C. Cutler, Phys. Rev. D **69**, 082005 (2004).
[3] P. Amaro-Seoane, J. R. Gair, M. Freitag, M. C. Miller, I. Mandel, C. J. Cutler, and S. Babak, Classical Quantum Gravity **24**, R113 (2007).
[4] J. R. Gair, Classical Quantum Gravity **26**, 094034 (2009).
[5] T. Damour, in *Three Hundred Years of Gravitation*, edited by S. W. Hawking and W. Israel (Cambridge University Press, Cambridge, England, 1987) Chap. 6, pp. 128–198.
[6] L. Blanchet, Living Rev. Relativ. **17**, 2 (2014).
[7] T. Futamase and Y. Itoh, Living Rev. Relativ. **10**, 2 (2007).
[8] L. Blanchet, in *Mass and Motion in General Relativity*, Fundamental Theories of Physics, Vol. 162, edited by L. Blanchet, A. Spallicci, and B. F. Whiting (Springer-Verlag, Berlin, 2011), p. 125.
[9] G. Schäfer, in *Mass and Motion in General Relativity*, Fundamental Theories of Physics, Vol. 162, edited by L. Blanchet, A. Spallicci, and B. F. Whiting (Springer-Verlag, Berlin, 2011), p. 167.
[10] F. Pretorius, in *Relativistic Objects in Compact Binaries: From Birth to Coalescence*, edited by M. Colpi (Springer-Verlag, Berlin, 2007).
[11] M. Hannam, S. Husa, J. G. Baker, M. Boyle, B. Brügmann, T. Chu, N. Dorband, F. Herrmann, I. Hinder, B. J. Kelly, L. E. Kidder, P. Laguna, K. D. Matthews, J. R. van Meter, H. P. Pfeiffer, D. Pollney, C. Reisswig, M. A. Scheel, and D. Shoemaker, Phys. Rev. D **79**, 084025 (2009).
[12] M. Hannam, Classical Quantum Gravity **26**, 114001 (2009).
[13] M. Hannam and I. Hawke, Gen. Relativ. Gravit. **43**, 465 (2011).
[14] M. Campanelli, C. O. Lousto, B. C. Mundim, H. Nakano, Y. Zlochower, and H.-P. Bischof, Classical Quantum Gravity **27**, 084034 (2010).
[15] N. T. Bishop, R. Gómez, S. Husa, L. Lehner, and J. Winicour, Phys. Rev. D **68**, 084015 (2003).
[16] N. T. Bishop, R. Gómez, L. Lehner, M. Maharaj, and J. Winicour, Phys. Rev. D **72**, 024002 (2005).
[17] C. F. Sopuerta, P. Sun, P. Laguna, and J. Xu, Classical Quantum Gravity **23**, 251 (2006).
[18] C. F. Sopuerta and P. Laguna, Phys. Rev. D **73**, 044028 (2006).
[19] C. O. Lousto, H. Nakano, Y. Zlochower, and M. Campanelli, Phys. Rev. Lett. **104**, 211101 (2010).
[20] C. O. Lousto and Y. Zlochower, Phys. Rev. Lett. **106**, 041101 (2011).
[21] E. Poisson, A. Pound, and I. Vega, Living Rev. Relativ. **14**, 7 (2011).
[22] R. Geroch and J. Traschen, Phys. Rev. D **36**, 1017 (1987).
[23] R. Steinbauer and J. A. Vickers, Classical Quantum Gravity **23**, R91 (2006).
[24] Y. Mino, M. Sasaki, and T. Tanaka, Phys. Rev. D **55**, 3457 (1997).
[25] T. C. Quinn and R. M. Wald, Phys. Rev. D **56**, 3381 (1997).
[26] S. Detweiler, Phys. Rev. Lett. **86**, 1931 (2001).

[27] S. E. Gralla and R. M. Wald, Classical Quantum Gravity **25**, 205009 (2008).
[28] S. E. Gralla, A. I. Harte, and R. M. Wald, Phys. Rev. D **80**, 024031 (2009).
[29] S. Detweiler, Classical Quantum Gravity **22**, S681 (2005).
[30] L. Barack, Classical Quantum Gravity **26**, 213001 (2009).
[31] L. Barack, in *Mass and Motion in General Relativity*, Fundamental Theories of Physics, Vol. 162, edited by L. Blanchet, A. Spallicci, and B. F. Whiting (Springer-Verlag, Berlin, 2011), p. 309.
[32] L. M. Burko, in *Mass and Motion in General Relativity*, edited by L. Blanchet, A. Spallicci, and B. F. Whiting (Springer-Verlag, Berlin, 2011), p. 395.
[33] S. Detweiler, in *Mass and Motion in General Relativity*, Fundamental Theories of Physics, Vol. 162, edited by L. Blanchet, A. Spallicci, and B. F. Whiting (Springer-Verlag, Berlin, 2011), p. 271.
[34] E. Poisson, in *Mass and Motion in General Relativity*, edited by L. Blanchet, A. Spallicci, and B. F. Whiting (Springer-Verlag, Berlin, 2011), p. 309.
[35] R. M. Wald, in *Mass and Motion in General Relativity*, Fundamental Theories of Physics, Vol. 162, edited by L. Blanchet, A. Spallicci, and B. F. Whiting (Springer-Verlag, Berlin, 2011), p. 253.
[36] N. Sago, L. Barack, and S. Detweiler, Phys. Rev. D **78**, 124024 (2008).
[37] L. Barack and A. Ori, Phys. Rev. D **61**, 061502(R) (2000).
[38] L. Barack, Phys. Rev. D **62**, 084027 (2000).
[39] L. Barack, Y. Mino, H. Nakano, A. Ori, and M. Sasaki, Phys. Rev. Lett. **88**, 091101 (2002).
[40] L. Barack and A. Ori, Phys. Rev. D **66**, 084022 (2002).
[41] L. Barack and A. Ori, Phys. Rev. D **67**, 024029 (2003).
[42] S. Detweiler and B. F. Whiting, Phys. Rev. D **67**, 024025 (2003).
[43] S. Detweiler, E. Messaritaki, and B. F. Whiting, Phys. Rev. D **67**, 104016 (2003).
[44] R. Haas and E. Poisson, Phys. Rev. D **74**, 044009 (2006).
[45] W. G. Anderson and A. G. Wiseman, Classical Quantum Gravity **22**, S783 (2005).
[46] M. Casals, S. R. Dolan, A. C. Ottewill, and B. Wardell, Phys. Rev. D **79**, 124043 (2009).
[47] M. Casals, S. Dolan, A. C. Ottewill, and B. Wardell, Phys. Rev. D **88**, 044022 (2013).
[48] B. Wardell, C. R. Galley, A. Zenginoğlu, M. Casals, S. R. Dolan, and A. C. Ottewill, Phys. Rev. D **89**, 084021 (2014).
[49] L. Barack and D. A. Golbourn, Phys. Rev. D **76**, 044020 (2007).
[50] I. Vega and S. Detweiler, Phys. Rev. D **77**, 084008 (2008).
[51] B. Wardell, in *Equations of Motion in Relativistic Gravity*, Fundamental Theories of Physics, Vol. 179, edited by D. Puetzfeld, C. Lmmerzahl, and B. Schutz (Springer International Publishing, Berlin, 2015), pp. 487–522.
[52] N. Warburton and L. Barack, Phys. Rev. D **83**, 124038 (2011).
[53] B. Wardell, I. Vega, J. Thornburg, and P. Diener, Phys. Rev. D **85**, 104044 (2012).
[54] L. Blanchet, S. Detweiler, A. Le Tiec, and B. F. Whiting, Phys. Rev. D **81**, 064004 (2010).
[55] A. G. Shah, T. S. Keidl, J. L. F. D.-H. Kim, and L. R. Price, Phys. Rev. D **83**, 064018 (2011).
[56] A. Heffernan, A. Ottewill, and B. Wardell, Phys. Rev. D **86**, 104023 (2012).
[57] N. K. Johnson-McDaniel, A. G. Shah, and B. F. Whiting, Phys. Rev. D **92**, 044007 (2015).
[58] K. Glampedakis and D. Kennefick, Phys. Rev. D **66**, 044002 (2002).
[59] L. Barack and C. O. Lousto, Phys. Rev. D **72**, 104026 (2005).
[60] L. Barack, A. Ori, and N. Sago, Phys. Rev. D **78**, 084021 (2008).
[61] J. L. Barton, D. J. Lazar, D. J. Kennefick, G. Khanna, and L. M. Burko, Phys. Rev. D **78**, 064042 (2008).
[62] R. Haas, Phys. Rev. D **75**, 124011 (2007).
[63] L. Barack and N. Sago, Phys. Rev. D **81**, 084021 (2010).
[64] C. Hopman and T. Alexander, Astrophys. J. **629**, 362 (2005).
[65] R. M. Wald, *General Relativity* (The University of Chicago Press, Chicago, 1984).
[66] D. R. Brill, P. L. Chrzanowski, C. M. Pereira, E. D. Fackerell, and J. R. Ipser, Phys. Rev. D **5**, 1913 (1972).
[67] S. A. Teukolsky, Astrophys. J. **185**, 635 (1973).
[68] I. Vega, B. Wardell, and P. Diener, Classical Quantum Gravity **28**, 134010 (2011).
[69] I. Vega, P. Diener, W. Tichy, and S. Detweiler, Phys. Rev. D **80**, 084021 (2009).
[70] P. Diener, I. Vega, B. Wardell, and S. Detweiler, Phys. Rev. Lett. **108**, 191102 (2012).
[71] I. Vega, B. Wardell, P. Diener, S. Cupp, and R. Haas, Phys. Rev. D **88**, 084021 (2013).
[72] L. Barack, D. A. Golbourn, and N. Sago, Phys. Rev. D **76**, 124036 (2007).
[73] S. R. Dolan and L. Barack, Phys. Rev. D **83**, 024019 (2011).
[74] S. R. Dolan, B. Wardell, and L. Barack, Phys. Rev. D **84**, 084001 (2011).
[75] S. R. Dolan and L. Barack, Phys. Rev. D **87**, 084066 (2013).
[76] W. Krivan, P. Laguna, and P. Papadopoulos, Phys. Rev. D **54**, 4728 (1996).
[77] P. A. Sundararajan, G. Khanna, and S. A. Hughes, Phys. Rev. D **76**, 104005 (2007).
[78] A. Zenginoğlu, Classical Quantum Gravity **25**, 145002 (2008).
[79] A. Zenginoğlu, Classical Quantum Gravity **25**, 195025 (2008).
[80] A. Zenginoğlu, J. Comput. Phys. **230**, 2286 (2011).
[81] A. Zenginoğlu and G. Khanna, Phys. Rev. X **1**, 021017 (2011).
[82] A. Zenginoğlu and L. E. Kidder, Phys. Rev. D **81**, 124010 (2010).
[83] A. Zenginoğlu and M. Tiglio, Phys. Rev. D **80**, 024044 (2009).
[84] S. Bernuzzi, A. Nagar, and A. Zenginoğlu, Phys. Rev. D **84**, 084026 (2011).
[85] S. Bernuzzi, A. Nagar, and A. Zenginoğlu, Phys. Rev. D **86**, 104038 (2012).
[86] B. Char, K. O. Geddes, W. M. Gentleman, and G. H. Gonnet, in *Computer Algebra*, Lecture Notes in Computer

Science Vol. 162, edited by J. A. van Hulzen (Springer-Verlag, 1983) pp. 101–115.
[87] A. Heffernan, A. Ottewill, and B. Wardell, Phys. Rev. D **89**, 024030 (2014).
[88] B. Wardell, I. Vega, J. Thornburg, and P. Diener, Phys. Rev. D **85**, 104044 (2012).
[89] Y. Mino, H. Nakano, and M. Sasaki, Prog. Theor. Phys. **108**, 1039 (2002).
[90] A. P. Prudnikov, Y. A. Brychkov, and O. I. Marichev, *Integrals and Series* (Gordon and Breach, New York, 1986), Vol. 1.
[91] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevD.95.084043 for Mathematica code deriving expressions for the puncture function and $m$-mode regularization parameters.
[92] S. E. Field, J. S. Hesthaven, and S. R. Lau, Phys. Rev. D **81**, 124030 (2010).
[93] J. L. Jaramillo, C. F. Sopuerta, and P. Canizares, Phys. Rev. D **83**, 061503 (2011).
[94] J. Thornburg, arXiv:1006.3788.
[95] A. Savitzky and M. J. E. Golay, Anal. Chem. **36**, 1627 (1964).
[96] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes*, 3rd ed. (Cambridge University Press, Cambridge, 2007).
[97] Y. Mino, Phys. Rev. D **67**, 084027 (2003).
[98] T. Hinderer and E. E. Flanagan, Phys. Rev. D **78**, 064028 (2008).
[99] L. M. Diaz-Rivera, E. Messaritaki, B. F. Whiting, and S. Detweiler, Phys. Rev. D **70**, 124018 (2004).
[100] L. Barack and N. Sago, Phys. Rev. Lett. **102**, 191101 (2009).
[101] M. W. Choptuik, Phys. Rev. D **44**, 3124 (1991).
[102] J. Thornburg, Phys. Rev. D **59**, 104007 (1999).
[103] N. Warburton (private communication).
[104] L. Barack, (private communication).
[105] G. Allen, T. Goodale, and E. Seidel, in *7th Symposium on the Frontiers of Massively Parallel Computation-Frontiers 99* (IEEE, New York, 1999).
[106] T. Goodale, G. Allen, G. Lanfermann, J. Massó, T. Radke, E. Seidel, and J. Shalf, in *Vector and Parallel Processing–VECPAR'2002, 5th International Conference,* Lecture Notes in Computer Science (Springer, Berlin, 2003).
[107] Cactus developers, Cactus Computational Toolkit.
[108] A. Dubey, A. Almgren, J. Bell, M. Berzins, S. Brandt, G. Bryan, P. Colella, D. Graves, M. Lijewski, F. Löffler, B. O'Shea, E. Schnetter, B. V. Straalen, and K. Weide, J. Parallel Distrib. Comput. **74**, 3217 (2014).
[109] E. Schnetter, S. H. Hawley, and I. Hawke, Classical Quantum Gravity **21**, 1465 (2004).
[110] F. Löffler, J. Faber, E. Bentivegna, T. Bode, P. Diener, R. Haas, I. Hinder, B. C. Mundim, C. D. Ott, E. Schnetter, G. Allen, M. Campanelli, and P. Laguna, Classical Quantum Gravity **29**, 115001 (2012).
[111] EinsteinToolkit, Einstein Toolkit: Open software for relativistic astrophysics.
[112] L. E. Kidder, S. E. Field, F. Foucart, E. Schnetter, S. A. Teukolsky, A. Bohn, N. Deppe, P. Diener, F. Hébert, J. Lippuner, J. Miller, C. D. Ott, M. A. Scheel, and T. Vincent, arXiv:1609.00098.
[113] J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, Texts in Applied Mathematics (Springer, New York, London, 2008).
[114] S. E. Field, J. S. Hesthaven, and S. R. Lau, Classical Quantum Gravity **26**, 165010 (2009).
[115] S. E. Field, J. S. Hesthaven, S. R. Lau, and A. H. Mroue, Phys. Rev. D **82**, 104051 (2010).
[116] J. D. Brown, P. Diener, S. E. Field, J. S. Hesthaven, F. Herrmann, A. H. Mroué, O. Sarbach, E. Schnetter, M. Tiglio, and M. Wagman, Phys. Rev. D **85**, 084004 (2012).
[117] K. Fan, W. Cai, and X. Ji, J. Comput. Phys. **227**, 2387 (2008).
[118] Éanna É. Flanagan and T. Hinderer, Phys. Rev. Lett. **109**, 071102 (2012).
[119] J. Brink, M. Geyer, and T. Hinderer, Phys. Rev. D **91**, 083001 (2015).
[120] J. Thornburg, GW Notes **5**, 3 (2011).
[121] J. Vines, D. Kunst, J. Steinhoff, and T. Hinderer, Phys. Rev. D **93**, 103008 (2016).
[122] N. Warburton, S. Akcay, L. Barack, J. R. Gair, and N. Sago, Phys. Rev. D **85**, 061501R (2012).
[123] A. C. Hindmarsh, *Scientific Computing*, IMACS Transactions on Scientific Computing (Elsevier Science Ltd, New York, 1983), Vol. 1.
[124] K. Radhakrishnan and A. C. Hindmarsh, in *Description and Use of LSODE, the Livermore Solver for Ordinary Differential Equations*, NASA Reference Publication 1327, Technical Report No. UCRL-ID-113855 (Lawrence Livermore National Laboratory, Livermore, CA, 1993).
[125] D. Goldberg, ACM Comput. Surv. **23**, 5 (1991).
[126] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward, ACM Trans. Math. Softw. **31**, 363 (2005).
[127] M. J. Berger, Ph. D. thesis, Stanford University, 1982, University Microfilms #DA 83-01196.
[128] M. J. Berger and J. Oliger, J. Comput. Phys. **53**, 484 (1984).
[129] M. J. Berger, SIAM J. Sci. Stat. Comput. **7**, 904 (1986).
[130] M. J. Berger and P. Colella, J. Comput. Phys. **82**, 64 (1989).
[131] R. Courant, K. Friedrichs, and H. Lewy, Math. Ann. **100**, 32 (1928) [(English translation IBM Journal of Research and Development **11**, 215 (1967)].
[132] U. M. Ascher, S. J. Ruuth, and B. T. R. Wetton, SIAM J. Numerical Analysis **32**, 797 (1995).
[133] U. M. Ascher, S. J. Ruuth, and R. J. Spiteri, Applied Numerical Mathematics **25**, 151 (1997).
[134] L. Pareschi and G. Russo, Recent Trends in Numerical Analysis **3**, 269 (2000).
[135] L. Pareschi and G. Russo, J. Sci. Comput. **25**, 129 (2005).
[136] S. Boscarino, SIAM J. Num. Anal. **45**, 1600 (2007).
[137] S. Boscarino, Applied Numerical Mathematics **59**, 1515 (2009).
[138] S. Boscarino and G. Russo, SIAM J. Sci. Comput. **31**, 1926 (2009).