# An 1+1 Dimensional Self-force Code for the Einstein Toolkit?

Peter Diener

Center for Computation & Technology
Department of Physics & Astronomy
Louisiana State University

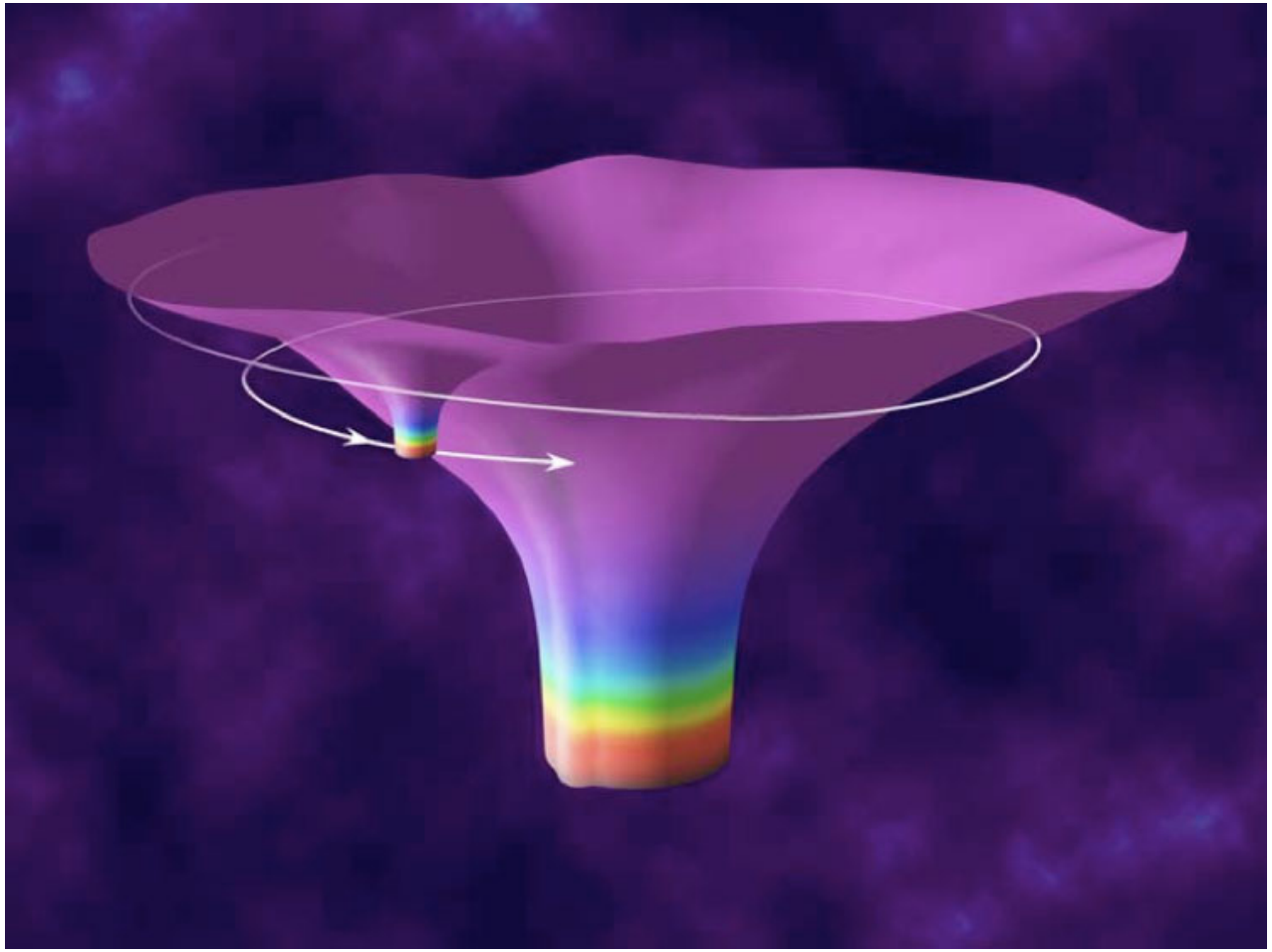
Supported by NSF grant PHY-130739

September 12, 2018
European Einstein Toolkit Workshop 2018
Instituto Superior Tecnico, Lisbon, Portugal

# Outline

- Extreme Mass Ratio Inspirals.
- The Self-force Problem.
- The Scalar Charge Playground.
- The Effective Source Approach.
- The 1+1D Code.
  1. Discontinuous Galerkin.
  2. Hyperboloidal Slices.
  3. Time Dependent Coordinates/Worldtube.
  4. Orbit Evolution.
  5. Other Infrastructure.
- Code Philosophy.
- Example Results.
- Outlook.

# Extreme Mass Ratio Inspirals.



Artist's impression of an EMRI. Credit NASA

A small compact object in orbit around a supermassive black hole.

# Extreme Mass Ratio Inspirals.

- Supermassive black holes are surrounded by a cluster of stars.
- Some of these will be black holes or neutron stars.
- Can be brought onto highly eccentric orbits by two-body interactions.
- Energy and angular momentum losses through gravitational wave emission shrinks the orbit until the small object plunges into the supermassive black hole.
- Eccentricity will decrease over time but will most likely still be significant just before the plunge.
- Such systems are expected to be very important events for the space based gravitational wave detector LISA.
- Analytically and numerically they can be handled using perturbation theory and the point particle approximation.
- The particle perturbs the spacetime and interacts with it's own perturbations to accelerate the orbit.

# The Self-force Problem.

To ease development of new numerical techniques we instead use the scalar charge case as a playground.
We wish to determine the self-forced motion and field of a particle with scalar charge

$$\begin{aligned} \Box\psi^{\rm ret} &= -4\pi q \int \delta^{(4)}(x - z(\tau))\, d\tau. \\ \frac{Du^\alpha}{d\tau} &= a^\alpha = \frac{q}{m}(g^{\alpha\beta} + u^\alpha u^\beta)\nabla_\beta \psi^{\rm R} \end{aligned}$$

2 general approaches:

- Compute enough "geodesic"-based self-forces and then use these to drive the motion of the particle. (Post-processing, fast, accurate self-forces, relies on slow orbit evolution)
- Compute the "true" self-force while simultaneously driving the motion. (Slow and expensive, less accurate self-forces)

# The Effective Source Approach.

The effective source is a general approach to self-force and self-consistent orbital evolution that doesn't use any delta functions.

Key ideas

- Compute a regular field, $\psi^{\mathrm{R}}$, such that the self-force is

$$F_\alpha = \nabla_\alpha \psi^{\mathrm{R}}|_{x=z},$$

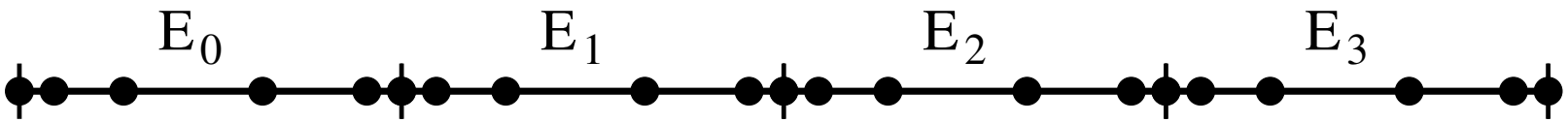
  where $\psi^{\mathrm{R}} = \psi^{\mathrm{ret}} - \psi^{\mathrm{S}}$, and $\psi^{\mathrm{S}}(x|z,u,a)$ can be approximated via local expansions: $\psi^{\mathrm{S}} = \tilde{\psi}^{\mathrm{S}} + O(\epsilon^n)$.

- The effective source, $S$, for the field equation for $\psi^{\mathrm{R}}$ is regular at the particle location.

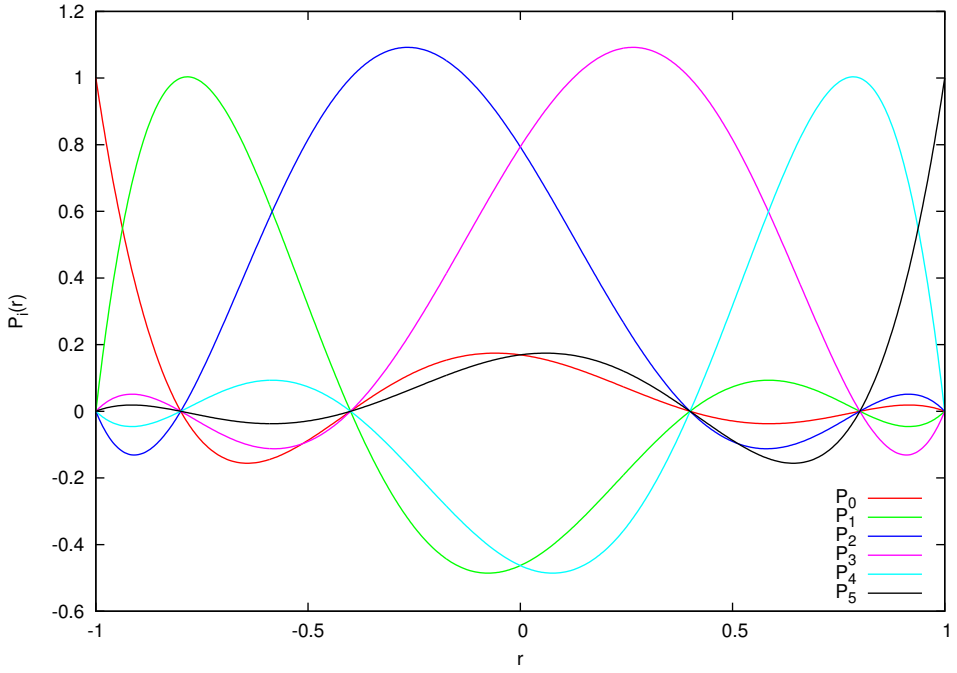
$$\Box\psi^{\mathrm{R}} = \Box\psi^{\mathrm{ret}} - \Box\tilde{\psi}^{\mathrm{S}} = S(x|z,u,a,\dot{a},\ddot{a})$$

  where $\Box\ \tilde{\psi}^{\mathrm{S}} = -4\pi q \int \delta^{(4)}(x - z(\tau))\, d\tau - S$.

# The 1+1D Code: Discontinuous Galerkin.




- Split the domain into $N$ $n$th order elements.
- Each element contains $n+1$ nodes.
- $u(t,x) \approx \sum_{i=0}^{n} \tilde{u}(t,x_i) P_i(x)$




- The numerical approximation is double valued at all element boundaries.
- Derivatives are approximated by multiplying the state vector in each element by a derivative matrix.
- Neighboring elements are glued together by numerical fluxes.

# The 1+1D Code: Discontinuous Galerkin.

- Numerical fluxes can be constructed in many different ways in order to maintain numerical stability and to guarantee that the jumps in the solution at the element boundaries converge to zero.
- We use fluxes based on a characteristic decomposition of the wave equation.

The convergence properties of the DG method for smooth solutions are

- Exponential with the order $n$ (with $N$ kept fixed).
- polynomial with the element size $1/N$ (with $n$ kept fixed).

As the DG scheme has discontinuities built in at the element boundaries, we retain these convergence properties even when the solution itself is non-smooth IF and only if, the non-smooth features can be placed at element boundaries.
(Hesthaven & Warburton, 2007)

# The 1+1D Code: Hyperboloidal Slices.

The code is 1+1 dimensional based on the spherical harmonic decomposition of the scalar wave equation in the Schwarzschild spacetime in tortoise coordinates $r_* = r + 2M \log(r/(2M) - 1)$ with a spherically harmonic decomposed effective source.

$$-\frac{\partial^2 \psi_{\ell m}}{\partial t^2} + \frac{\partial^2 \psi_{\ell m}}{\partial r_*^2} - V_\ell(r)\psi_{\ell m} = S^{\rm eff}_{\ell m}.$$

As $r_* \in [-\infty, \infty]$ we split the domain into three regions. In the inner ($r_* \in [-\infty, T_1]$) and outer ($r_* \in [T_2, \infty]$) regions we introduce new coordinates $(\tau, \rho)$ used in Bernuzzi, Nagar & Zenginoğlu (2011).

$$\begin{aligned} t &= \tau + h(\rho) \\ r_* &= \rho/\Omega(\rho) \end{aligned}$$

where $h(\rho)$ and $\Omega(\rho)$ are chosen suitably (hyperboloidal layers) in each region to make the inner boundary ($\rho_{\rm min}$) coincide with the horizon $H$ and the outer boundary ($\rho_{\rm max}$) coincide with $\mathscr{I}^+$.

# The 1+1D Code: Time Dependent Coordinates/Worldtube

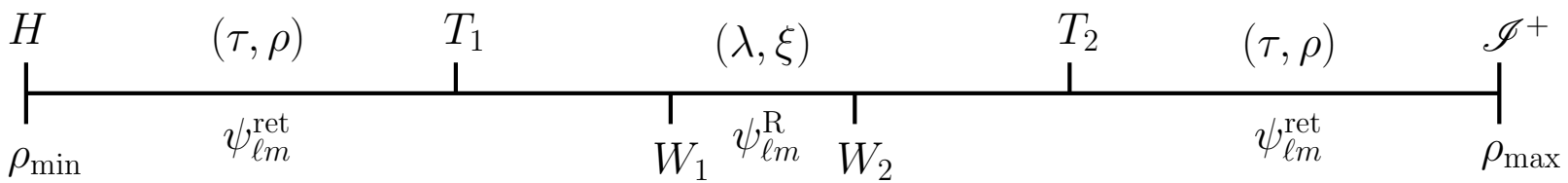
In the middle region ($r_* \in [T_1, T_2]$) we introduce a time dependent coordinate transformation (Field, Hesthaven & Lau, 2009)

$$\begin{aligned} t &= \lambda \\ r_* &= T_1 + \frac{r_*^p - T_1}{\xi^p - T_1}(\xi - T_1) + \\ &\quad \frac{(T_2 - r_*^p)(\xi^p - T_1) - (r_*^p - T_1)(T_2 - \xi^p)}{(\xi^p - T_1)(T_2 - \xi^p)(T_2 - T_1)}(\xi - T_1)(\xi - \xi^p) \end{aligned}$$

where $r_*^p$ is the time-dependent particle location. This satisfies $r_*(\lambda, T_1) = T_1$, $r_*(\lambda, \xi^p) = r_*^p$, $r_*(\lambda, T_2) = T_2$.

In addition we use the world tube approach so that we evolve $\psi^{\rm R}_{\ell m} = \psi^{\rm ret}_{\ell m} - \psi^{\rm S}_{\ell m}$ in the region $r_* \in [W_1, W_2]$ (where typically $W_1 > T_1$ and $W_2 < T_2$), while elsewhere we evolve $\psi^{\rm ret}_{\ell m}$.

The location of $T_1$, $W_1$, $W_2$ and $T_2$ is of course chosen to coincide with element boundaries.



# The 1+1D Code: Orbital Evolution.

- Orbital evolution in Schwarzschild can be done either by direct integration of the geodesic equations or by the osculating orbit equations.
- Geodesic equations: $z^\alpha$ and $u^\alpha$ are evolved.
- Osculating orbit equations: $p$ (semilatus rectum), $\chi$, $\phi$, $\alpha$ and $\beta$ are evolved. Here the radial coordinate is given by

$$r = \frac{pM}{1 + \alpha \sin\chi + \beta \cos\chi}$$

while the eccentricity is given by

$$e = \sqrt{\alpha^2 + \beta^2}.$$

- The advantage of the osculating orbit equations is that many quantities evolve on the radiation reaction timescale rather than the orbital timescale.

# The 1+1D Code: Other Infrastructure.

- The Method of Lines (MoL) is used for time integration. Currently low storage, 5 stage, 4th order and 8 stage, 5th order, continuous Runge-Kutta methods have been implemented. An Adams-Bashford-Moulton multivalue method is being implemented.
- Routines for extracting the self-force and then summing up the spherical harmonics modes have been implemented.
- Routines for reading in initial data from frequency domain codes have been implemented.
- An interface to Barry Wardell's effective source for a scalar charge moving on a Schwarzschild background has been implemented.

# Code Philosphy.

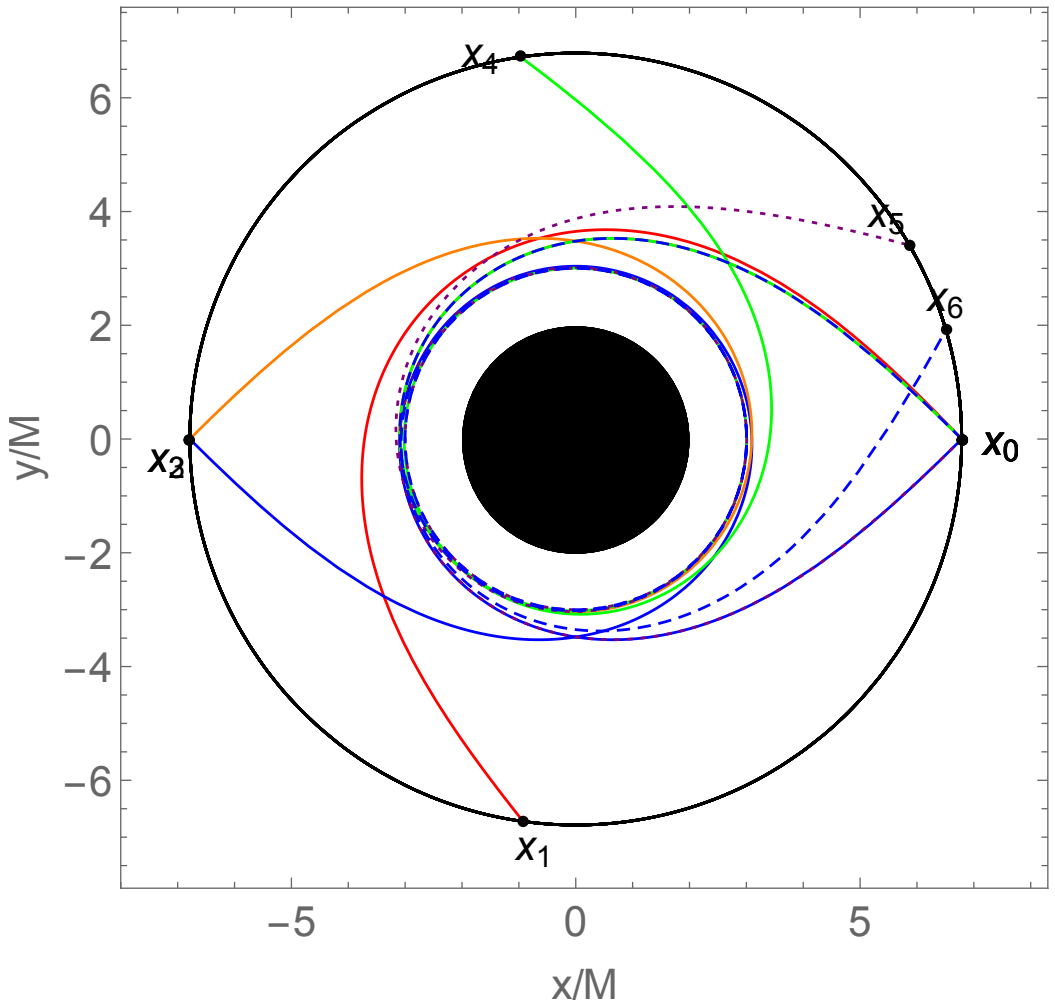
- The code is written in modern Fortran, i.e. it uses object oriented programming features of Fortran 2008.
- The code is highly modular. Inspired by Cactus but implemented using abstract Fortran classes, inheritance polymorphism, and type bound procedures (the fortran equivalent of C++ member functions).
- Example:
  - The time integrators gets initialized with an array of pointers to the abstract equation class.
  - These can point to equations of either ODE or PDE type.
  - Calculation of RHS's and updates to variables are performed using type bound procedures provided by the equations.
  - Hence the time integrator does not need to know anything about the number or type of the variables.
- The code uses modules and submodules, to clearly separate the interface from the implementation.
- Communication of fundamental variables between different equations are done via 'base' classes (similar to ADMBase).

# Example Results.

$$p = 6.7862,\ e = 0.0,\ A = 0.05,\ \sigma = 1.8,\ t_0 = 800$$

$t_1 = 30.1867$
$t_2 = 55.5379$
$t_3 = 55.5379$
$t_4 = 80.7708$
$t_5 = 101.779$
$t_6 = 106.003$

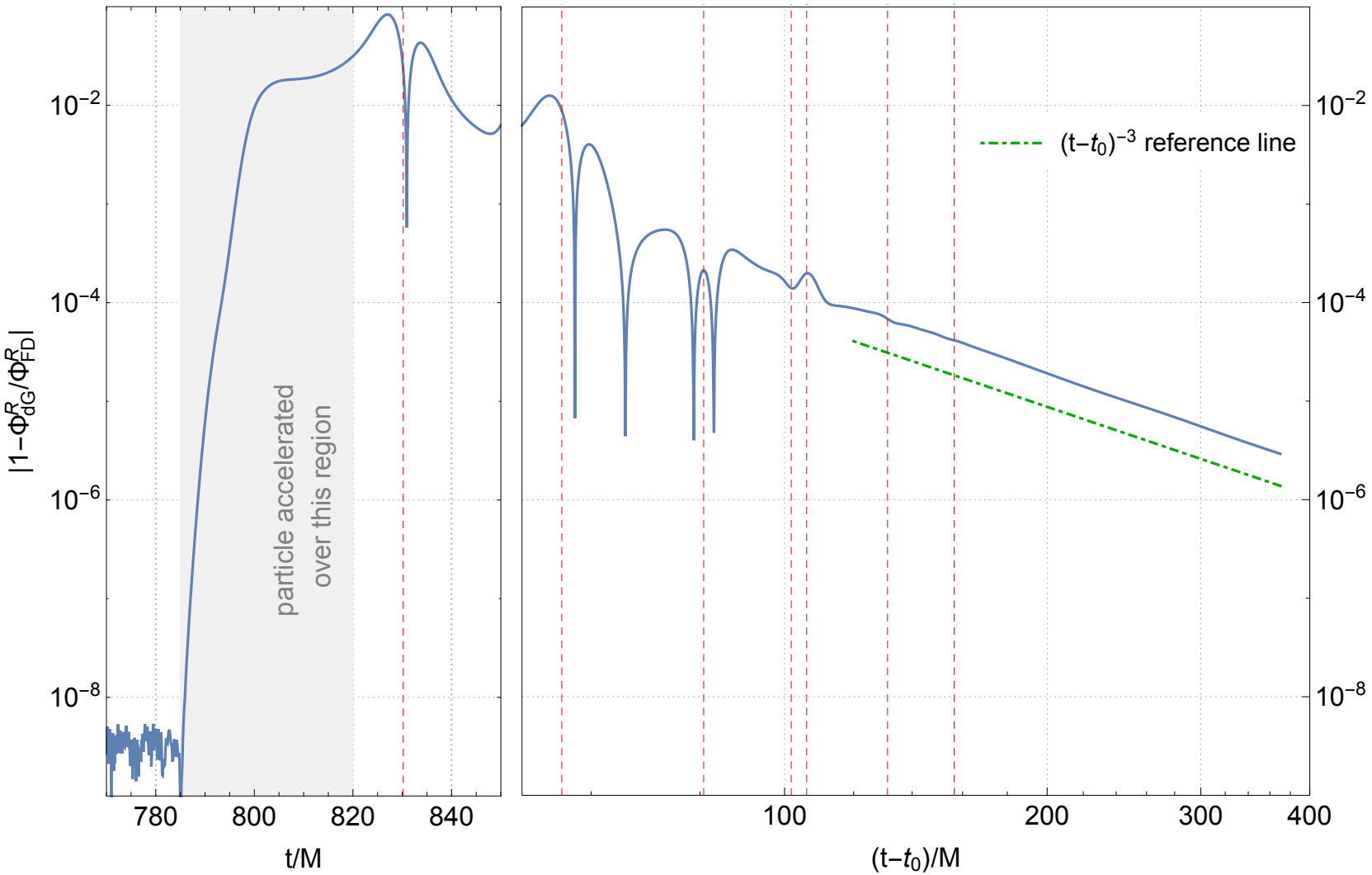
Evolve up to $\ell = 65$.
DG order up to 96.




$$\Omega(t) = \Omega_0 + Ae^{-\frac{(t-t_0)^2}{\sigma^2}}$$

# Example Results.

$$p = 6.7862,\ e = 0.0,\ A = 0.05,\ \sigma = 1.8,\ t_0 = 800$$



# Outlook.

- The code is designed to be very modular.
- An REU student implemented the Teukolsky equation (in Schwarzschild) this summer.
- Regge-Wheeler-Zerilli and Lorenz gauge gravitational perturbations codes in Schwarzschild are in the pipeline.
- Also plans to generalize to Kerr (requires evolving coupled modes).
- Even though the code was developed for self-force simulations, it might also be useful for other purposes.
- The plan is to release the code as open source.
- The question is whether the ET community would like it to be included in the ET.