Towards the Self-consistent Evolution of a Scalar Charge Around a Schwarzschild Black Hole (Yet Again)

Peter Diener[1]
in collaboration with
Barry Wardell[2] and Niels Warburton[2]

[1]Louisiana State University
[2]University College Dublin

June 18, 2019
22nd Capra Meeting 2019, CBPF, Rio de Janeiro, Brazil

The problem.

We wish to determine the self-forced motion and field (e.g. energy and angular momentum fluxes) of a particle with scalar charge

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

Two 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 <u>while</u> simultaneously driving the motion. (Potentially slow and expensive, potentially less accurate self-forces)

Effective source approach.

... 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, ψ^{R}, such that the self-force is

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

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

- The effective source, S, for the field equation for ψ^{R} is regular at the particle location

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

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

Status at last years Capra.

- 1+1D Discontinuous Galerkin code (with time dependent coordinates as well as hyperboloidal slices) was able to do self-consistent evolutions under certain conditions.
  - No $\ddot{a}$.
  - With RK4: no $\dot{a}$ and no extrapolation to higher modes.
  - With ABMV: with $\dot{a}$ but ℓ_{max} fairly modest.
- In other cases an instability sets in.
- The code was able to do analytically prescribed highly accelerated orbits where $\dot{a}$ and $\ddot{a}$ are needed.
- The code, developed over several years had grown increasingly messy and unnecessarily complicated.
- With the plan to eventually release the code as open source it was decided that a redesign and complete rewrite was necessary.
- This will also allow easier extension to other systems.

The new design.

- Relies on object oriented programming ideas to expose and exploit modularity whenever possible.
- Implemented in modern Fortran 2003/2008.
- One of the basic concepts is an abstract 'Equation' class that knows nothing about the actual equations but defines the interface to certain type bound procedures (like C++ member functions) that any other Equation class has to provide.
- On top of this different types of Equation classes that know about the data structures needed (i.e. ODE or PDE equations) can be defined.
- On top of these actual equations systems (geodesic evolution, osculating orbits evolution and scalar wave equation) can finally be defined.
- The time integrator need only know about the type bound procedures as defined in the abstract equation class (implemented in the actual equation classes) and hence is completely agnostic about the underlying data structures.
- Communication between equations are done through external data types where different equation classes can write and read data without knowing about each other.

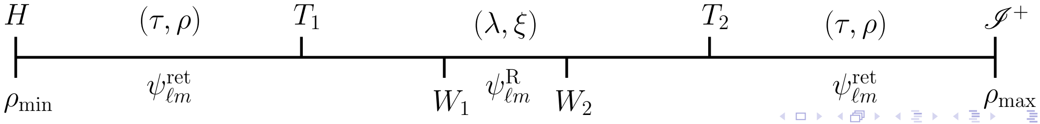
The abstract equation class

```
type, abstract :: equation
  integer :: ntmp
  character(:), allocatable :: ename
contains
  procedure (eq_init_interface), deferred, pass :: init
  procedure (eq_rhs_interface), deferred, pass :: rhs
  procedure (eq_set_to_zero_interface), deferred, pass :: set_to_zero
  procedure (eq_update_vars_interface), deferred, pass :: update_vars
  procedure (eq_save_globals_1), deferred, pass :: save_globals_1
  procedure (eq_save_globals_2), deferred, pass :: save_globals_2
  procedure (eq_load_globals), deferred, pass :: load_globals
  procedure (eq_output), deferred, pass :: output
end type equation
```

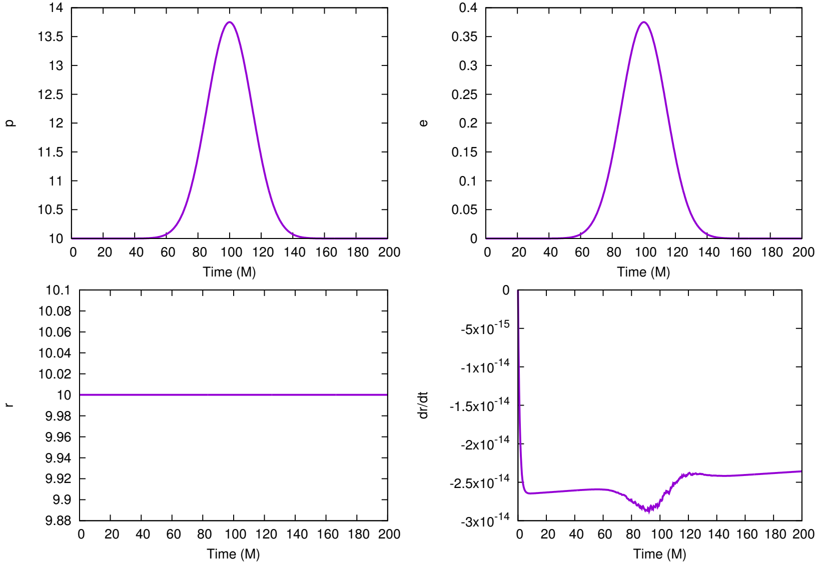
Other classes can then extend this class, provide some of the routines and defer other routines to the next level.

The code.

- Solves the spherical harmonic decomposed scalar wave equation in a Schwarzschild spacetime with a scalar effective source.
- Uses the Discontinuous Galerkin method for spatial discretization.
- Uses the method of lines approach and supports a number of time integrators.
- Uses a world-tube approach.
- Uses hyperboloidal slices, placing the computational domain boundaries at the horizon and $\mathscr{I}^+$.
- Uses a time dependent coordinate transformation to place the particle at a fixed coordinate location.
- The effective source include acceleration terms.
- Can read in frequency domain code initial data for small ℓ modes.
- Can evolve geodesics directly or through the osculating orbits framework.

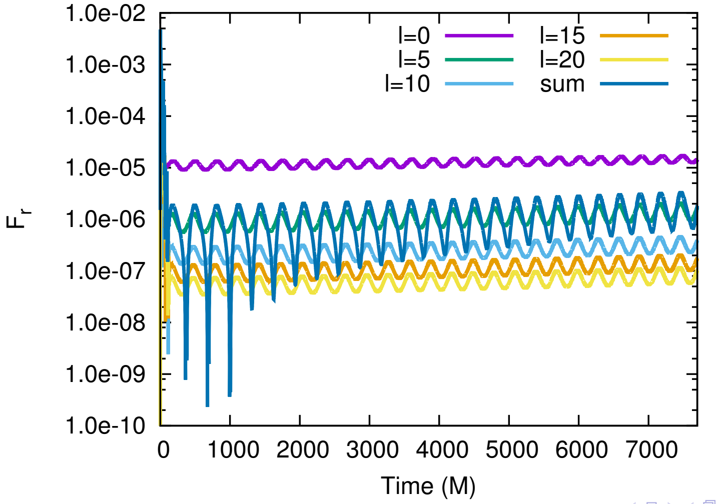


Test of the osculating orbits framework.



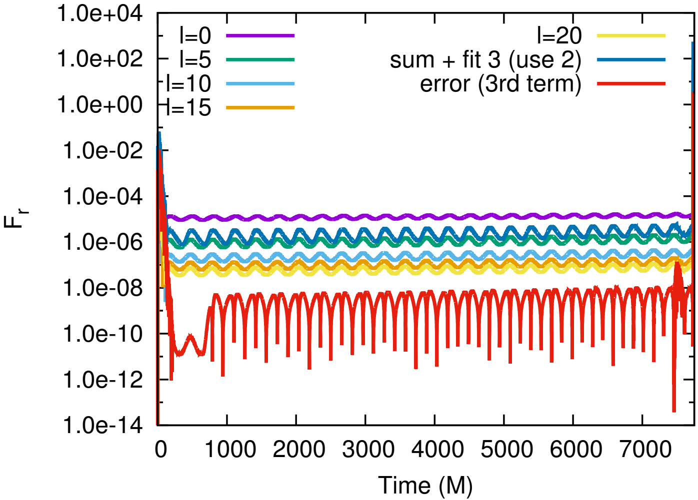
The state of self-consistent evolution now.

$p = 9.9$, $e = 0.1$, $q = 1/8$. Only a^μ passed in to the effective source. No fit to high ℓ.



The state of self-consistent evolution now.

$p = 9.9$, $e = 0.1$, $q = 1/8$. Only a^{μ} passed in to the effective source. Fit to high ℓ .



The state of self-consistent evolution now.

- Very similar to last year, but now with a much nicer code.
- We still have instabilities when we pass in $\dot{a}$ and/or $\ddot{a}$ as well as when we fit the higher ℓ-modes.
- Even using the Adams-Bashford-Moulton multi-value (ABMV) time integrator only extends the runs a bit before they go unstable.
- Fitting high ℓ-modes is important for accuracy so those instabilities need to be fixed.
- We still need to investigate how sensitive the results are to leaving out $\dot{a}$ and $\ddot{a}$.
- Have recently implemented some new Hermite extrapolating smoothing derivatives. First test did not look too promising but still inconclusive.
- May have to think outside the box to come up with ways to stabilize evolutions with back reaction.

Conclusions and Outlook.

- After redesign and rewrite the code is much nicer, easier to maintain and extend.
- Before we had separate codes for scalar in Schwarzschild, scalar in Kerr, Lorenz gauge, Regge-Wheeler-Zerilli and Teukolsky.
- Now they can all be implemented as different equation classes and share a common infrastructure.
- A basic Teukolsky code (without effective source) has already been implemented (Sarah Skinner).
- Samuel Cupp is working on the Lorenz gauge metric perturbation code.
- Have to fix back-reaction instabilities.
- Almost ready for release as open source software. Will be added to both the Black Hole Perturbation Toolkit and the Einstein Toolkit.
- If anybody is interested in using the code before the release, please talk to me.