Fast Self-Forced Inspirals into a Rotating Black Hole

Philip Lynch, Maarten van de Meent, Niels Warburton

UCD RELATIVITY GROUP

UCD DUBLIN

UCD School of Mathematics and Statistics

IRISH RESEARCH COUNCIL
An Chomhairle um Thaighde in Éirinn

Self-Forced Inspirals

Fast Self-Forced Inspirals

Contents

- Motivation
- Osculating Geodesics
- Near Identity Transformations
- Application to Kerr Spacetime
- Results
- Future Work

Motivation

Motivation

Method	Description	Fast?	Accurate?	Application
Kludge Models	Variety of analytic and numeric approximate models	Yes, very! ✓	Not quite adiabatic order	• Detection of loudest EMRI signals • LISA mock data challenges
Flux Driven Inspirals	Balance local changes in E & L with fluxes at ∞	Yes! ✓	Adiabatic order	• EMRI signal detection
Self-Force	Compute the local force on the orbiting body due to its own gravitational field	Nope!	Post-Adiabatic (with 2nd order) ✓	• Accurate parameter extraction • Strong tests of GR

Self-Force EMRI Waveforms in 3 “Easy” Steps

Can be sped up using NITs

1. Gravitational Self Force

- $a^{(1)}_{Dis} + a^{(1)}_{Cons}$
- Interpolate over Parameter Space

2. Trajectory

- Two-Timescale EoM
- **Osculating Geodesics**

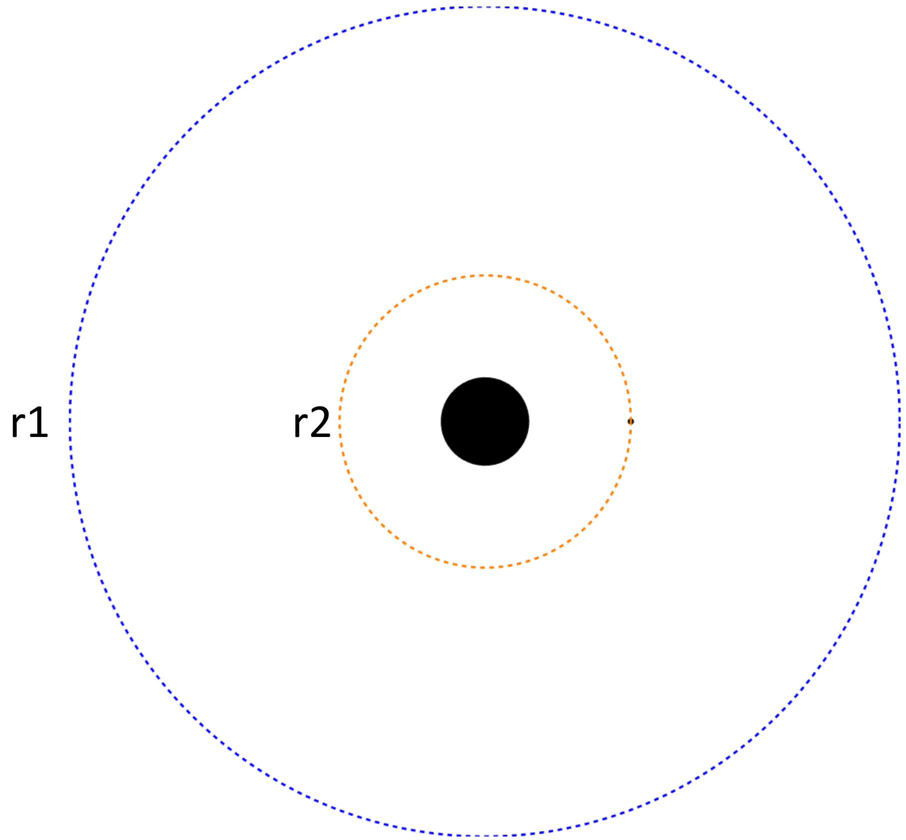
3. Waveform

- Quadrupole Formula
- Teukolsky–Based Methods

Goal: Extend NITs to Rotating Black Holes

Osculating Geodesics

Equatorial Geodesic Motion in Kerr Spacetime



$$\frac{d^2x^\alpha}{d\tau^2} + \Gamma^\alpha_{\beta\gamma}\frac{dx^\beta}{d\tau}\frac{dx^\gamma}{d\tau} = 0$$

- Spin Parameter: $a = \frac{J}{M}$
- Semilatus Rectum: $p = \frac{2r_1 r_2}{r_1 + r_2}$
- Eccentricity: $e = \frac{r_1 - r_2}{r_1 + r_2}$

Osculating Geodesics

- Each Geodesic can be identified with constants of motion:
- $P_j = \{p, e\}$
- And an initial phase $q_{i0} = \{\psi_{r0}\}$
- $I^A = \{P_j, q_{i0}\} \rightarrow I^A(\lambda) = \{P_j(\lambda), q_{i0}(\lambda)\}$
- Extrinsic Quantities due to symmetries: $S_K = \{t, \phi\}$

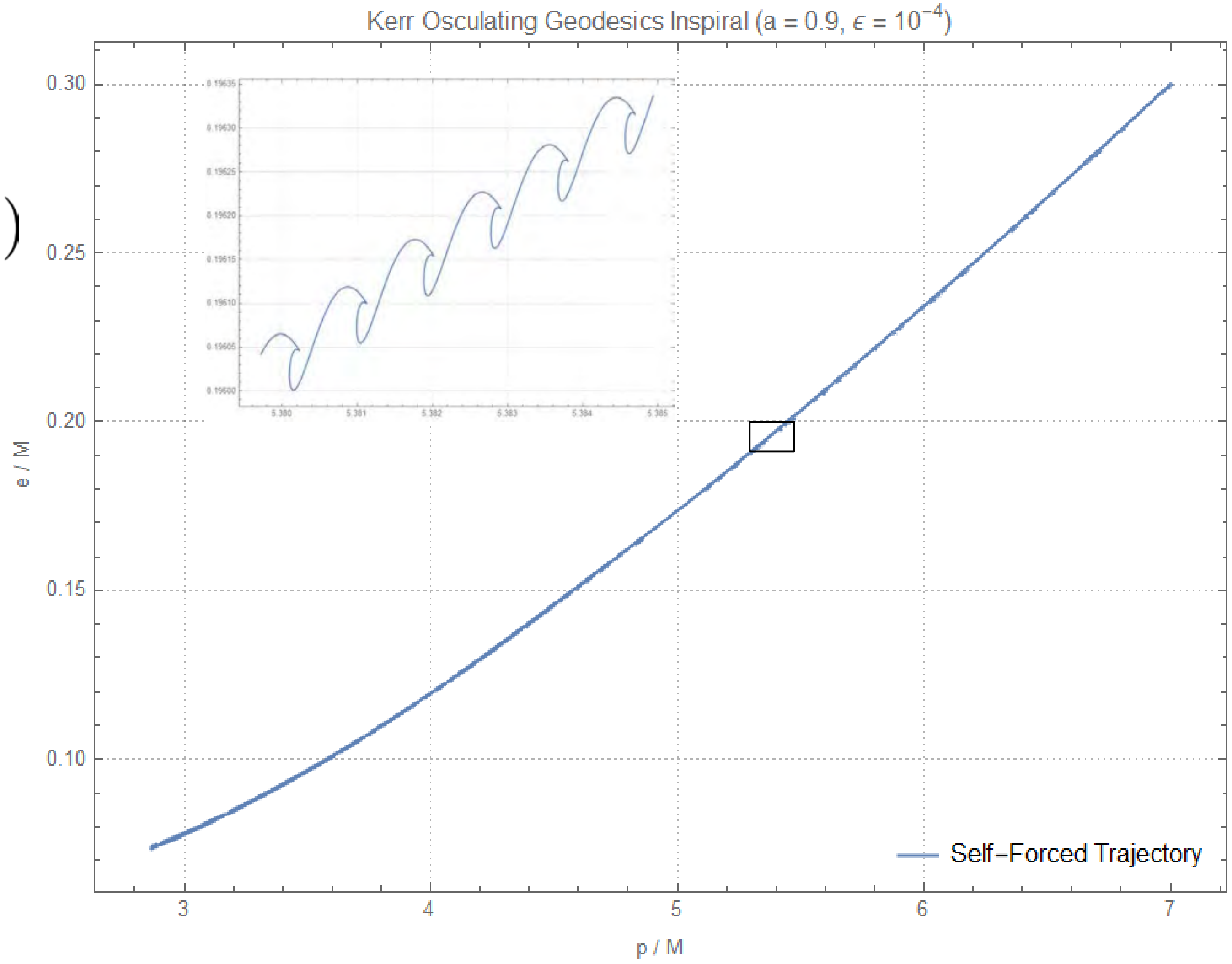
Forced Motion in Kerr Spacetime

$$\dot{P}_j = 0 + \epsilon F_j^{(1)}(\vec{P}, \vec{q} : a^\alpha)$$

$$\dot{q}_i = \Omega_i(\vec{P}, \vec{q}) + \epsilon f_i^{(1)}(\vec{P}, \vec{q} : a^\alpha)$$

$$\dot{S}_k = s_k(\vec{P}, \vec{q})$$

- Equations found by Gair et al [*arxiv.org/abs/1012.5111*].
- Take minutes to hours due to small oscillations



Near Identity Transformations

Near Identity (Averaging) Transformations

Transform to new variables

$$\tilde{P}_j = P_j + \epsilon Y_j^{(1)}(\vec{P},\vec{q}) + \epsilon^2 Y_j^{(2)}(\vec{P},\vec{q})\mathcal{O}(\epsilon^3),$$

$$\tilde{q}_i = q_i + \epsilon X_i^{(1)}(\vec{P},\vec{q}) + \epsilon^2 X_i^{(2)}(\vec{P},\vec{q})\mathcal{O}(\epsilon^3),$$

$$\tilde{S}_k = S_k + Z_k^{(0)}(\vec{P},\vec{q}) + \epsilon Z_i^{(1)}(\vec{P},\vec{q})\mathcal{O}(\epsilon^2).$$

To obtain equations of motion independent of $\vec{q}$

$$\dot{\tilde{P}}_j = 0 + \epsilon \tilde{F}_j^{(1)}(\vec{\tilde{P}}) + \epsilon^2 \tilde{F}_j^{(2)}(\vec{\tilde{P}})$$

$$\dot{\tilde{q}}_i = \Omega(\vec{\tilde{P}}) + \epsilon \tilde{f}_i^{(1)}(\vec{\tilde{P}}) + \epsilon^2 \tilde{f}_i^{(2)}(\vec{\tilde{P}})$$

$$\dot{\tilde{S}}_k = \tilde{s}_k^{(0)}(\vec{\tilde{P}}) + \epsilon \tilde{s}_k^{(1)}(\vec{\tilde{P}})$$

Order ϵ Example: Solving for $Y_j^{(1)}$

- Use EoM and NIT

$$\tilde{F}_j^{(1)} = \frac{1}{\epsilon}\dot{\tilde{P}}_j = \frac{1}{\epsilon}\left(\dot{P}_j + \epsilon\dot{Y}_j\right) = F_j^{(1)} + \frac{dY_j}{dq_i}\Omega_i$$

- Split into avg and osc

$$A(\vec{\tilde{P}}, \vec{\tilde{q}}) = \langle A\rangle(\vec{\tilde{P}}) + \sum_{\vec{\kappa}\neq\vec{0}} A_{\vec{\kappa}}(\vec{\tilde{P}})e^{i\vec{\kappa}\cdot\vec{\tilde{q}}}$$

- Cancel osc pieces

$$\tilde{F}_j^{(1)} = \left\langle F_j^{(1)}\right\rangle + \breve{F}_j^{(1)} + i(\vec{\kappa}\cdot\vec{\Omega})\breve{Y}_j.$$

$$Y_j^{(1)} \equiv \sum_{\vec{\kappa}\neq\vec{0}} \frac{i}{\vec{\kappa}\cdot\vec{\Omega}} F_{j,\vec{\kappa}}^{(1)} e^{i\vec{\kappa}\cdot\vec{q}}.$$

- Full details in van de Meent & Warburton [*arxiv.org/abs/1802.05281*]

NITs Cont'd

$$\dot{\tilde{P}}_j = 0 + \epsilon \tilde{F}_j^{(1)}(\vec{\tilde{P}}) + \epsilon^2 \tilde{F}_j^{(2)}(\vec{\tilde{P}})$$

$$\dot{\tilde{q}}_i = \Omega(\vec{\tilde{P}}) + \epsilon \tilde{f}_i^{(1)}(\vec{\tilde{P}}) + \epsilon^2 \tilde{f}_i^{(2)}(\vec{\tilde{P}})$$

$$\dot{\tilde{S}}_k = \tilde{s}_k^{(0)}(\vec{\tilde{P}}) + \epsilon \tilde{s}_k^{(1)}(\vec{\tilde{P}})$$

Independent of $\vec{q}$

$$\tilde{F}_j^{(1)} = \left\langle F_j^{(1)} \right\rangle, \tilde{f}_i^{(1)} = \left\langle f_i^{(1)} \right\rangle, \tilde{s}_k^{(0)} = \left\langle s_k^{(0)} \right\rangle,$$

$$\tilde{F}_j^{(2)} = \left\langle F_j^{(2)} \right\rangle + \left\langle \frac{\partial \breve{Y}_j^{(1)}}{\partial \tilde{q}_i} \breve{f}_i^{(1)} \right\rangle + \left\langle \frac{\partial \breve{Y}_j^{(1)}}{\partial \tilde{P}_k} \breve{F}_k^{(1)} \right\rangle,$$

$$\tilde{f}_i^{(2)} = 0, \text{ and}$$

$$\tilde{s}_k^{(1)} = \left\langle s_k^{(1)} \right\rangle - \left\langle \frac{\partial \breve{s}_k^{(0)}}{\partial \tilde{P}_j} \breve{Y}_j^{(1)} \right\rangle - \left\langle \frac{\partial \breve{s}_k^{(0)}}{\partial \tilde{q}_i} \breve{X}_i^{(1)} \right\rangle.$$

Application to Kerr Spacetime

Osculating Geodesics in Kerr Spacetime

$$\dot{P}_j = \epsilon F_j^{(1)}(\vec{P}, \vec{q} : a^\alpha)$$

$$\dot{q}_i = \Omega_i(\vec{P}, \vec{q}) + \epsilon f_i^{(1)}(\vec{P}, \vec{q} : a^\alpha)$$

$$\dot{S}_k = s_k(\vec{P}, \vec{q})$$

- Differentiation w.r.t Mino Time λ
- $\vec{P} = \{\mathcal{E}, \mathcal{L}, \mathcal{K}\}$
- $\vec{q} = \{\psi_r, \psi_z\}$
- $\vec{S} = \{t, \phi\}$

Problem

- Ω_i depend on $\vec{q}$
- NIT'd EoM would still depend on $\vec{q}$

New Equations of Motion

- Replace phases with action angles q_r and q_z

$$\frac{dq_i}{d\lambda} = \Upsilon_i(\vec{P})$$

- Where Υ_i is the Mino time fundamental frequency

- $r(q_r)$ and $z(q_z)$ are known for geodesics

$$\dot{P}_j = \epsilon F_j^{(1)}(\vec{P}, \vec{q} : a^\alpha)$$

$$\dot{q}_i = \Upsilon_i(\vec{P}) + \epsilon f_i^{(1)}(\vec{P}, \vec{q} : a^\alpha)$$

$$\dot{S}_k = s_k(\vec{P}, \vec{q})$$

NIT'd Equatorial Kerr Equations

$$\dot{\tilde{P}}_j = 0 + \epsilon \tilde{F}_j^{(1)}(\vec{\tilde{P}}) + \epsilon^2 \tilde{F}_j^{(2)}(\vec{\tilde{P}})$$

$$\dot{\tilde{q}}_i = \Omega(\vec{\tilde{P}}) + \epsilon \tilde{f}_i^{(1)}(\vec{\tilde{P}})$$

$$\dot{\tilde{S}}_k = \tilde{s}_k^{(0)}(\vec{\tilde{P}}) + \epsilon \tilde{s}_k^{(1)}(\vec{\tilde{P}})$$

- Differentiation w.r.t λ
- $\vec{P} = \{p, e\}$,
- $\vec{q} = \{q_r\}$,
- $\vec{\Omega} = \{\Upsilon_r\}$
- $\vec{S} = \{t, \phi\}$

$$\tilde{F}_j^{(1)} = \left\langle F_j^{(1)} \right\rangle, \tilde{f}_i^{(1)} = \left\langle f_i^{(1)} \right\rangle, \tilde{s}_k^{(0)} = \left\langle s_k^{(0)} \right\rangle$$

$$\tilde{F}_j^{(2)} = \sum_{n \neq 0} \frac{i}{n\Upsilon_r} \left[\left(\frac{\partial F_{j,n}^{(1)}}{\partial \tilde{P}_k} - \frac{\partial \Upsilon_r}{\partial \tilde{P}_k} \frac{F_{j,n}^{(1)}}{\Upsilon_r} \right) F_{k,-n}^{(1)} + inF_{j,n} f_{r,-n,-m}^{(1)} \right];$$

$$\tilde{s}_k^{(1)} = \sum_{n \neq 0} \frac{-i}{n\Upsilon_r} \left[\frac{\partial s_{k,n}^{(0)}}{\partial \tilde{P}_j} F_{j,-n}^{(1)} + s_{k,n}^{(0)} \left(inf_{r,-n,-m}^{(1)} + \frac{\partial \Upsilon_r}{\partial \tilde{P}_j} \frac{F_{j,-n}^{(1)}}{\Upsilon_r} \right) \right]$$

Steps Involved in Applying the NIT

Offline Steps

- Use fast Fourier transforms to find modes of F_j, f_i, s_k + derivatives at a given (a,p,e,x)
- Combine these modes to find $\tilde{F}_j^{(1)}, \tilde{F}_j^{(2)}, \tilde{f}_i^{(1)}, \tilde{s}_k^{(0)}$ *and* $\tilde{s}_k^{(1)}$
- Repeat across the parameter space
- Interpolate the points using cubic splines
- Save the interpolants

Online Steps

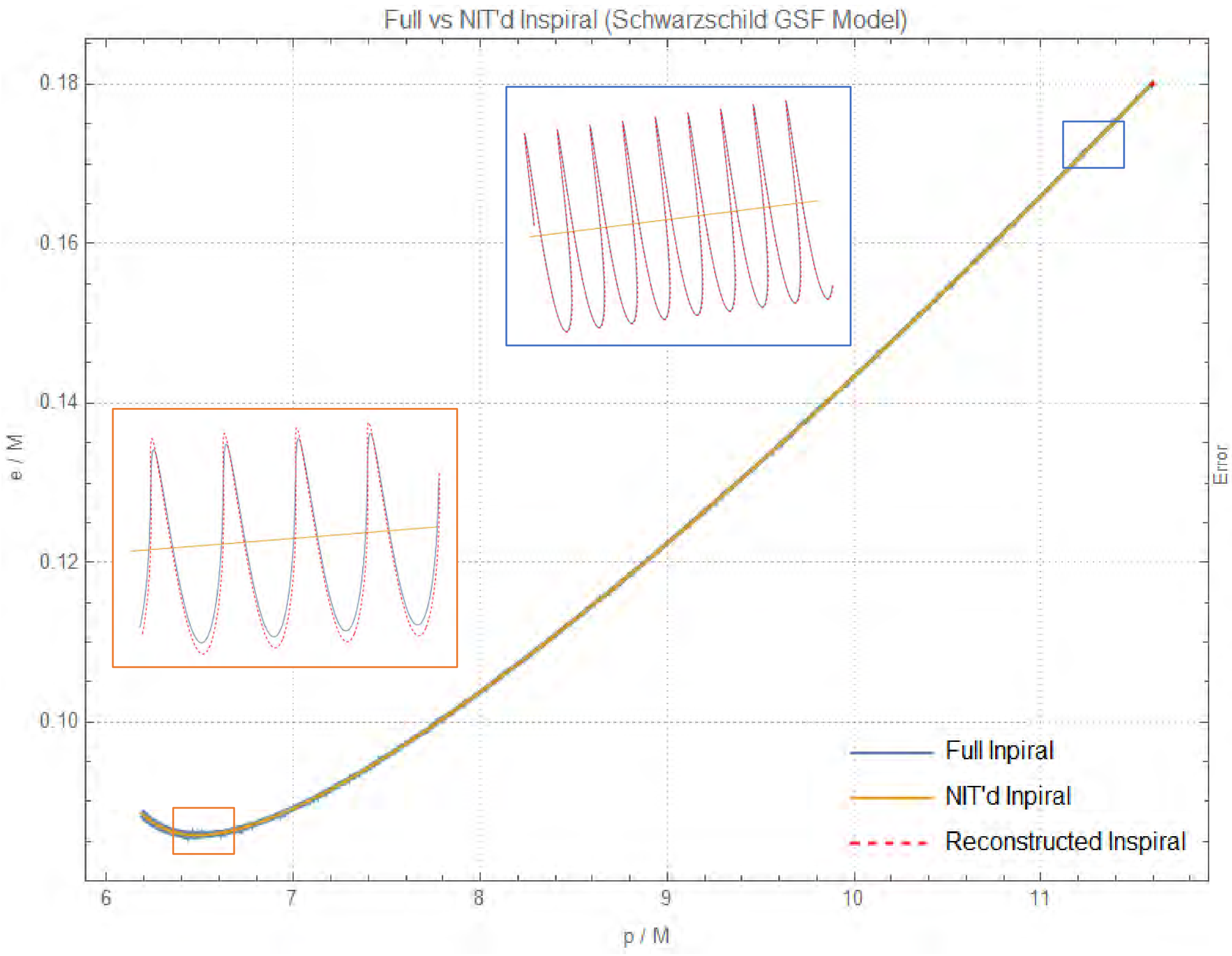
- Load interpolants
- Set initial conditions
- Numerically solve EoM

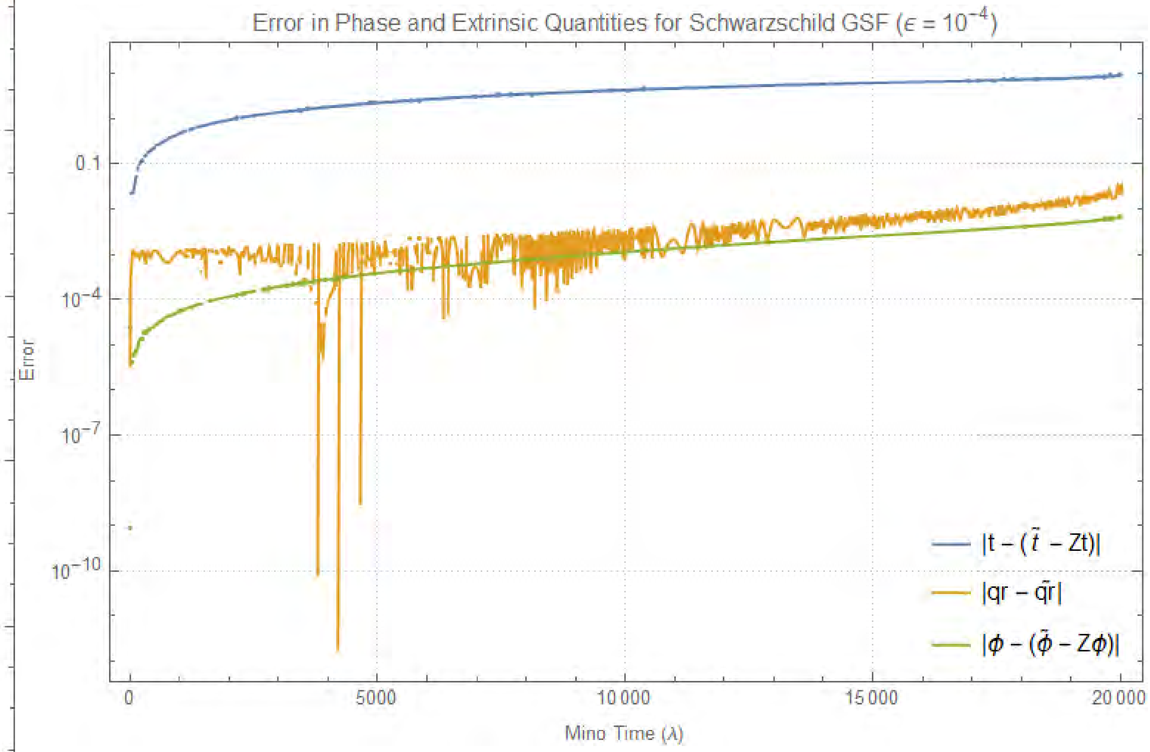
Results

Schwarzschild GSF Model

- First order dissipative and conservative force
- $a^{(1)}{}^{r}_{cons} = \sum_{n=0}^{\bar{n}} A_n(p,e)\cos(n\,\psi_r)$
- $A_n(p,e) = \mathrm{p}^{-2}\sum_{j=n}^{\bar{J}}\sum_{k=0}^{\bar{k}} a_{jk}^{n}\, e^{j} p^{-k}$
- Only valid from $6 \leq\ p \leq 12$ and $0 \leq e \leq 0.2$
- Set spin parameter a to 0.1

Schwarzschild GSF Results

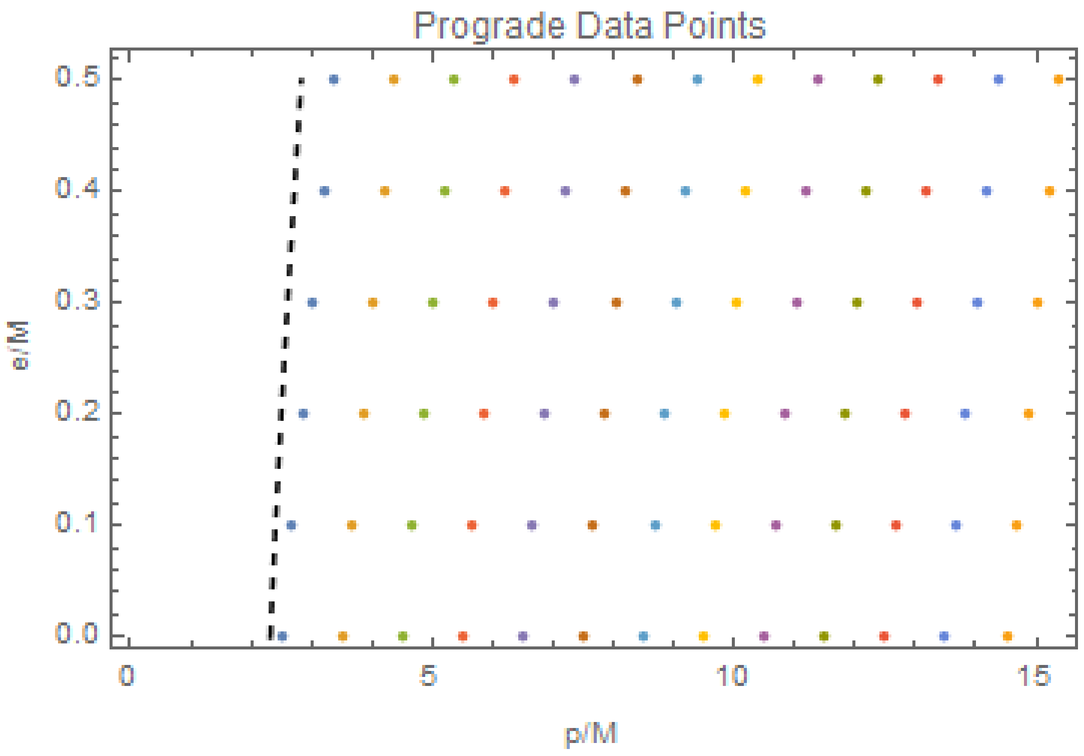




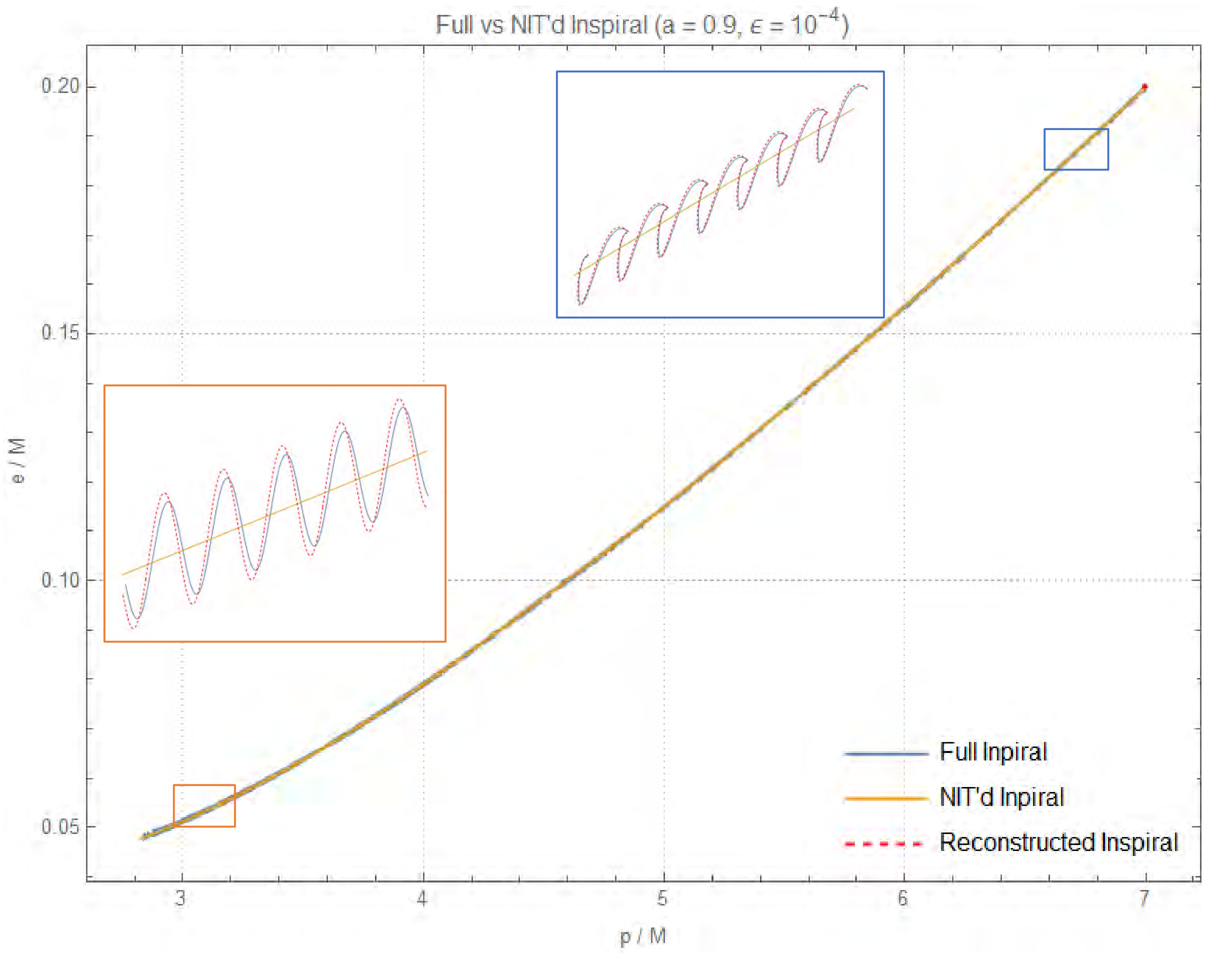
Interpolated Equatorial Kerr GSF Model

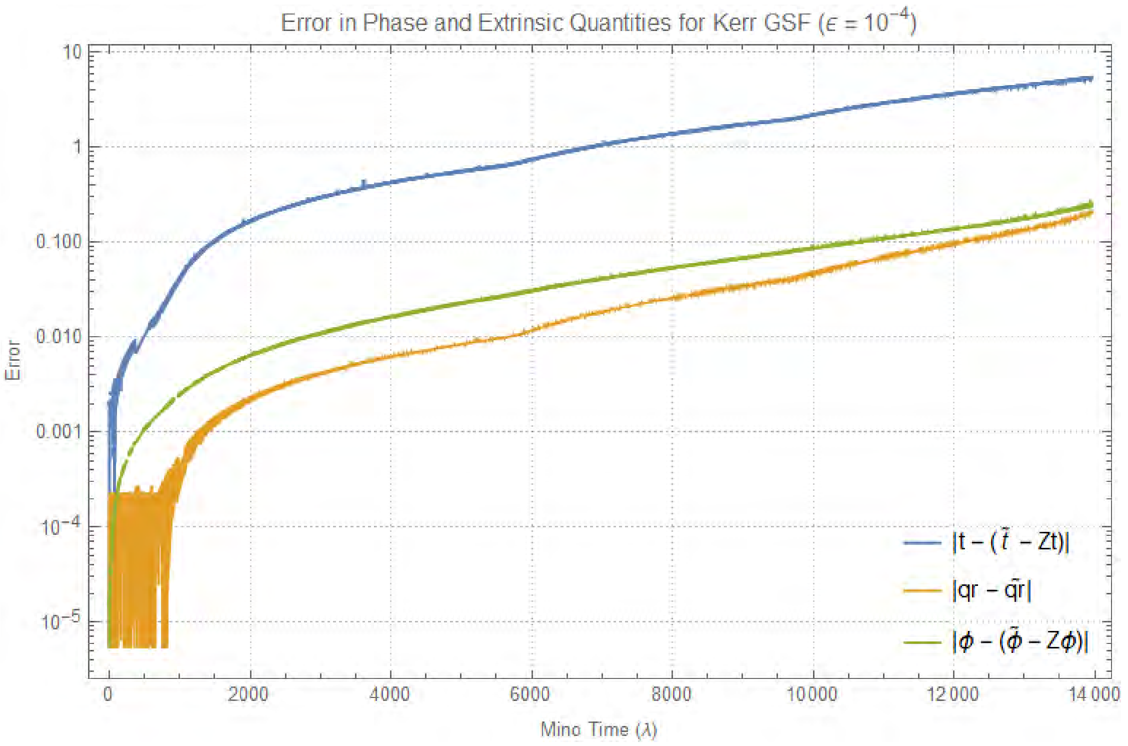
- Prograde Equatorial Data for a = 0.9
- $y = p - 1.75\, e$: 2.5 to 14.5
- e: 0 to 0.5

- Cubic spline interpolation



Kerr GSF Model (a = 0.9)





Runtime

Mass Ratio	Osculating Geodesics	NIT	Relative Speed Up
10^{-2}	52s	4.4s	$\sim \times 12$
10^{-3}	4mins 45s	2.9s	$\sim \times 98$
10^{-4}	35mins	5.3s	$\sim \times 392$
10^{-5}	4.25hrs	2.9s	$\sim \times 5280$
10^{-6}	???	3.9s	???

Future Work

To Do

- Better Kerr GSF Model
- Generate waveforms → Calculate Mismatch
- Generic model and Effect of Transient Resonance
- Two Timescale Expansions

Questions