Fast and accurate evaluation of black hole Green's functions using surrogate models

Chad Galley, California Institute of Technology

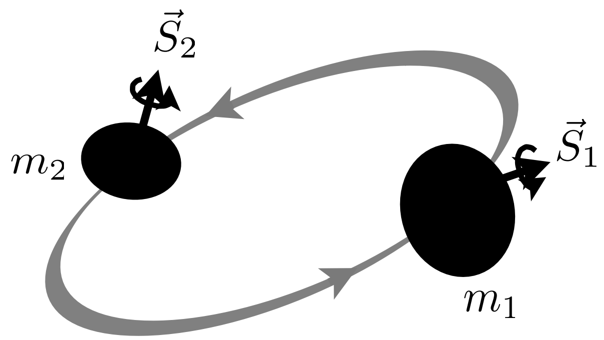
with Barry Wardell (University College Dublin)

GR21, New York City, July 11, 2016

Caltech

Compact binaries

Numerical relativity has proven success approximating compact binary evolutions

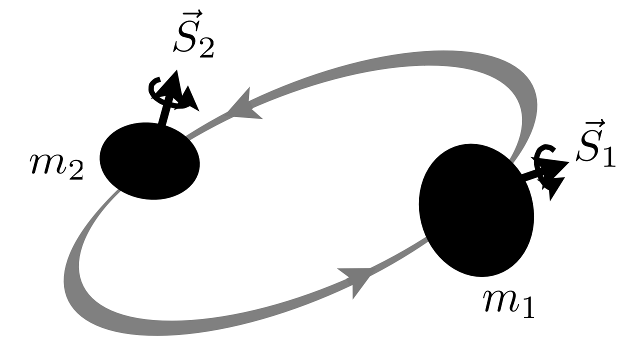


- Very good for mass ratios up to ~1/10
- Challenging for much smaller mass ratios

Compact binaries

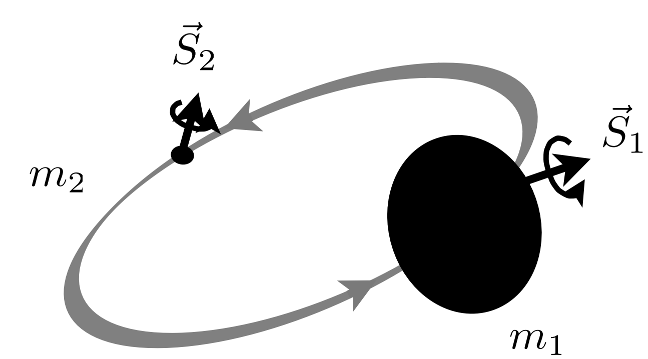
Numerical relativity has proven success approximating compact binary evolutions

- Very good for mass ratios up to ~1/10
- Challenging for much smaller mass ratios



Complementary description provided by self-force framework where mass ratio is an expansion parameter for perturbation theory

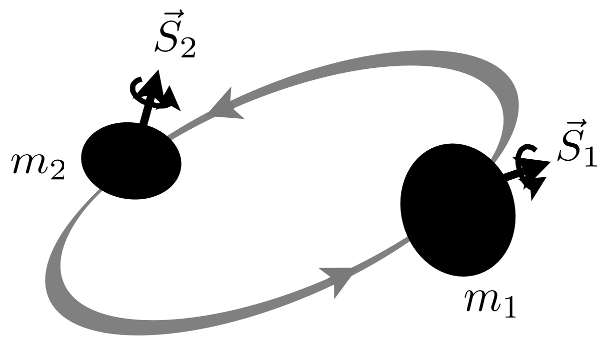
- Very good for mass ratios up to ~1/10
- Challenging for comparable mass ratios



Compact binaries

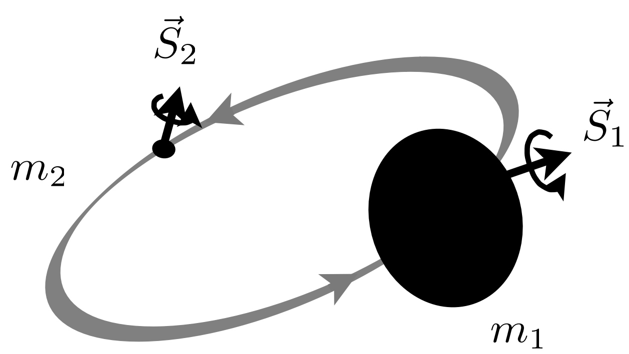
Numerical relativity has proven success approximating compact binary evolutions

- Very good for mass ratios up to ~1/10
- Challenging for much smaller mass ratios

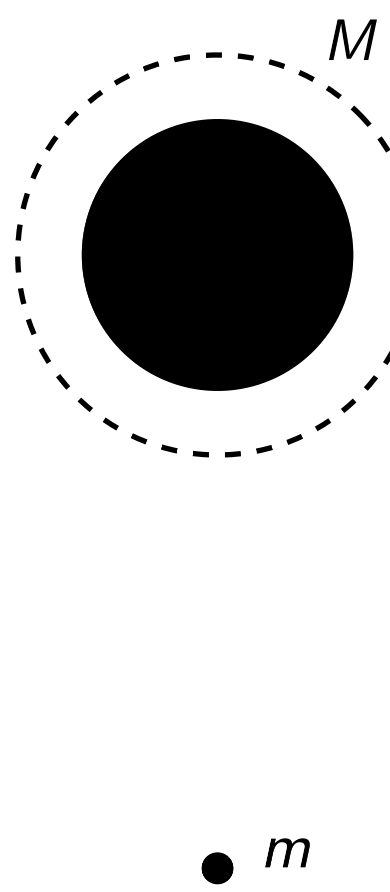


Complementary description provided by self-force framework where mass ratio is an expansion parameter for perturbation theory

- Very good for mass ratios up to ~1/10
- Challenging for comparable mass ratios

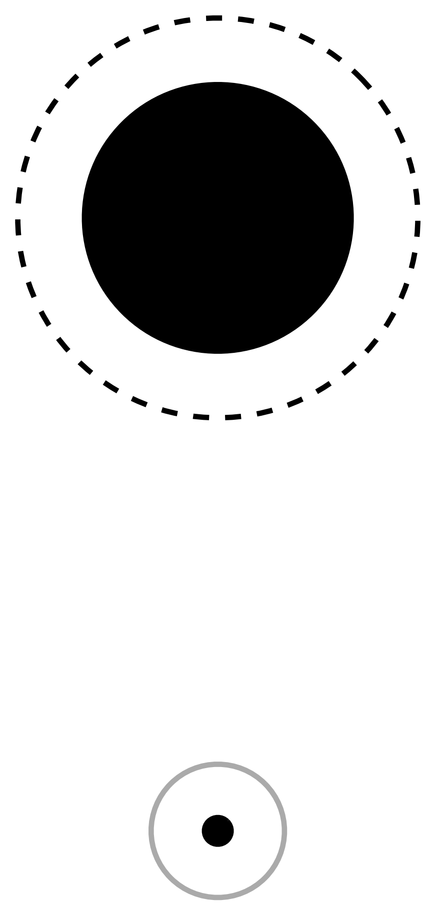


What is self-force?



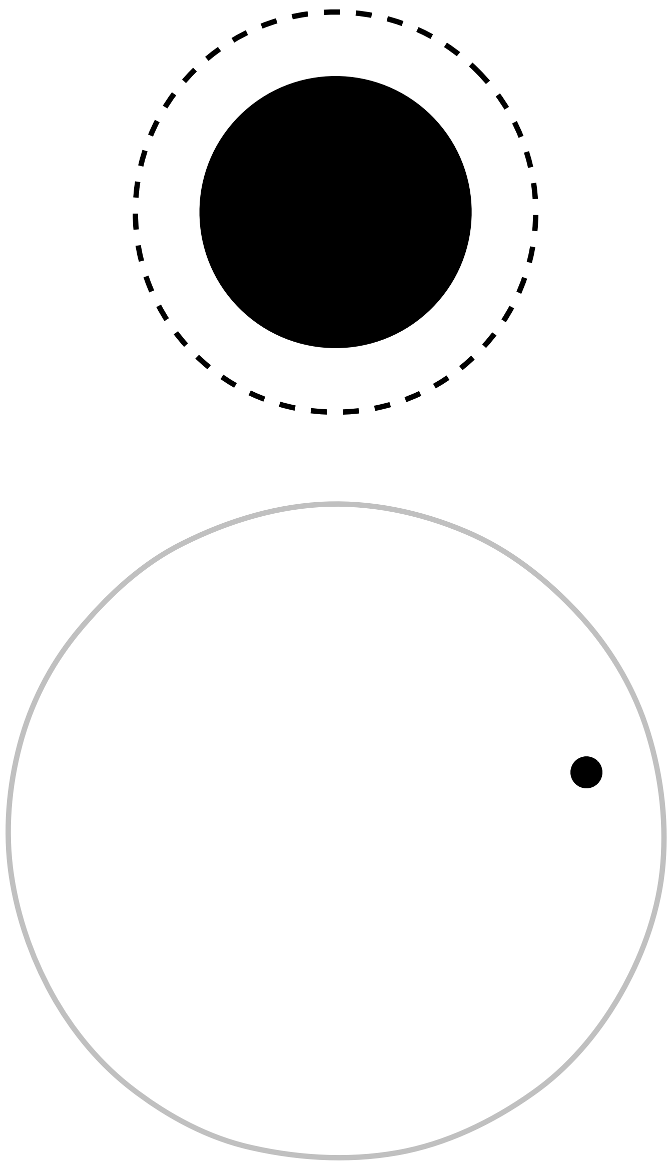
Self-force is like
radiation reaction
but history-dependent

What is self-force?



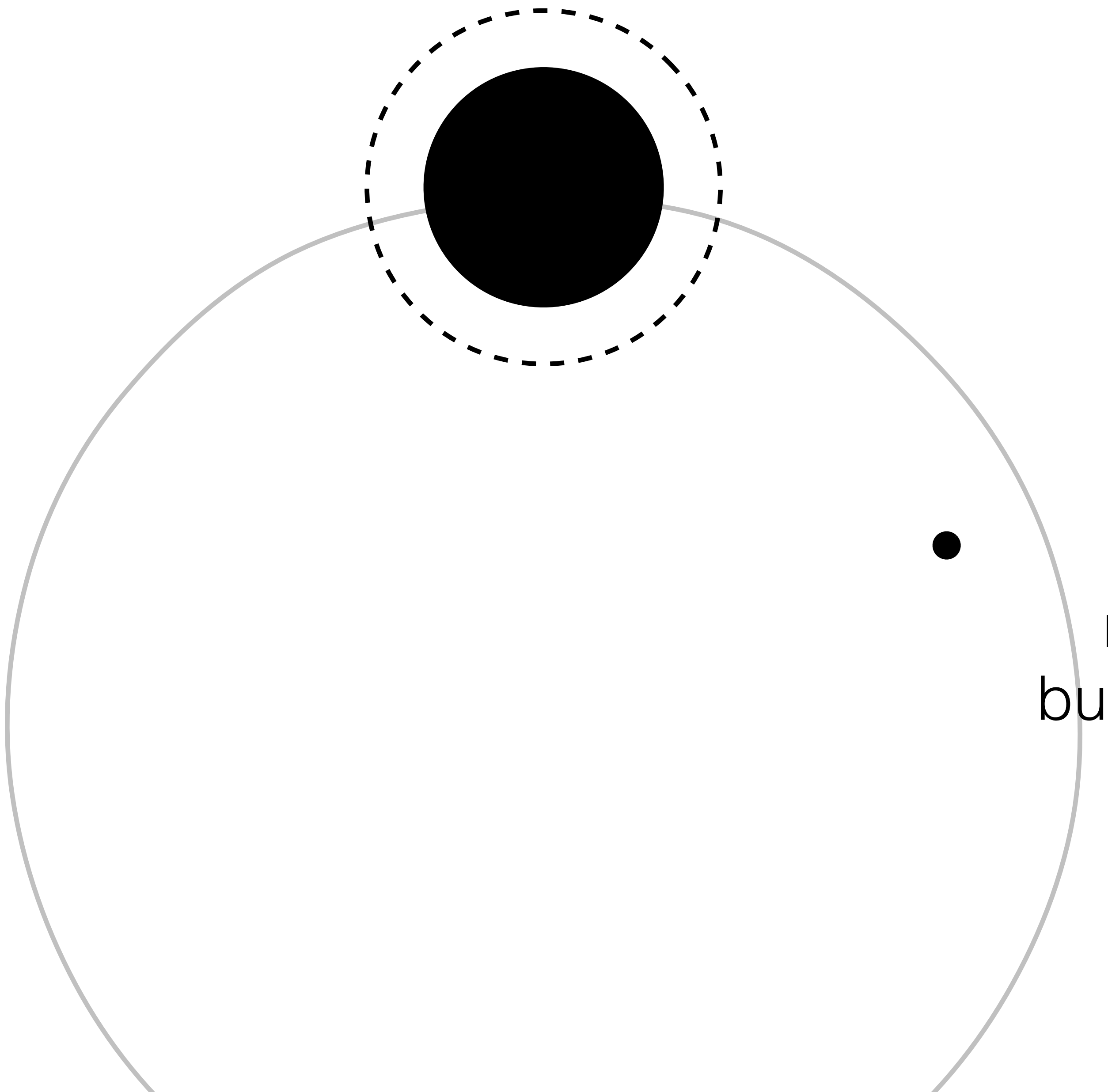
Self-force is like
radiation reaction
but history-dependent

What is self-force?



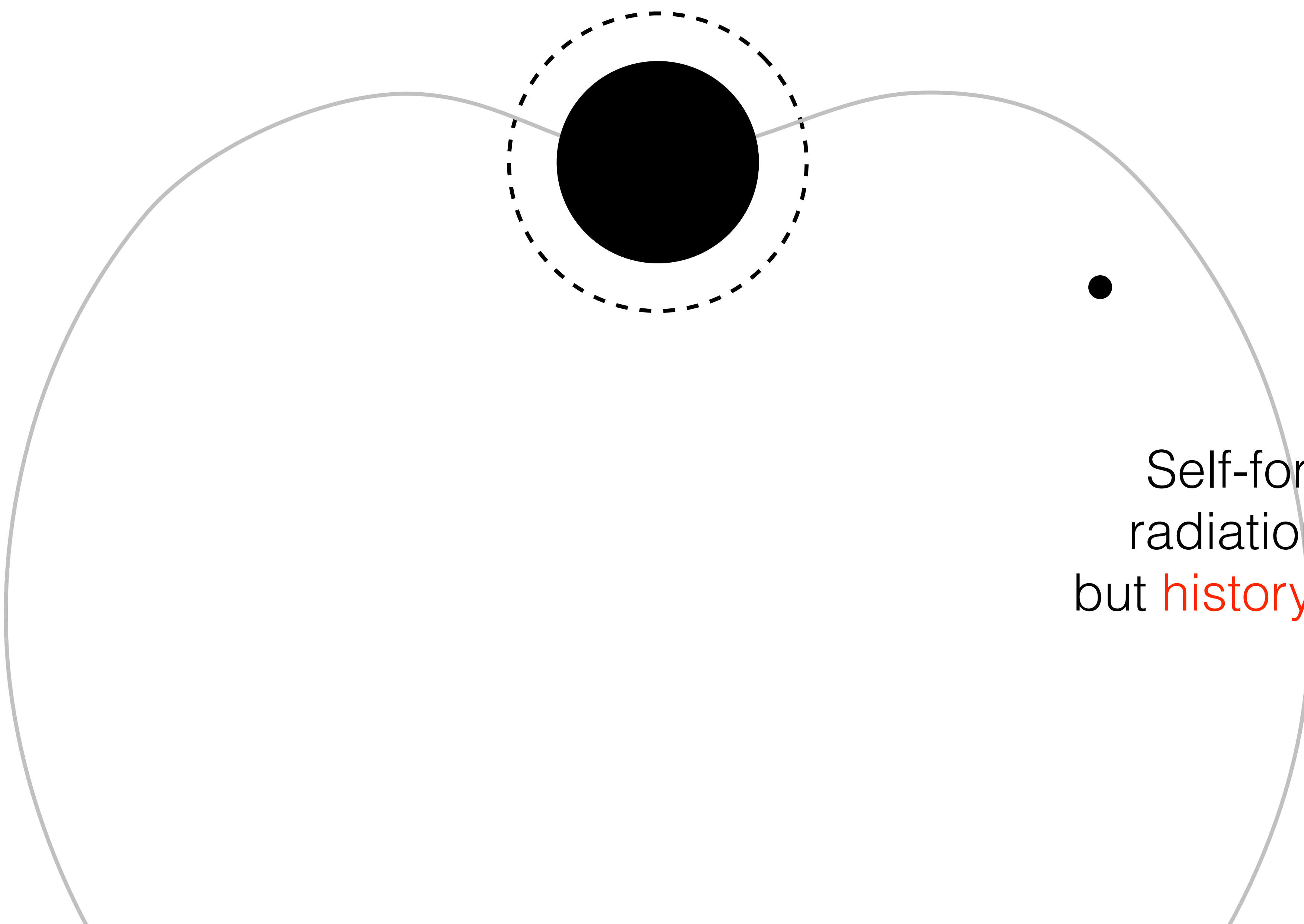
Self-force is like
radiation reaction
but history-dependent

What is self-force?



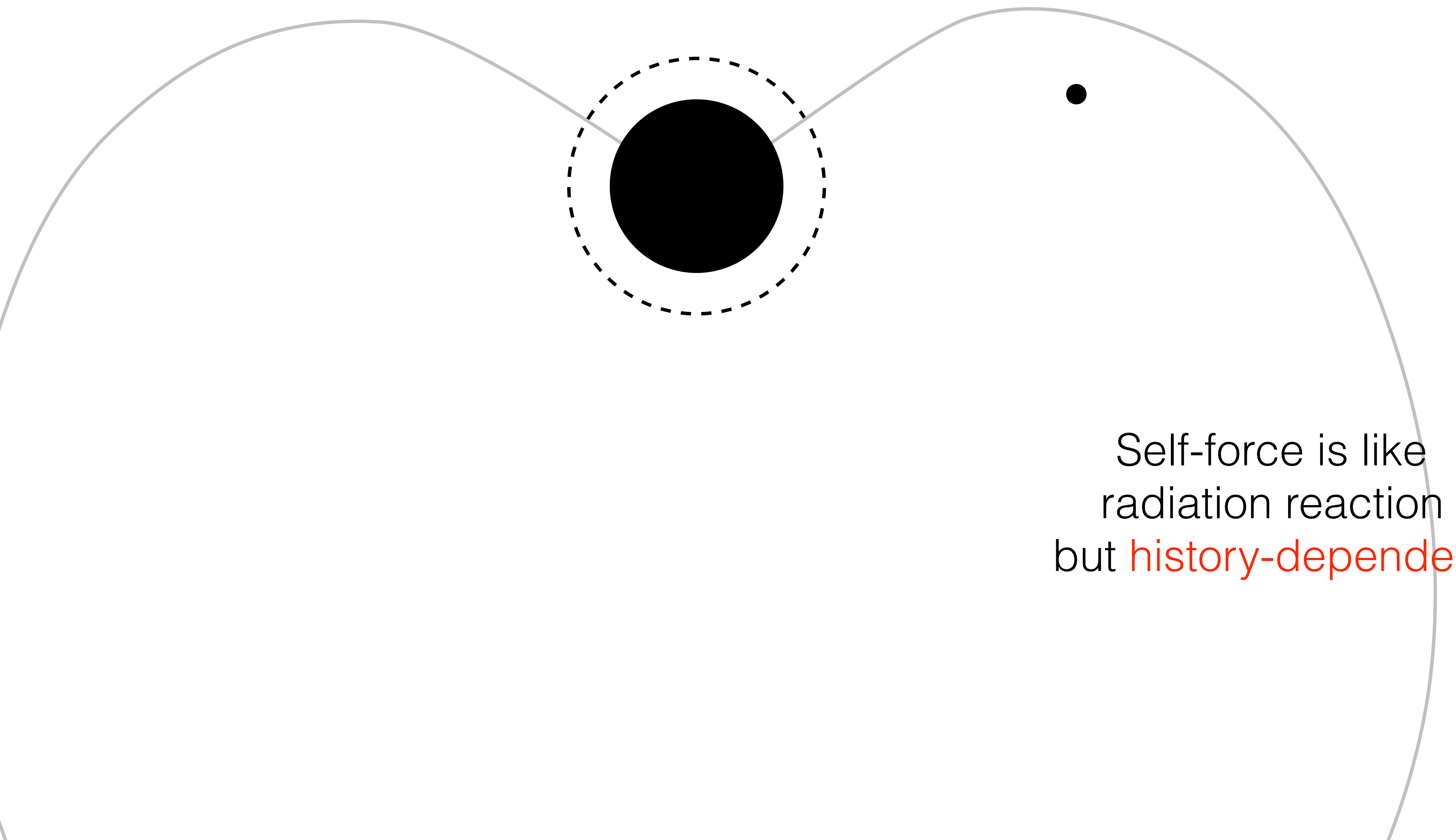
Self-force is like
radiation reaction
but history-dependent

What is self-force?



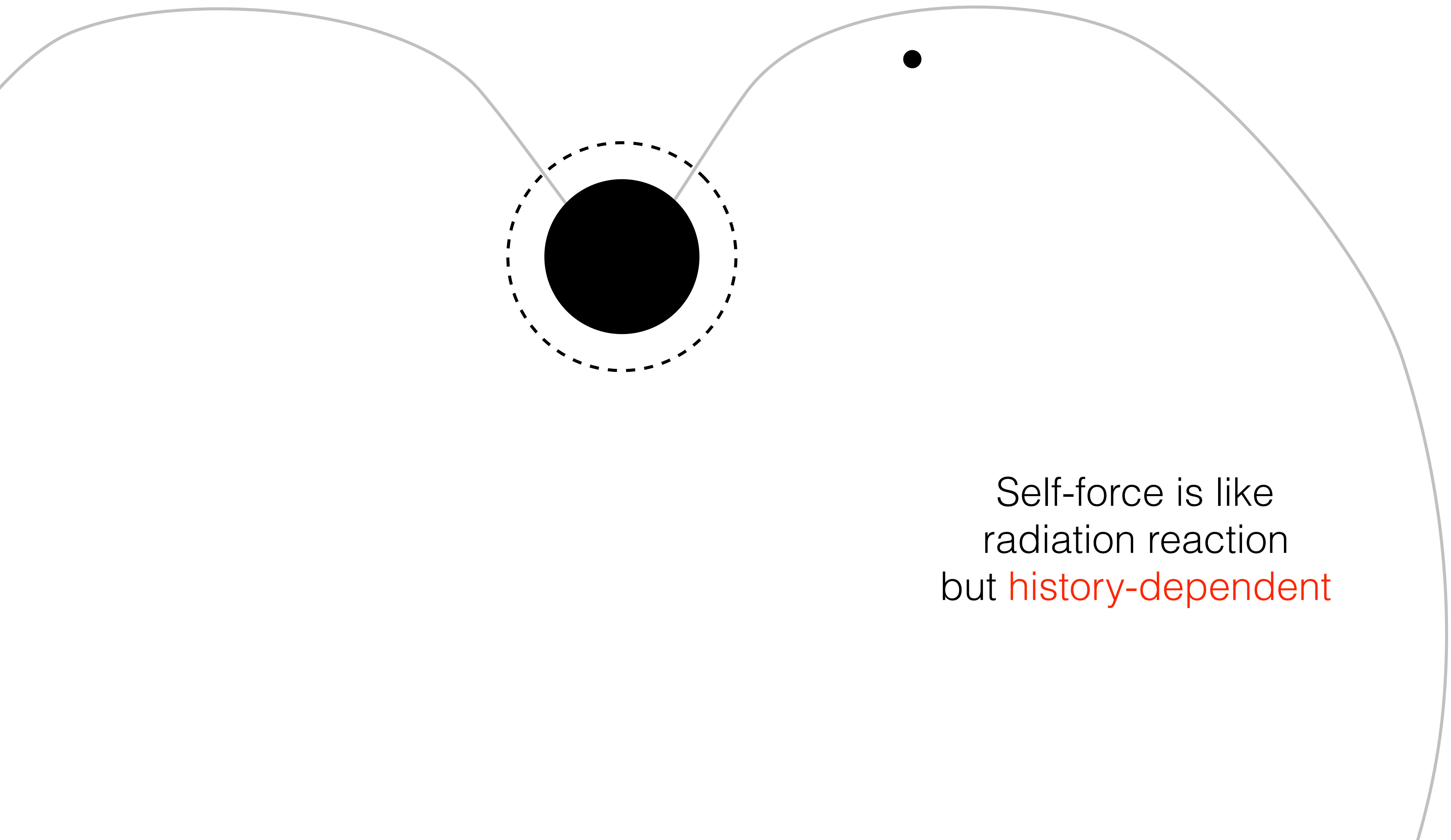
Self-force is like
radiation reaction
but history-dependent

What is self-force?



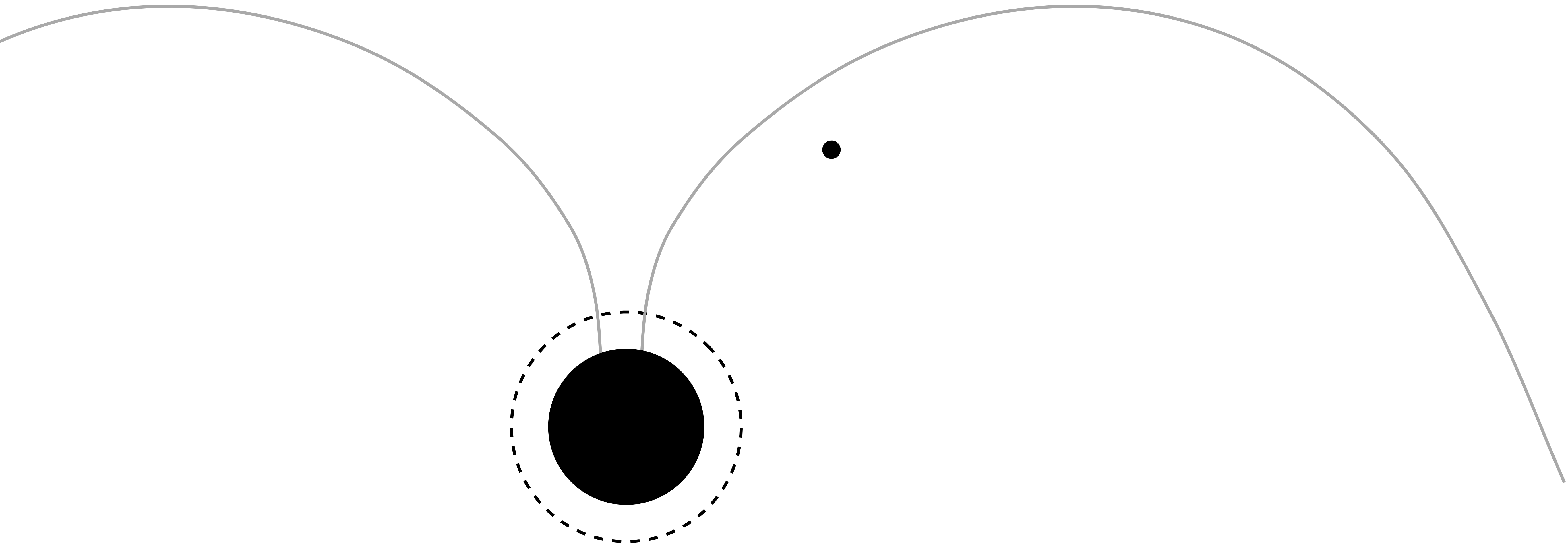
Self-force is like
radiation reaction
but history-dependent

What is self-force?



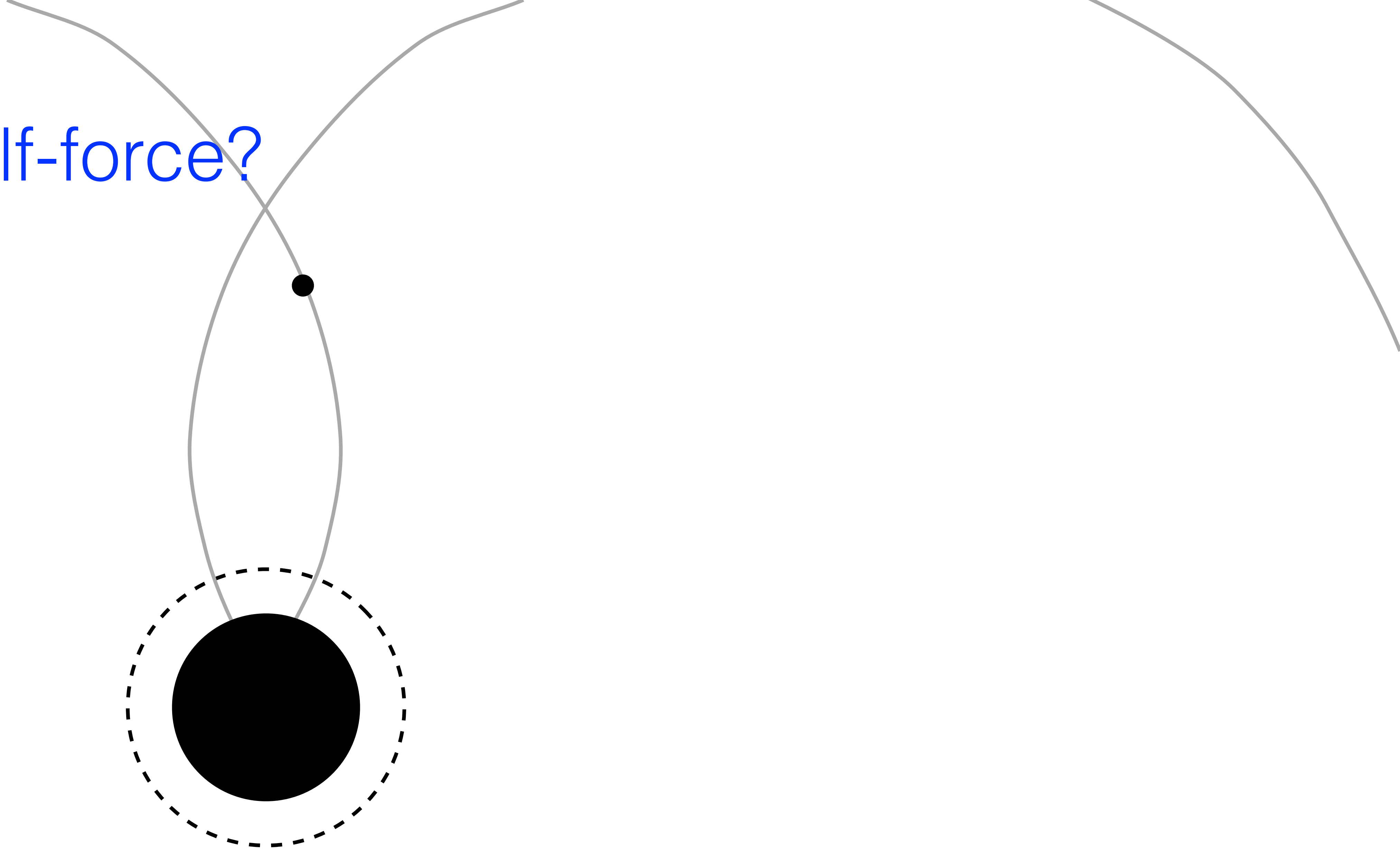
Self-force is like
radiation reaction
but history-dependent

What is self-force?



Self-force is like
radiation reaction
but history-dependent

What is self-force?



Self-force is like
radiation reaction
but history-dependent

Self-force

Gravitational self-force is described by the MiSaTaQuWa equation

Mino, Sasaki, Tanaka (96); Quinn, Wald (96)

$$a^\mu = \frac{m}{2} P^{\mu\nu\alpha\beta} \lim_{\epsilon \to 0^+} \int_{-\infty}^{\tau-\epsilon} d\tau' \, \nabla_\nu G^{\mathrm{ret}}_{\alpha\beta\gamma'\delta'}(z^\mu(\tau), z^{\mu'}(\tau')) u^{\gamma'} u^{\delta'}$$

Self-force

Gravitational self-force is described by the <u>MiSaTaQuWa equation</u>

Mino, Sasaki, Tanaka (96); Quinn, Wald (96)

$$a^\mu = \frac{m}{2} P^{\mu\nu\alpha\beta} \lim_{\epsilon\to 0^+} \int_{-\infty}^{\tau-\epsilon} d\tau' \, \nabla_\nu G^{\mathrm{ret}}_{\alpha\beta\gamma'\delta'}(z^\mu(\tau), z^{\mu'}(\tau')) u^{\gamma'} u^{\delta'}$$

Scalar self-force is described by the <u>Quinn equation</u>

Quinn (00)

$$a^\mu = \frac{q^2}{12\pi m} P^{\mu\nu} \frac{Da_\nu}{d\tau} + \frac{q^2}{m} P^{\mu\nu} \lim_{\epsilon\to 0^+} \int_{-\infty}^{\tau-\epsilon} d\tau' \, (a^\mu + P^{\mu\nu}\nabla_\nu) G^{\mathrm{ret}}(z^\mu(\tau), z^{\mu'}(\tau'))$$

Self-force

Gravitational self-force is described by the MiSaTaQuWa equation
Mino, Sasaki, Tanaka (96); Quinn, Wald (96)

$$a^{\mu} = \frac{m}{2} P^{\mu\nu\alpha\beta} \lim_{\epsilon \to 0^+} \int_{-\infty}^{\tau-\epsilon} d\tau' \, \nabla_{\nu} G^{\text{ret}}_{\alpha\beta\gamma'\delta'}(z^{\mu}(\tau), z^{\mu'}(\tau')) u^{\gamma'} u^{\delta'}$$

Scalar self-force is described by the Quinn equation
Quinn (00)

$$a^{\mu} = \frac{q^2}{12\pi m} P^{\mu\nu} \frac{Da_{\nu}}{d\tau} + \frac{q^2}{m} P^{\mu\nu} \lim_{\epsilon \to 0^+} \int_{-\infty}^{\tau-\epsilon} d\tau' \, (a^{\mu} + P^{\mu\nu} \nabla_{\nu}) G^{\text{ret}}(z^{\mu}(\tau), z^{\mu'}(\tau'))$$

The Green's function is central to understanding and describing self-force effects

$$\Box_x G(x, x') = \frac{\delta^4(x - x')}{\sqrt{-g(x)}} \quad \Longrightarrow \quad \phi(x) = \int d^4x' \, \sqrt{-g(x')} G(x, x') J(x')$$

- x is field point; x' is source/base point; G is a biscalar/bitensor

Green's functions

Green's functions

What are the advantages of using Green's functions?

- Compute only once for every source
- Nearly all physical quantities of interest are calculated via convolution integrals
- Arbitrary motion for self-force *Wardell, CRG et al (14)*
- Geometric interpretation *Zenginoglu & CRG (12), Wardell, CRG et al (14)*
- Higher-order self-force
- Self-consistent (higher-order) self-forced evolution
- Self-consistent inspiral waveforms
- Arguably straightforward to implement once known

(show movie)

Credit: B. Wardell

Numerical Green's functions

$$\Box_x G(x,x') = \frac{\delta^4(x-x')}{\sqrt{-g(x)}} \quad \longrightarrow$$ Narrow Gaussian

Zenginoglu & CRG (12)

Numerical Green's functions

$$\Box_x G(x,x') = \frac{\delta^4(x-x')}{\sqrt{-g(x)}} \quad \longrightarrow$$ Narrow Gaussian

Zenginoglu & CRG (12)

Solve a (symmetry-reduced) homogeneous wave equation with narrow Gaussian in initial data

Wardell, CRG et al (14)

Numerical Green's functions

$$\Box_x G(x,x') = \frac{\delta^4(x-x')}{\sqrt{-g(x)}} \quad \longrightarrow$$ Narrow Gaussian

Zenginoglu & CRG (12)

Solve a (symmetry-reduced) homogeneous wave equation with narrow Gaussian in initial data

Wardell, CRG et al (14)

Numerical Green's functions are globally valid approximations but utilizing analytic approximations at early and late times is extremely helpful for self-force calculations

- Quasi-local expansions *Ottewill & Wardell (08); Wardell's thesis*
- Pade approximants *Casals et al (09)*
- Method of matched expansions *Anderson & Wiseman (05); Casals et al (13)*

Numerical Green's functions

$$\Box_x G(x,x') = \frac{\delta^4(x-x')}{\sqrt{-g(x)}} \quad \longrightarrow$$ Narrow Gaussian
Zenginoglu & CRG (12)

Solve a (symmetry-reduced) homogeneous wave equation with narrow Gaussian in initial data
Wardell, CRG et al (14)

Numerical Green's functions are globally valid approximations but utilizing analytic approximations at early and late times is extremely helpful for self-force calculations

- Quasi-local expansions *Ottewill & Wardell (08); Wardell's thesis*
- Pade approximants *Casals et al (09)*
- Method of matched expansions *Anderson & Wiseman (05); Casals et al (13)*

When these analytical approximations (e.g., in Schwarzschild) are available we use numerical Green's functions for intermediate times

What are the disadvantages of using Green's functions?

- Analytically, very difficult to calculate in curved spacetimes
  Casals, Nolan (16)
- Numerically, computationally expensive
- Large data sets (*G* depends on two space-time points!)

What are the disadvantages of using Green's functions?

- Analytically, very difficult to calculate in curved spacetimes
  Casals, Nolan (16)
- Numerically, computationally expensive
- Large data sets (*G* depends on two space-time points!)

Single mode	~300MB (~2 GB)
101 modes	~30GB (~200 GB)
Kerr	~(300MB)*10*10*10^4 = 300 GB (~2TB)
Gravity	~3TB (~20TB)

What are the disadvantages of using Green's functions?

- Analytically, very difficult to calculate in curved spacetimes
  Casals, Nolan (16)
- Numerically, computationally expensive
- Large data sets (*G* depends on two space-time points!)

Single mode	~300MB (~2 GB)
101 modes	~30GB (~200 GB)
Kerr	~(300MB)*10*10*10^4 = 300 GB (~2TB)
Gravity	~3TB (~20TB)

Goal:

Find a way for Green's functions to be efficient and accurate to use for practical self-force and related computations.

The goal

To quickly predict accurate solutions to the Green's function wave equation that are otherwise too slow and too large for practical use

The goal

To quickly predict accurate solutions to the Green's function wave equation that are otherwise too slow and too large for practical use

The method

Use *reduced-order modeling* techniques to reduce large sets of pre-computed Green's function data to its essential components

The goal

To quickly predict accurate solutions to the Green's function wave equation that are otherwise too slow and too large for practical use

The method

Use *reduced-order modeling* techniques to reduce large sets of pre-computed Green's function data to its essential components

- *Reduced basis (compression in source & field points)*
- *Empirical interpolation (compression in time)*
- *Fits to predict new Green's function values*

Field, CRG et al (14)

The goal

To quickly predict accurate solutions to the Green's function wave equation that are otherwise too slow and too large for practical use

The method

Use *reduced-order modeling* techniques to reduce large sets of pre-computed Green's function data to its essential components

- *Reduced basis (compression in source & field points)*
- *Empirical interpolation (compression in time)*
- *Fits to predict new Green's function values*

Field, CRG et al (14)

The result

An accurate *surrogate model* to generate new Green's function data on demand

Surrogate models for gravitational waveforms have been built successfully for:

- Non-spinning Effective One-Body (*EOBNRv2*)
  Field, CRG, et al PRX (14)
- Spin-aligned Effective One-Body (*SEOBNRv2*)
  Purrer (15)
- Non-spinning Numerical Relativity (*SpEC*)
  Blackman, Field, CRG et al PRL (15)
- 4d precession, Numerical Relativity (*SpEC*) *(see J. Blackman's talk)*
  (in prep)
- Tidal Effective One-Body *(see S. Bernuzzi's talk)*
  (in prep)

Surrogate models for gravitational waveforms have been built successfully for:

- Non-spinning Effective One-Body (*EOBNRv2*)
  Field, CRG, et al PRX (14)
- Spin-aligned Effective One-Body (*SEOBNRv2*)
  Purrer (15)
- Non-spinning Numerical Relativity (*SpEC*)
  Blackman, Field, CRG et al PRL (15)
- 4d precession, Numerical Relativity (*SpEC*) *(see J. Blackman's talk)*
  (in prep)
- Tidal Effective One-Body *(see S. Bernuzzi's talk)*
  (in prep)

However, some steps for building a Green's function surrogate are necessarily different than for waveforms

- Provides one with dynamics, field content, and waveforms
- Source and field points are time-dependent for worldline convolutions

Surrogate building: Initial stuff

see Wardell, CRG et al (14) for details of numerical implementation

- Coordinates: (t, r_*, θ, ϕ)

Surrogate building: Initial stuff

see Wardell, CRG et al (14)
for details of numerical implementation

- Coordinates: (t, r_*, θ, ϕ)

- Mode decomposition means building a surrogate for each mode

$$G(x^\alpha, x'^\alpha) = \frac{1}{rr'} \sum_{\ell=0}^{\infty} (2\ell+1) P_\ell(\cos\theta) G_\ell(t - t'; r_*, r'_*)$$

Surrogate building: Initial stuff

see Wardell, CRG et al (14)
for details of numerical implementation

- Coordinates: (t, r_*, θ, ϕ)

- Mode decomposition means building a surrogate for each mode

$$G(x^\alpha, x'^\alpha) = \frac{1}{rr'} \sum_{\ell=0}^{\infty} (2\ell+1) P_\ell(\cos\theta) G_\ell(t - t'; r_*, r'_*)$$

- Treat radial coordinates of source and field points as "parameters" for the model

$$\vec{\lambda} = (r_*, r'_*)$$

Surrogate building: Initial stuff

see Wardell, CRG et al (14) for details of numerical implementation

- Coordinates: (t, r_*, θ, ϕ)

- Mode decomposition means building a surrogate for each mode

$$G(x^\alpha, x'^\alpha) = \frac{1}{rr'} \sum_{\ell=0}^{\infty} (2\ell+1) P_\ell(\cos\theta) G_\ell(t-t'; r_*, r'_*)$$

- Treat radial coordinates of source and field points as "parameters" for the model

$$\vec{\lambda} = (r_*, r'_*)$$

- Solutions stored to disk at increments

$$\Delta t = 0.1M$$

$$\Delta r_* = \Delta r'_* = 0.1M$$

Surrogate building: Initial stuff

see Wardell, CRG et al (14) for details of numerical implementation

- Coordinates: (t, r_*, θ, ϕ)

- Mode decomposition means building a surrogate for each mode

$$G(x^\alpha, x'^\alpha) = \frac{1}{rr'} \sum_{\ell=0}^{\infty} (2\ell + 1) P_\ell(\cos\theta) G_\ell(t - t'; r_*, r'_*)$$

- Treat radial coordinates of source and field points as "parameters" for the model

$$\vec{\lambda} = (r_*, r'_*)$$

- Solutions stored to disk at increments

$$\Delta t = 0.1M$$

$$\Delta r_* = \Delta r'_* = 0.1M$$

- ~30GB saved to disk

Surrogate building: Initial stuff

see Wardell, CRG et al (14) for details of numerical implementation

- Coordinates: (t, r_*, θ, ϕ)

- Mode decomposition means building a surrogate for each mode

$$G(x^\alpha, x'^\alpha) = \frac{1}{rr'} \sum_{\ell=0}^{\infty} (2\ell+1) P_\ell(\cos\theta) G_\ell(t-t'; r_*, r'_*)$$

- Treat radial coordinates of source and field points as "parameters" for the model

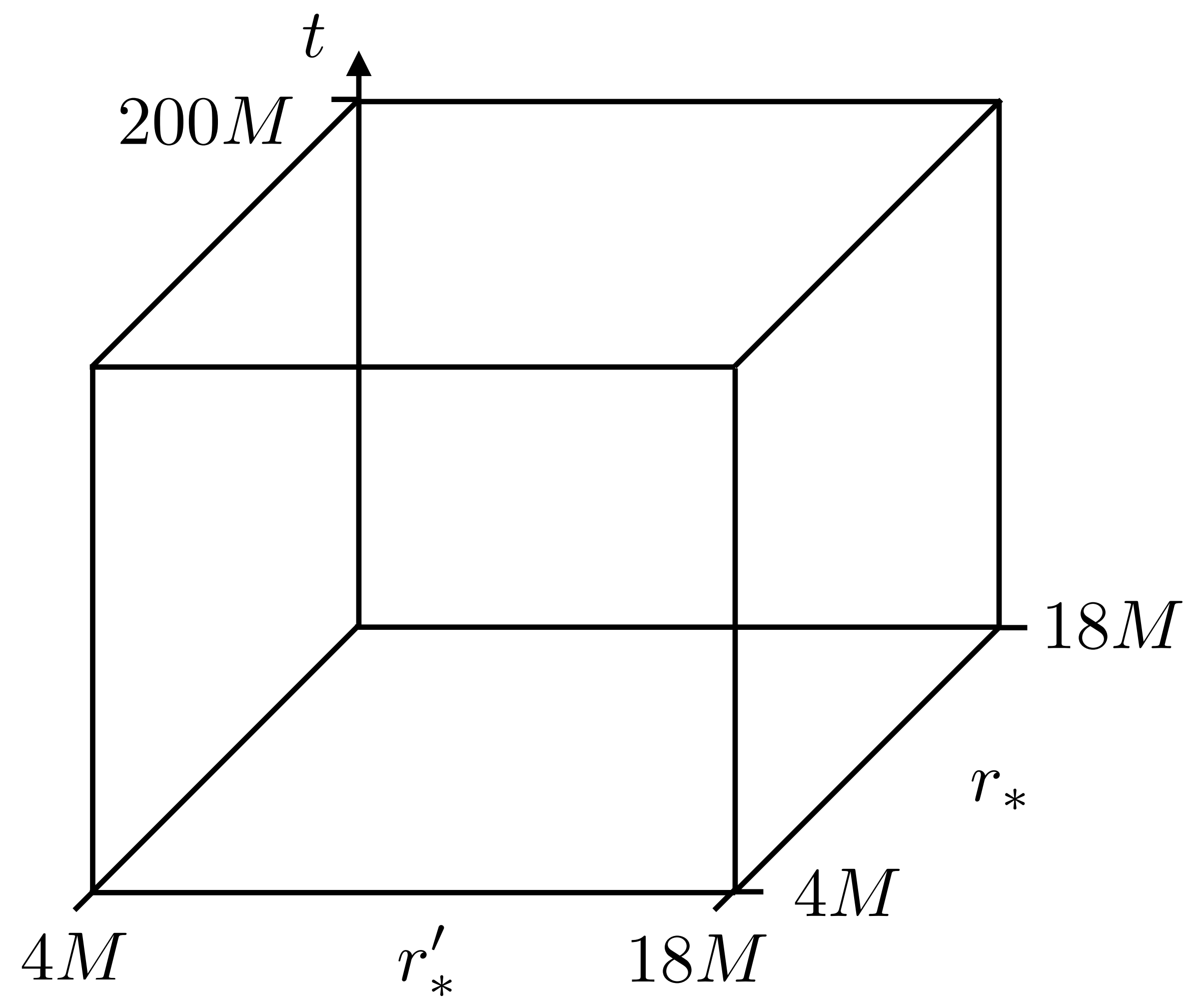
$$\vec{\lambda} = (r_*, r'_*)$$

- Solutions stored to disk at increments

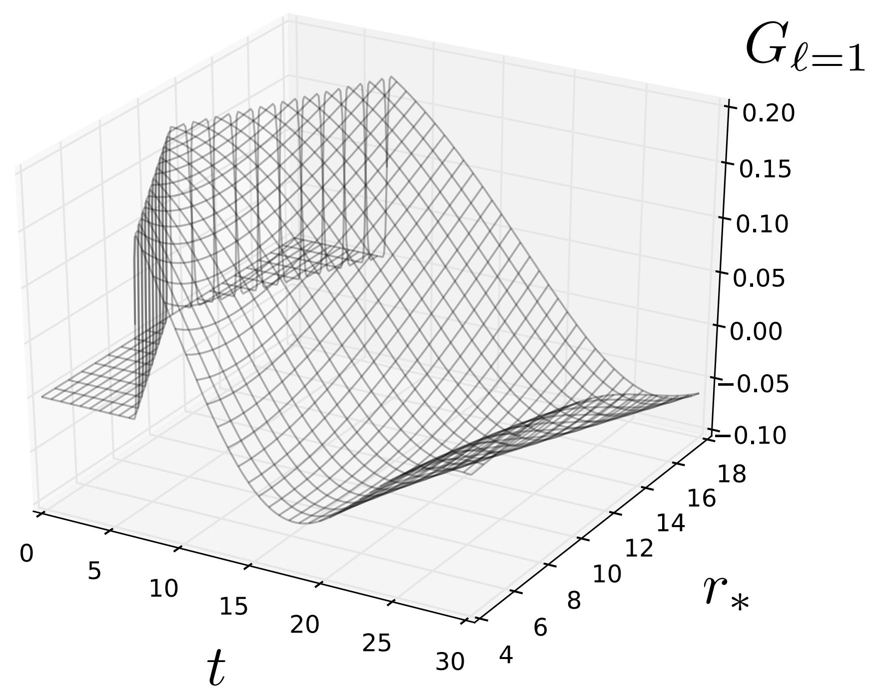
$$\Delta t = 0.1M$$

$$\Delta r_* = \Delta r'_* = 0.1M$$

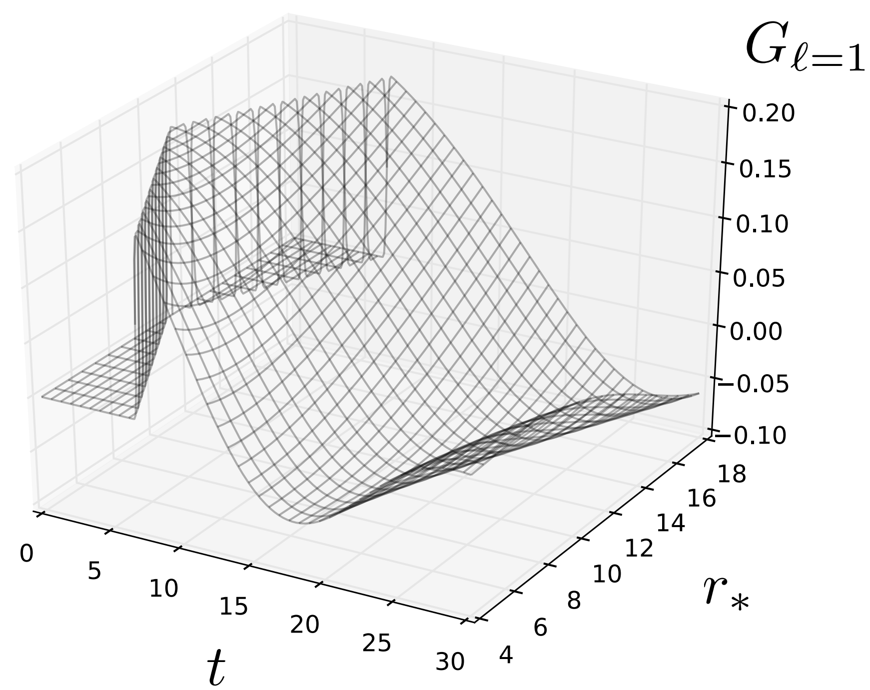
- ~30GB saved to disk



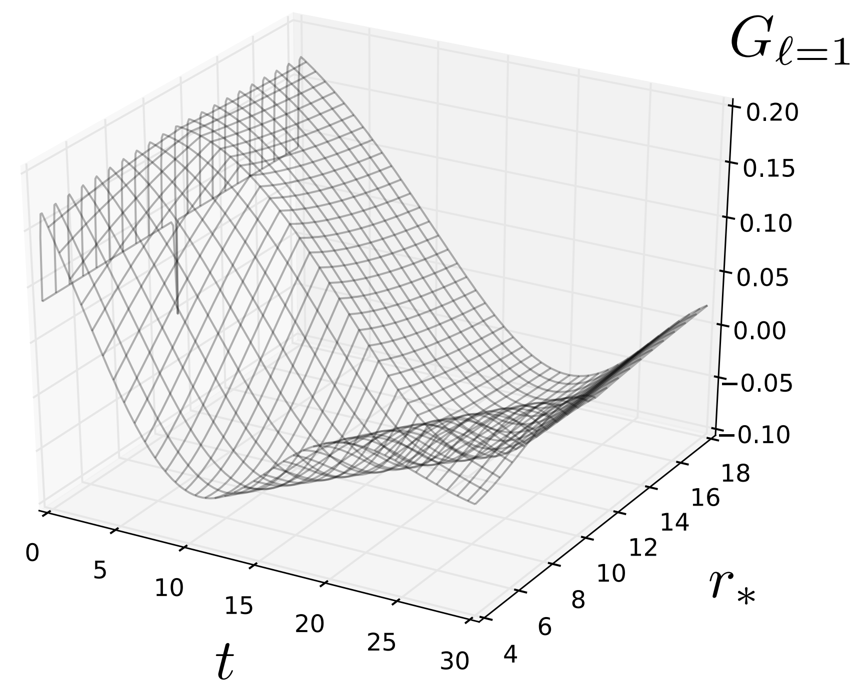
- Reduce known features by analytically time-shifting each series by light travel time from source point to field point, $t \to t - |r_* - r'_*|$



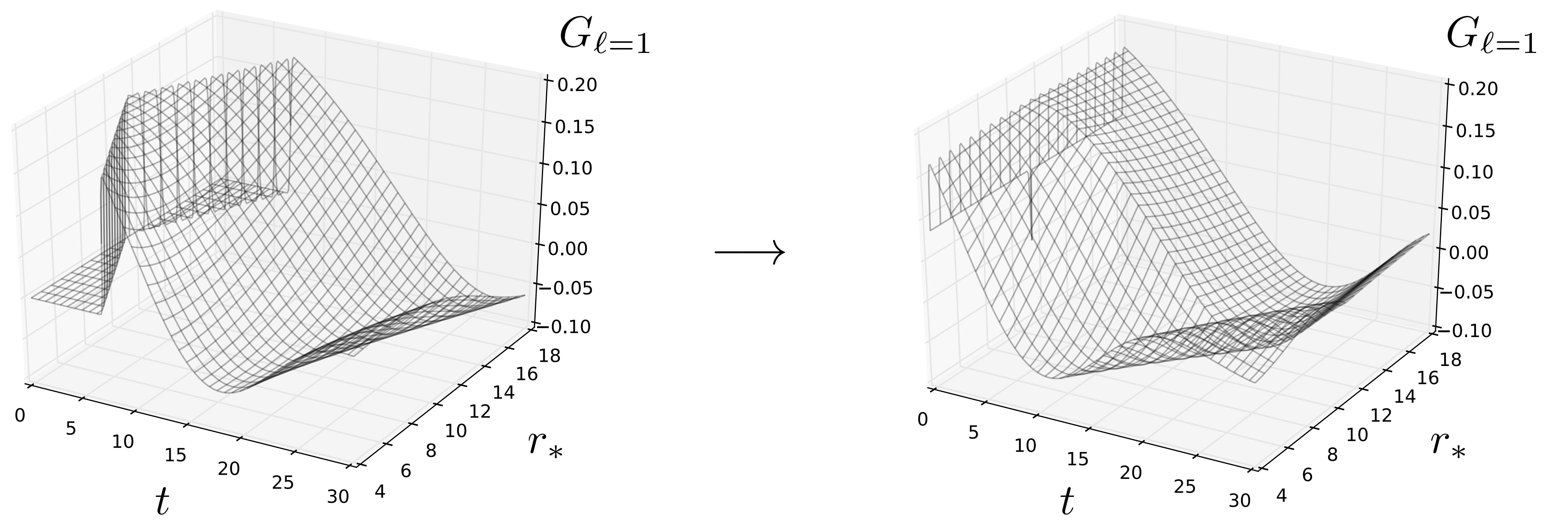
- Reduce known features by analytically time-shifting each series by light travel time from source point to field point, $t \to t - |r_* - r'_*|$



$\longrightarrow$



- Reduce known features by analytically time-shifting each series by light travel time from source point to field point, $t \to t - |r_* - r'_*|$



- In addition, to reduce high-frequency noise we introduce a smoothing factor *Wardell, CRG et al (14)*

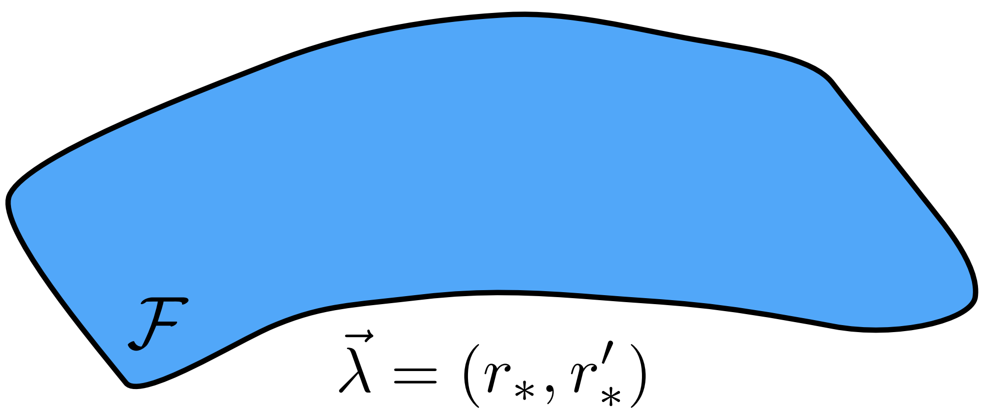
$$G(x^\alpha, x'^\alpha) \approx \frac{1}{rr'} \sum_{\ell=0}^{\ell_{\max}} P_\ell(\cos\theta)(2\ell+1) e^{-\ell^2/2\ell_{\rm cut}^2} G_\ell(t-t'; r_*, r'_*)$$

$\ell_{\max} = 100$ $\qquad\qquad$ $\ell_{\rm cut} = \ell_{\max}/5$

1) Reduced basis via greedy algorithm

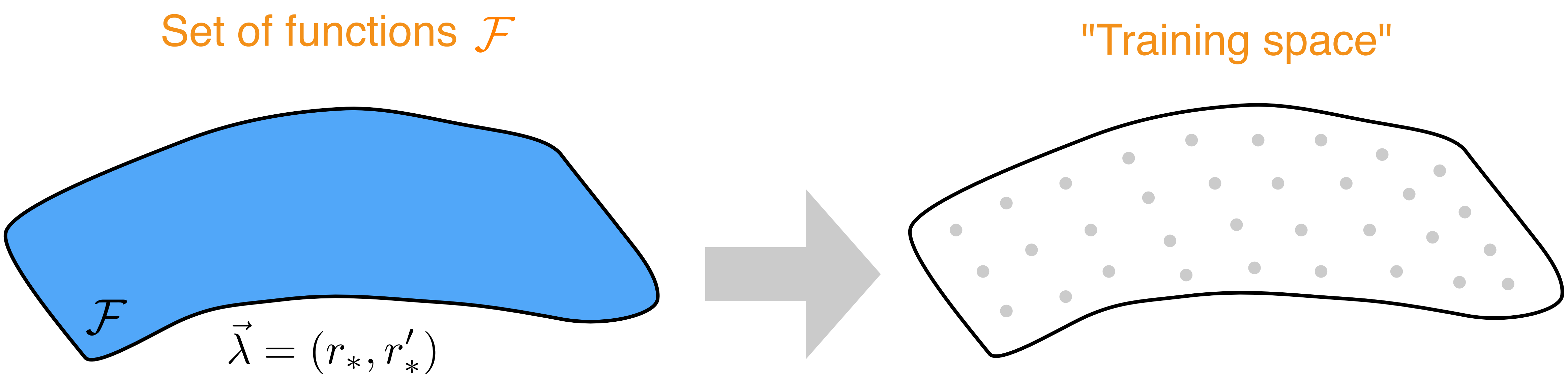
Can find a linear approximation space that is nearly optimal

Set of functions $\mathcal{F}$



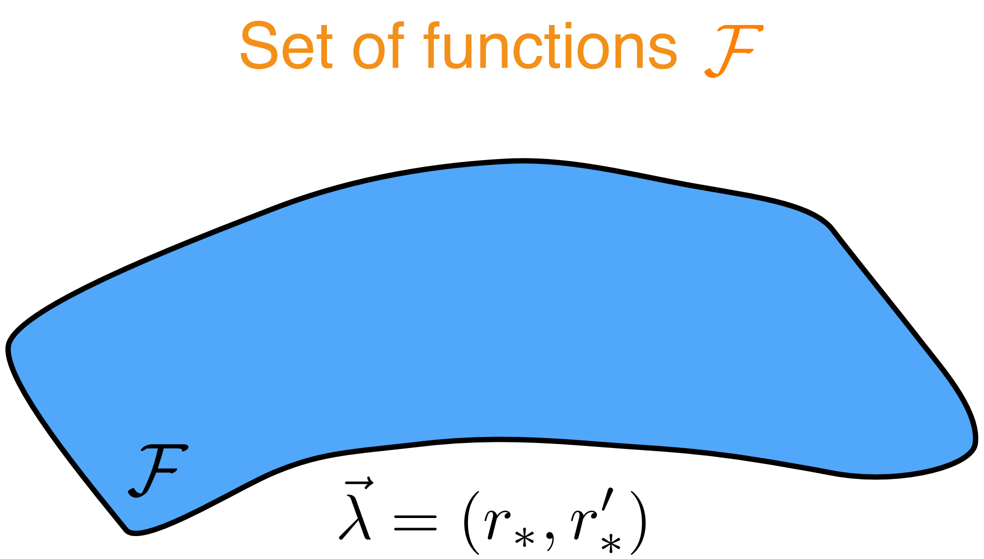
1) Reduced basis via greedy algorithm

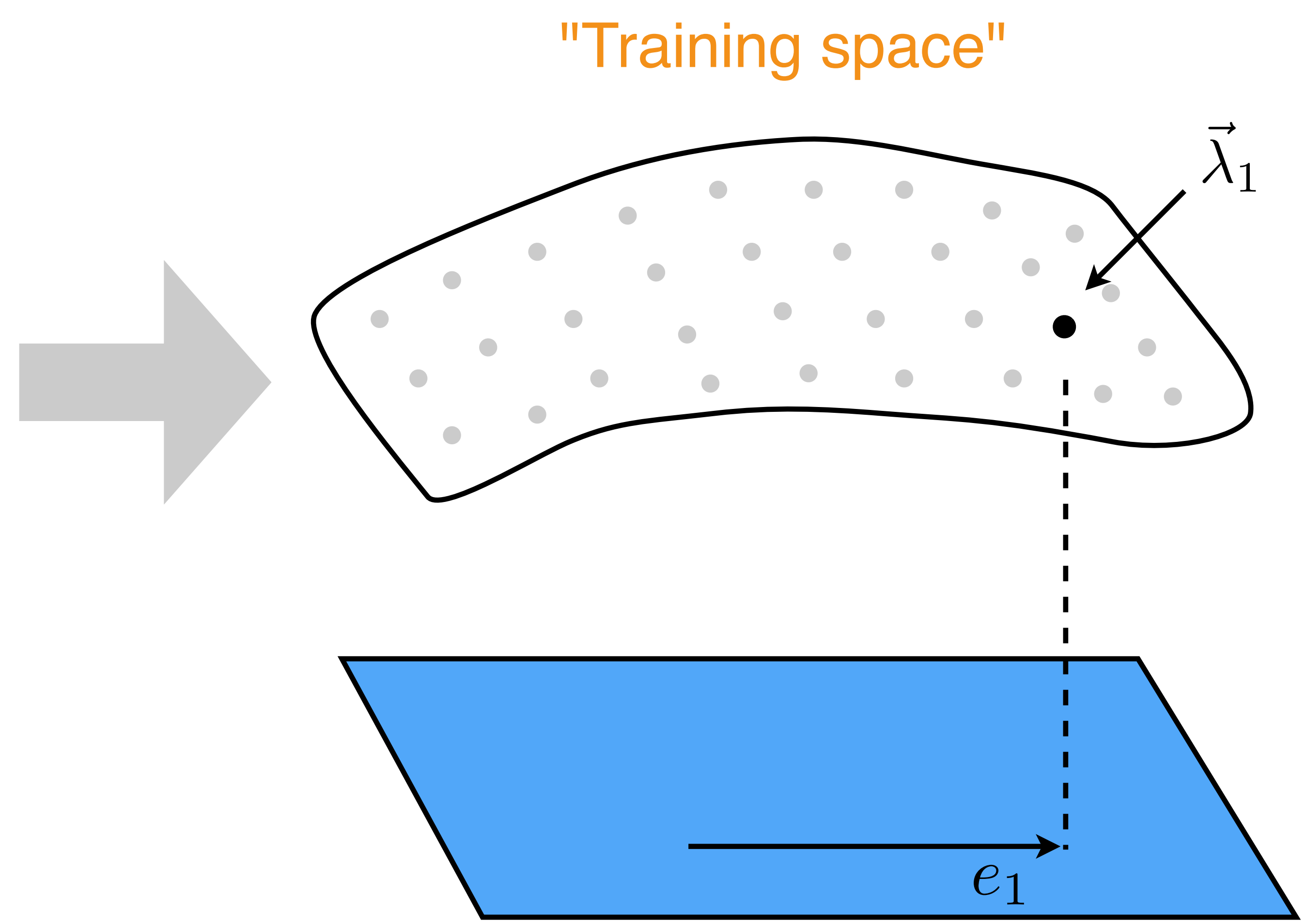
Can find a linear approximation space that is nearly optimal



1) Reduced basis via greedy algorithm

Can find a linear approximation space that is nearly optimal





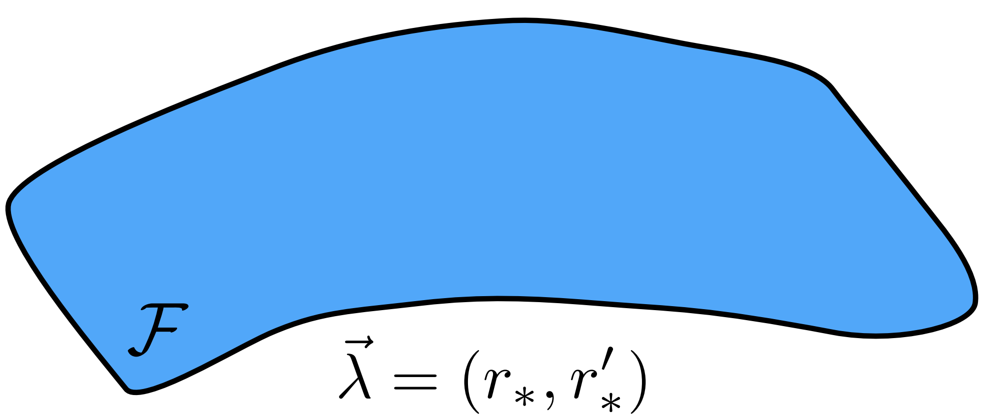
1) Choose any parameter,

$$e_1 = G_\ell(t; \vec{\lambda}_1), \; C_1 = \{e_1\}$$

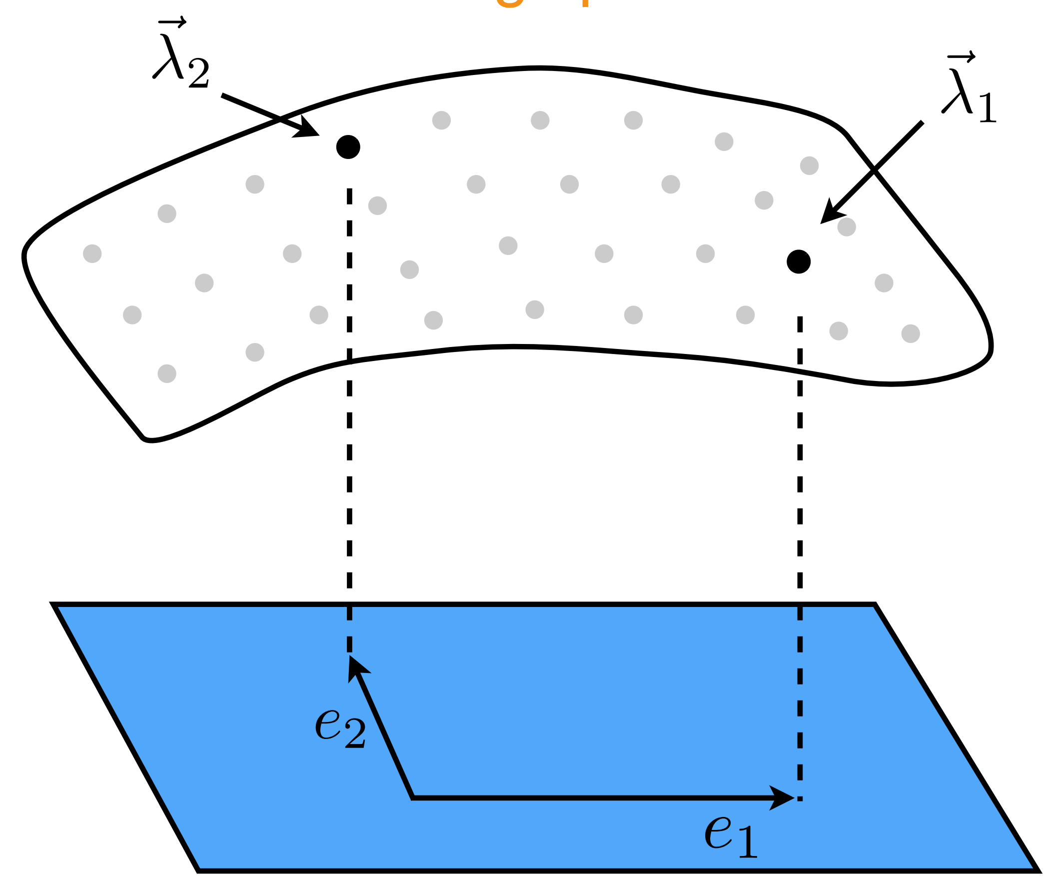
1) Reduced basis via greedy algorithm

Can find a linear approximation space that is nearly optimal

Set of functions $\mathcal{F}$



"Training space"



1) Choose any parameter,

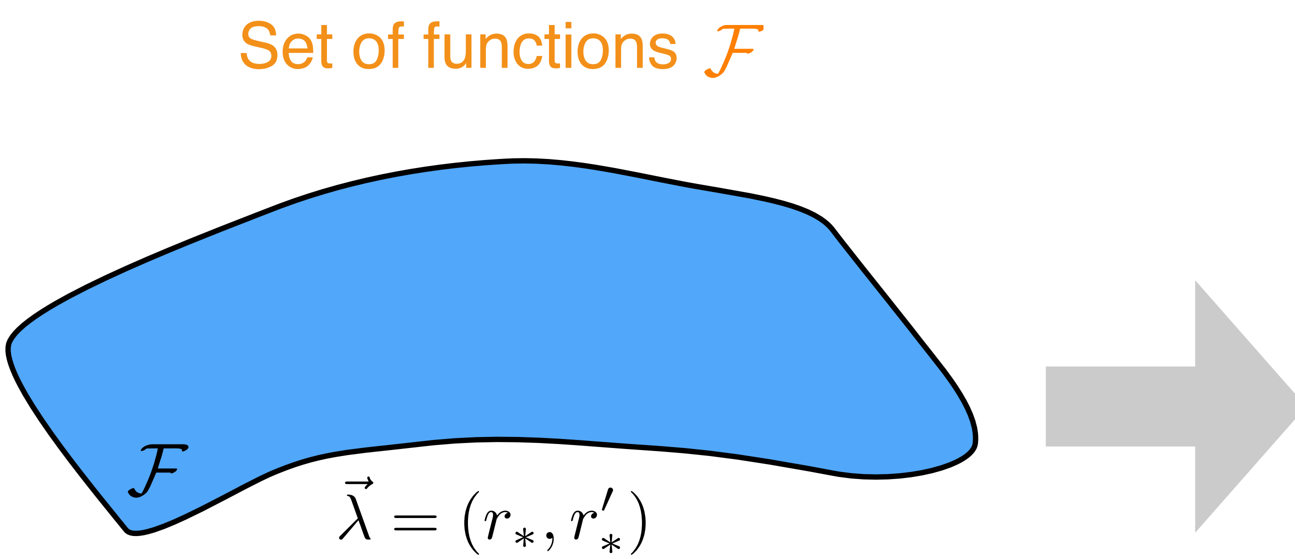
$$e_1 = G_\ell(t; \vec{\lambda}_1), \; C_1 = \{e_1\}$$

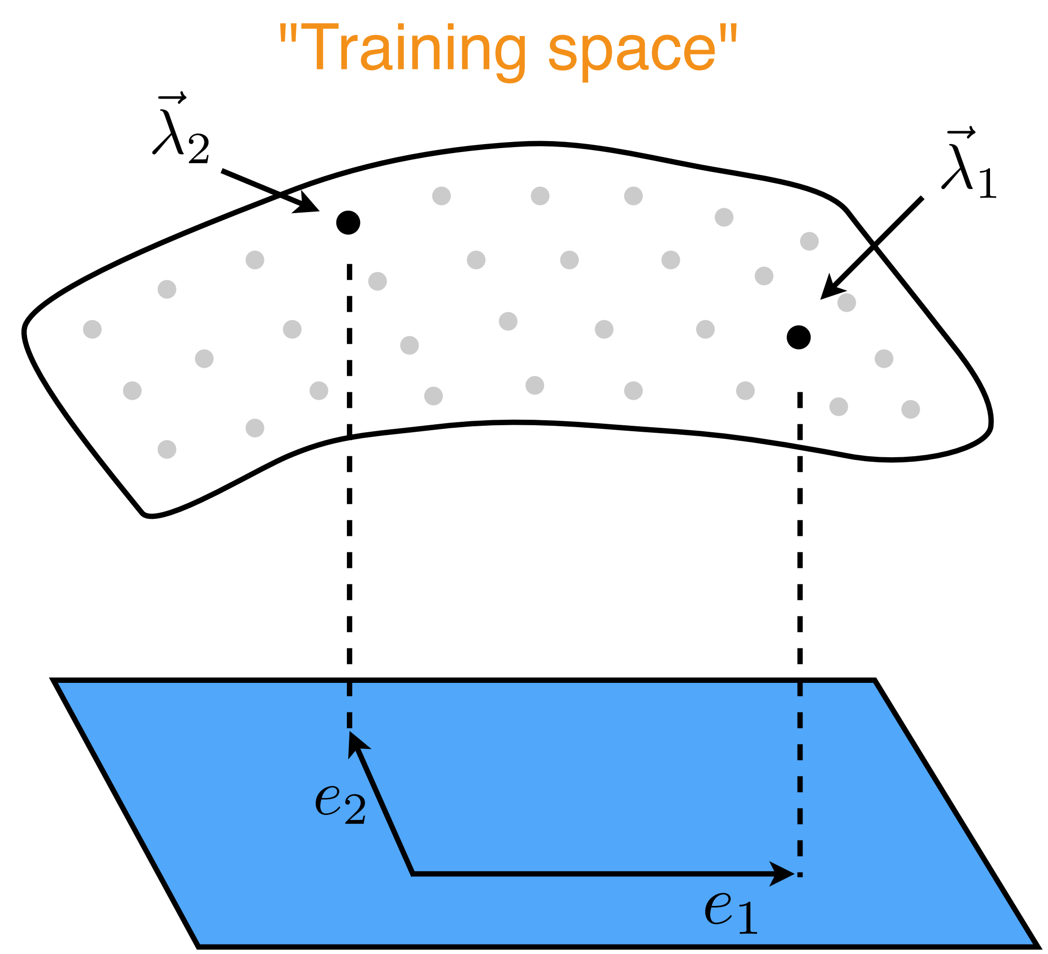
2) Greedy search - Find the parameter that maximizes:

$$\max_t \left| G_\ell(t; \vec{\lambda}) - P_1[G_\ell(t; \vec{\lambda})] \right|, \; P_1[\cdot] = e_1 \langle e_1, \cdot \rangle$$

1) Reduced basis via greedy algorithm

Can find a linear approximation space that is nearly optimal





1) Choose any parameter,

$$e_1 = G_\ell(t; \vec{\lambda}_1), \; C_1 = \{e_1\}$$

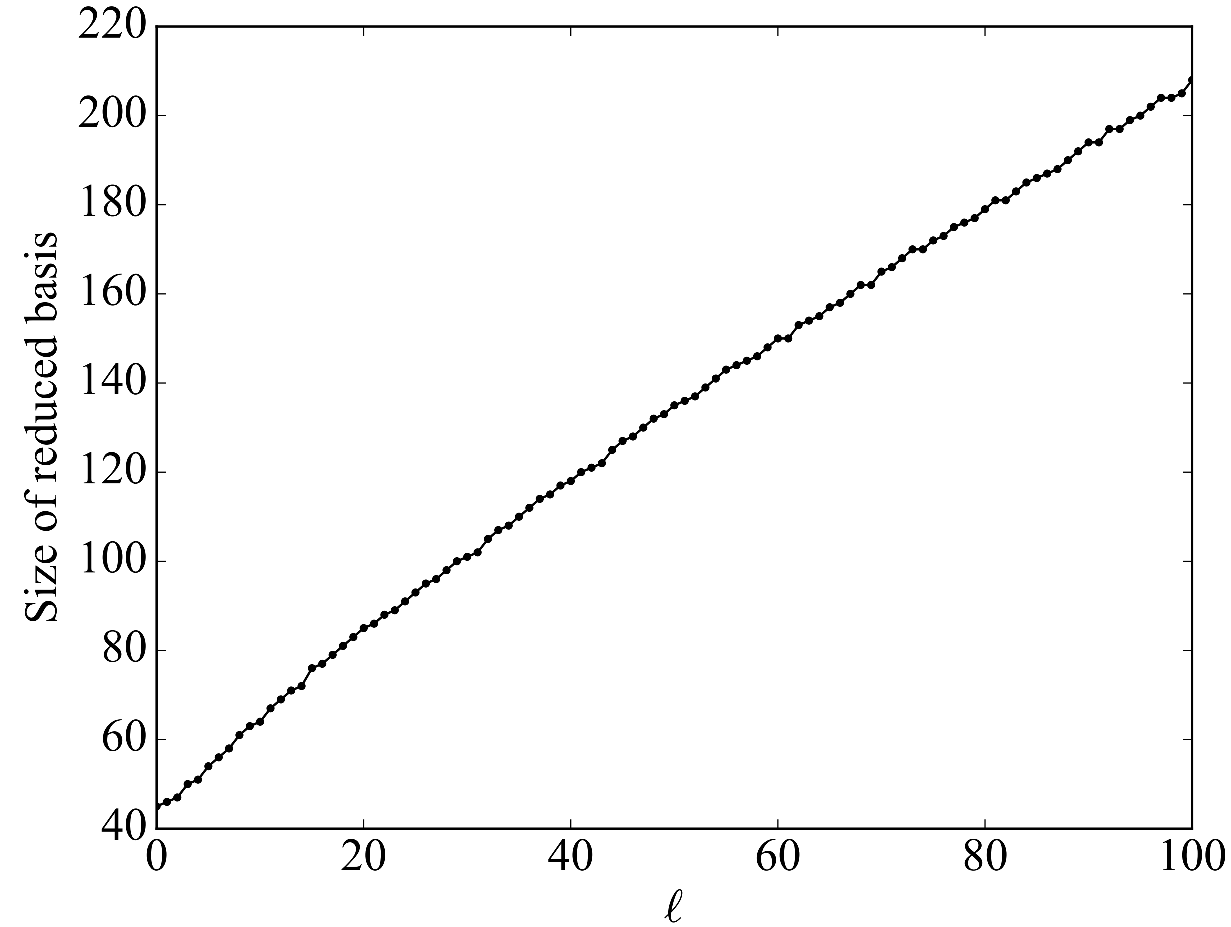
2) <u>Greedy search</u> - Find the parameter that maximizes:

$$\max_t \left| G_\ell(t; \vec{\lambda}) - P_1[G_\ell(t; \vec{\lambda})] \right|, \; P_1[\cdot] = e_1 \langle e_1, \cdot \rangle$$

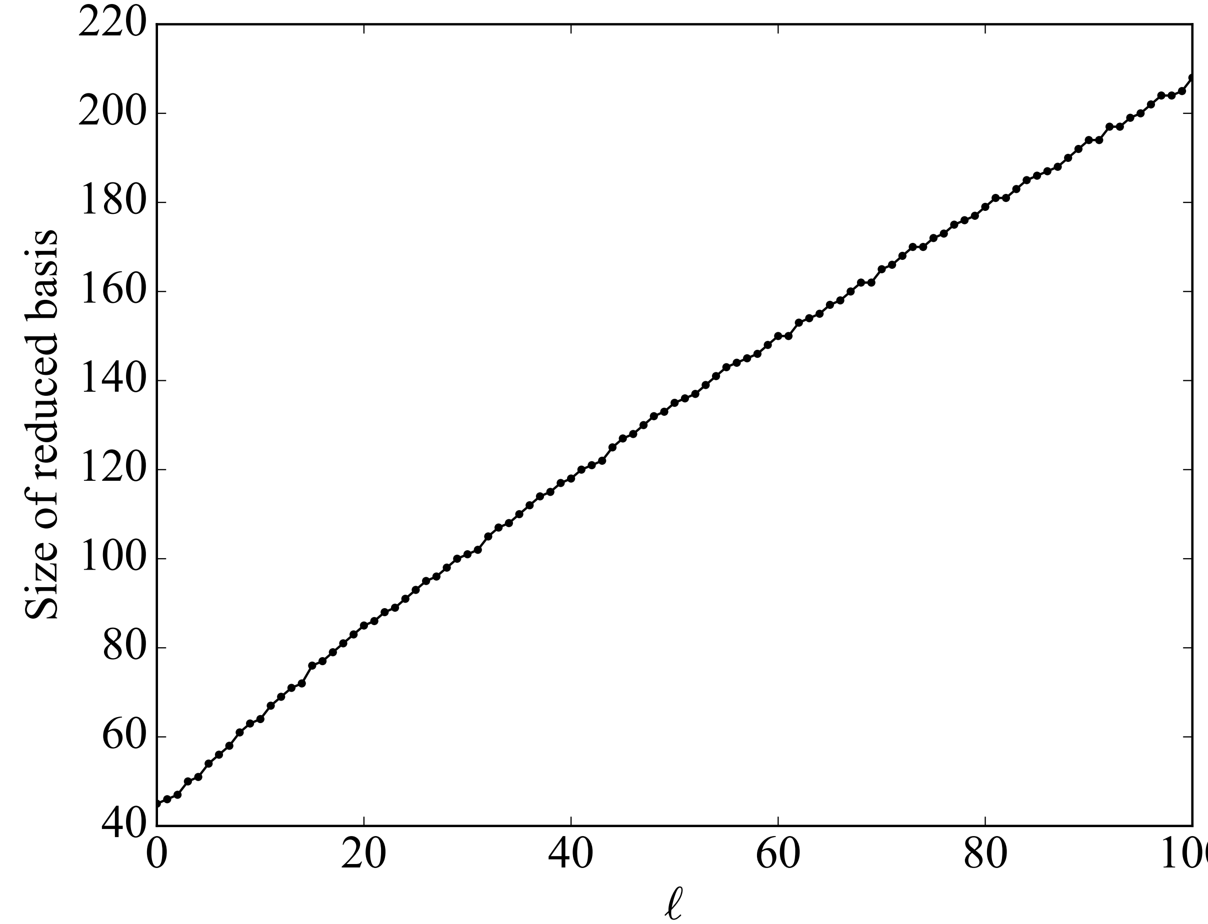
3) Orthogonalization to get basis vector e_2

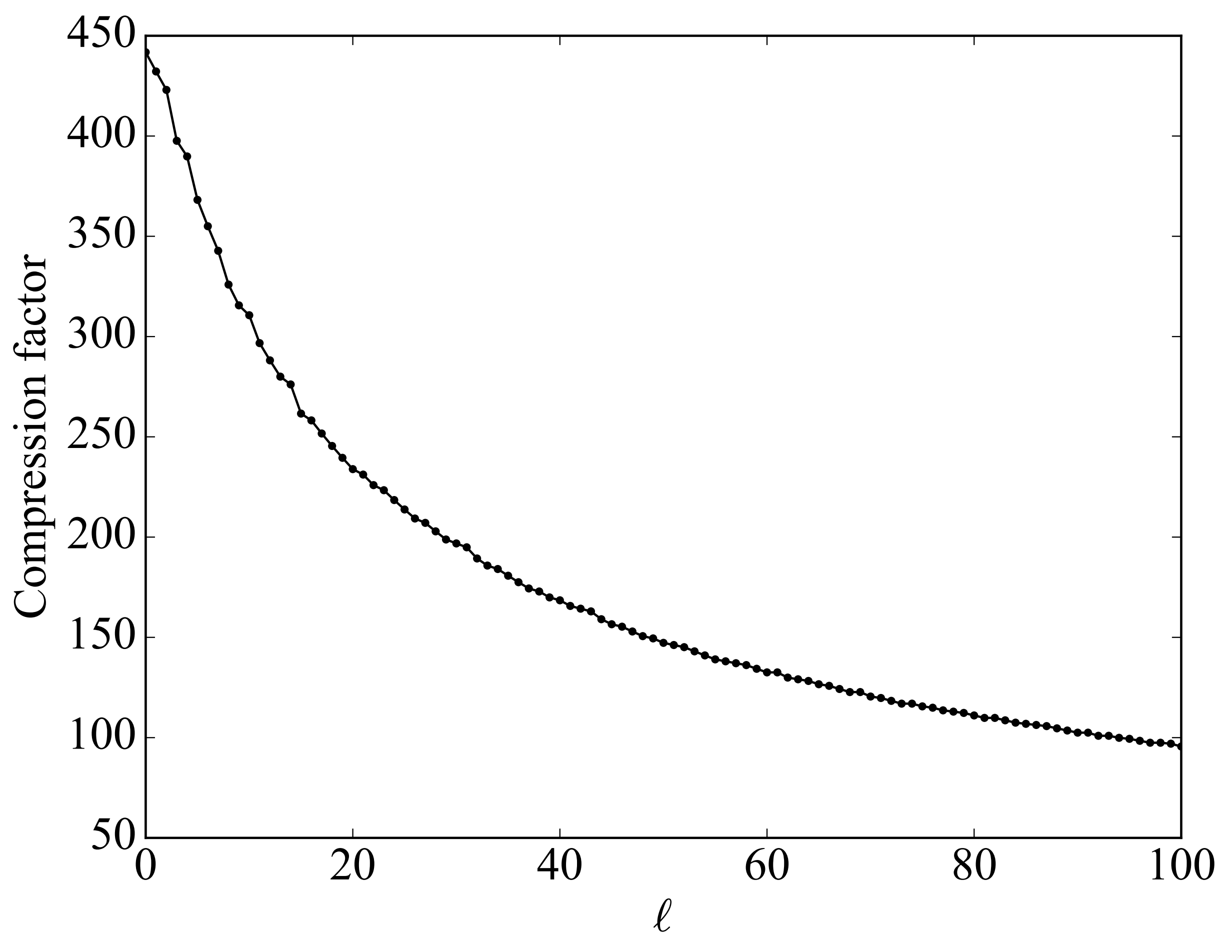
$$C_2 = \{e_1, e_2\}, \; C_1 \subset C_2$$

Basis size grows nearly linearly with mode number



Basis size grows nearly linearly with mode number



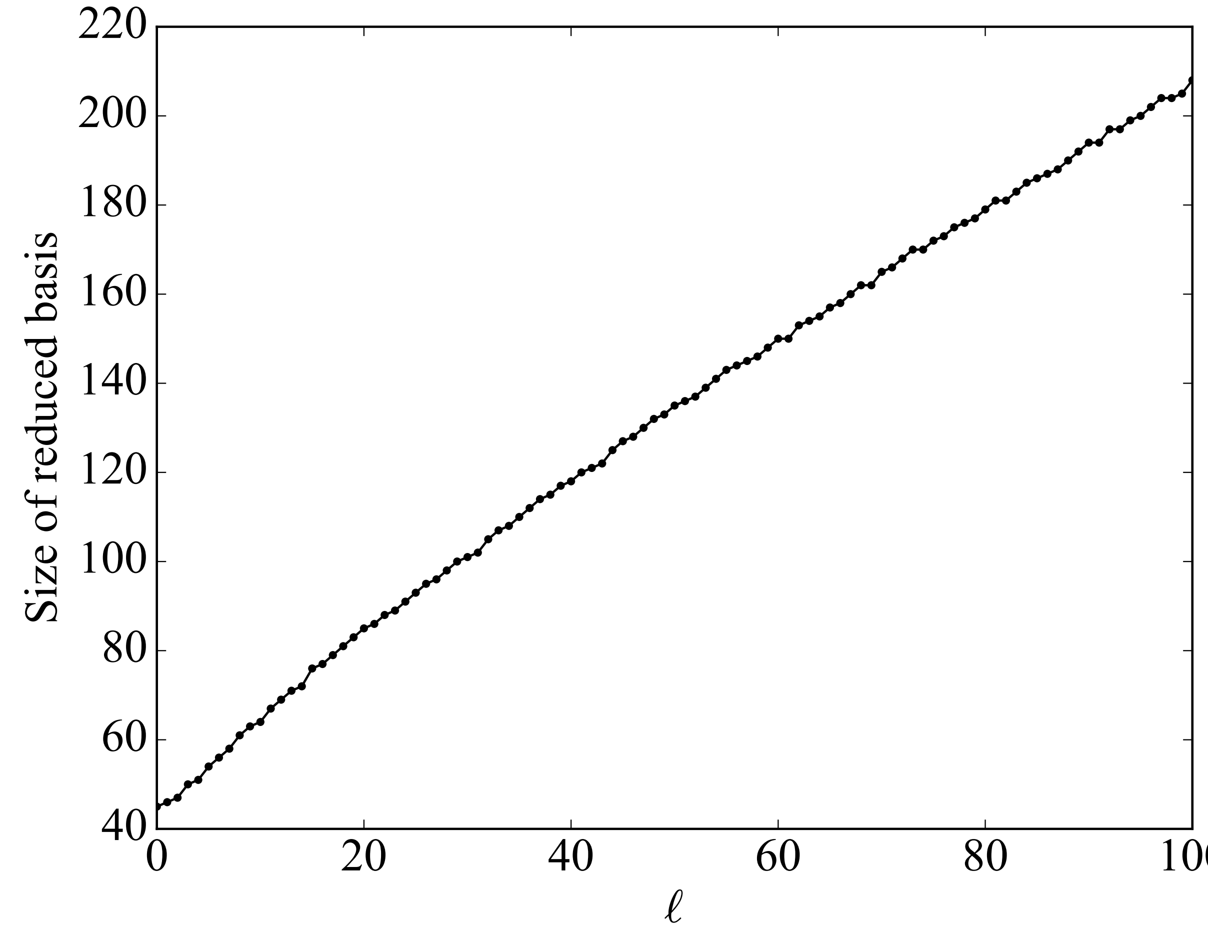


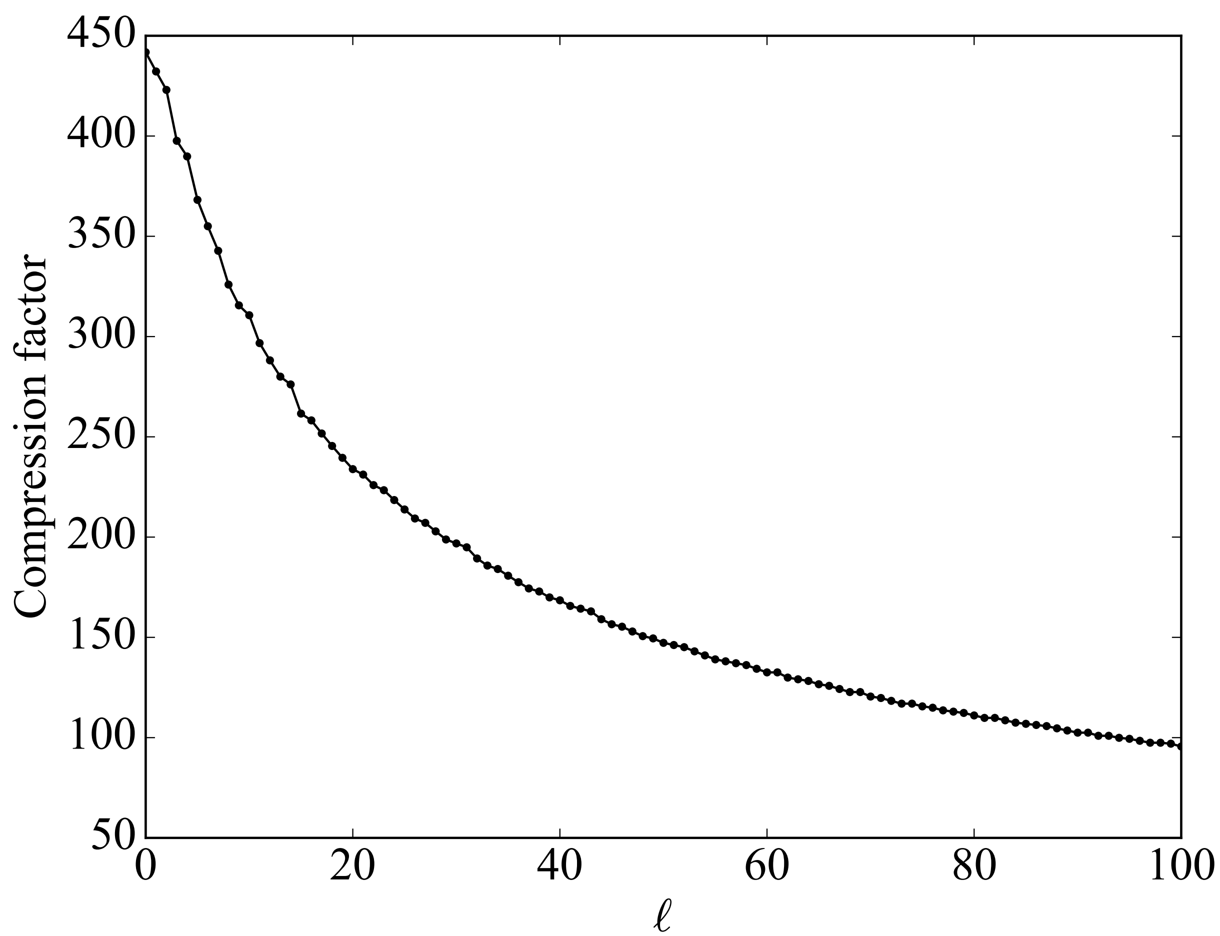
Total compression factor:

$$C_{\rm total} = (\ell_{\rm max} + 1) \left(\sum_{\ell=0}^{\ell_{\rm max}} \frac{1}{C_\ell} \right)^{-1}$$

$$\approx 151$$

Basis size grows nearly linearly with mode number





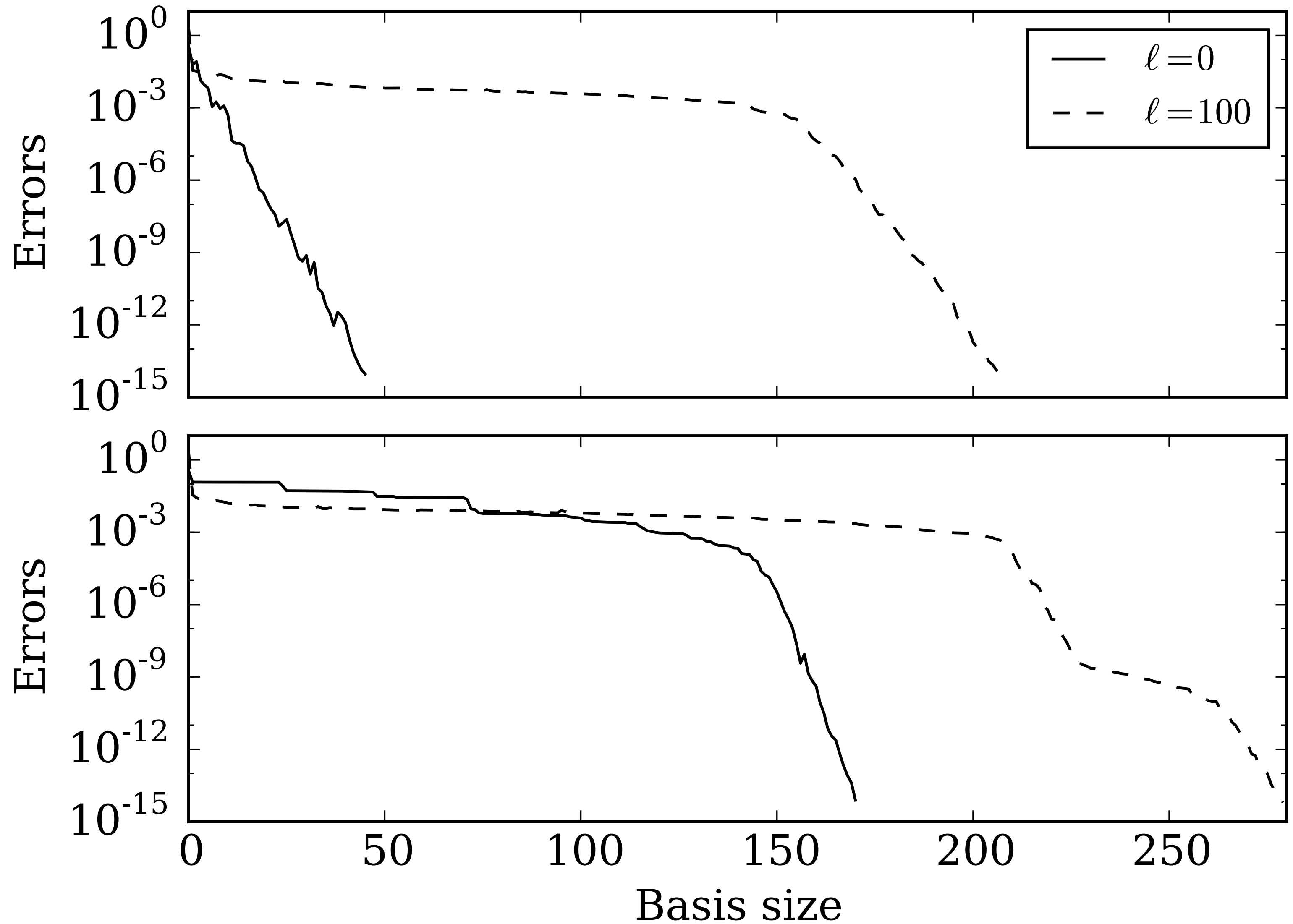
Total compression factor:

$$C_{\text{total}} = (\ell_{\max} + 1) \left(\sum_{\ell=0}^{\ell_{\max}} \frac{1}{C_\ell} \right)^{-1}$$

$$\approx 151$$

Less than 1% of the data is needed to capture all features up to numerical round-off errors

Shifting by time-of-arrival



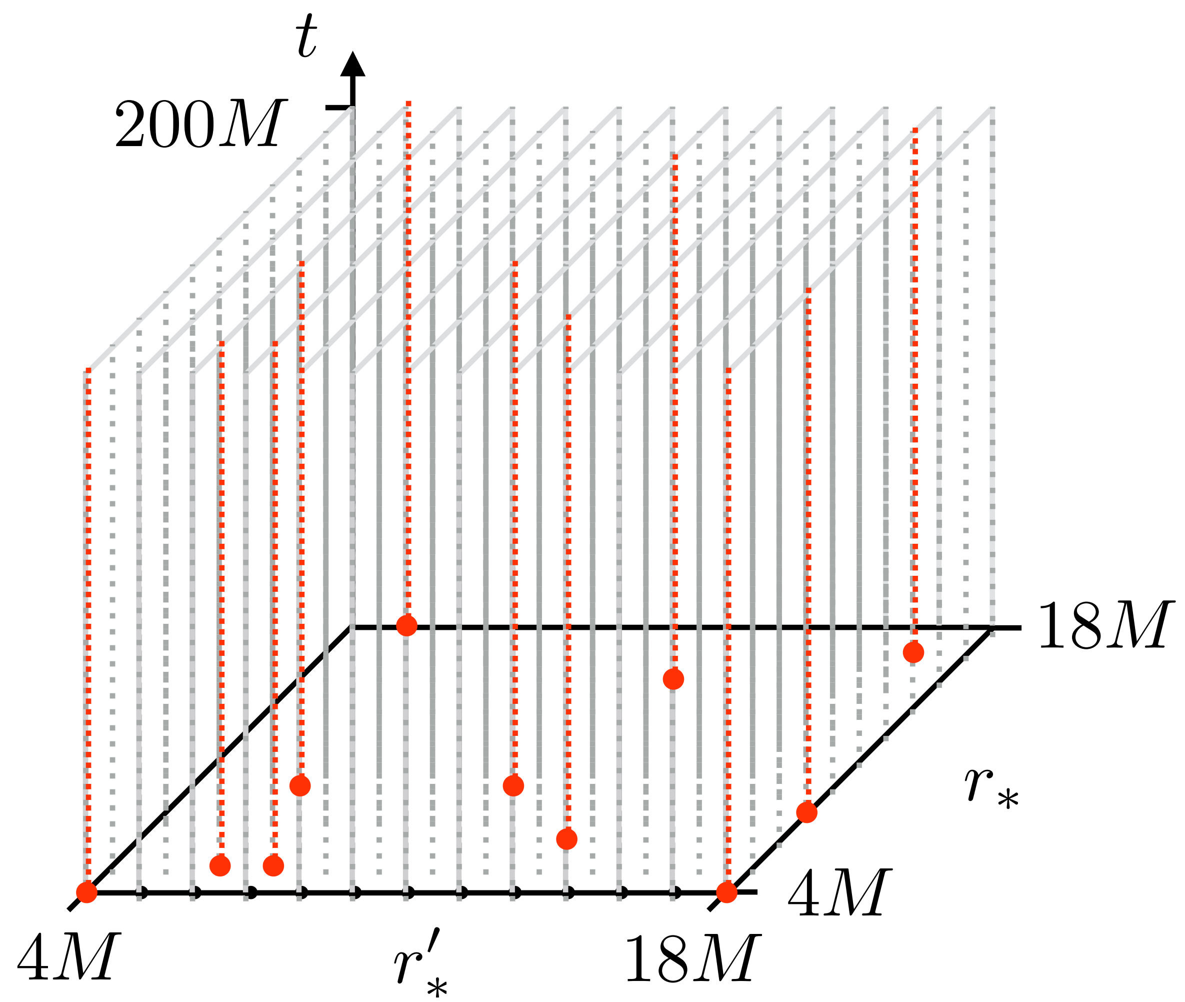
Not shifting by time-of-arrival

2) Empirical interpolation

Barrault+ (04)
Maday+ (09)

RB approximation:

$$G_\ell(t;\vec{\lambda}) \approx \sum_{i=1}^{N_\ell} C_i^\ell(\vec{\lambda}) e_i^\ell(t)$$



2) Empirical interpolation

Barrault+ (04)
Maday+ (09)

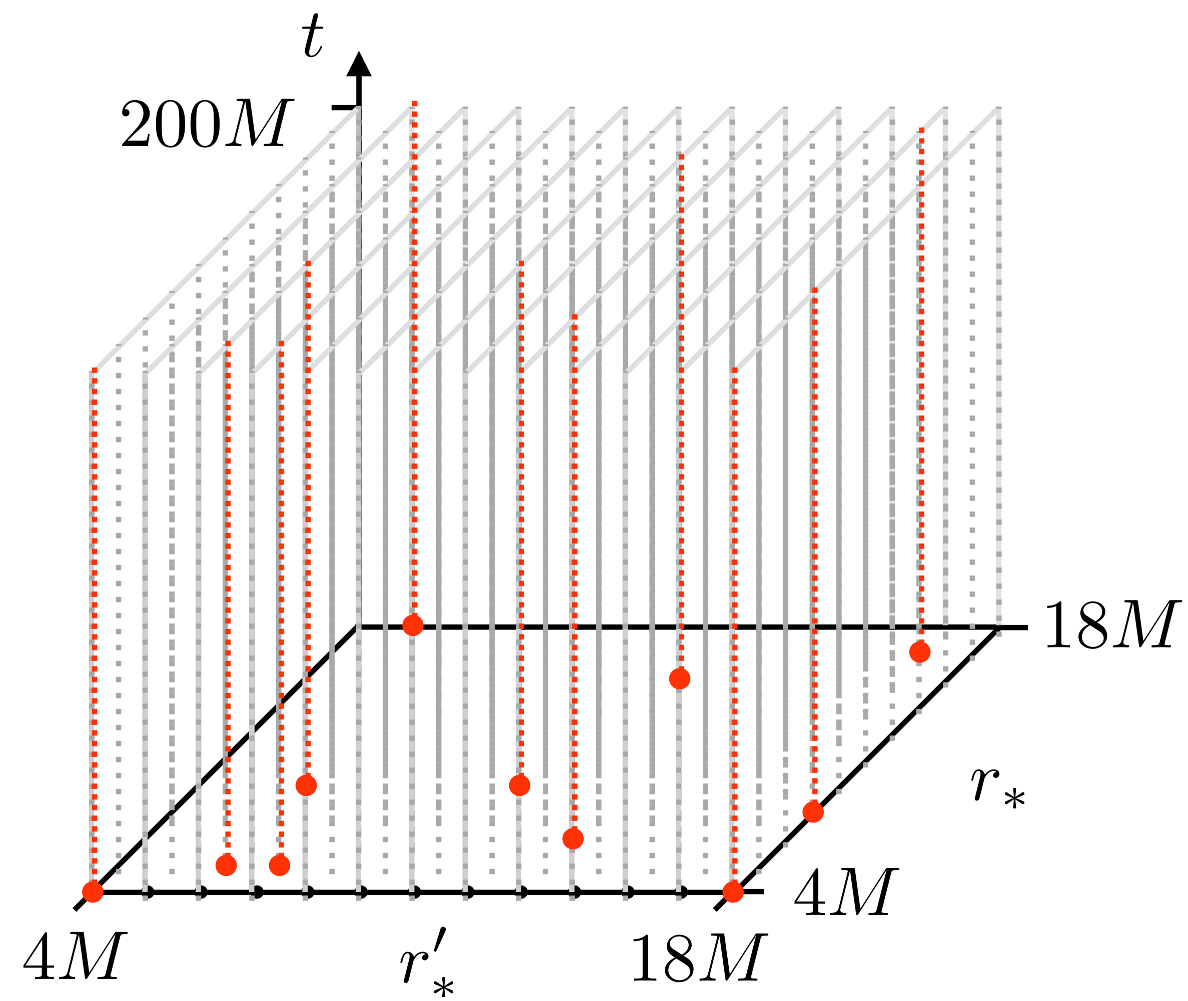
RB approximation:

$$G_\ell(t;\vec{\lambda}) \approx \sum_{i=1}^{N_\ell} C_i^\ell(\vec{\lambda}) e_i^\ell(t)$$

At n time subsamples of data, $\{T_i\}_{i=1}^{n}$ the coefficients can be solved

$$C_i^\ell(\vec{\lambda}) = \sum_{j=1}^{N_\ell} \left(V_\ell^{-1}\right)_{ij} G_\ell(T_j;\vec{\lambda})$$

$$(V_\ell)_{ij} \equiv e_i(T_j)$$



2) Empirical interpolation

Barrault+ (04)
Maday+ (09)

RB approximation:

$$G_\ell(t;\vec{\lambda}) \approx \sum_{i=1}^{N_\ell} C_i^\ell(\vec{\lambda}) e_i^\ell(t)$$

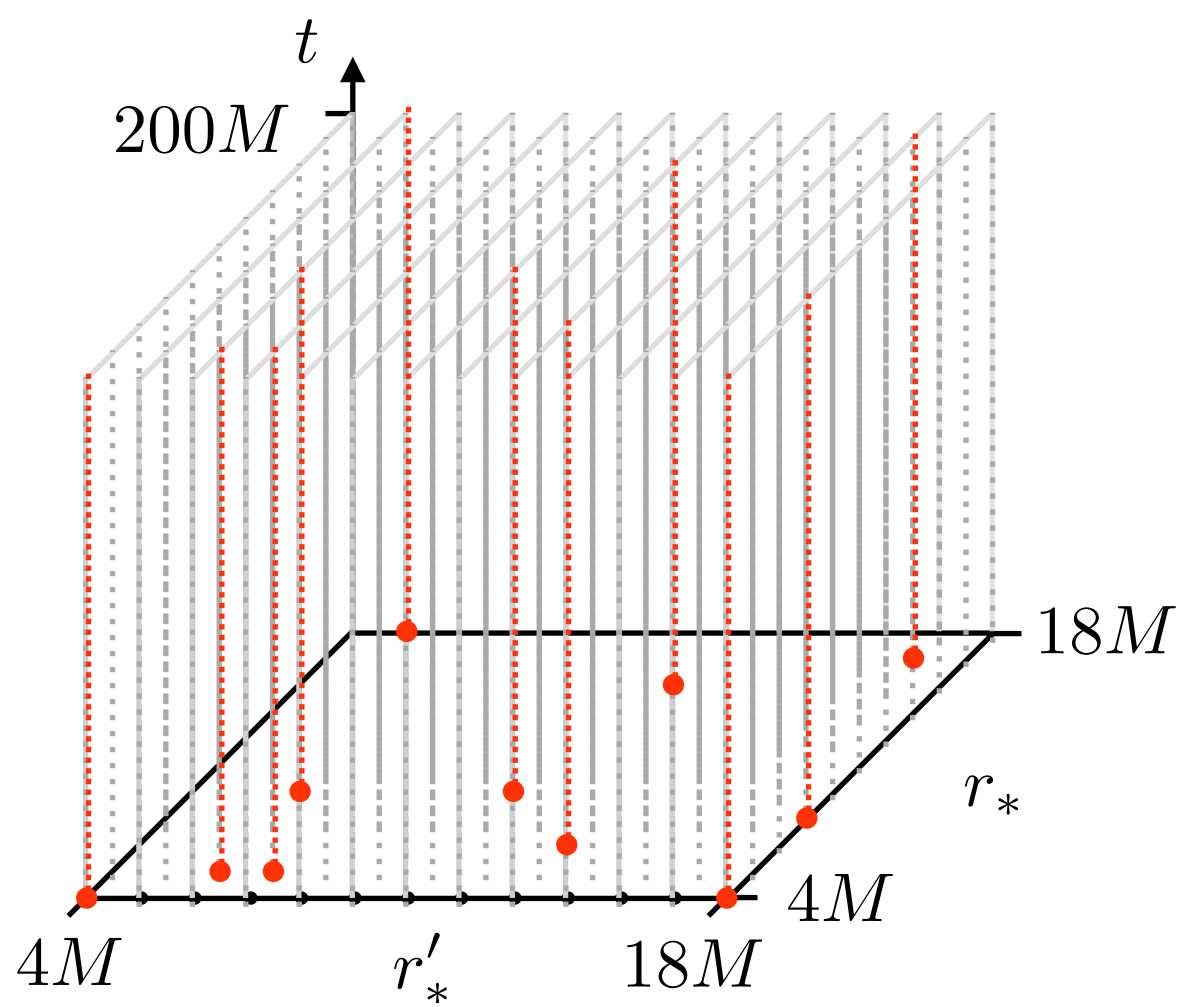
At n time subsamples of data, $\{T_i\}_{i=1}^n$ the coefficients can be solved

$$C_i^\ell(\vec{\lambda}) = \sum_{j=1}^{N_\ell} \left(V_\ell^{-1}\right)_{ij} G_\ell(T_j;\vec{\lambda})$$

$$(V_\ell)_{ij} \equiv e_i(T_j)$$

$$G_\ell(t;\vec{\lambda}) \approx \sum_{j=1}^{N_\ell} B_j^\ell(t) G_\ell(T_j;\vec{\lambda})$$

$$B_j^\ell(t) \equiv \sum_{i=1}^{N_\ell} e_i^\ell(t) \left(V_\ell^{-1}\right)_{ij}$$



2) Empirical interpolation

Barrault+ (04)
Maday+ (09)

RB approximation:

$$G_\ell(t;\vec{\lambda}) \approx \sum_{i=1}^{N_\ell} C_i^\ell(\vec{\lambda}) e_i^\ell(t)$$

At n time subsamples of data, $\{T_i\}_{i=1}^n$ the coefficients can be solved

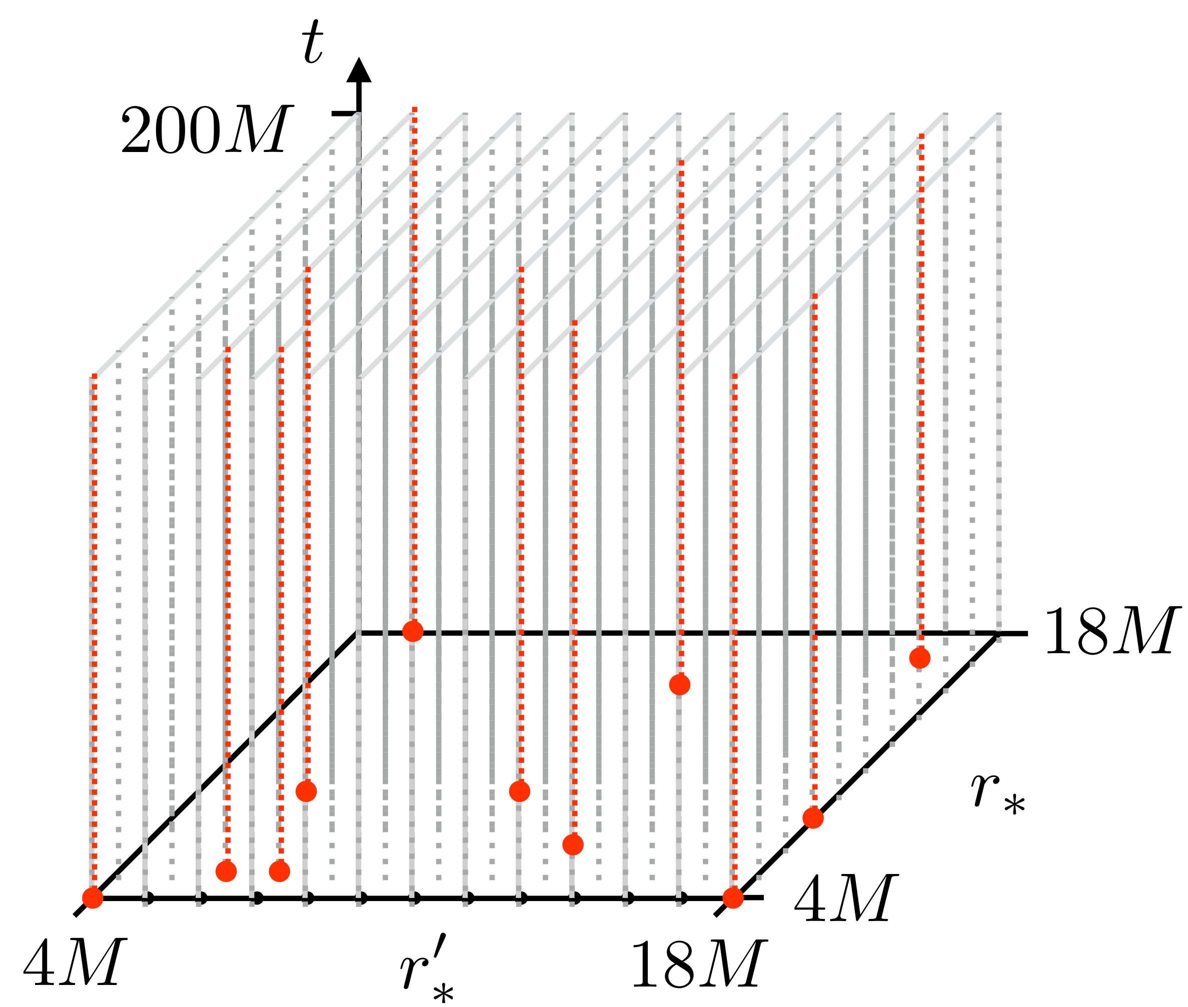
$$C_i^\ell(\vec{\lambda}) = \sum_{j=1}^{N_\ell} \left(V_\ell^{-1}\right)_{ij} G_\ell(T_j;\vec{\lambda})$$

$$(V_\ell)_{ij} \equiv e_i(T_j)$$

$$G_\ell(t;\vec{\lambda}) \approx \sum_{j=1}^{N_\ell} B_j^\ell(t) G_\ell(T_j;\vec{\lambda})$$

$$B_j^\ell(t) \equiv \sum_{i=1}^{N_\ell} e_i^\ell(t) \left(V_\ell^{-1}\right)_{ij}$$

Find the interpolation nodes through another greedy algorithm that minimizes the interpolation error



2) Empirical interpolation

Barrault+ (04)
Maday+ (09)

RB approximation:

$$G_\ell(t;\vec{\lambda}) \approx \sum_{i=1}^{N_\ell} C_i^\ell(\vec{\lambda}) e_i^\ell(t)$$

At n time subsamples of data, $\{T_i\}_{i=1}^{n}$ the coefficients can be solved

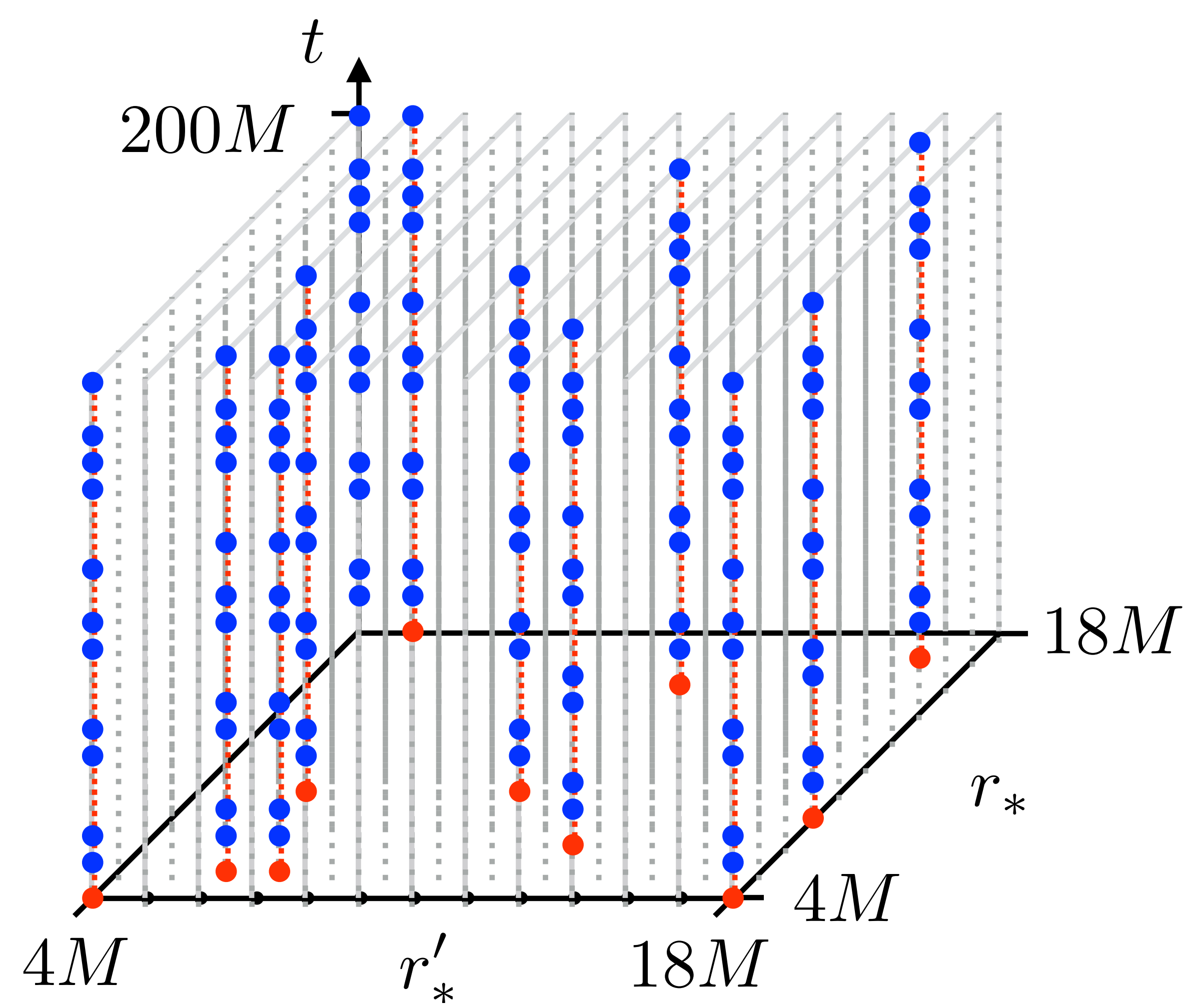
$$C_i^\ell(\vec{\lambda}) = \sum_{j=1}^{N_\ell} \left(V_\ell^{-1}\right)_{ij} G_\ell(T_j;\vec{\lambda})$$

$$(V_\ell)_{ij} \equiv e_i(T_j)$$

$$G_\ell(t;\vec{\lambda}) \approx \sum_{j=1}^{N_\ell} B_j^\ell(t) G_\ell(T_j;\vec{\lambda})$$

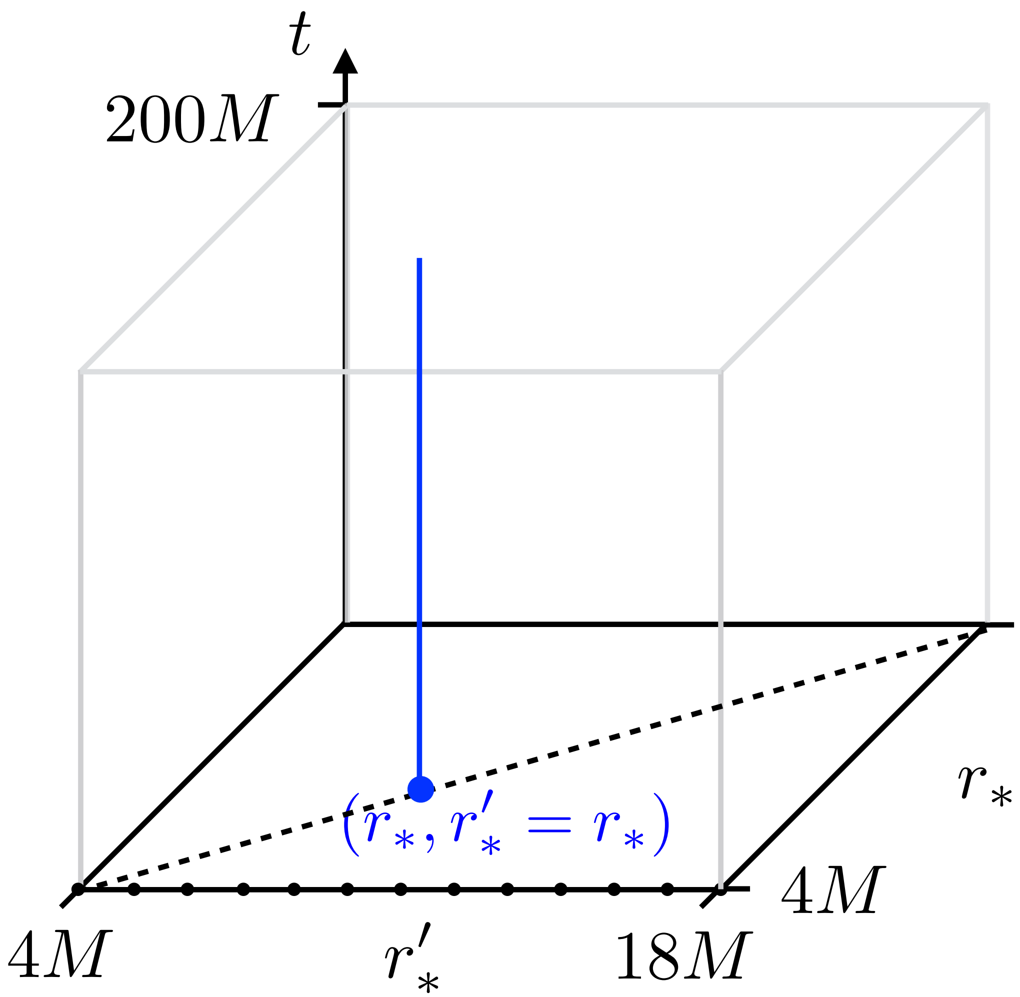
$$B_j^\ell(t) \equiv \sum_{i=1}^{N_\ell} e_i^\ell(t) \left(V_\ell^{-1}\right)_{ij}$$

Find the interpolation nodes through another greedy algorithm that minimizes the interpolation error



3) Parametric fitting — r^*= constant

Mostly interested in computing Green's function on worldlines for self-force

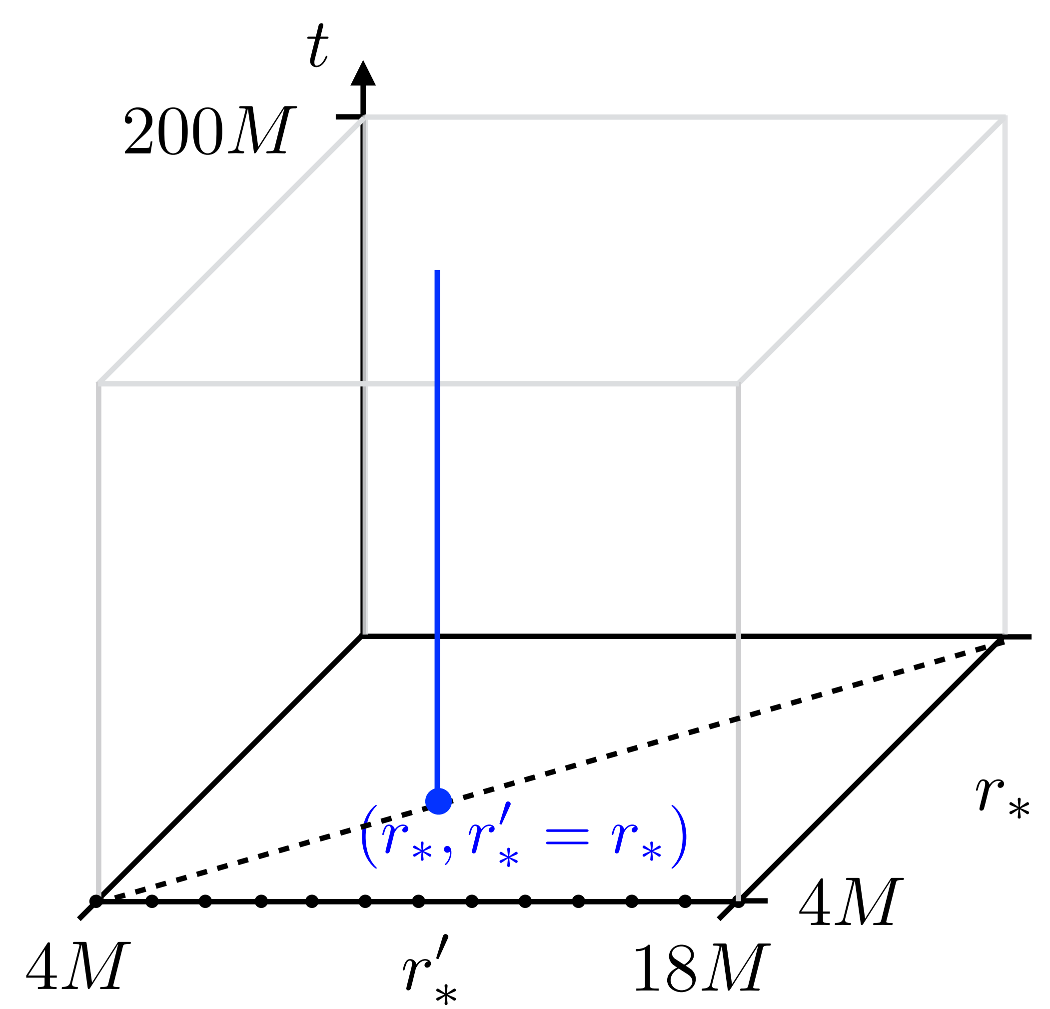


3) Parametric fitting — r^*= constant

Mostly interested in computing Green's function on worldlines for self-force

The surrogate is simple to evaluate on worldlines with r_* = constant

- static worldlines
- circular geodesics
- accelerated circular worldlines



3) Parametric fitting — r^*= constant

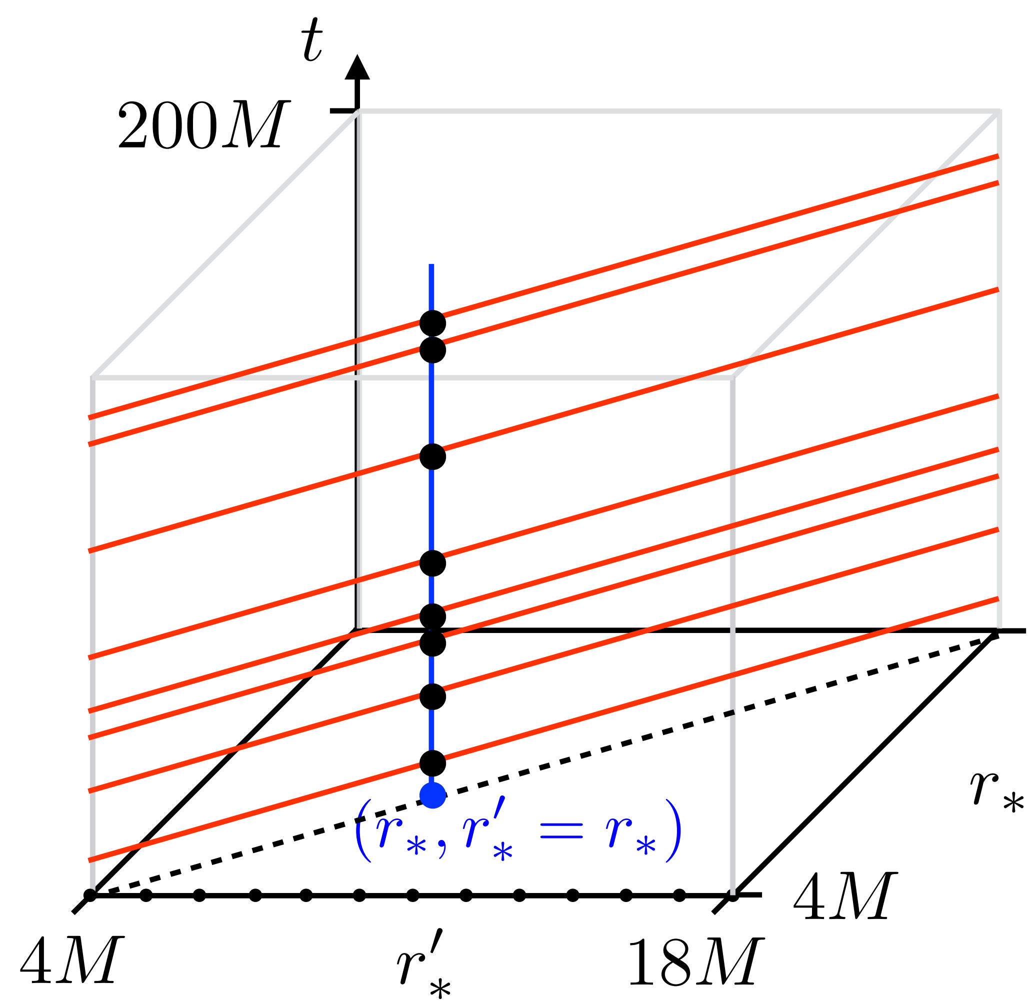
Mostly interested in computing Green's function on worldlines for self-force

The surrogate is simple to evaluate on worldlines with r_* = constant

- static worldlines
- circular geodesics
- accelerated circular worldlines

$$G^S_\ell(t; r_*, r_*) = \sum_{j=1}^{N_\ell} B^\ell_j(t) G^{\text{spline}}_\ell(T_j; r_*, r'_* = r_*)$$

$$G_S(x, x') = \frac{1}{r^2} \sum_{\ell=0}^{\ell_{\max}} (2\ell+1) e^{-\ell^2/2\ell^2_{\text{cut}}} \times P_\ell(\cos\gamma(t)) G^S_\ell(t; r_*, r_*)$$

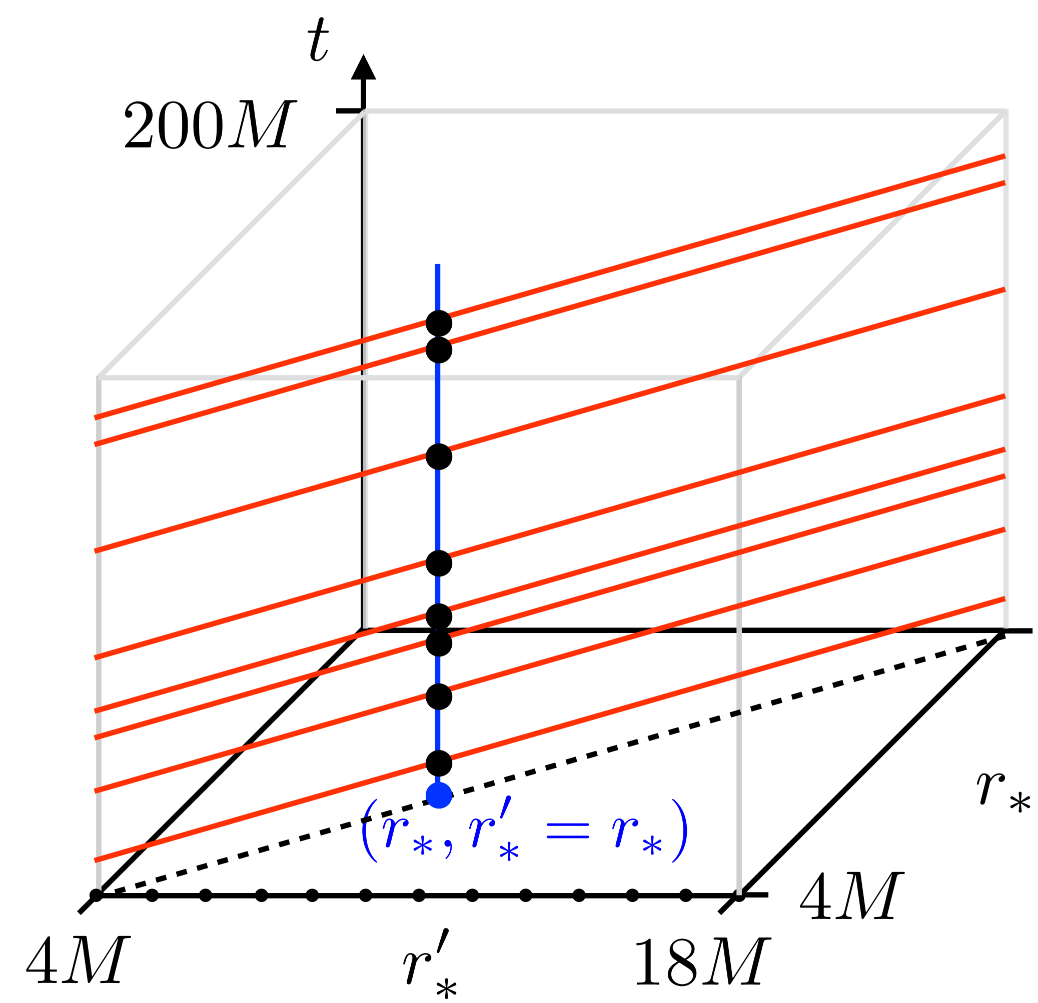


3) Parametric fitting — r^*= constant

Mostly interested in computing Green's function on worldlines for self-force

The surrogate is simple to evaluate on worldlines with r_* = constant

- static worldlines
- circular geodesics
- accelerated circular worldlines



$$G^S_\ell(t; r_*, r_*) = \sum_{j=1}^{N_\ell} B^\ell_j(t) G^{\text{spline}}_\ell(T_j; r_*, r'_* = r_*)$$

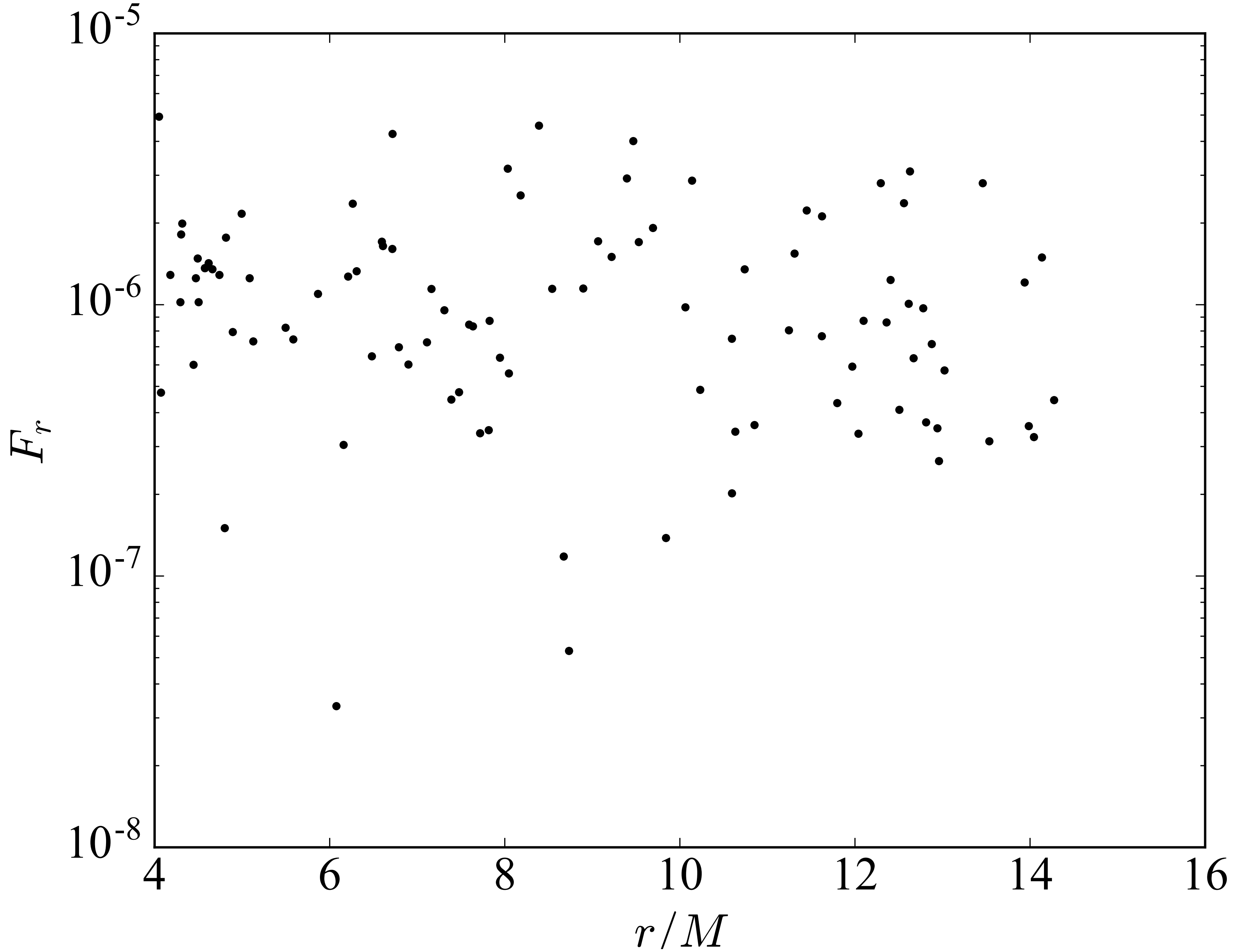
$$G_S(x, x') = \frac{1}{r^2} \sum_{\ell=0}^{\ell_{\max}} (2\ell + 1) e^{-\ell^2/2\ell^2_{\text{cut}}} \times P_\ell(\cos\gamma(t)) G^S_\ell(t; r_*, r_*)$$

<u>Total compression</u>: 30GB /192MB = 156

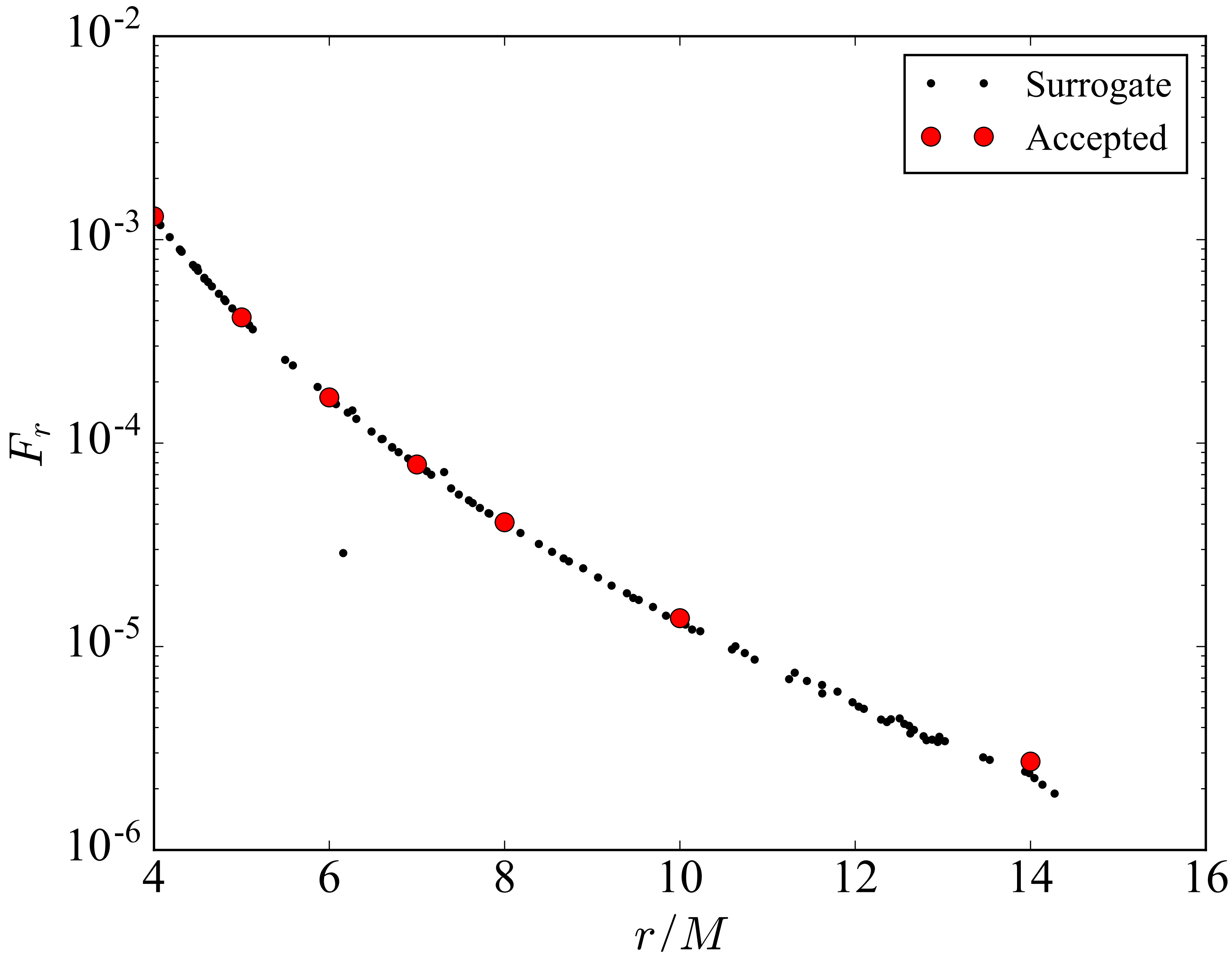
<u>Speed-up</u>: 380s / 0.5s = 760*x*

Static self-force

Analytically derived and found to be zero *Wiseman (00)*



Self-force on circular geodesics



Accepted or “truth” values computed in *Diaz-Rivera et al (04)*

3) Parametric fitting — generic case

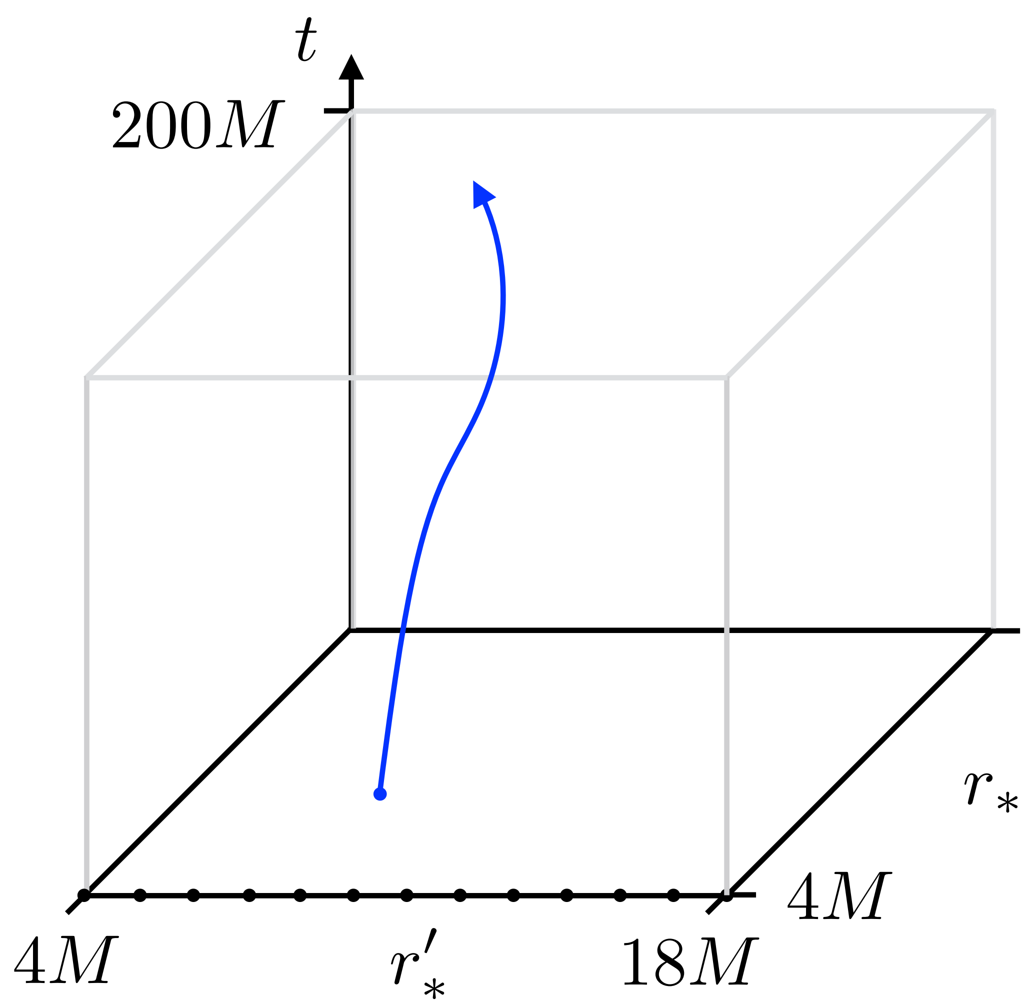
More general worldlines

$$z^\mu(t) = \big(t, r(t), \pi/2, \gamma(t)\big)$$

destroy the affine nature
of the model (i.e., separability)

$$G_\ell(t; r_*(t), r'_*) \approx \sum_{j=1}^{N_\ell} B^\ell_j(t) G_\ell(T_j; r_*(t), r'_*)$$

As a result, fitting is done on-the-fly
on a worldline



3) Parametric fitting — generic case

More general worldlines

$$z^\mu(t) = \big(t, r(t), \pi/2, \gamma(t)\big)$$

destroy the affine nature
of the model (i.e., separability)

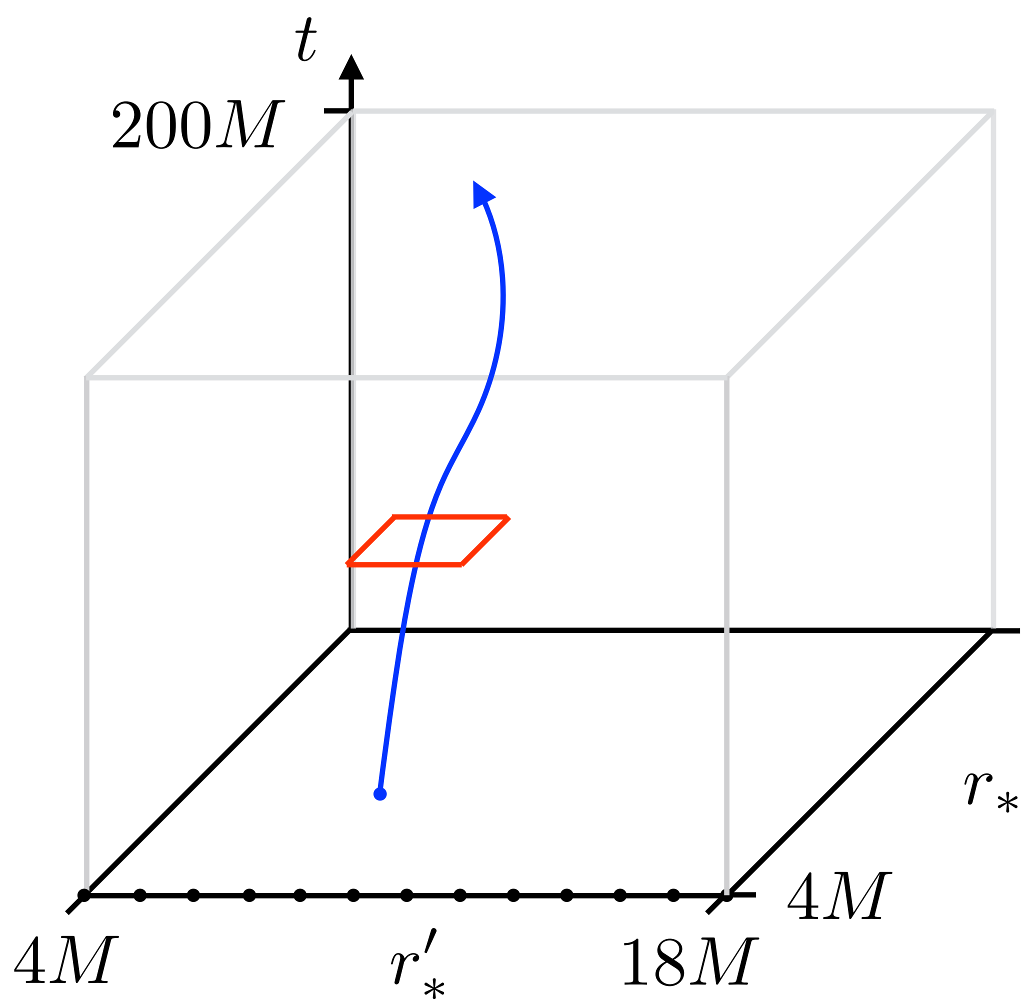
$$G_\ell(t; r_*(t), r'_*) \approx \sum_{j=1}^{N_\ell} B_j^\ell(t) G_\ell(T_j; r_*(t), r'_*)$$

As a result, fitting is done on-the-fly
on a worldline

At a given time step, t_k, reconstruct
the Green's function data in a small
patch around the worldline

Fit with spline and interpolate to $r_*(t_k)$

Repeat for all time steps



3) Parametric fitting — generic case

More general worldlines

$$z^\mu(t) = \big(t, r(t), \pi/2, \gamma(t)\big)$$

destroy the affine nature
of the model (i.e., separability)

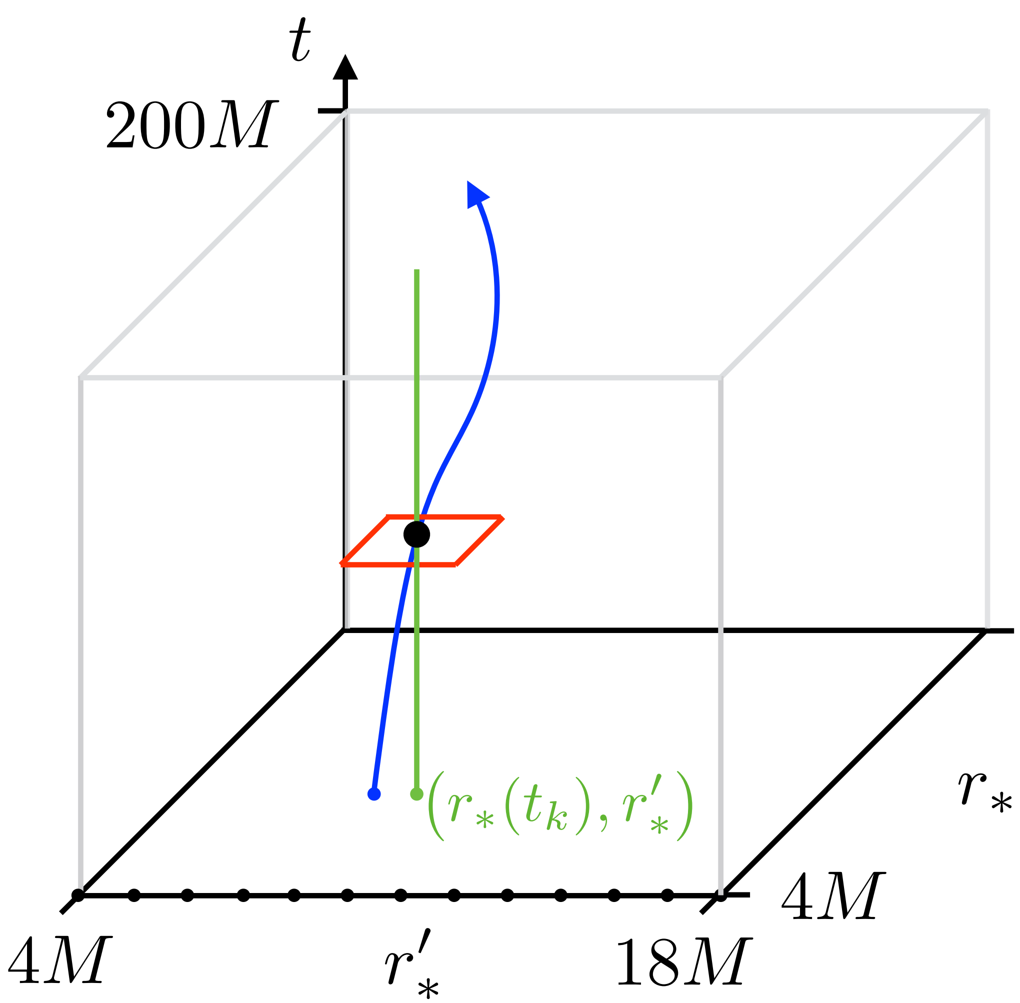
$$G_\ell(t; r_*(t), r'_*) \approx \sum_{j=1}^{N_\ell} B^\ell_j(t) G_\ell(T_j; r_*(t), r'_*)$$

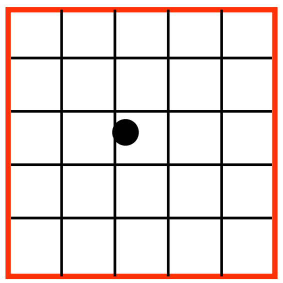
As a result, fitting is done on-the-fly
on a worldline

At a given time step, t_k, reconstruct
the Green's function data in a small
patch around the worldline

Fit with spline and interpolate to $r_*(t_k)$

Repeat for all time steps





3) Parametric fitting — generic case

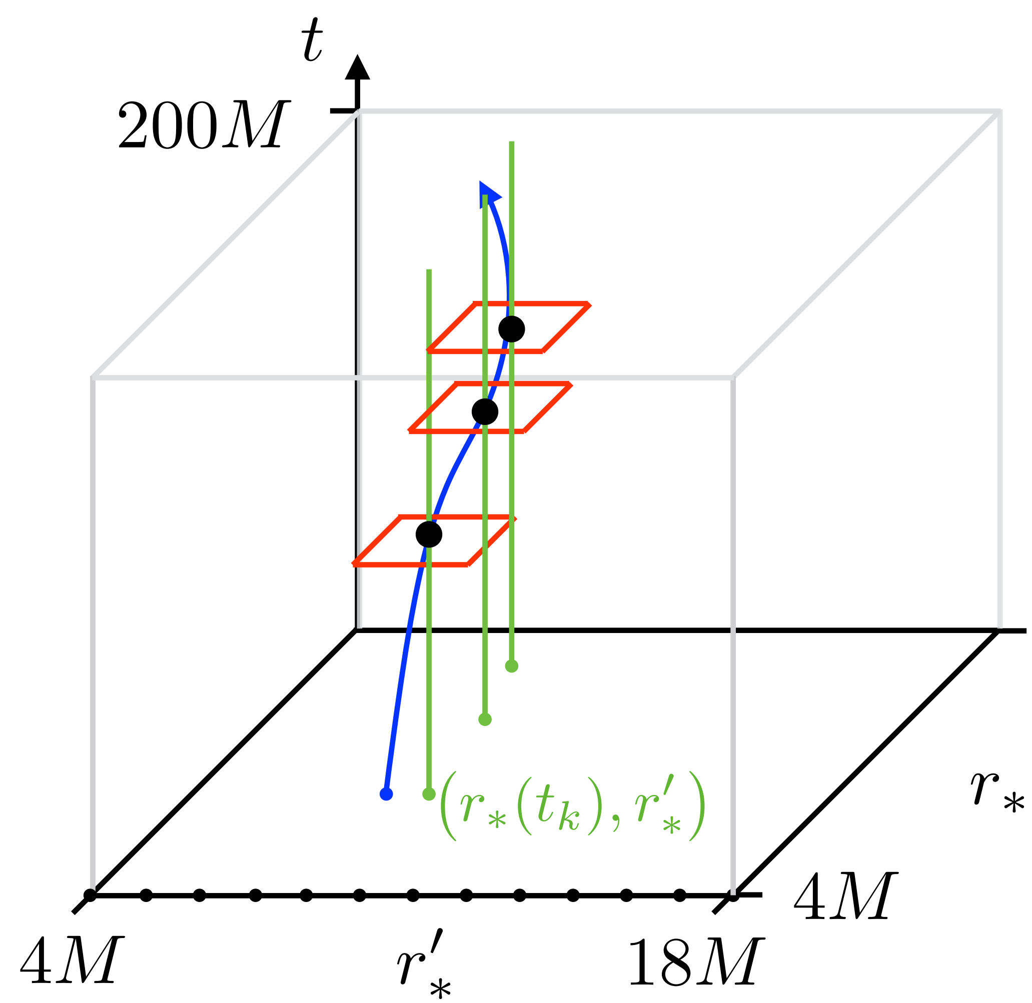
More general worldlines

$$z^\mu(t) = \big(t, r(t), \pi/2, \gamma(t)\big)$$

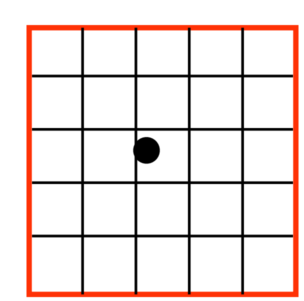
destroy the affine nature
of the model (i.e., separability)

$$G_\ell(t; r_*(t), r'_*) \approx \sum_{j=1}^{N_\ell} B_j^\ell(t) G_\ell(T_j; r_*(t), r'_*)$$

As a result, fitting is done on-the-fly
on a worldline



At a given time step, t_k, reconstruct
the Green's function data in a small
patch around the worldline



Fit with spline and interpolate to $r_*(t_k)$

Repeat for all time steps

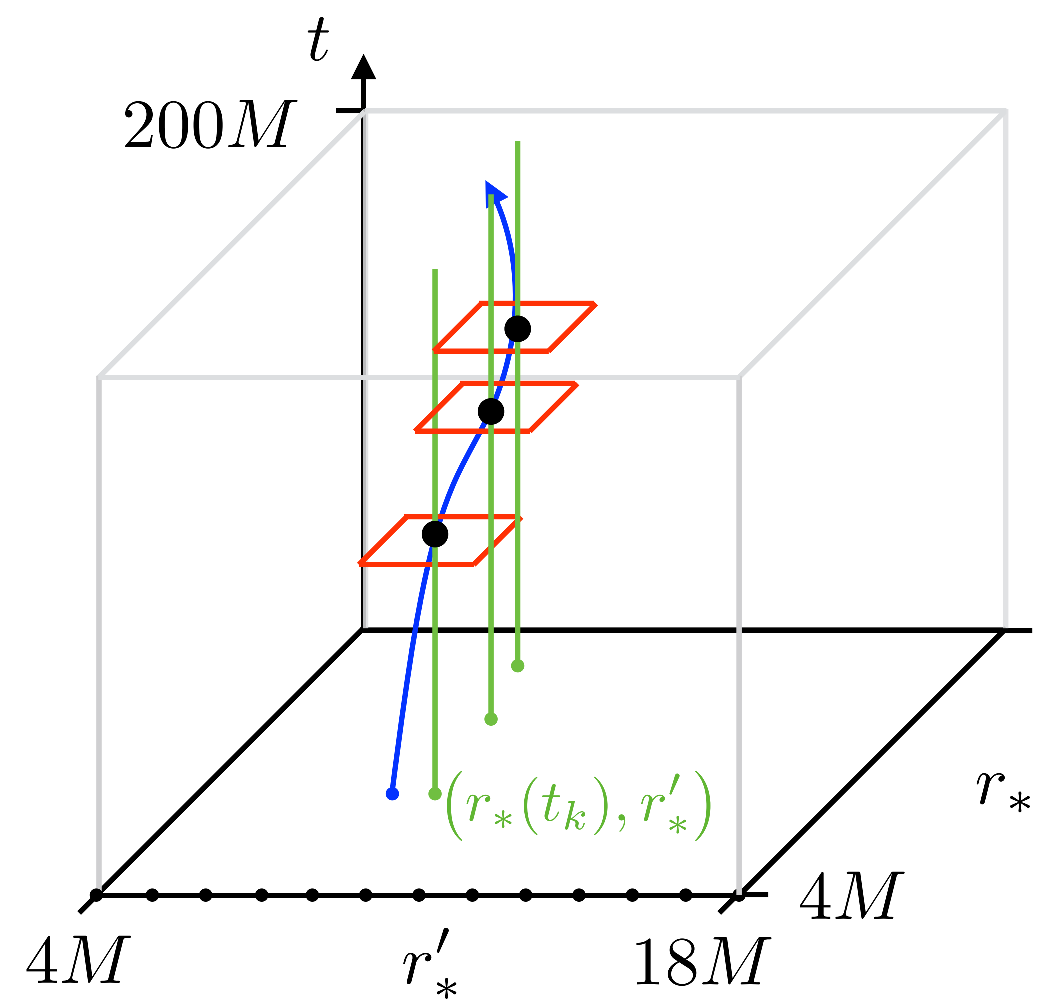
3) Parametric fitting — generic case

More general worldlines

$$z^\mu(t) = \big(t, r(t), \pi/2, \gamma(t)\big)$$

destroy the affine nature
of the model (i.e., separability)

$$G_\ell(t; r_*(t), r'_*) \approx \sum_{j=1}^{N_\ell} B_j^\ell(t) G_\ell(T_j; r_*(t), r'_*)$$

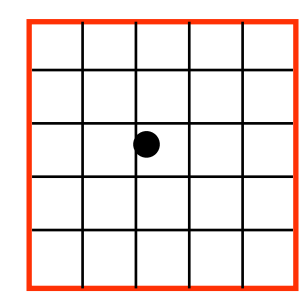


As a result, fitting is done on-the-fly
on a worldline

At a given time step, t_k, reconstruct
the Green's function data in a small
patch around the worldline

Fit with spline and interpolate to $r_*(t_k)$

Slower and need to store to disk all data at each T_j…



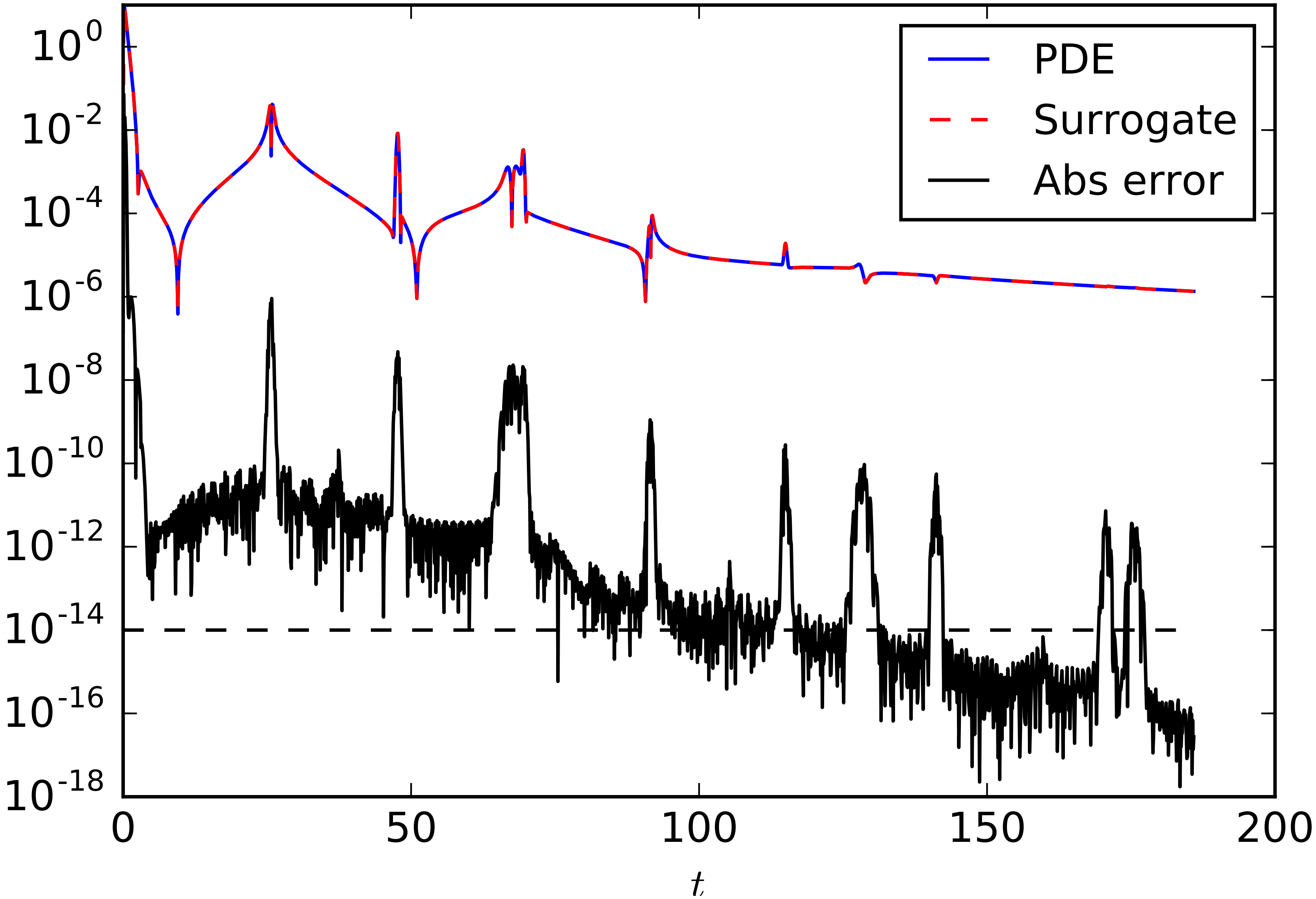
Repeat for all time steps

Surrogate accuracy, speed-up, and size

Eccentric geodesic orbit ($e = 0.5$, $p = 7.2$)

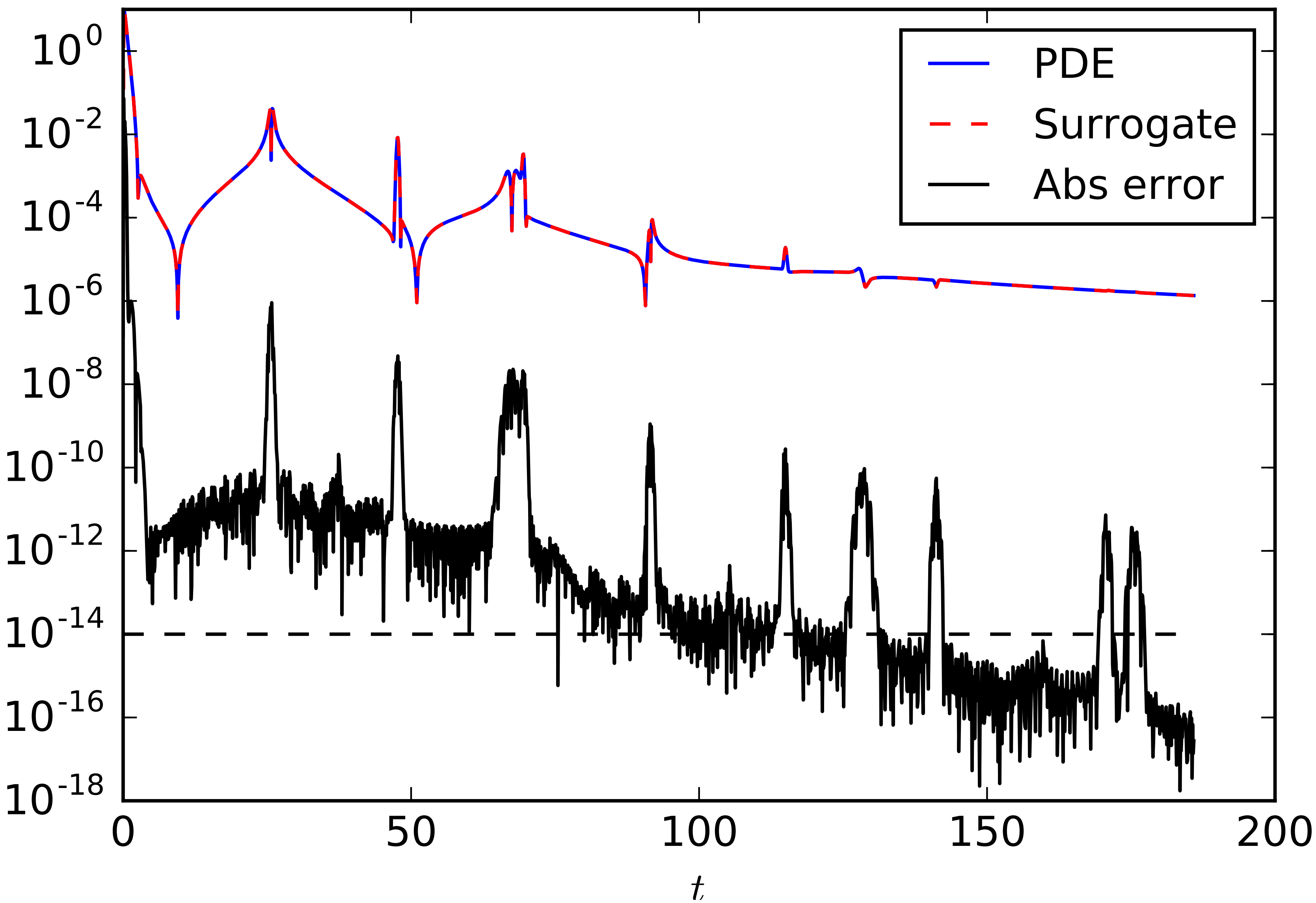
Surrogate accuracy, speed-up, and size

Eccentric geodesic orbit (*e = 0.5, p = 7.2*)



Surrogate accuracy, speed-up, and size

Eccentric geodesic orbit (*e = 0.5, p = 7.2*)

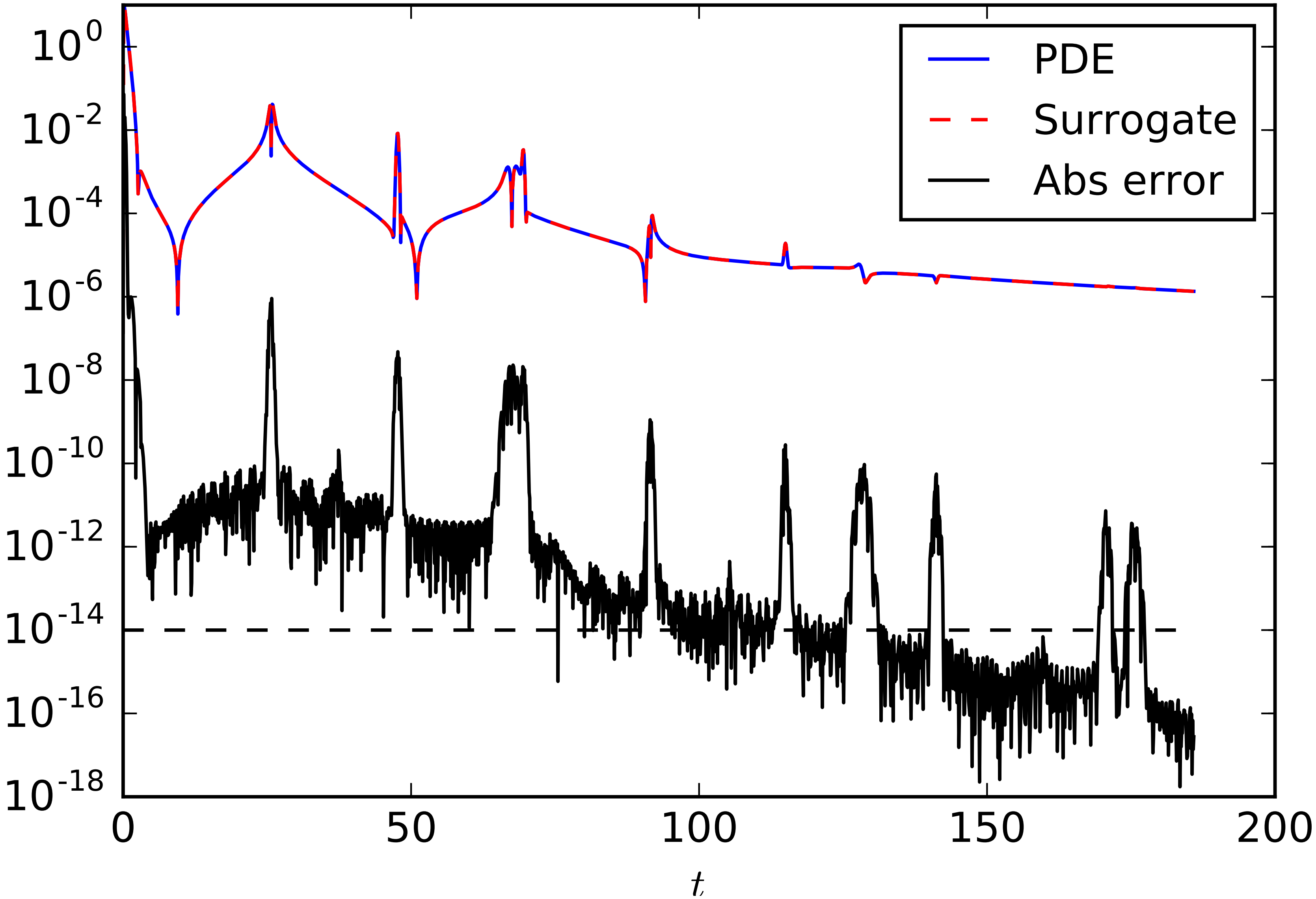


PDE	~380s
Surrogate	~25s
Speed-up*	~15x

* *PDE in C++*
Surrogate in Python

Surrogate accuracy, speed-up, and size

Eccentric geodesic orbit (*e = 0.5, p = 7.2*)



PDE	~380s
Surrogate	~25s
Speed-up*	~15x

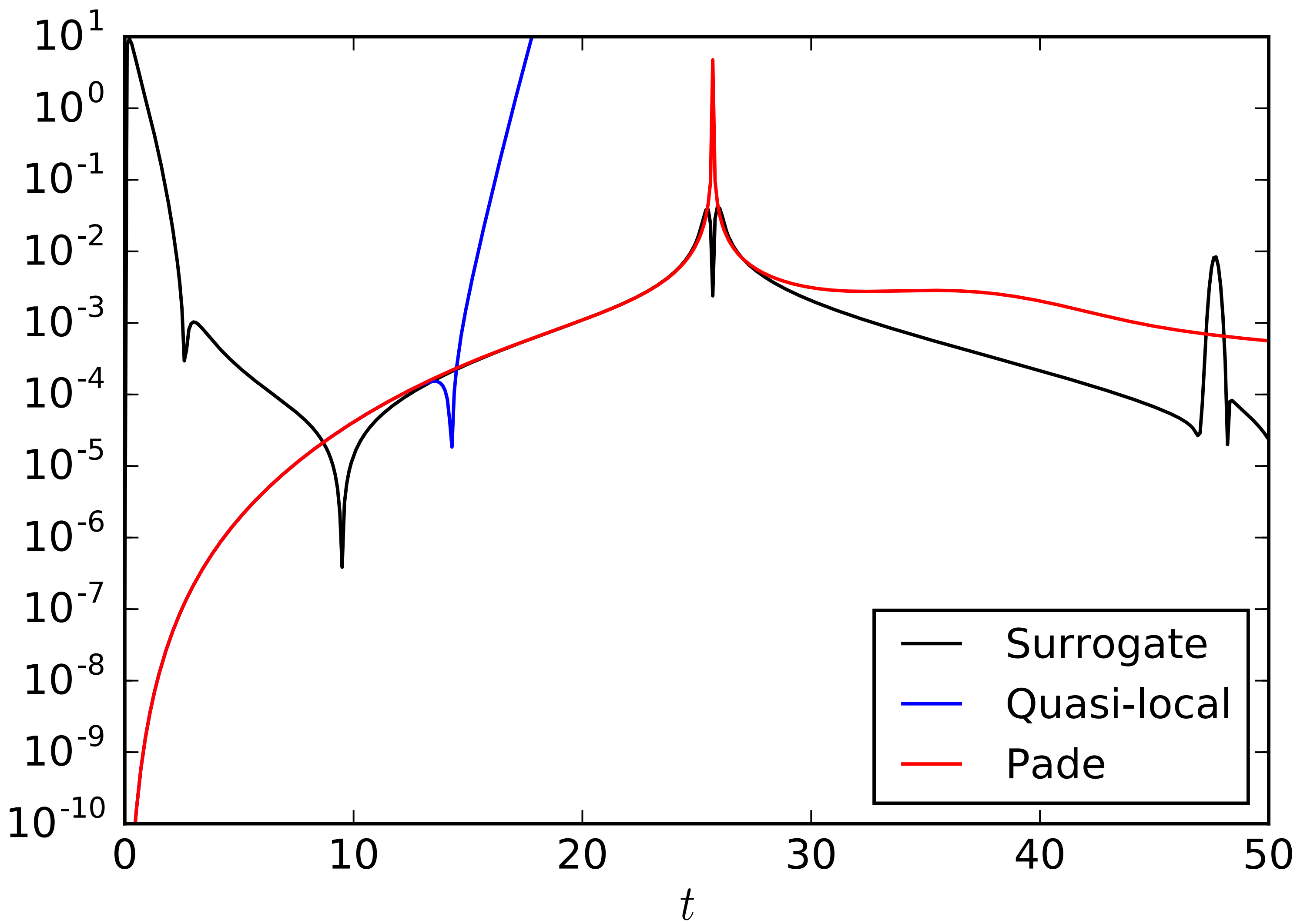
* *PDE in C++*
Surrogate in Python

Physical memory: 30GB reduced to 2GB

Surrogate self-force evaluation

$$
\begin{aligned}
&G_{\rm ret}(z^{\mu}, z^{\mu'}) \\
&\approx \theta(\tau - \tau_{\rm ql})\mathrm{Pade}[V_{\rm ql}(z^{\mu}, z^{\mu'})] \\
&\quad + \theta(\tau_{\rm ql} - \tau)\theta(\tau - \tau_{\rm br})G_{\rm surr}(z^{\mu}, z^{\mu'}) \\
&\quad + \theta(\tau_{\rm br} - \tau)G_{\rm br}(z^{\mu}, z^{\mu'})
\end{aligned}
$$

- Method of matched expansions
  Anderson & Wiseman (05); Casals et al (13)
- Quasi-local expansions
  Ottewill & Wardell (08); Wardell's thesis
- Pade approximants *Casals et al (09)*

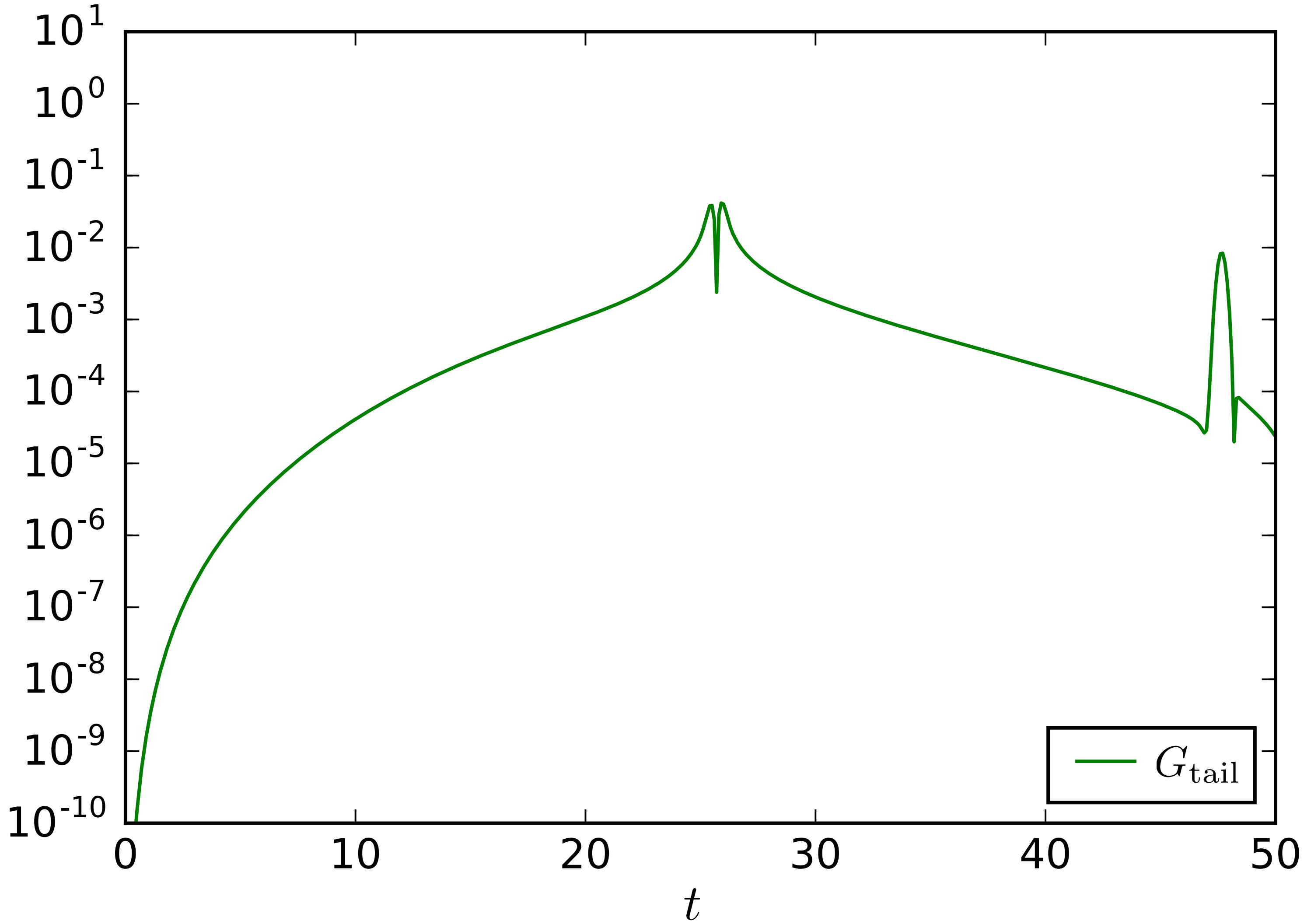


Eccentric geodesic orbit (*e = 0.5, p = 7.2*)

Surrogate self-force evaluation

$$
\begin{aligned}
&G_{\rm ret}(z^\mu, z^{\mu'}) \\
&\approx \theta(\tau - \tau_{\rm ql})\mathrm{Pade}[V_{\rm ql}(z^\mu, z^{\mu'})] \\
&\quad + \theta(\tau_{\rm ql} - \tau)\theta(\tau - \tau_{\rm br})G_{\rm surr}(z^\mu, z^{\mu'}) \\
&\quad + \theta(\tau_{\rm br} - \tau)G_{\rm br}(z^\mu, z^{\mu'})
\end{aligned}
$$

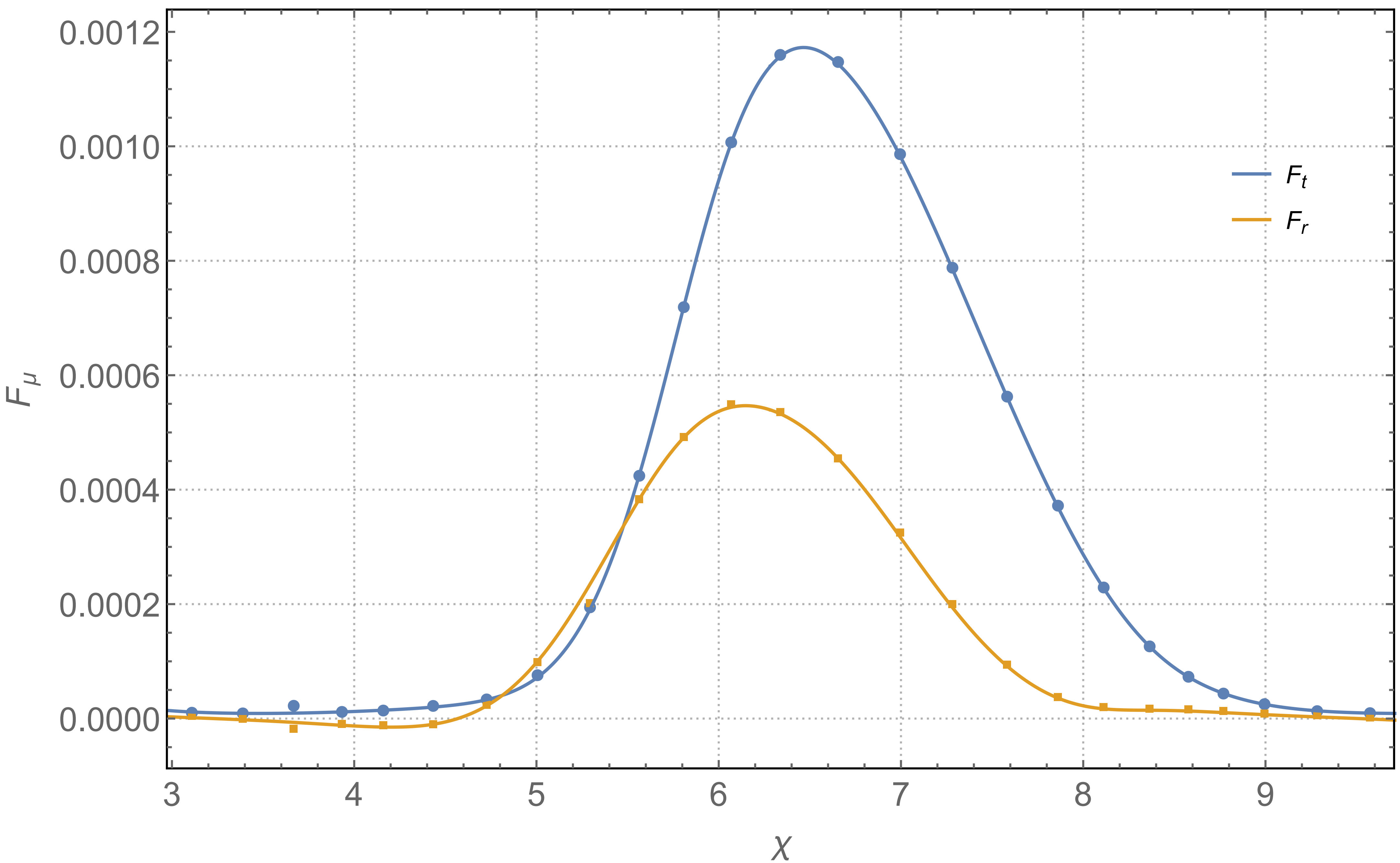
- Method of matched expansions
  Anderson & Wiseman (05); Casals et al (13)
- Quasi-local expansions
  Ottewill & Wardell (08); Wardell's thesis
- Pade approximants *Casals et al (09)*



Eccentric geodesic orbit (*e = 0.5, p = 7.2*)

Surrogate self-force evaluation

Eccentric geodesic orbit (*e = 0.5, p = 7.2*)

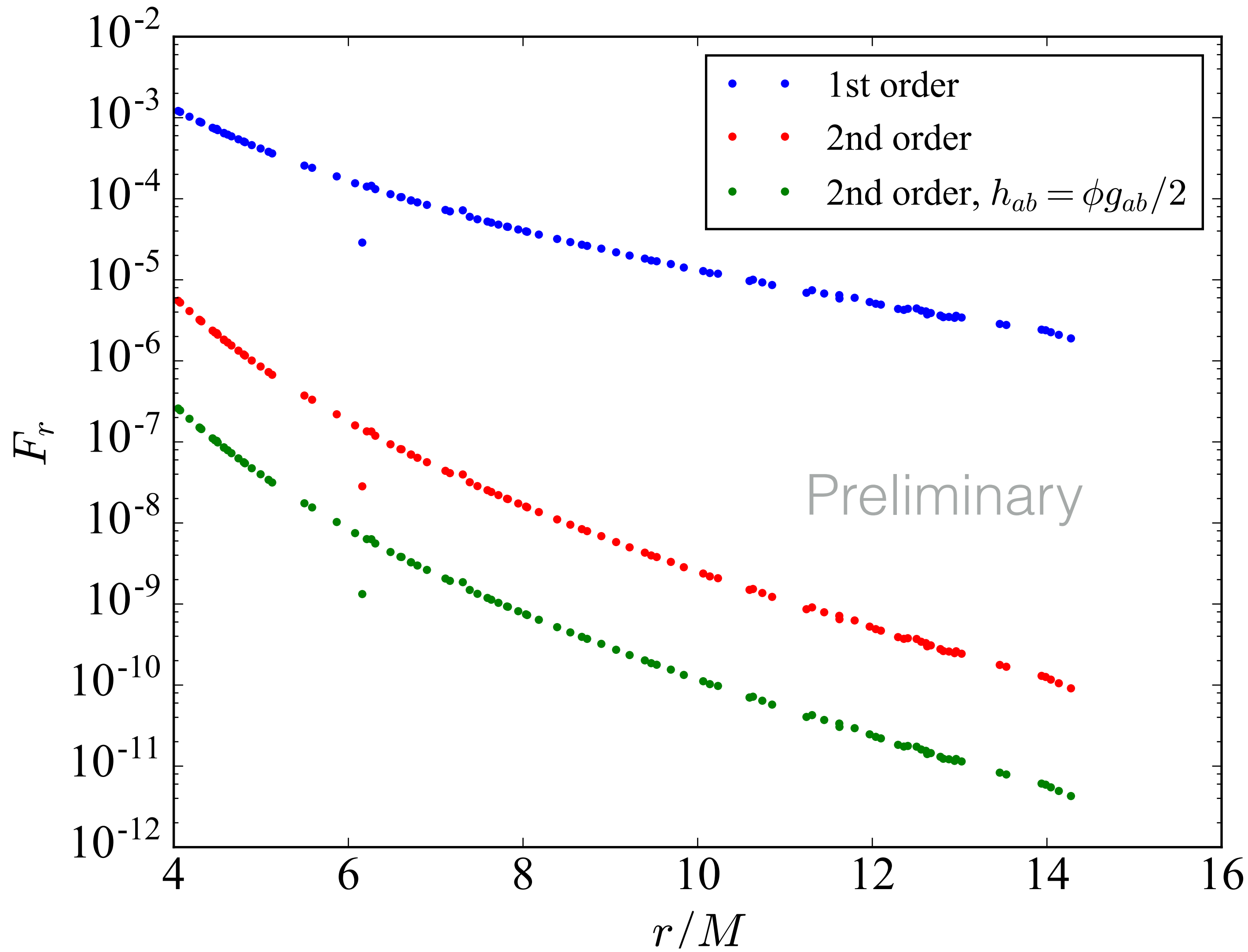


Applications

- Higher-order self-force and radiation *CRG (12a, 12b)*

$$ma^\mu \supset P^{\mu\nu}\left(\lim_{\epsilon\to 0^+}\int_{-\infty}^{\tau-\epsilon} d\tau'\,\nabla_\nu G_{\rm ret}(z^\mu, z^{\mu'})\right)\left(\lim_{\epsilon\to 0^+}\int_{-\infty}^{\tau-\epsilon} d\tau''\,G_{\rm ret}(z^\mu, z^{\mu''})\right),$$

$$P^{\mu\nu}\lim_{\epsilon\to 0^+}\int_{-\infty}^{\tau-\epsilon} d\tau'\,\nabla_\nu G_{\rm ret}(z^\mu, z^{\mu'})\lim_{\epsilon\to 0^+}\int_{-\infty}^{\tau'-\epsilon} d\tau''\,G_{\rm ret}(z^{\mu'}, z^{\mu''})$$



Applications

- Higher-order self-force and radiation *CRG (12a, 12b)*

$$ma^\mu \supset P^{\mu\nu}\left(\lim_{\epsilon\to 0^+}\int_{-\infty}^{\tau-\epsilon} d\tau'\,\nabla_\nu G_{\rm ret}(z^\mu, z^{\mu'})\right)\left(\lim_{\epsilon\to 0^+}\int_{-\infty}^{\tau-\epsilon} d\tau''\,G_{\rm ret}(z^\mu, z^{\mu''})\right),$$

$$P^{\mu\nu}\lim_{\epsilon\to 0^+}\int_{-\infty}^{\tau-\epsilon} d\tau'\,\nabla_\nu G_{\rm ret}(z^\mu, z^{\mu'})\,\lim_{\epsilon\to 0^+}\int_{-\infty}^{\tau'-\epsilon} d\tau''\,G_{\rm ret}(z^{\mu'}, z^{\mu''})$$

- Self-consistent evolution

$$\begin{aligned}\frac{d^2 z^\mu}{d\tau^2} + \Gamma^\mu_{\alpha\beta}\frac{dz^\alpha}{d\tau}\frac{dz^\beta}{d\tau} \approx q^2 P^{\mu\nu}\int_{-\infty}^{\tau} d\tau' \Big\{ &\theta(\tau' - \tau_{\rm ql}){\rm Pade}\big[\nabla_\nu V_{\rm ql}(z^\mu, z^{\mu'})\big] \\ &+ \theta(\tau_{\rm ql} - \tau')\theta(\tau' - \tau_{\rm br})\nabla_\nu G_{\rm surr}(z^\mu, z^{\mu'}) \\ &+ \theta(\tau_{\rm br} - \tau')G_{\rm br}(z^\mu, z^{\mu'})\Big\} + \text{local terms}\end{aligned}$$

Applications

- Higher-order self-force and radiation *CRG (12a, 12b)*

$$ma^\mu \supset P^{\mu\nu}\left(\lim_{\epsilon\to 0^+}\int_{-\infty}^{\tau-\epsilon} d\tau'\,\nabla_\nu G_{\rm ret}(z^\mu, z^{\mu'})\right)\left(\lim_{\epsilon\to 0^+}\int_{-\infty}^{\tau-\epsilon} d\tau''\,G_{\rm ret}(z^\mu, z^{\mu''})\right),$$

$$P^{\mu\nu}\lim_{\epsilon\to 0^+}\int_{-\infty}^{\tau-\epsilon} d\tau'\,\nabla_\nu G_{\rm ret}(z^\mu, z^{\mu'})\;\lim_{\epsilon\to 0^+}\int_{-\infty}^{\tau'-\epsilon} d\tau''\,G_{\rm ret}(z^{\mu'}, z^{\mu''})$$

- Self-consistent evolution

$$\begin{aligned}\frac{d^2 z^\mu}{d\tau^2} + \Gamma^\mu_{\alpha\beta}\frac{dz^\alpha}{d\tau}\frac{dz^\beta}{d\tau} \approx q^2 P^{\mu\nu}\int_{-\infty}^{\tau} d\tau'\Big\{&\theta(\tau'-\tau_{\rm ql})\mathrm{Pade}\big[\nabla_\nu V_{\rm ql}(z^\mu, z^{\mu'})\big]\\ &+\theta(\tau_{\rm ql}-\tau')\theta(\tau'-\tau_{\rm br})\nabla_\nu G_{\rm surr}(z^\mu, z^{\mu'})\\ &+\theta(\tau_{\rm br}-\tau')G_{\rm br}(z^\mu, z^{\mu'})\Big\} + \text{local terms}\end{aligned}$$

- Higher-order, self-consistent evolutions

Applications

- Higher-order self-force and radiation *CRG (12a, 12b)*

$$ma^\mu \supset P^{\mu\nu}\left(\lim_{\epsilon\to 0^+}\int_{-\infty}^{\tau-\epsilon} d\tau'\, \nabla_\nu G_{\rm ret}(z^\mu, z^{\mu'})\right)\left(\lim_{\epsilon\to 0^+}\int_{-\infty}^{\tau-\epsilon} d\tau''\, G_{\rm ret}(z^\mu, z^{\mu''})\right),$$

$$P^{\mu\nu}\lim_{\epsilon\to 0^+}\int_{-\infty}^{\tau-\epsilon} d\tau'\, \nabla_\nu G_{\rm ret}(z^\mu, z^{\mu'})\, \lim_{\epsilon\to 0^+}\int_{-\infty}^{\tau'-\epsilon} d\tau''\, G_{\rm ret}(z^{\mu'}, z^{\mu''})$$

- Self-consistent evolution

$$\frac{d^2 z^\mu}{d\tau^2} + \Gamma^\mu_{\alpha\beta}\frac{dz^\alpha}{d\tau}\frac{dz^\beta}{d\tau} \approx q^2 P^{\mu\nu}\int_{-\infty}^{\tau} d\tau' \Big\{\theta(\tau'-\tau_{\rm ql}){\rm Pade}\big[\nabla_\nu V_{\rm ql}(z^\mu, z^{\mu'})\big]$$
$$+\,\theta(\tau_{\rm ql}-\tau')\theta(\tau'-\tau_{\rm br})\nabla_\nu G_{\rm surr}(z^\mu, z^{\mu'})$$
$$+\,\theta(\tau_{\rm br}-\tau')G_{\rm br}(z^\mu, z^{\mu'})\Big\} + \text{local terms}$$

- Higher-order, self-consistent evolutions

- Self-consistent field/waveform and at higher orders

$$\phi(x^\alpha) = q\int_{-\infty}^{\tau_{\rm ret}(x)} d\tau' \Big\{\theta(\tau'-\tau_{\rm br})G_{\rm surr}(x, z^{\mu'}) + +\theta(\tau_{\rm br}-\tau')G_{\rm br}(x, z^{\mu'})\Big\}$$

- Comparing errors in osculating orbits and self-consistent evolutions (via two derivatives of the Green's function) *Pound (unpublished)*

$$F^{\mu}_{\mathrm{hist}}(\tau) = q^2 P^{\mu\nu} \lim_{\epsilon\to 0^+} \int_{-\infty}^{\tau-\epsilon} d\tau' \, \nabla_\nu G_{\mathrm{ret}}(z^\mu, z^{\mu'})$$

- Studying and visualizing basic wave propagation in black hole spacetimes

- Many similar applications in gravity plus others (e.g., NS-BH inspirals)

Summary & Outlook

- Green's function methods have many advantages to offer but significant challenges to overcome to be practical
- Reduced-order surrogate modeling offers a promising way to use Green's functions efficiently and accurately for self-force calculations

Summary & Outlook

- Green's function methods have many advantages to offer but significant challenges to overcome to be practical
- Reduced-order surrogate modeling offers a promising way to use Green's functions efficiently and accurately for self-force calculations
- For a generic worldline, the surrogate is more than 15x faster to evaluate than solving the wave equation, with little loss of accuracy
- For r^* = constant worldlines, the surrogate is more than 750x faster to evaluate than solving the wave equation and ~150x smaller data

Summary & Outlook

- Green's function methods have many advantages to offer but significant challenges to overcome to be practical

- Reduced-order surrogate modeling offers a promising way to use Green's functions efficiently and accurately for self-force calculations

- For a generic worldline, the surrogate is more than 15x faster to evaluate than solving the wave equation, with little loss of accuracy

- For r^* = constant worldlines, the surrogate is more than 750x faster to evaluate than solving the wave equation and ~150x smaller data

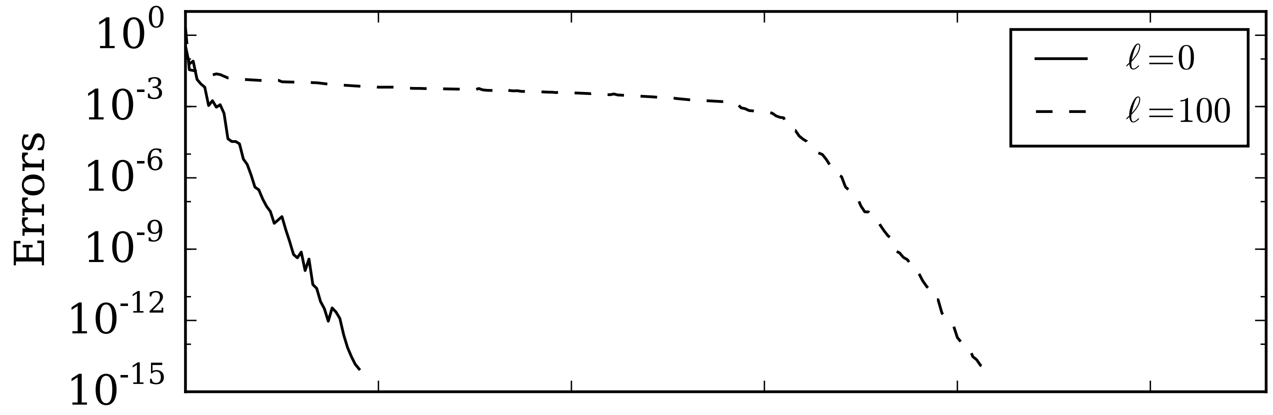
- Extending to Kerr spacetime is straightforward but may involve (much?) larger data sets because of extra parameters and reduced symmetry

- How to compute Green's function for gravitational perturbations?
  - Lorenz gauge has unstable non-radiative modes…
  - Accuracy and speed of "metric" reconstruction from curvature scalars?

Extra slides

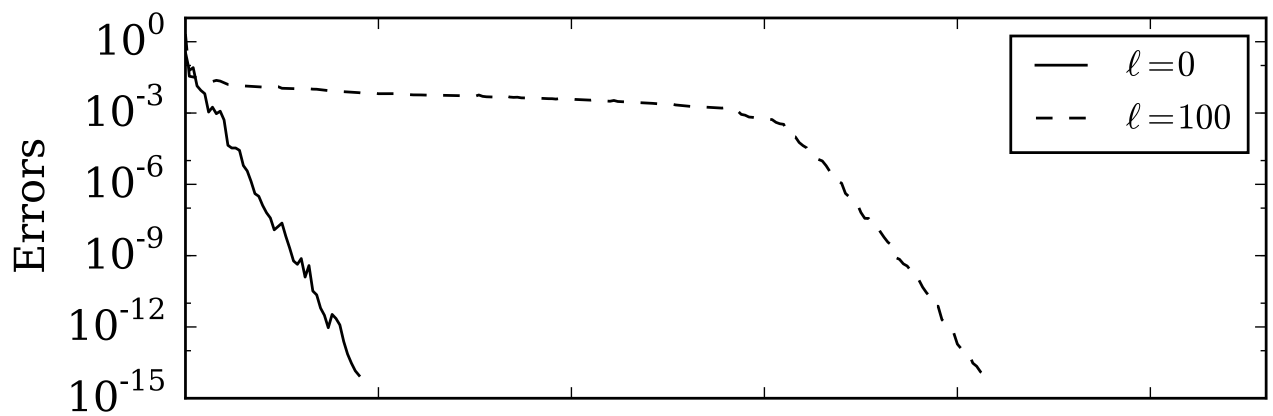
Improving the surrogate building strategy

The plateau in the max projection errors often hints that a different representation of the data may generate a more compact basis



Improving the surrogate building strategy

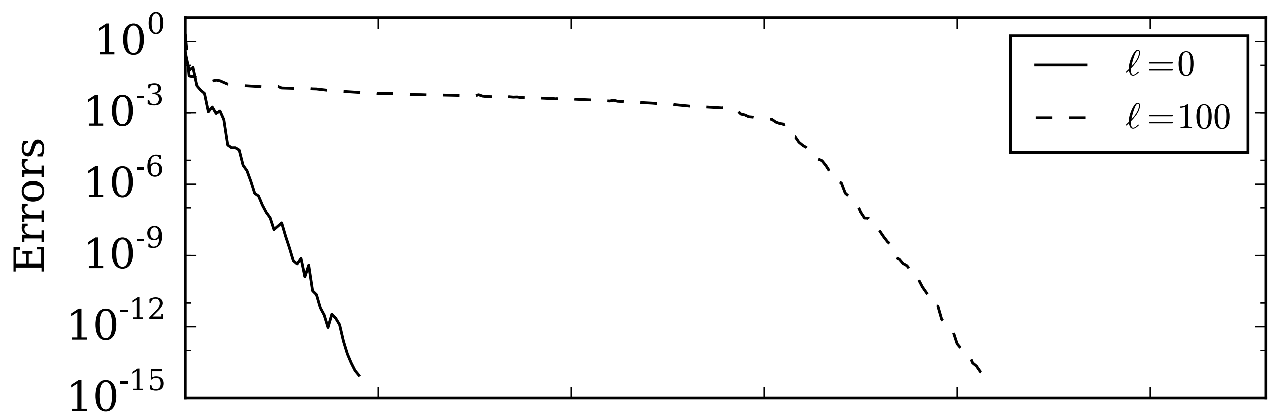
The plateau in the max projection errors often hints that a different representation of the data may generate a more compact basis



- Amplitude and phase representation of real-valued oscillating data via Hilbert transform
  - "Rippling" is a problem
  - Phase at initial times is difficult to estimate
  - Total basis sizes are often larger

Improving the surrogate building strategy

The plateau in the max projection errors often hints that a different representation of the data may generate a more compact basis



- Amplitude and phase representation of real-valued oscillating data via Hilbert transform
  - "Rippling" is a problem
  - Phase at initial times is difficult to estimate
  - Total basis sizes are often larger

- Some other way to represent the data?

Different and useful ways to parametrize the data?

- A more "natural" parametrization might be $\lambda = r'_*$ and regard (t, r_*) as the physical dimension

$$G_\ell(t, r_*; r'_*) \approx \sum_{i=1}^{N_\ell} B_i^\ell(t, r_*) G_\ell(T_i, R_{*i}; r'_*)$$

Different and useful ways to parametrize the data?

- A more "natural" parametrization might be $\lambda = r'_*$ and regard (t, r_*) as the physical dimension

$$G_\ell(t, r_*; r'_*) \approx \sum_{i=1}^{N_\ell} B_i^\ell(t, r_*) G_\ell(T_i, R_{*i}; r'_*)$$

$$F^\mu \sim q^2 P^{\mu\nu} \sum_{\ell=0}^{\ell_{\max}} \frac{1}{r'}(2\ell+1)e^{-\ell^2/2\ell_{\rm cut}^2} \int dt'\, P_\ell(\cos\gamma(t)) \sum_{i=1}^{N_\ell} B_i^\ell(t, r_*(t)) G_\ell(T_i, R_{*i}; r'_*)$$

Different and useful ways to parametrize the data?

- A more "natural" parametrization might be $\lambda = r'_*$ and regard (t, r_*) as the physical dimension

$$G_\ell(t, r_*; r'_*) \approx \sum_{i=1}^{N_\ell} B_i^\ell(t, r_*) G_\ell(T_i, R_{*i}; r'_*)$$

$$F^\mu \sim q^2 P^{\mu\nu} \sum_{\ell=0}^{\ell_{\rm max}} \frac{1}{r'}(2\ell+1) e^{-\ell^2/2\ell_{\rm cut}^2} \int dt'\, P_\ell(\cos\gamma(t)) \sum_{i=1}^{N_\ell} B_i^\ell(t, r_*(t)) G_\ell(T_i, R_{*i}; r'_*)$$

$$= q^2 P^{\mu\nu} \sum_{\ell=0}^{\ell_{\rm max}} \frac{1}{r'}(2\ell+1) e^{-\ell^2/2\ell_{\rm cut}^2} \sum_{i=1}^{N_\ell} G_\ell(T_i, R_{*i}; r'_*) \int dt'\, P_\ell(\cos\gamma(t)) B_i^\ell(t, r_*(t))$$

Different and useful ways to parametrize the data?

- A more "natural" parametrization might be $\lambda = r'_*$ and regard (t, r_*) as the physical dimension

$$G_\ell(t, r_*; r'_*) \approx \sum_{i=1}^{N_\ell} B_i^\ell(t, r_*) G_\ell(T_i, R_{*i}; r'_*)$$

$$F^\mu \sim q^2 P^{\mu\nu} \sum_{\ell=0}^{\ell_{\max}} \frac{1}{r'}(2\ell+1) e^{-\ell^2/2\ell_{\mathrm{cut}}^2} \int dt'\, P_\ell(\cos\gamma(t)) \sum_{i=1}^{N_\ell} B_i^\ell(t, r_*(t)) G_\ell(T_i, R_{*i}; r'_*)$$

$$= q^2 P^{\mu\nu} \sum_{\ell=0}^{\ell_{\max}} \frac{1}{r'}(2\ell+1) e^{-\ell^2/2\ell_{\mathrm{cut}}^2} \sum_{i=1}^{N_\ell} G_\ell(T_i, R_{*i}; r'_*) \int dt'\, P_\ell(\cos\gamma(t)) B_i^\ell(t, r_*(t))$$

- There are some hints that including mode number may provide significant data reduction but not yet known how to evaluate surrogate

Different and useful ways to parametrize the data?

- A more "natural" parametrization might be $\lambda = r'_*$ and regard (t, r_*) as the physical dimension

$$G_\ell(t, r_*; r'_*) \approx \sum_{i=1}^{N_\ell} B_i^\ell(t, r_*) G_\ell(T_i, R_{*i}; r'_*)$$

$$F^\mu \sim q^2 P^{\mu\nu} \sum_{\ell=0}^{\ell_{\max}} \frac{1}{r'}(2\ell+1) e^{-\ell^2/2\ell_{\rm cut}^2} \int dt'\, P_\ell(\cos\gamma(t)) \sum_{i=1}^{N_\ell} B_i^\ell(t, r_*(t)) G_\ell(T_i, R_{*i}; r'_*)$$

$$= q^2 P^{\mu\nu} \sum_{\ell=0}^{\ell_{\max}} \frac{1}{r'}(2\ell+1) e^{-\ell^2/2\ell_{\rm cut}^2} \sum_{i=1}^{N_\ell} G_\ell(T_i, R_{*i}; r'_*) \int dt'\, P_\ell(\cos\gamma(t)) B_i^\ell(t, r_*(t))$$

- There are some hints that including mode number may provide significant data reduction but not yet known how to evaluate surrogate

Maybe try "invasive" approaches that project the wave equation onto the small vector space spanned by the basis