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)

Capra 19, Meudon, Paris, June 29, 2016

Caltech

Green's functions

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

Green's functions

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

What are the advantages of using Green's functions?

- Compute only once
- Nearly all physical quantities of interest are calculated via convolution integrals
- Arbitrary motion for self-force
- Geometric interpretation *(see also J. Thornburg's talk)*
- Higher-order self-force
- Self-consistent (higher-order) self-forced evolution
- Self-consistent inspiral waveforms
- Arguably straightforward to implement once known

What are the disadvantages of using Green's functions?

- Computationally expensive
- Large data sets (*G* is a bitensor!)
- Gravitational perturbations: Instabilities? Metric reconstruction?

What are the disadvantages of using Green’s functions?

- Computationally expensive
- Large data sets (*G* is a bitensor!)
- Gravitational perturbations: Instabilities? Metric reconstruction?

“Direct integration of the MiSaTaQuWa equation:

> *“Possible in principle, not efficient in practice.”*
> — *Mino (2004)*”

— Isoyama (2016)

What are the disadvantages of using Green’s functions?

- Computationally expensive
- Large data sets (*G* is a bitensor!)
- Gravitational perturbations: Instabilities? Metric reconstruction?

“Direct integration of the MiSaTaQuWa equation:

> *“Possible in principle, not efficient in practice.”*
> *— Mino (2004)*”

— Isoyama (2016)

Goal:

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

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

- 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

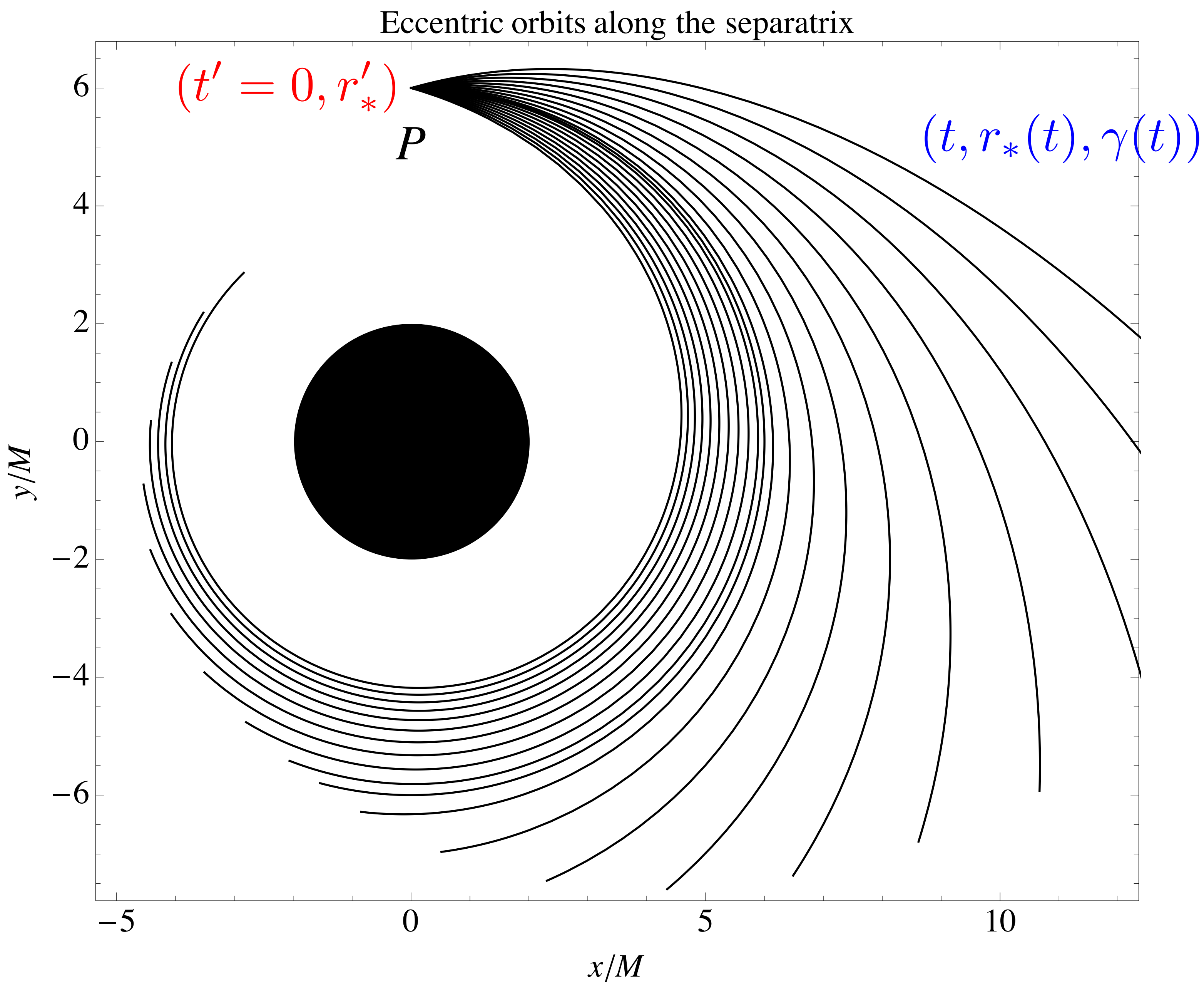
Zenginoglu & CRG (12)

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

- 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



Wardell, 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 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 used 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*)
  (in prep)

Surrogate models for gravitational waveforms have been used 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*)
  (in prep)

However, the steps for building a Green's function surrogate are necessarily a little 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)

- Coordinates: (t, r_*, θ, ϕ)

Surrogate building: Initial stuff

see Wardell, CRG et al (14)

- 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)

- 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)

- 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)

- 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)

- 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

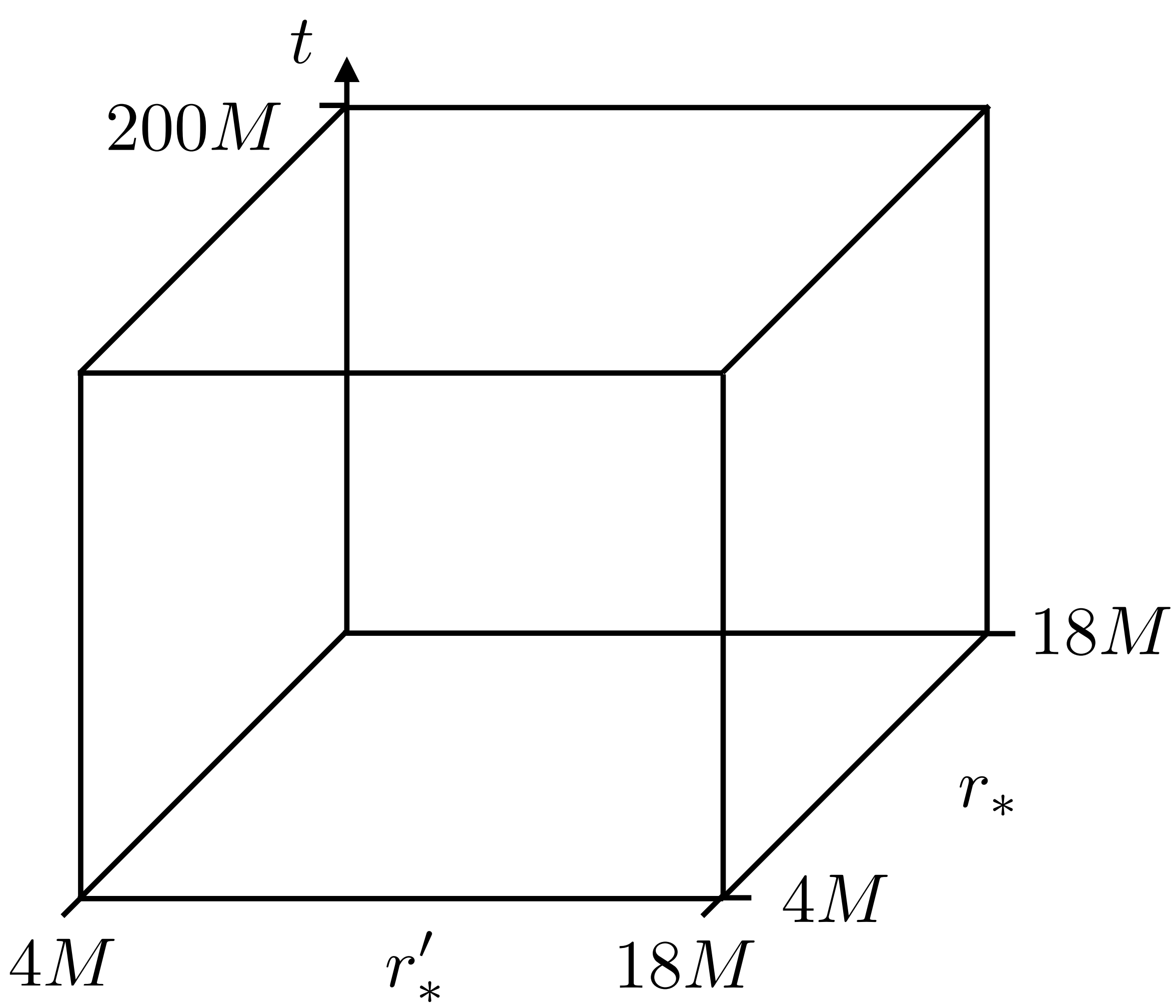
$$\vec{\lambda} = (r_*, r'_*)$$

- Solutions stored to disk at increments

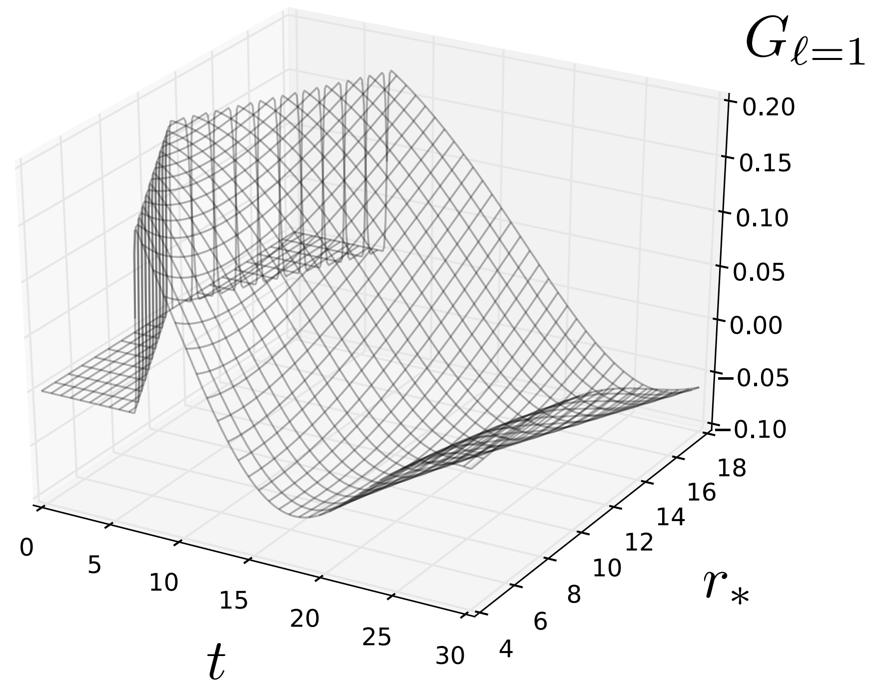
$$\Delta t = 0.1M$$

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

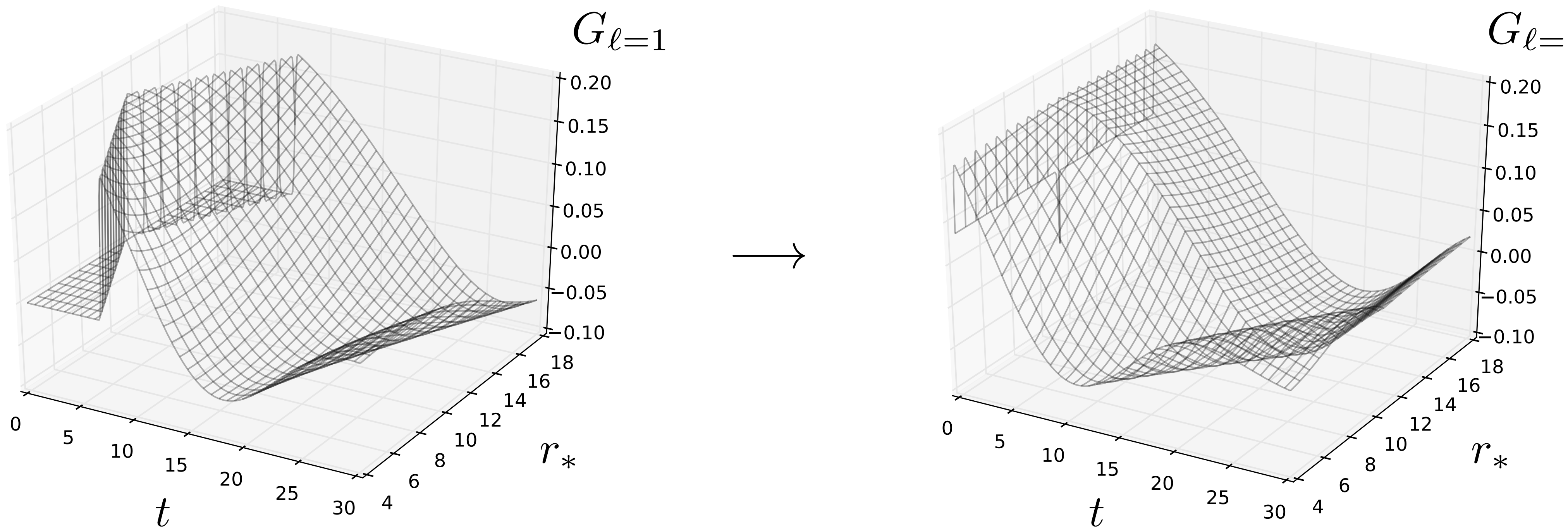
- ~30GB saved to disk



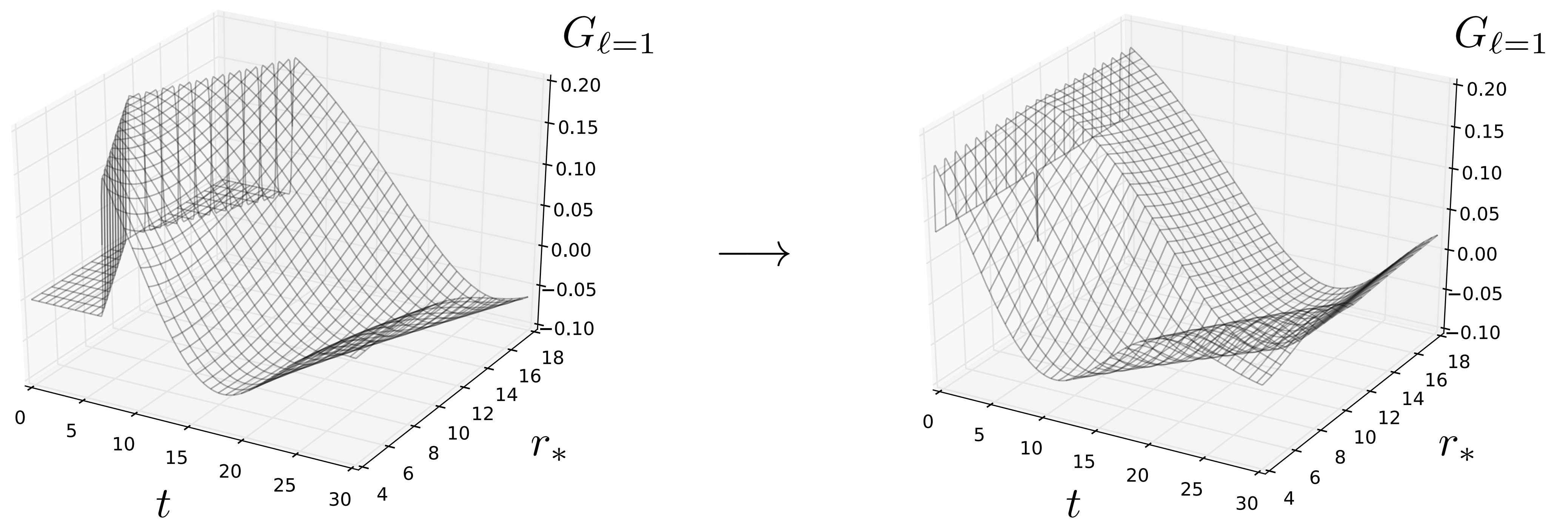
- Reduce known features by analytically time-shifting each series by light travel time from source point to field point, $t \to t - |r_* - r'_*|$



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



- 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, because only a finite number of modes can be computed we introduce a smoothing factor *Wardell, CRG et al (14)*

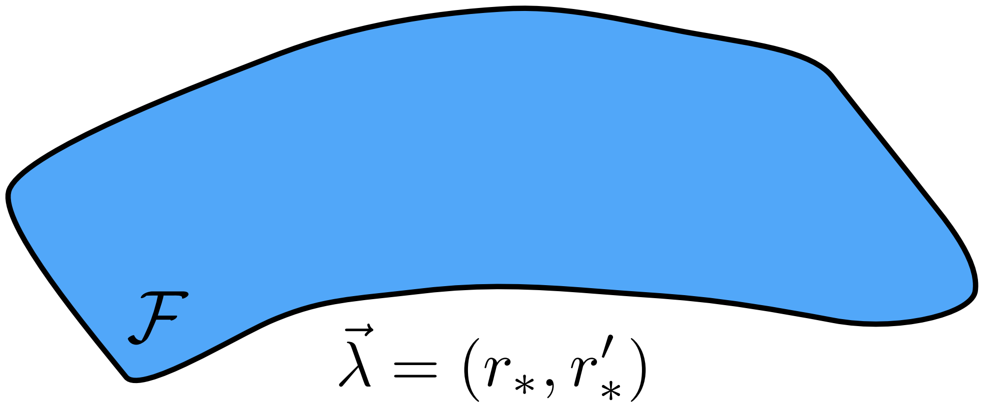
$$G(x^\alpha, x'^\alpha) \approx \frac{1}{rr'} \sum_{\ell=0}^{\ell_{\max}} P_\ell(\cos\theta)(2\ell+1)e^{-\ell^2/2\ell_{\rm cut}^2} G_\ell(t-t'; r_*, r'_*)$$

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

1) Reduced basis via greedy algorithm

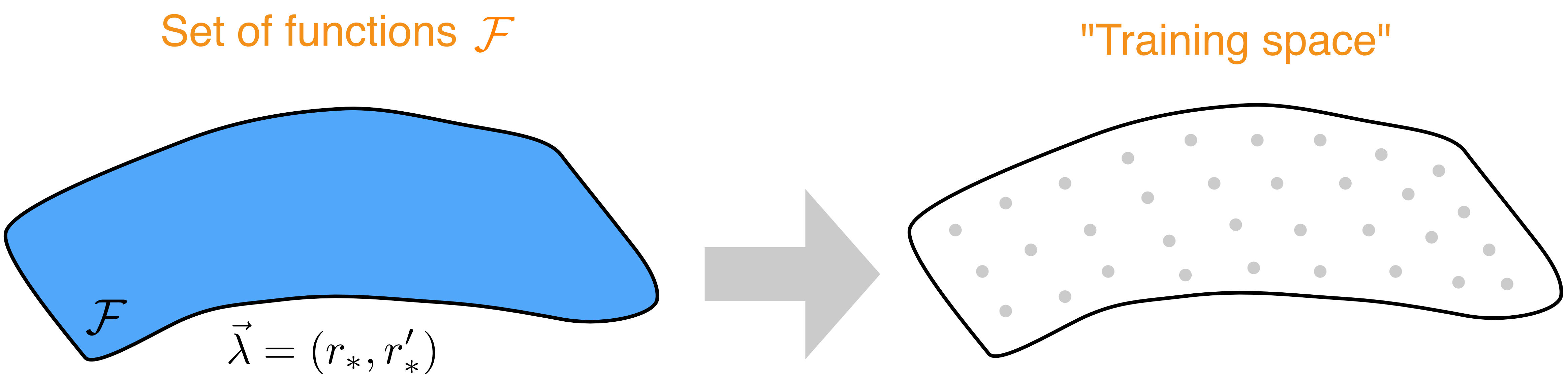
Can find a linear approximation space that is nearly optimal

Set of functions $\mathcal{F}$



1) Reduced basis via greedy algorithm

Can find a linear approximation space that is nearly optimal

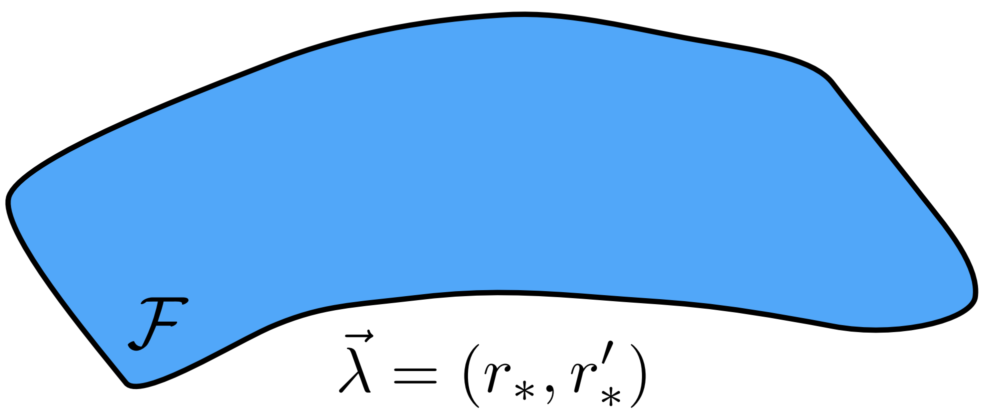


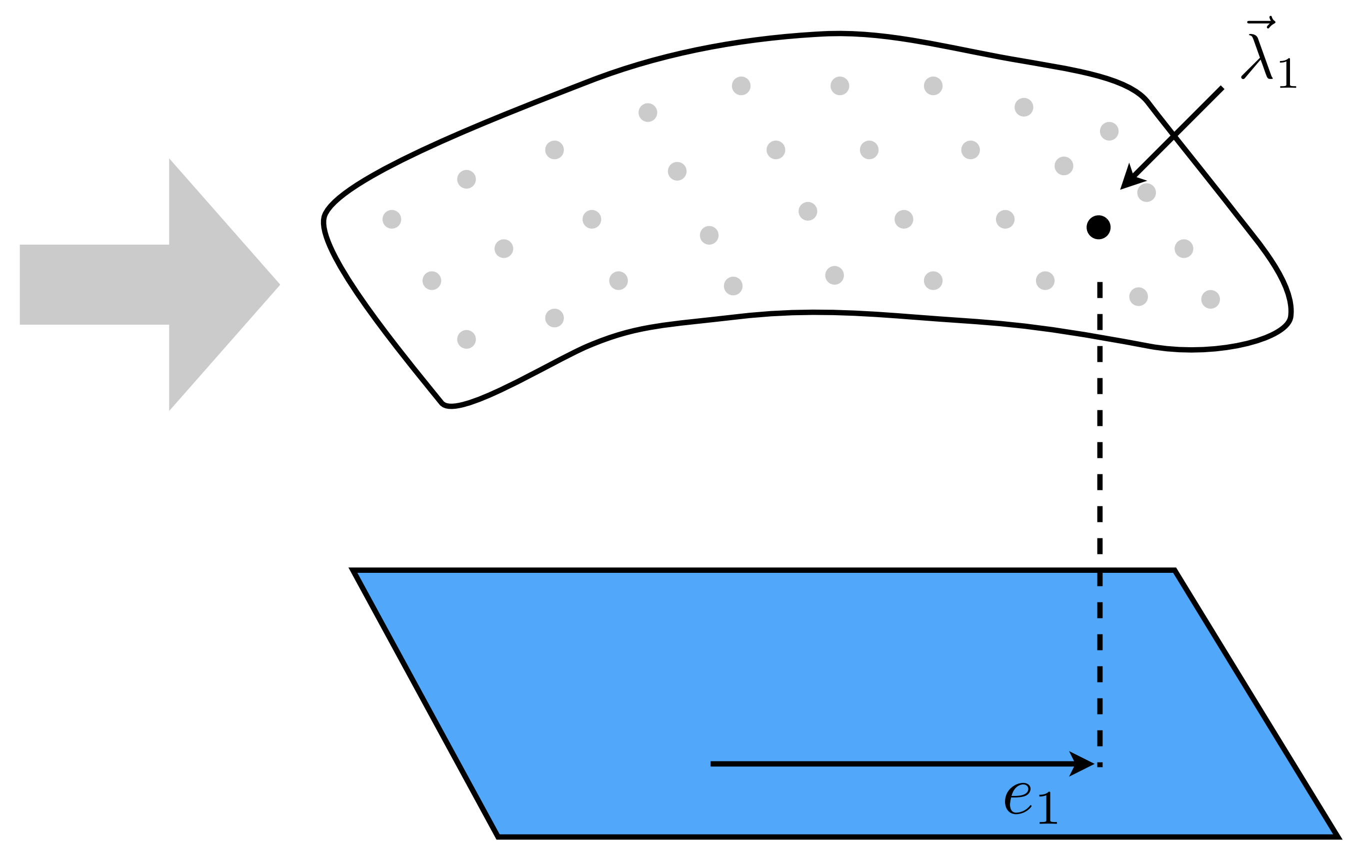
1) Reduced basis via greedy algorithm

Can find a linear approximation space that is nearly optimal

Set of functions $\mathcal{F}$

"Training space"



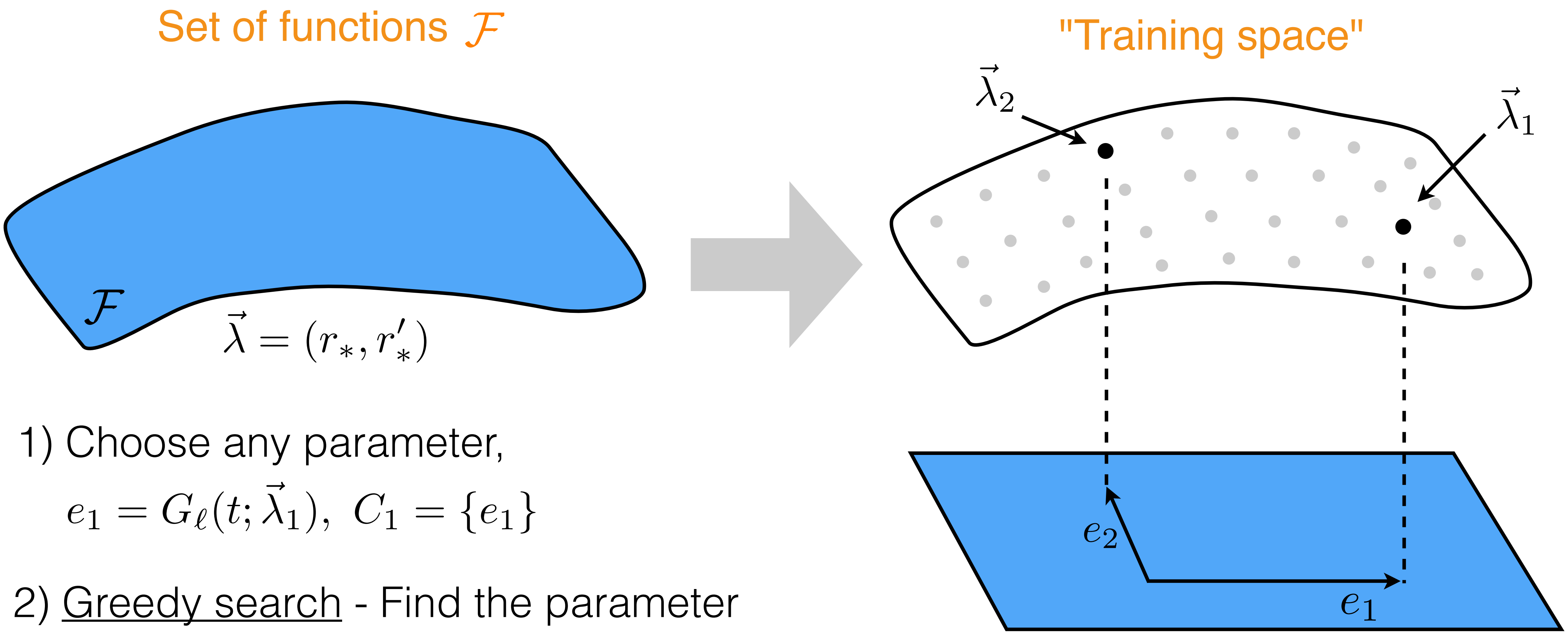


1) Choose any parameter,

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

1) Reduced basis via greedy algorithm

Can find a linear approximation space that is nearly optimal



1) Choose any parameter,

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

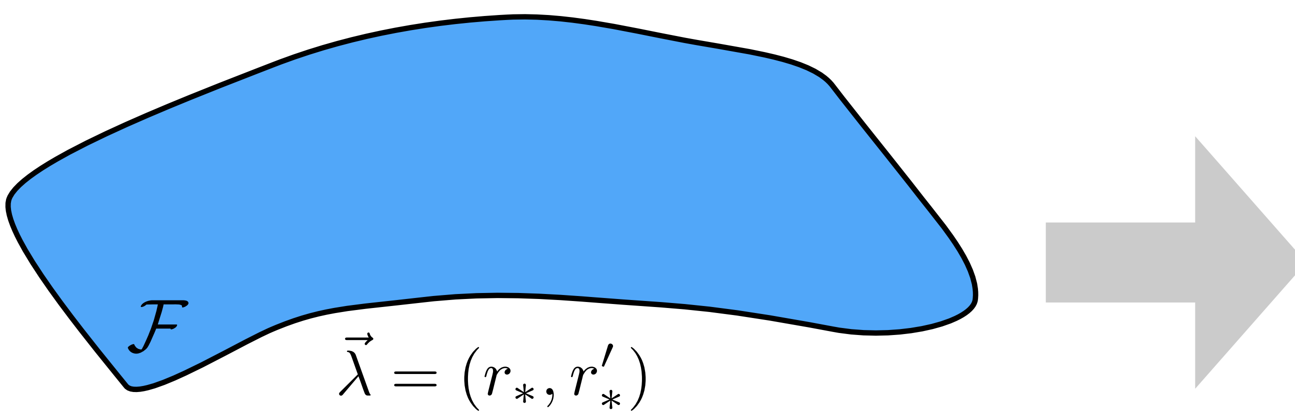
2) Greedy search - Find the parameter that maximizes:

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

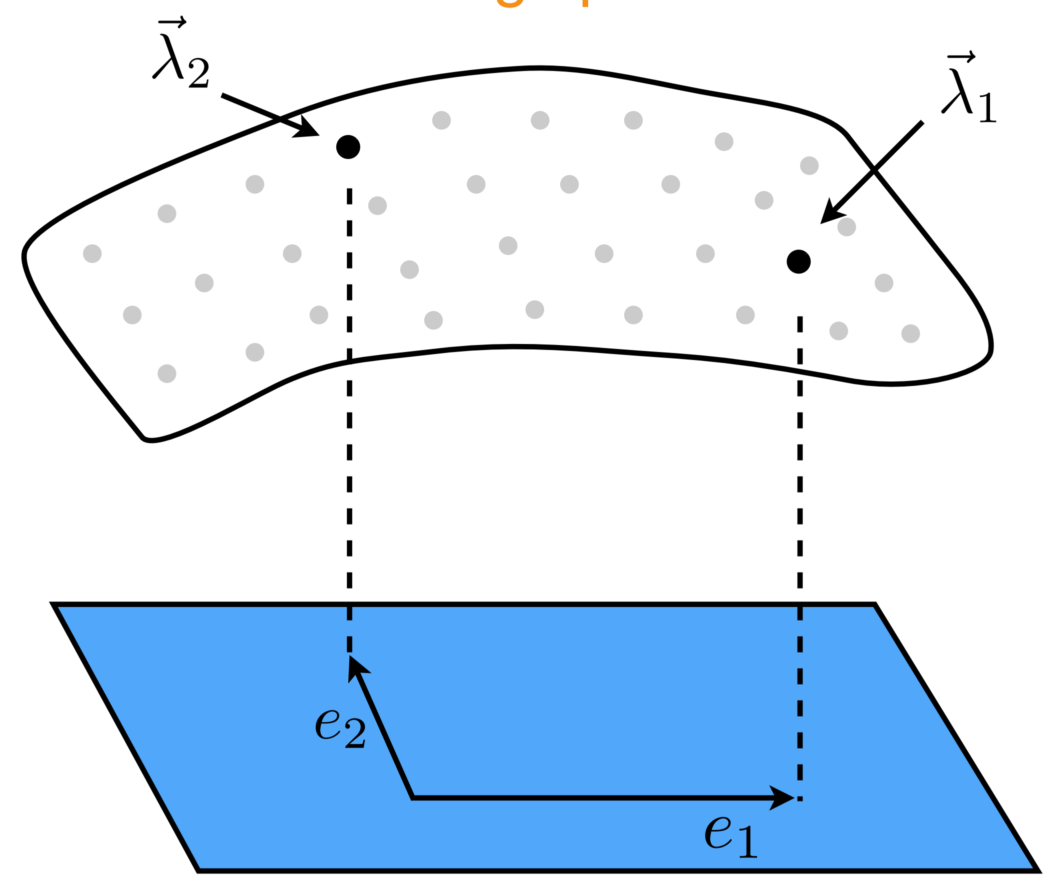
1) Reduced basis via greedy algorithm

Can find a linear approximation space that is nearly optimal

Set of functions $\mathcal{F}$



"Training space"



1) Choose any parameter,

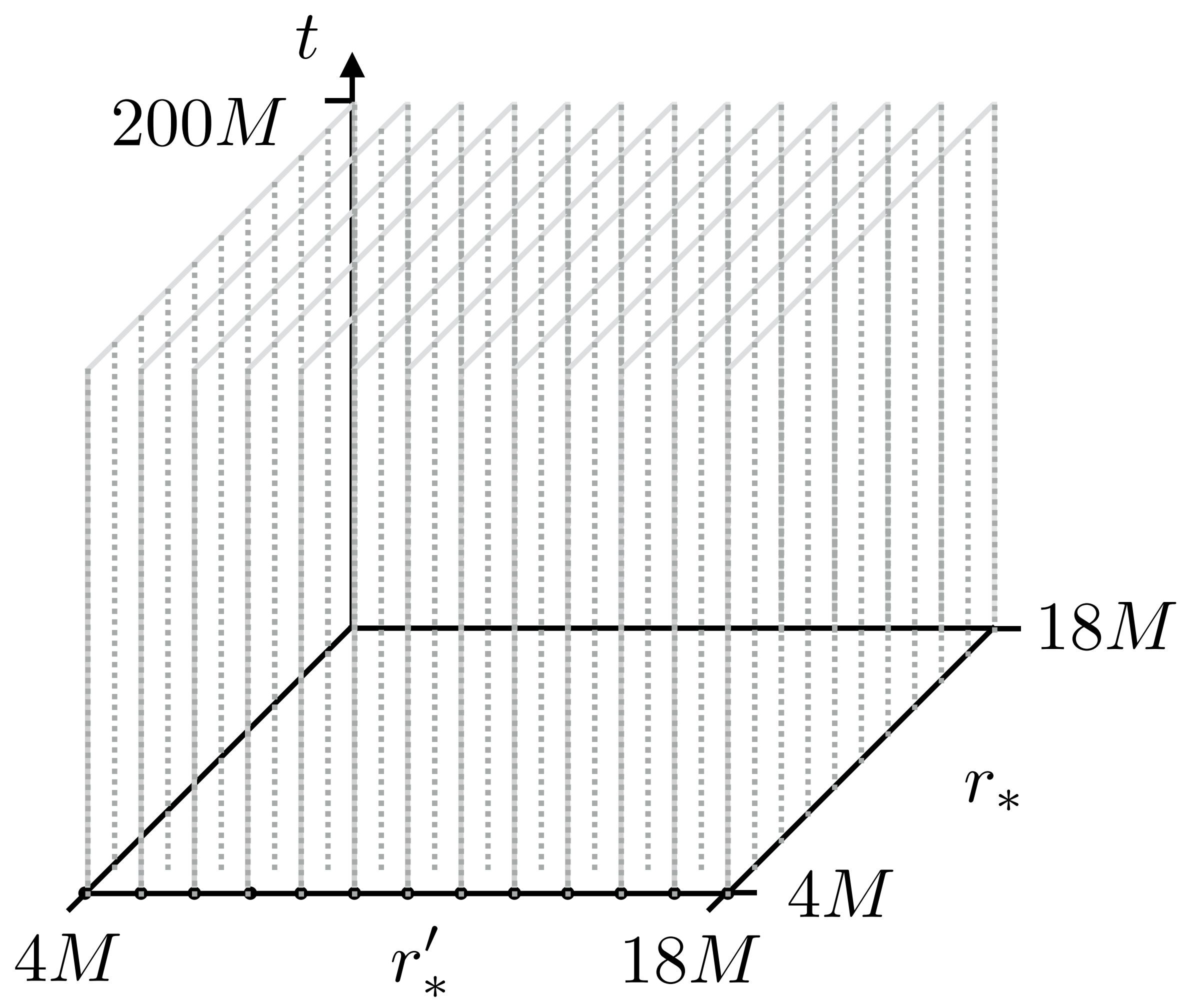
$$e_1 = G_\ell(t; \vec{\lambda}_1),\ C_1 = \{e_1\}$$

2) Greedy search - Find the parameter that maximizes:

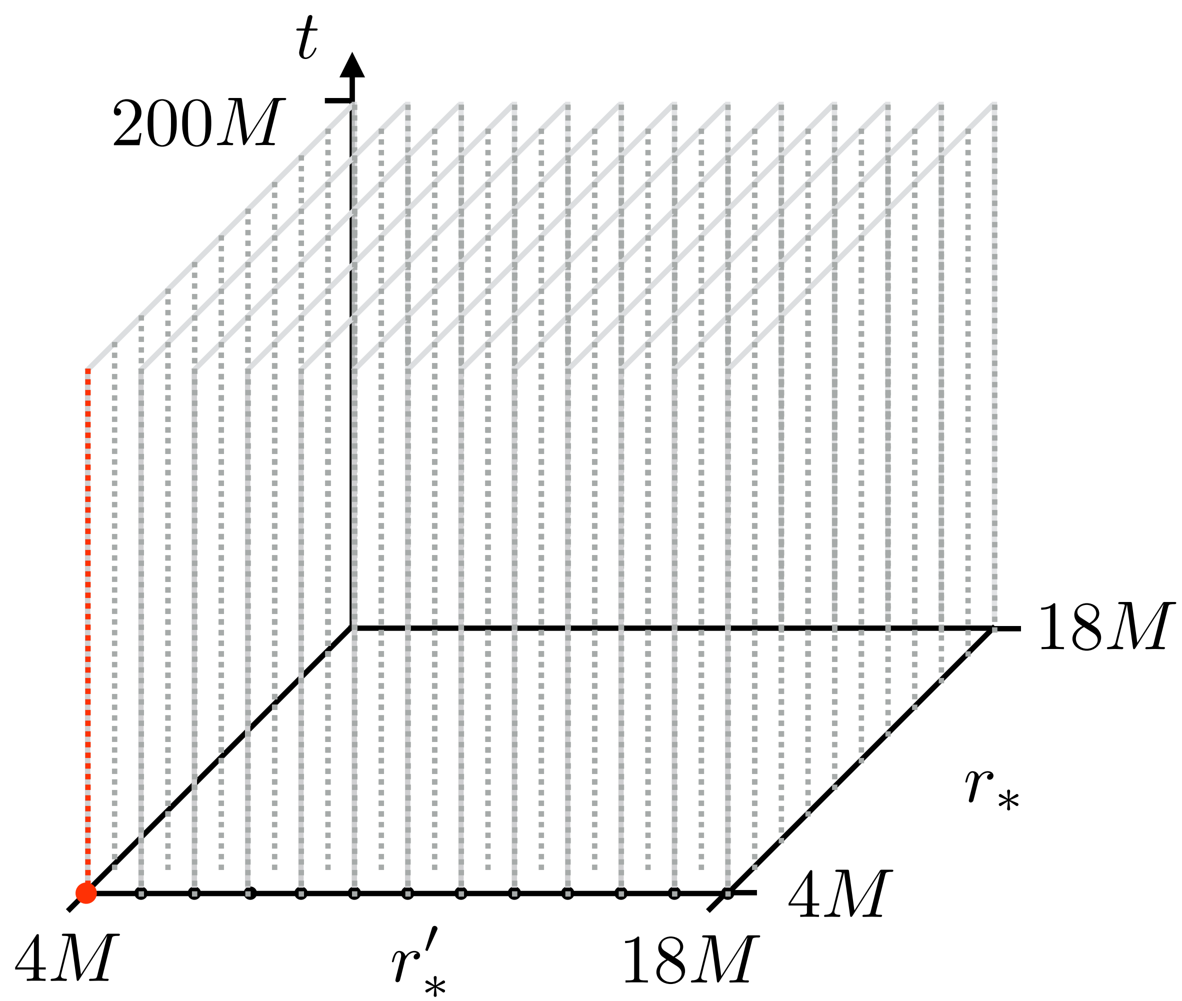
$$\max_t \left| G_\ell(t; \vec{\lambda}) - P_1[G_\ell(t; \vec{\lambda})] \right|,\ P_1[\cdot] = e_1 \langle e_1, \cdot \rangle$$

3) Orthogonalization to get basis vector e_2

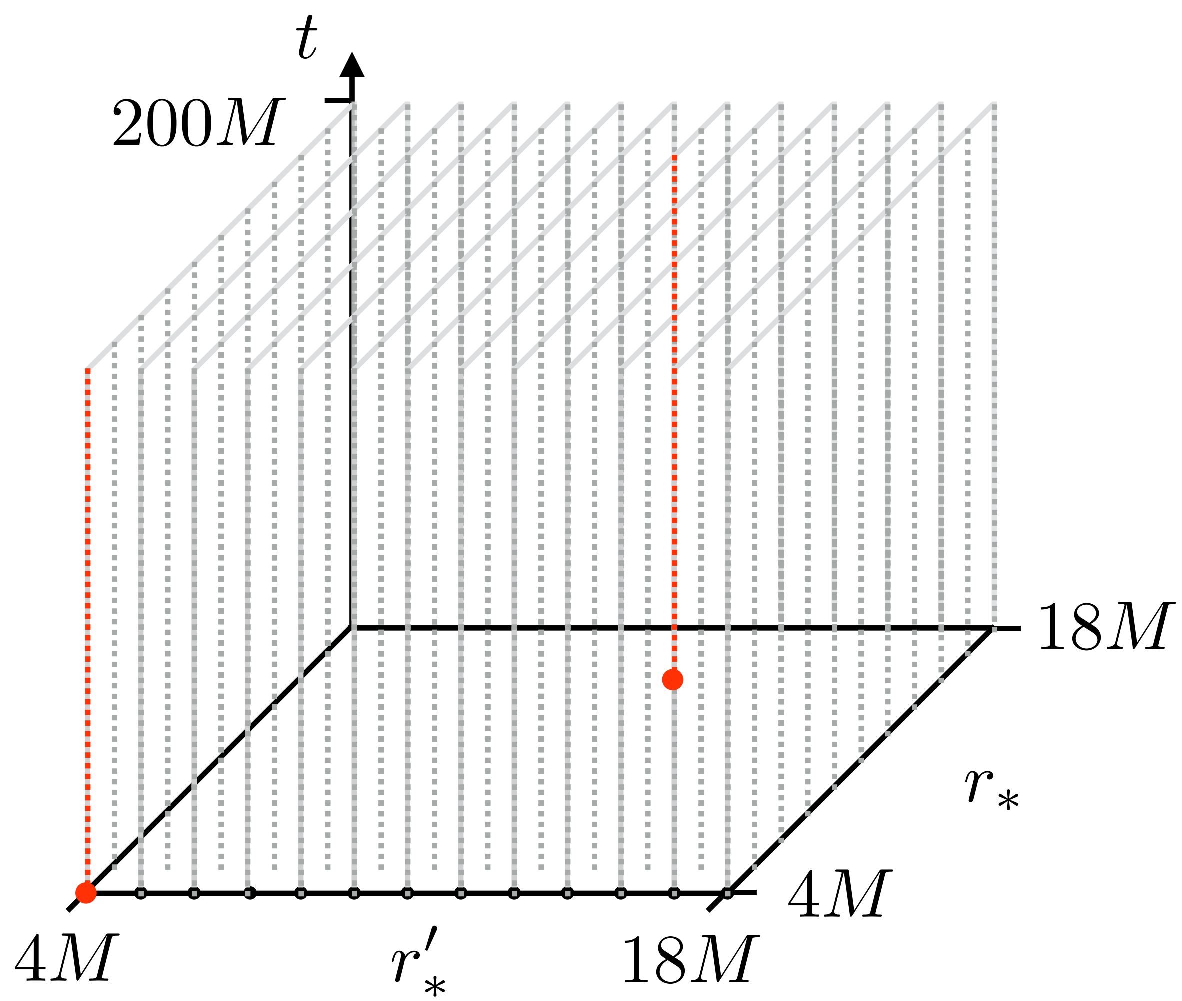
$$C_2 = \{e_1, e_2\},\ C_1 \subset C_2$$



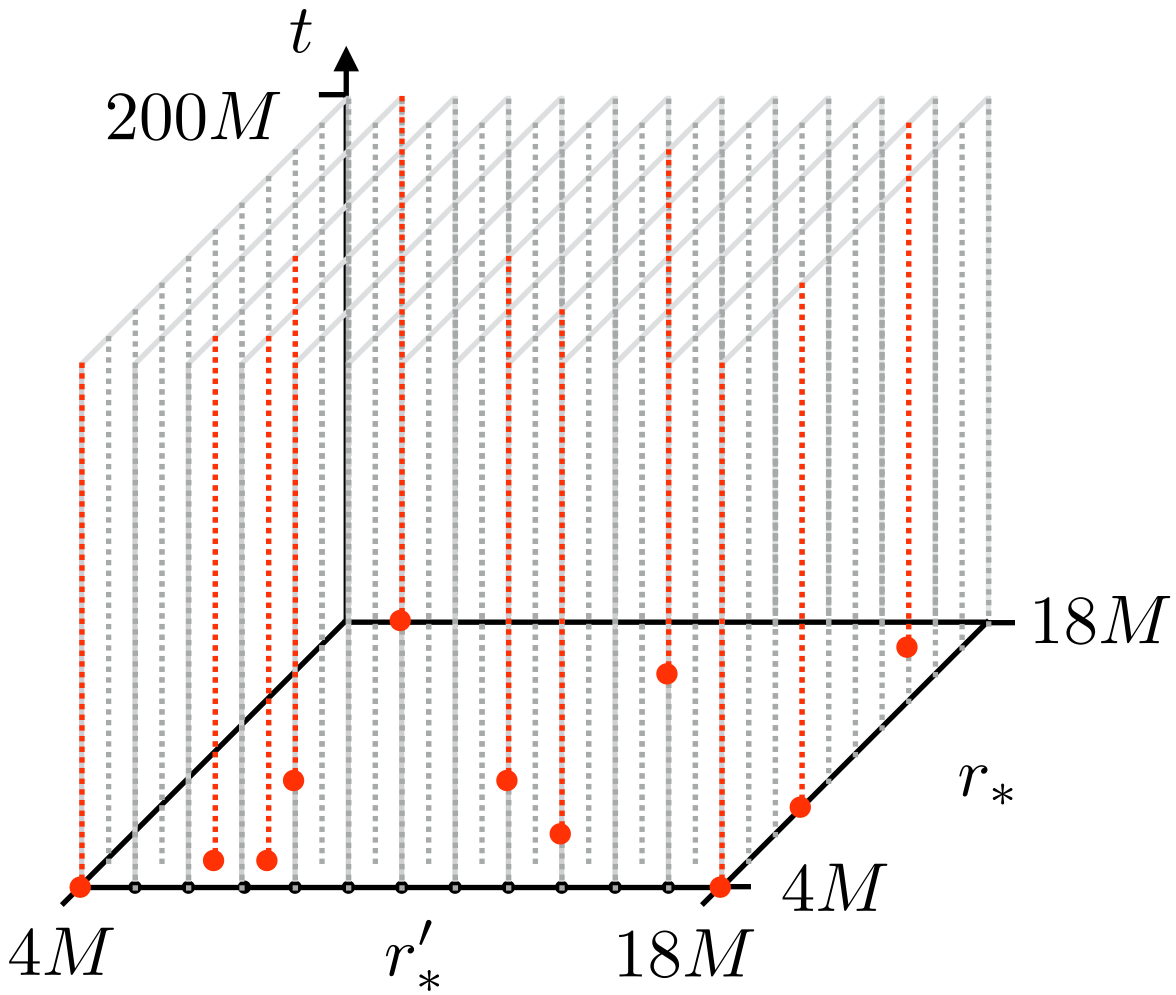
t
$200M$
$18M$
r_*
$4M$
$4M$
r'_*
$18M$



t
$200M$
$18M$
r_*
$4M$
$4M$
r'_*
$18M$

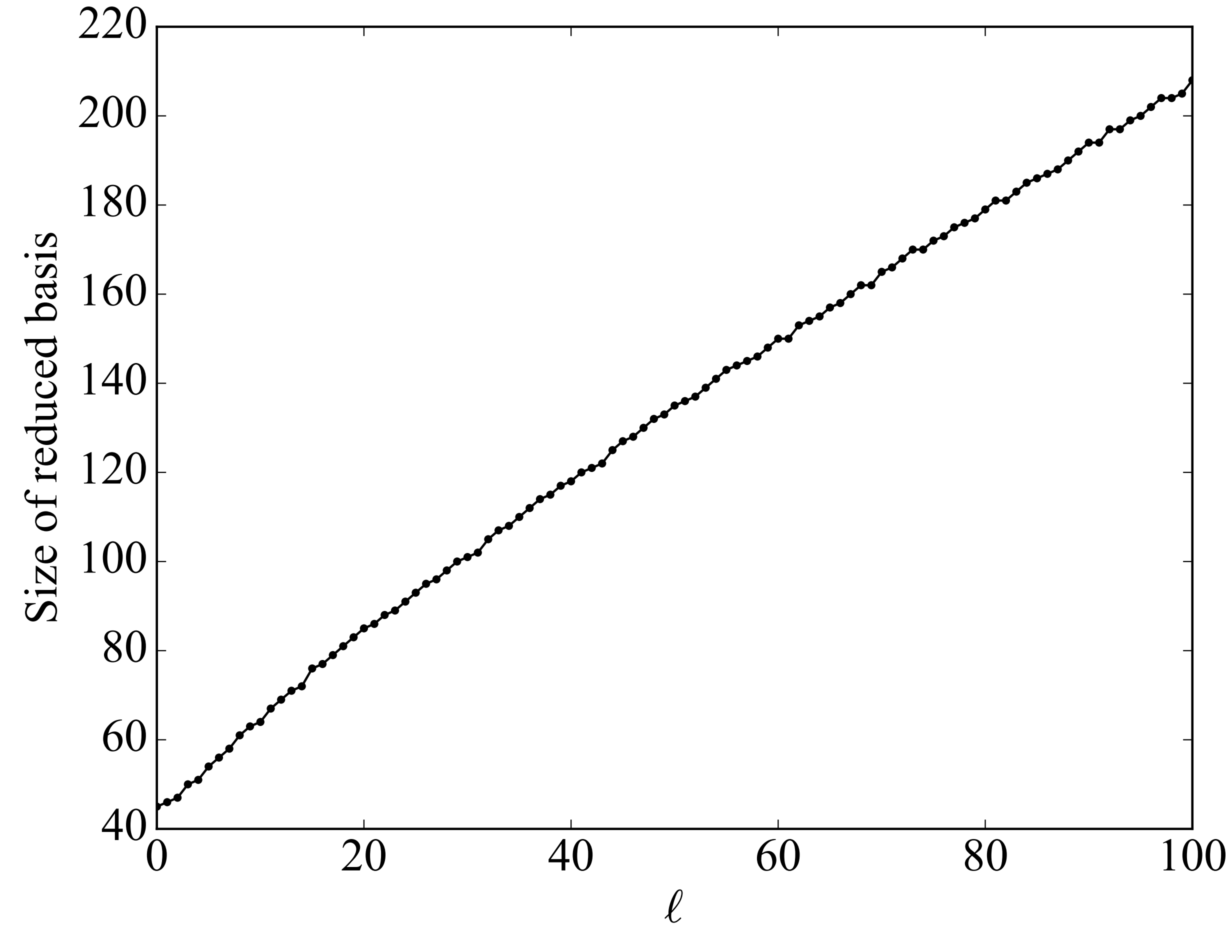


t
$200M$
$18M$
r_*
$4M$
$4M$
r'_*
$18M$

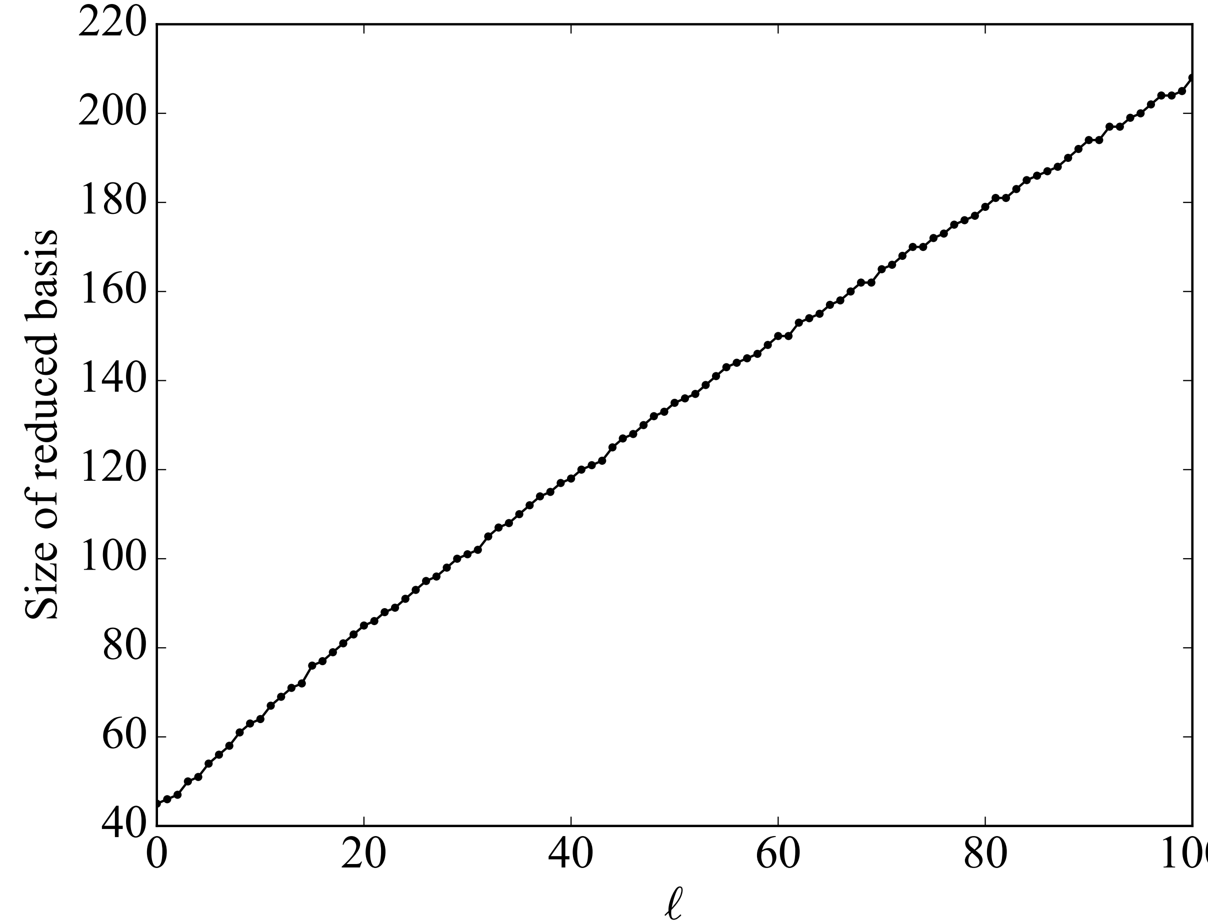


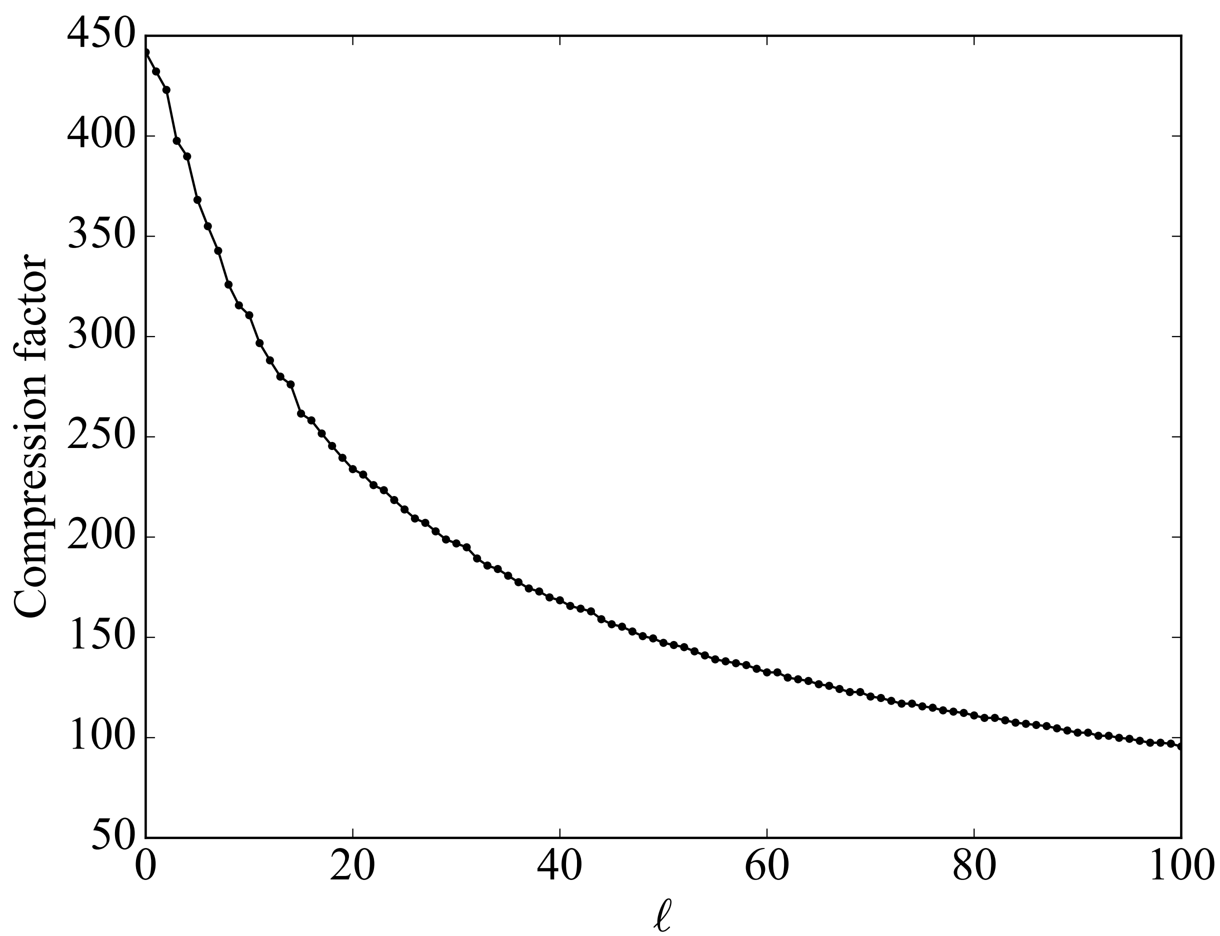
t
200M
18M
r_*
4M
4M
r'_*
18M

Basis size grows nearly linearly
mode number



Basis size grows nearly linearly mode number



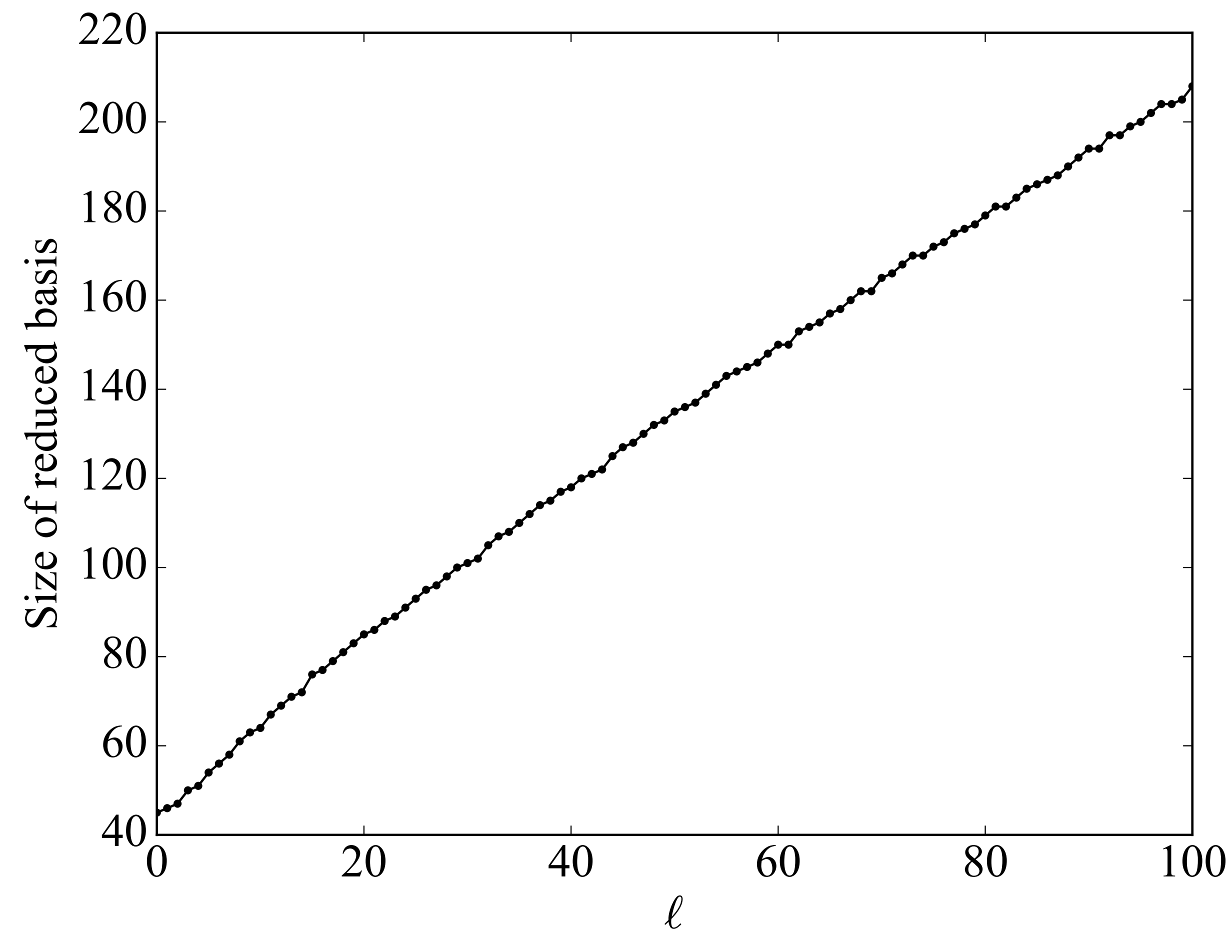


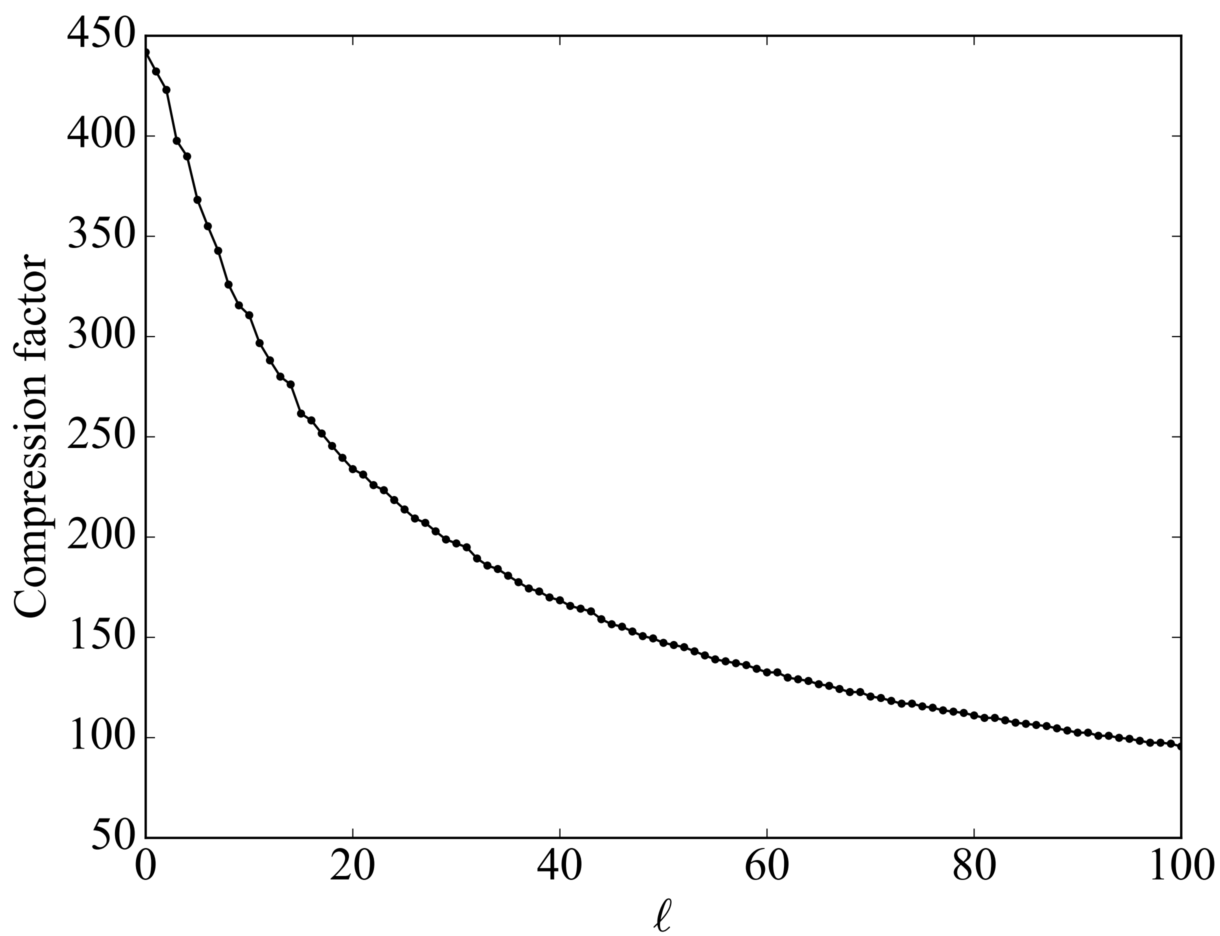
Total compression factor:

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

$$\approx 151$$

Basis size grows nearly linearly mode number



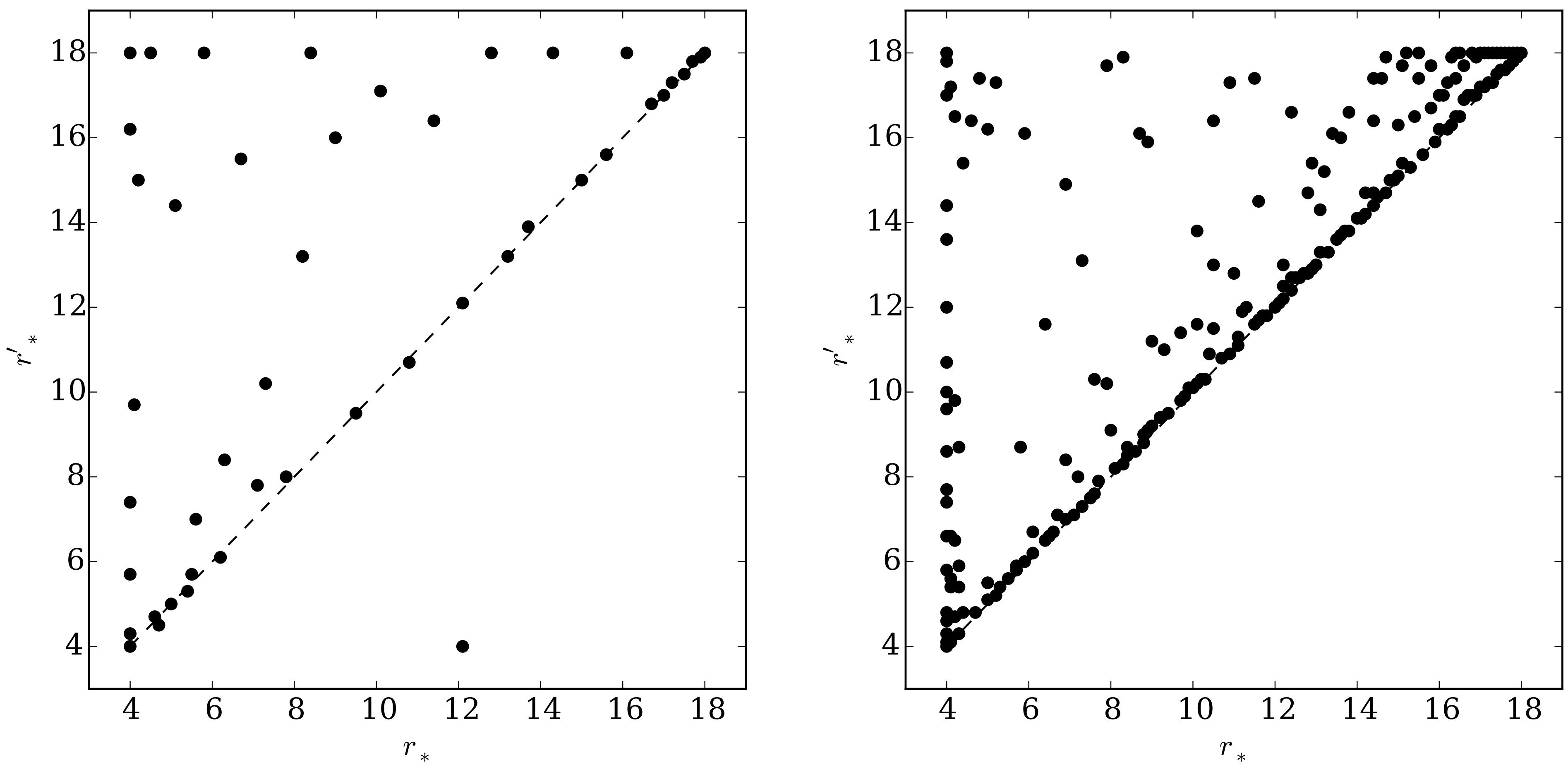


Total compression factor:

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

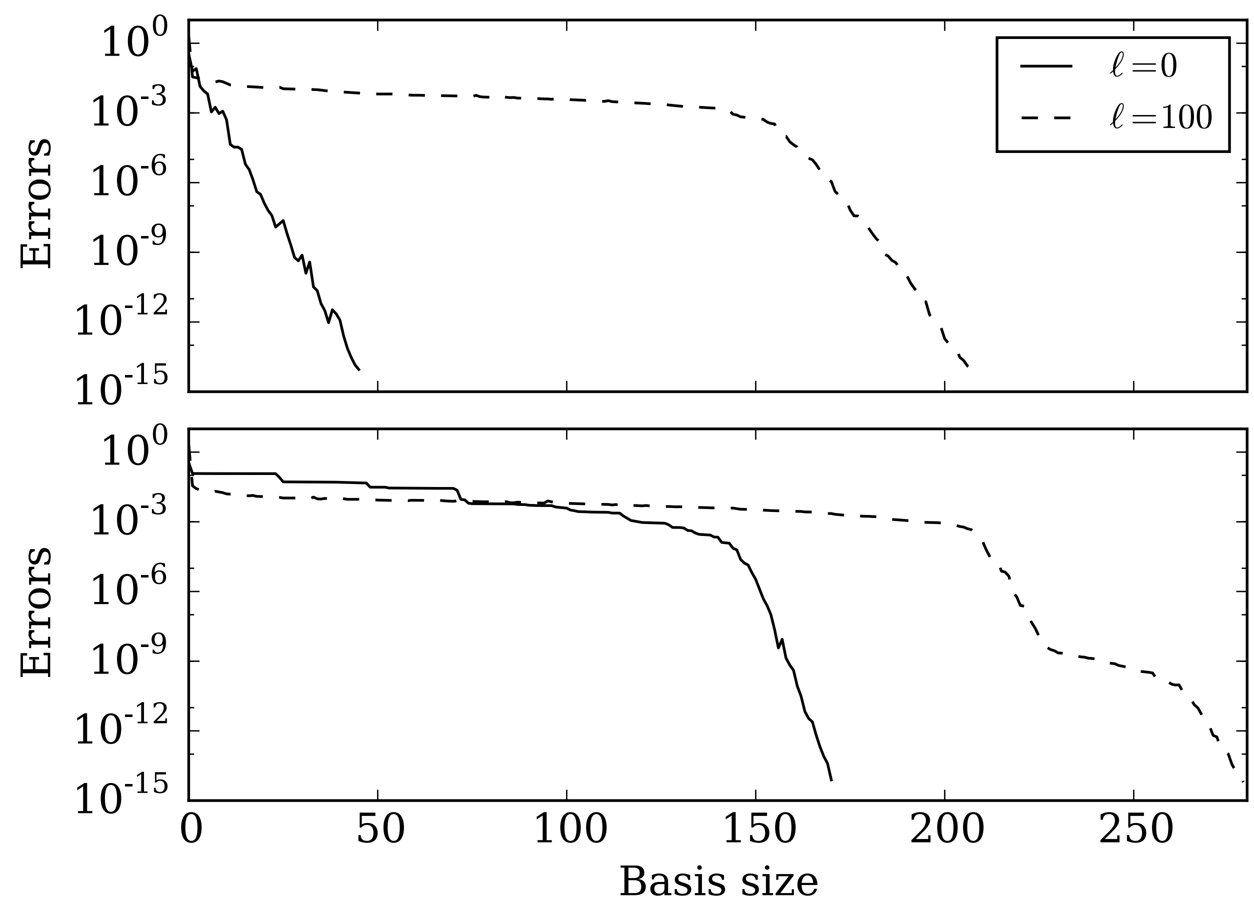
$$\approx 151$$

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:

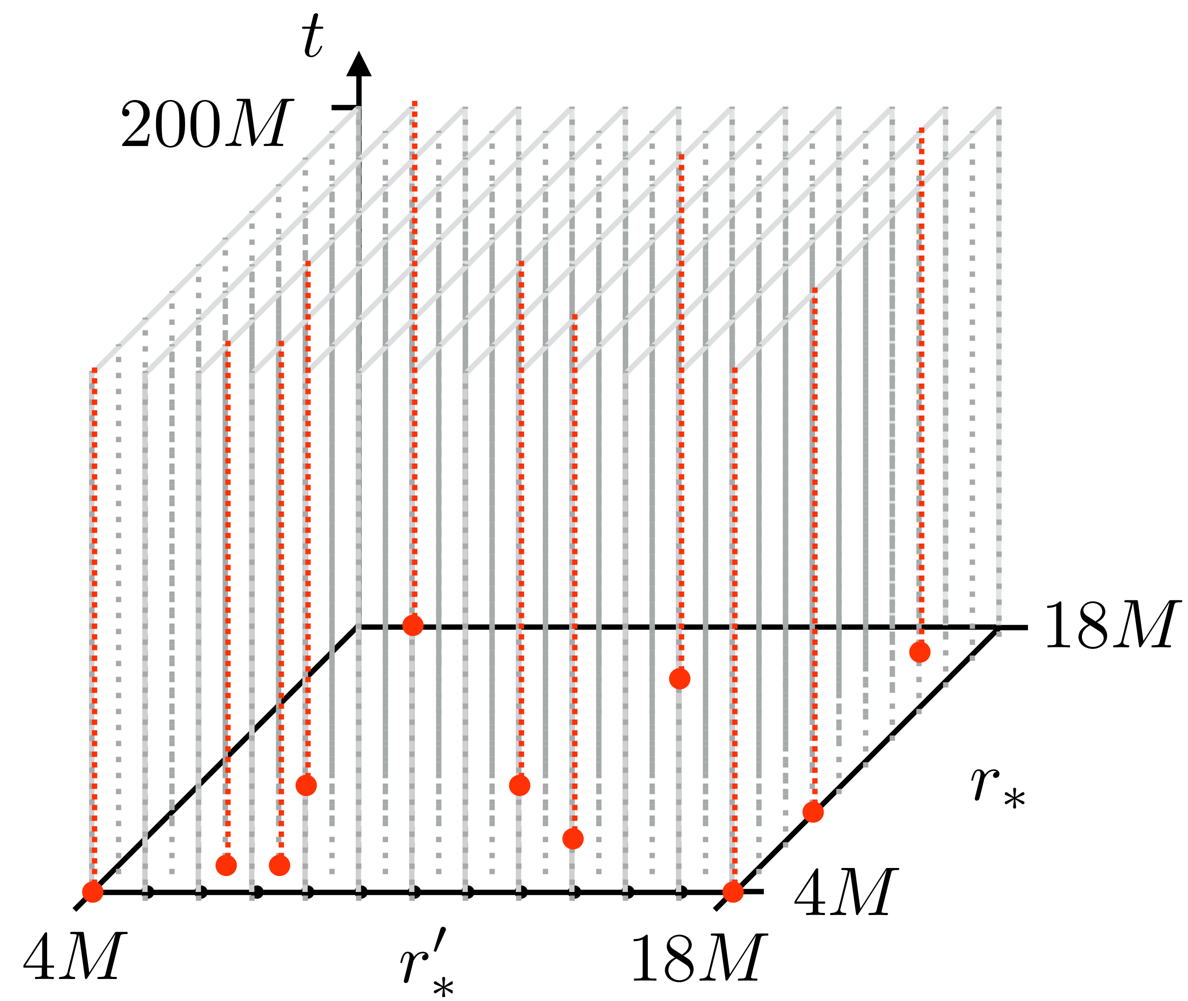


2) Empirical interpolation

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)

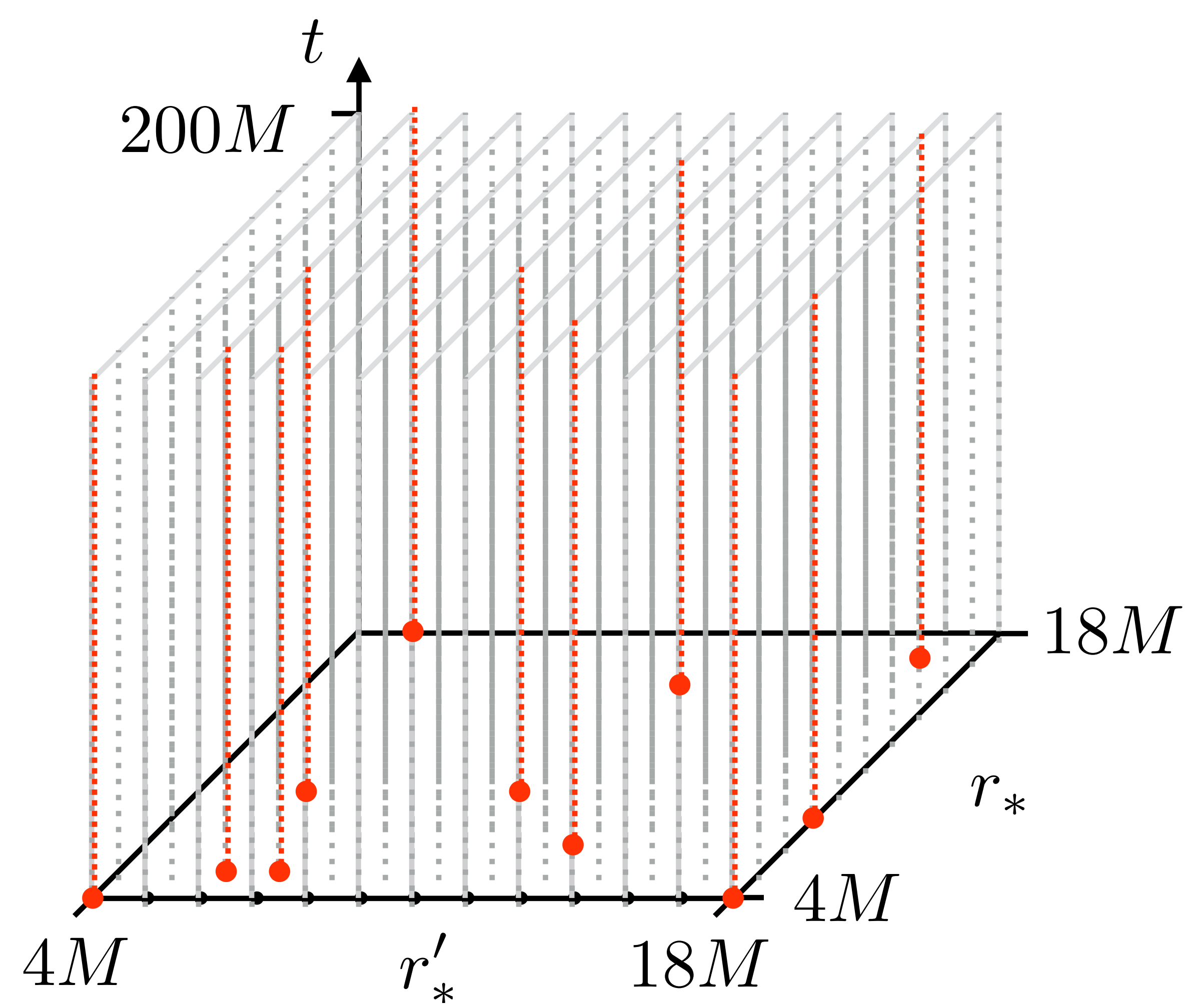
RB approximation:

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

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:

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

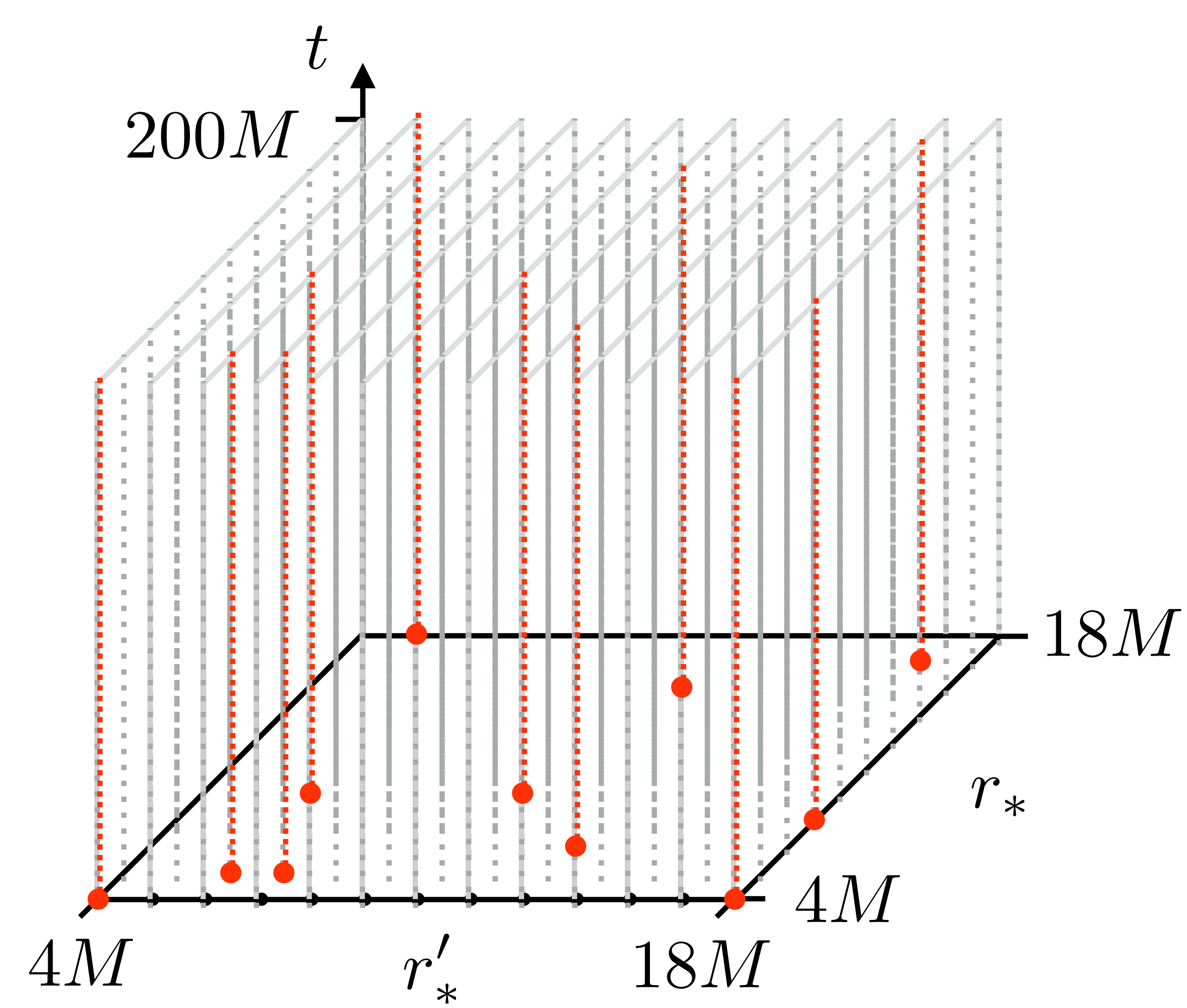
At n time subsamples of data, $\{T_i\}_{i=1}^n$ the coefficients can be solved

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

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

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

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



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

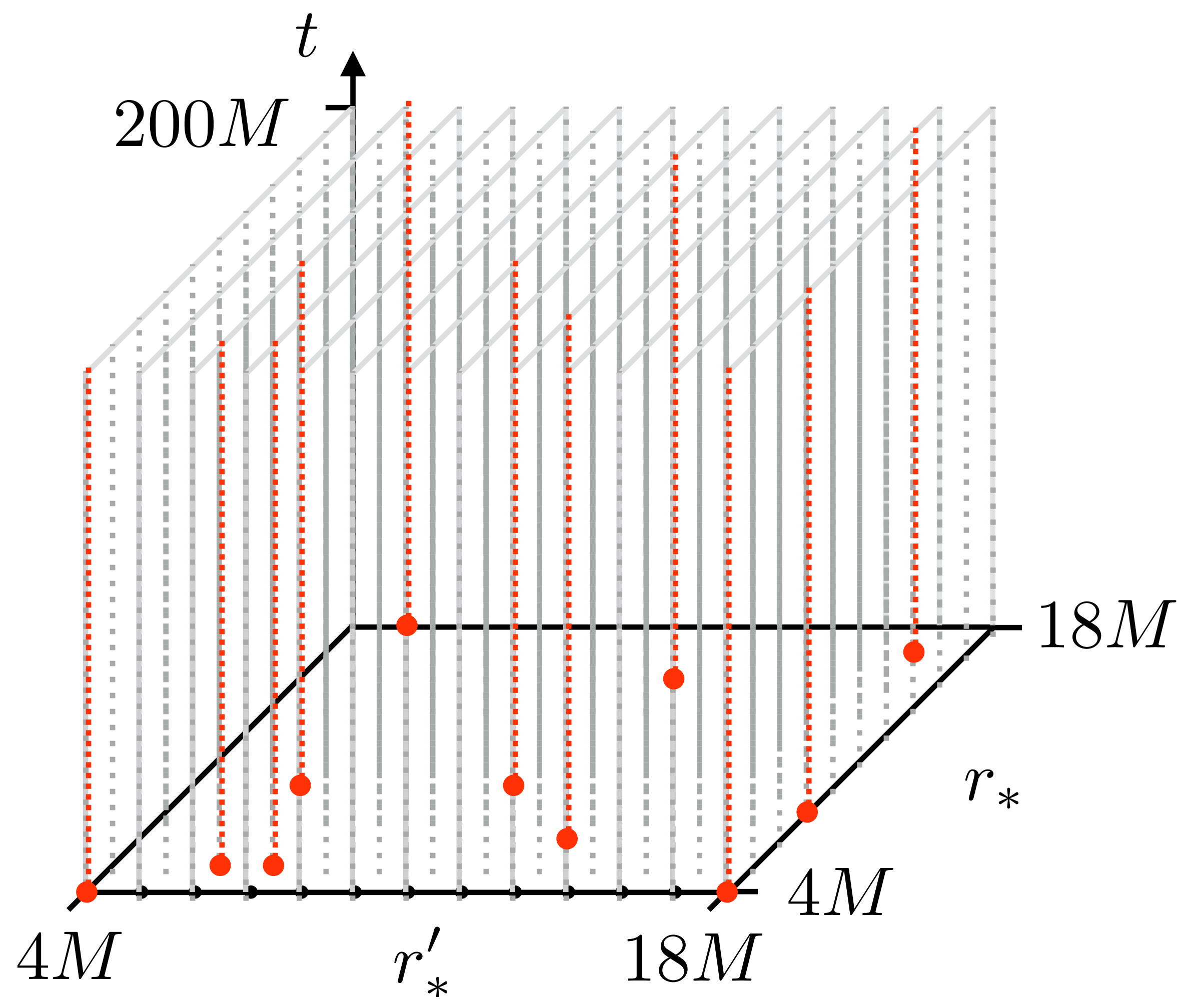
$$C_i^\ell(\vec{\lambda}) = \sum_{j=1}^{N_\ell} \left(V_\ell^{-1}\right)_{ij} G_\ell(T_j;\vec{\lambda})$$

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

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

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

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

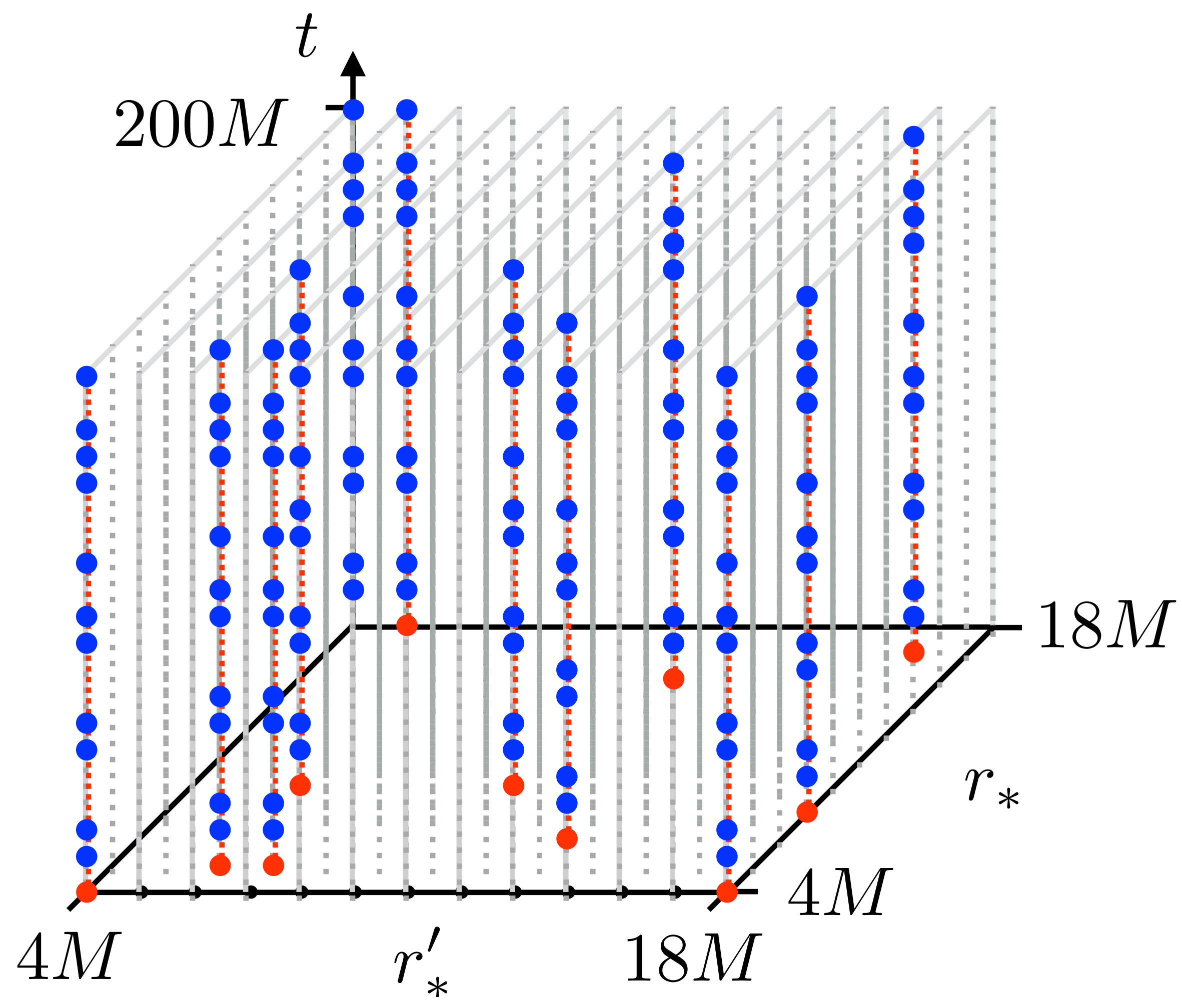
$$C_i^\ell(\vec{\lambda}) = \sum_{j=1}^{N_\ell} \left(V_\ell^{-1}\right)_{ij} G_\ell(T_j;\vec{\lambda})$$

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

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

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

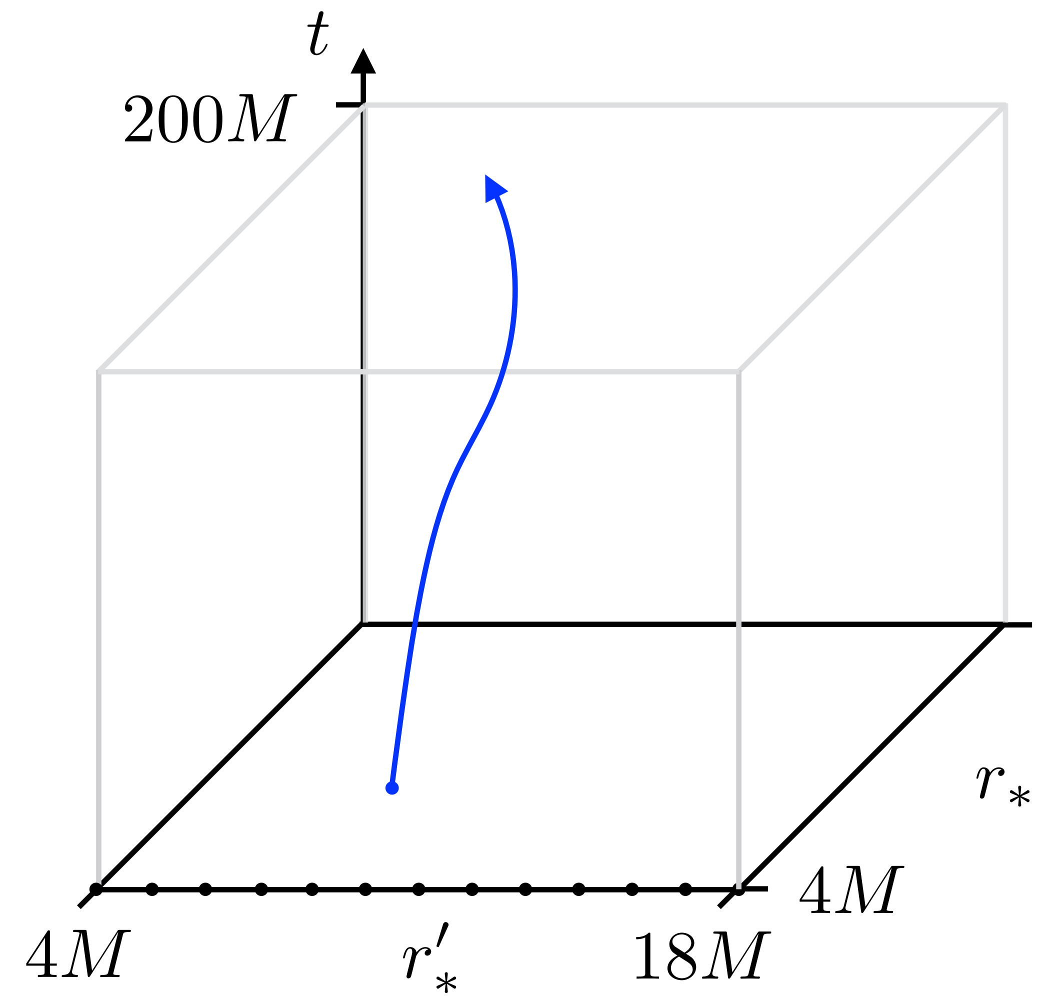
Find the interpolation nodes through another greedy algorithm that minimizes the interpolation error



3) Parametric fitting

Mostly interested in finding Green's function on worldlines

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



3) Parametric fitting

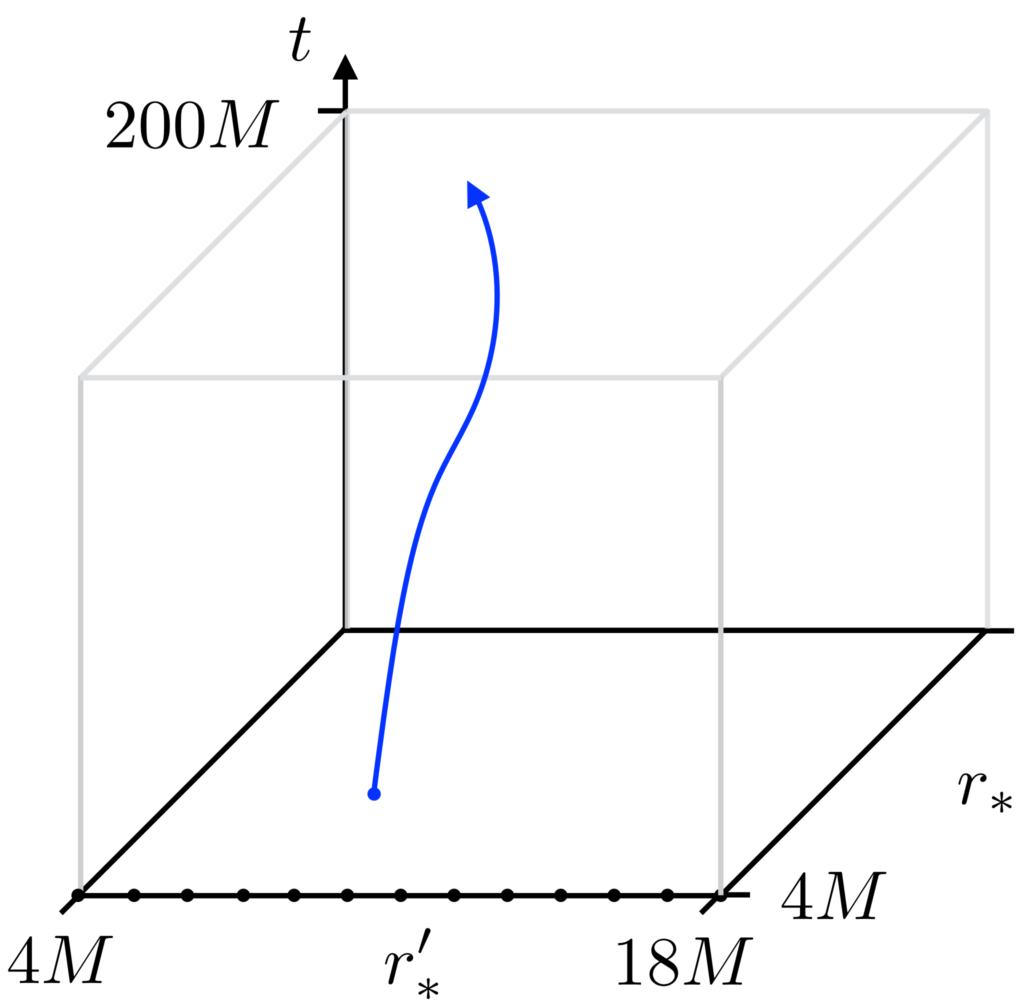
Mostly interested in finding Green's function on worldlines

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

But destroys 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



3) Parametric fitting

Mostly interested in finding Green's function on worldlines

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

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

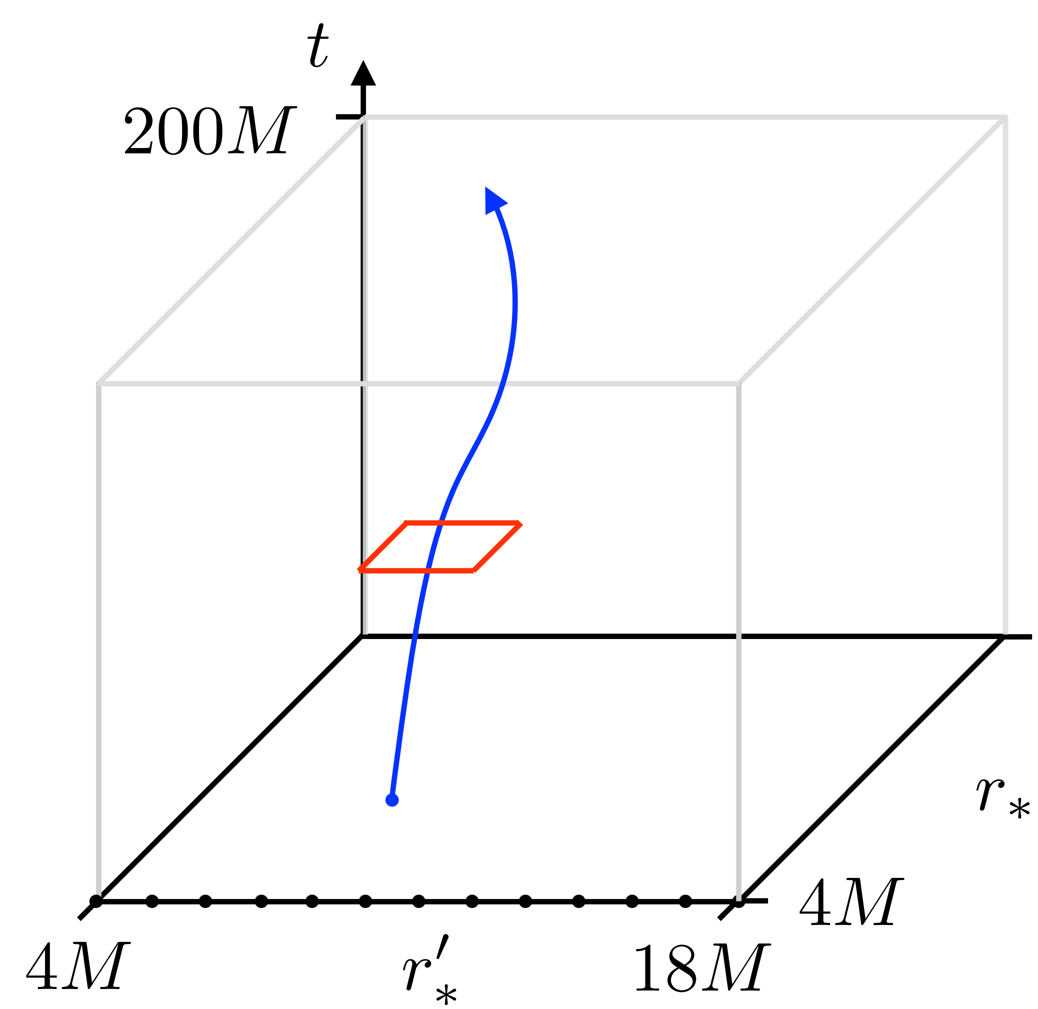
$$G_\ell(t; r_*(t), r'_*) \approx \sum_{j=1}^{N_\ell} B_j^\ell(t) G_\ell(T_j; r_*(t), r'_*)$$

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

Mostly interested in finding Green's function on worldlines

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

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

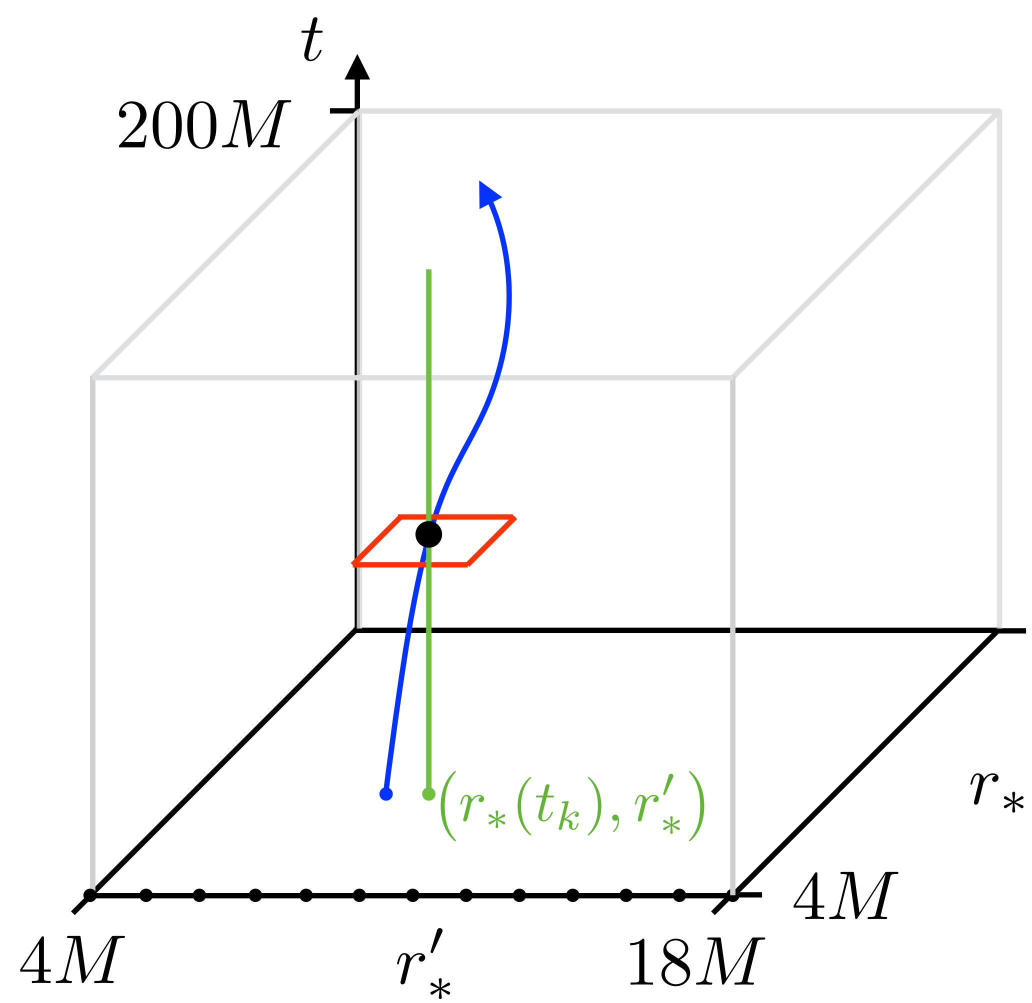
$$G_\ell(t; r_*(t), r'_*) \approx \sum_{j=1}^{N_\ell} B_j^\ell(t) G_\ell(T_j; r_*(t), r'_*)$$

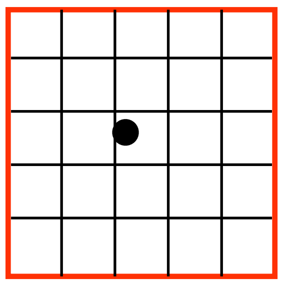
As a result, fitting is done on-the-fly on a worldline

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

Mostly interested in finding Green's function on worldlines

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

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

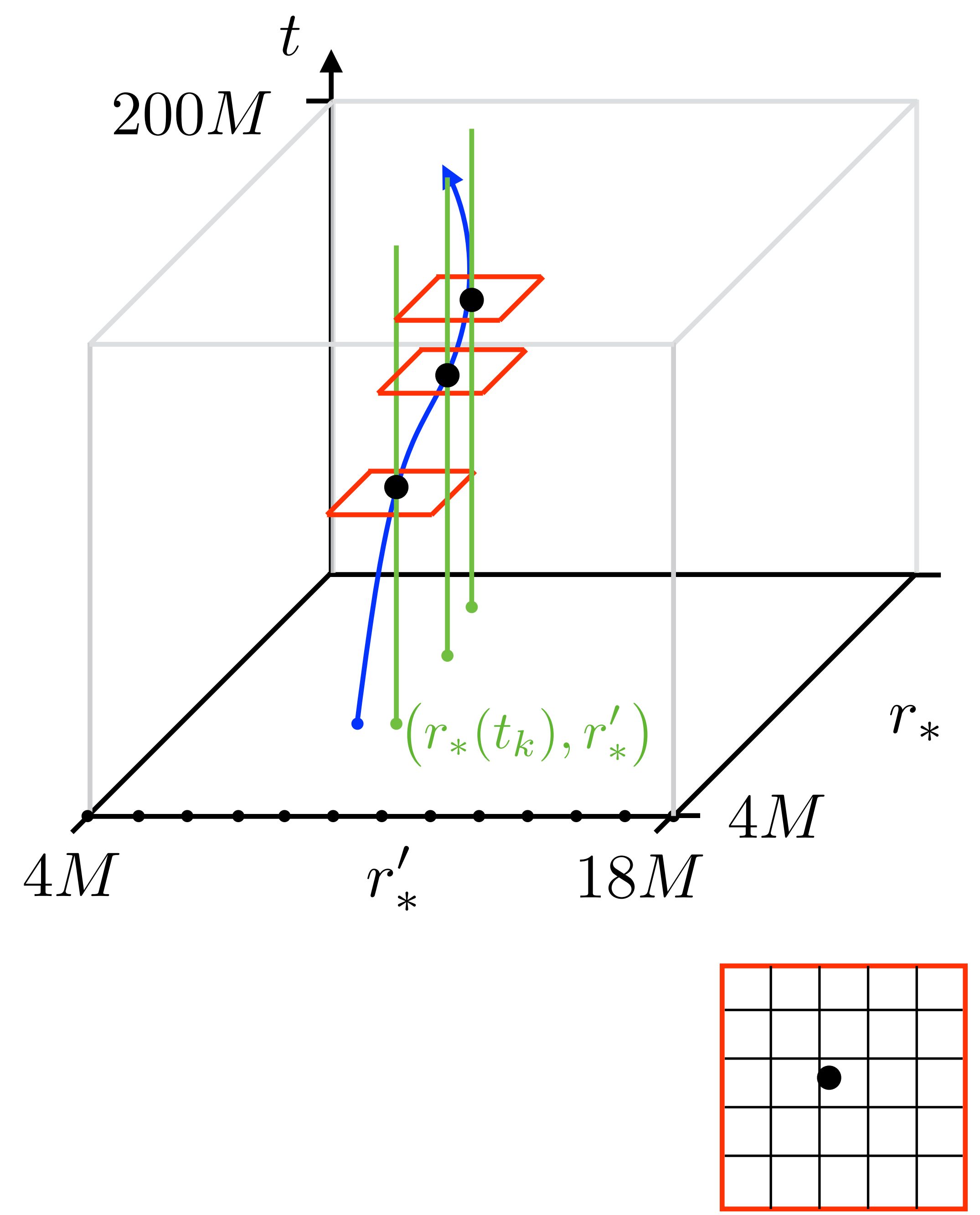
$$G_\ell(t; r_*(t), r'_*) \approx \sum_{j=1}^{N_\ell} B_j^\ell(t) G_\ell(T_j; r_*(t), r'_*)$$

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

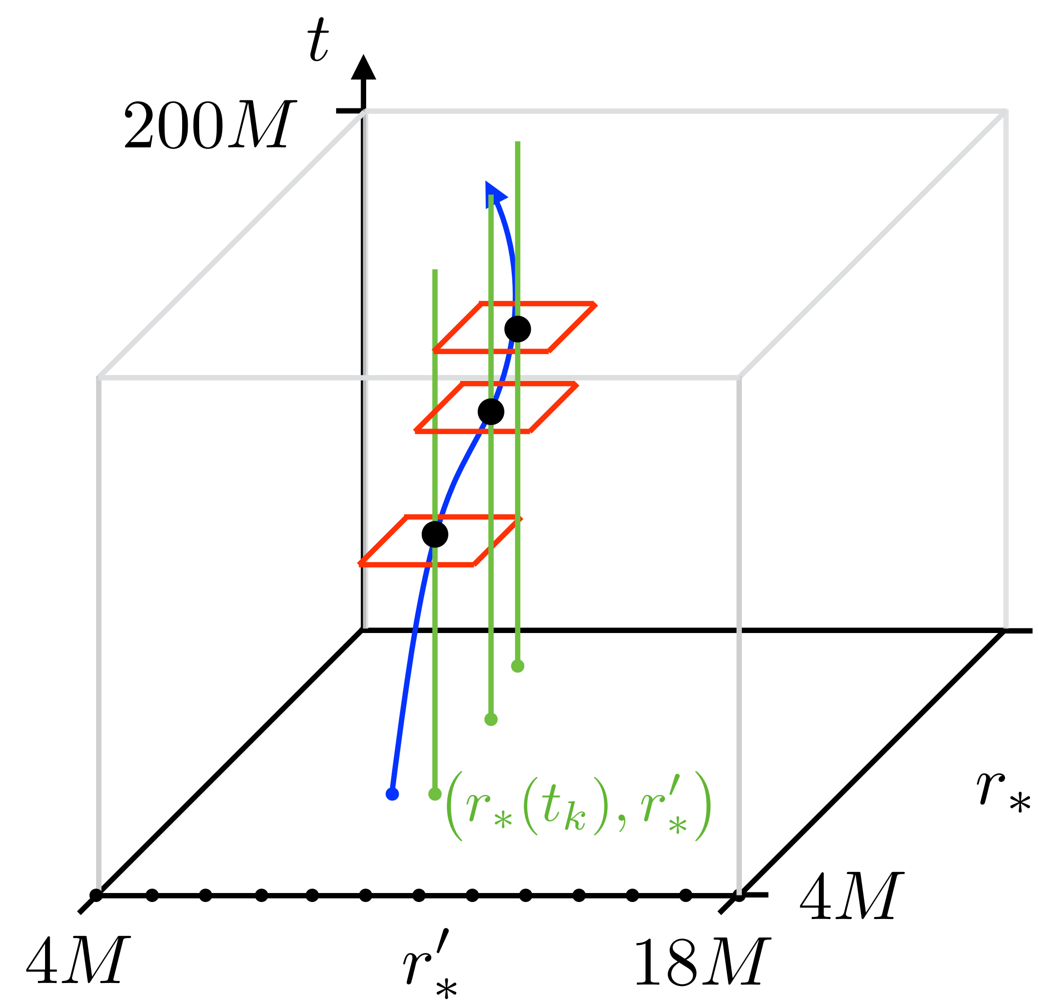
Mostly interested in finding Green's function on worldlines

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

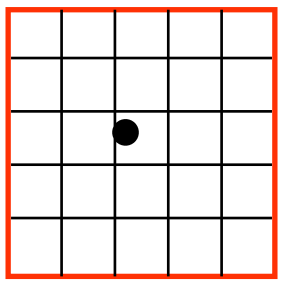
But destroys the affine nature of the model (i.e., separability)

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

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 $r_*(t_k)$

But have to store to disk all data at each T_j…

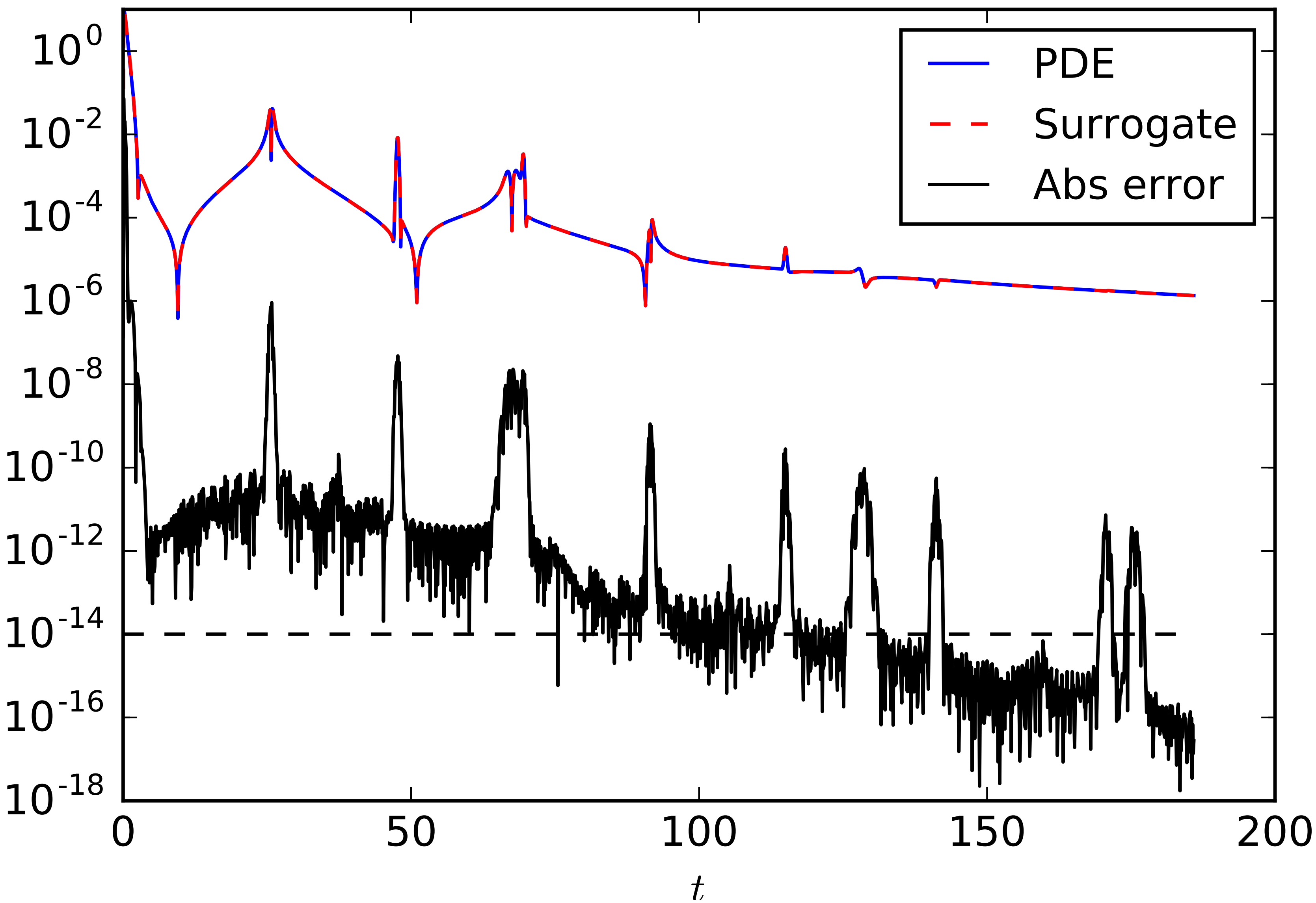
Repeat for all time steps

Surrogate accuracy, speed-up, and size

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

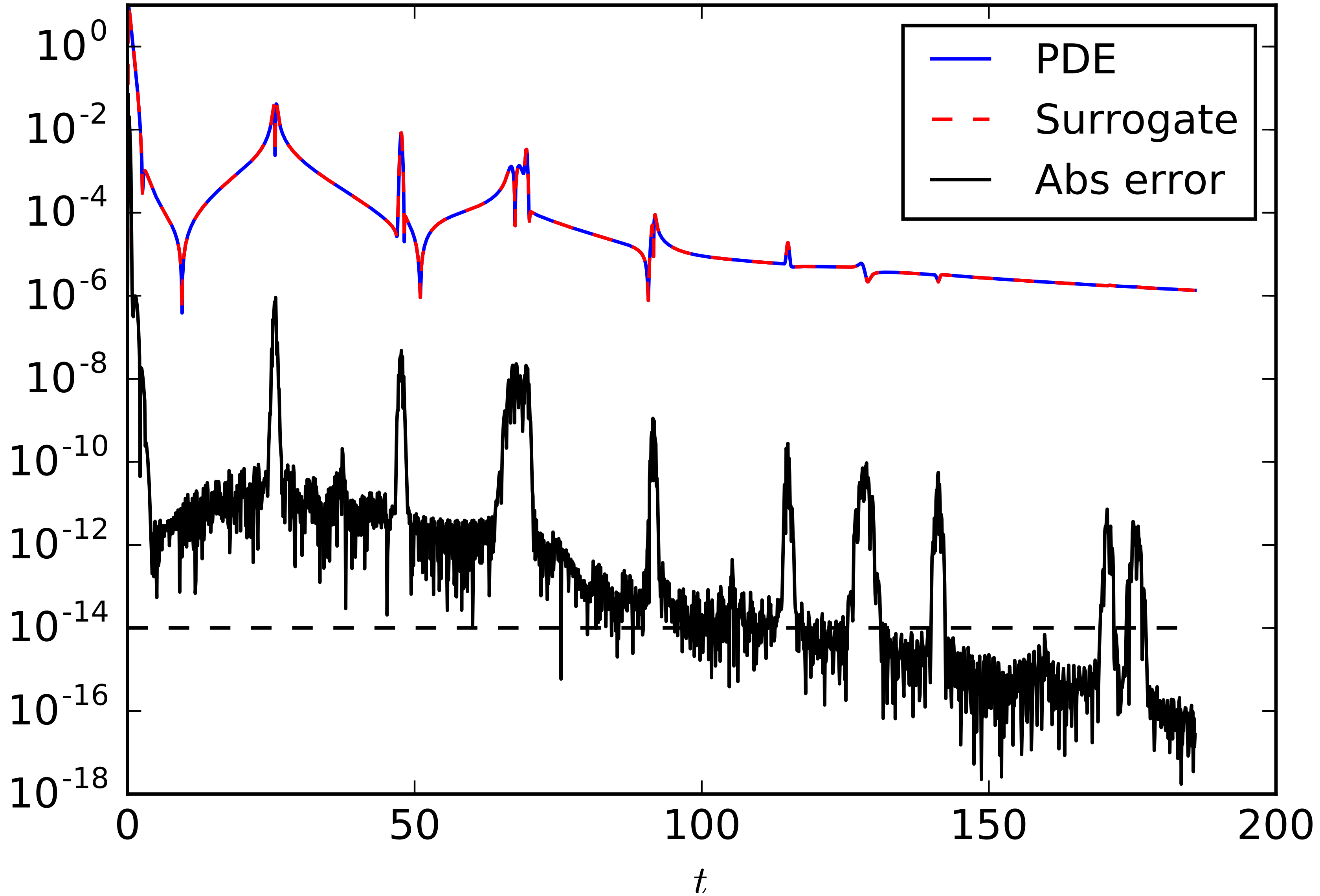
Surrogate accuracy, speed-up, and size

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



Surrogate accuracy, speed-up, and size

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

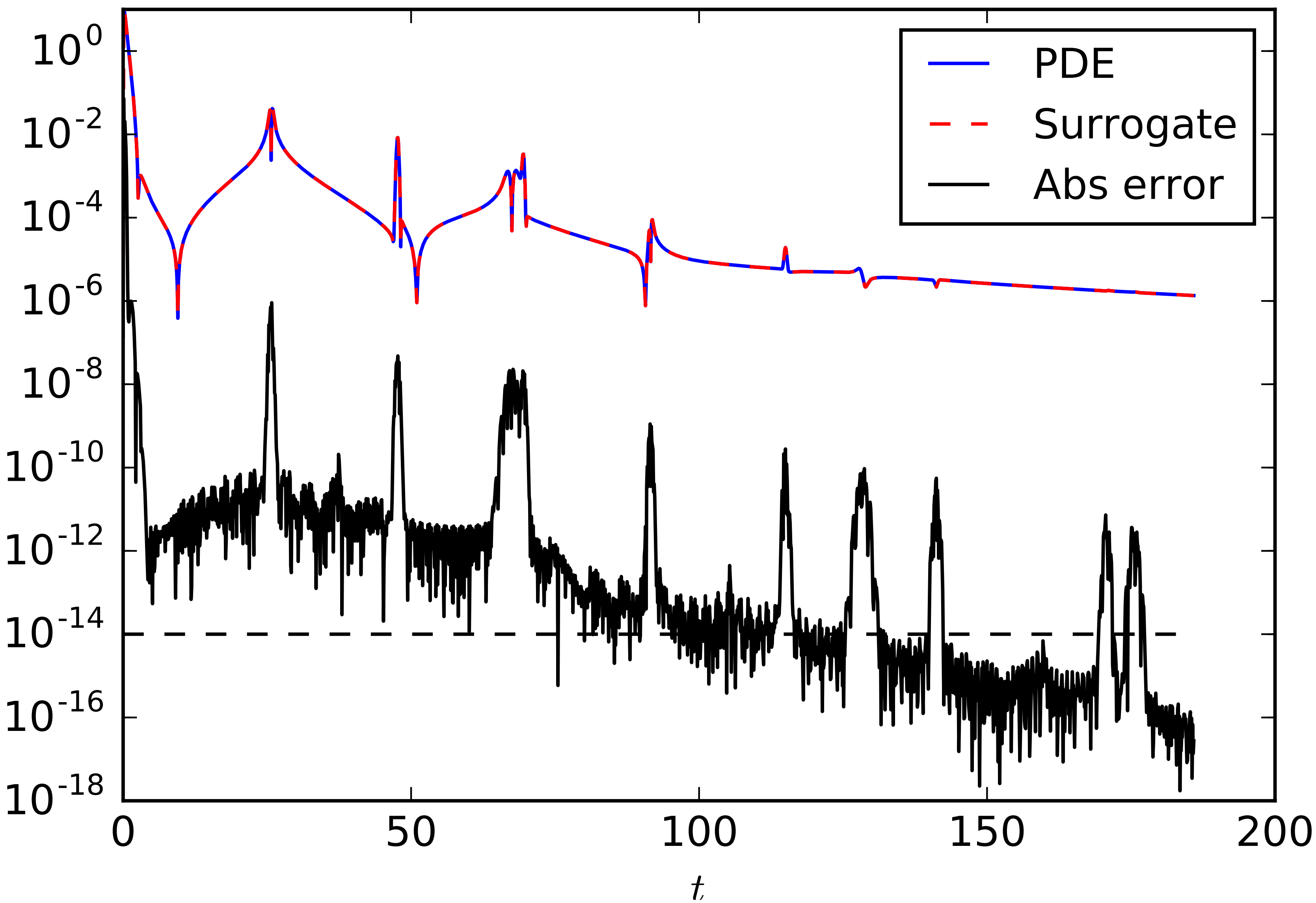


PDE	~380s
Surrogate	~25s
Speed-up*	~15x

**But not quite an apples-to-apples comparison.*

Surrogate accuracy, speed-up, and size

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



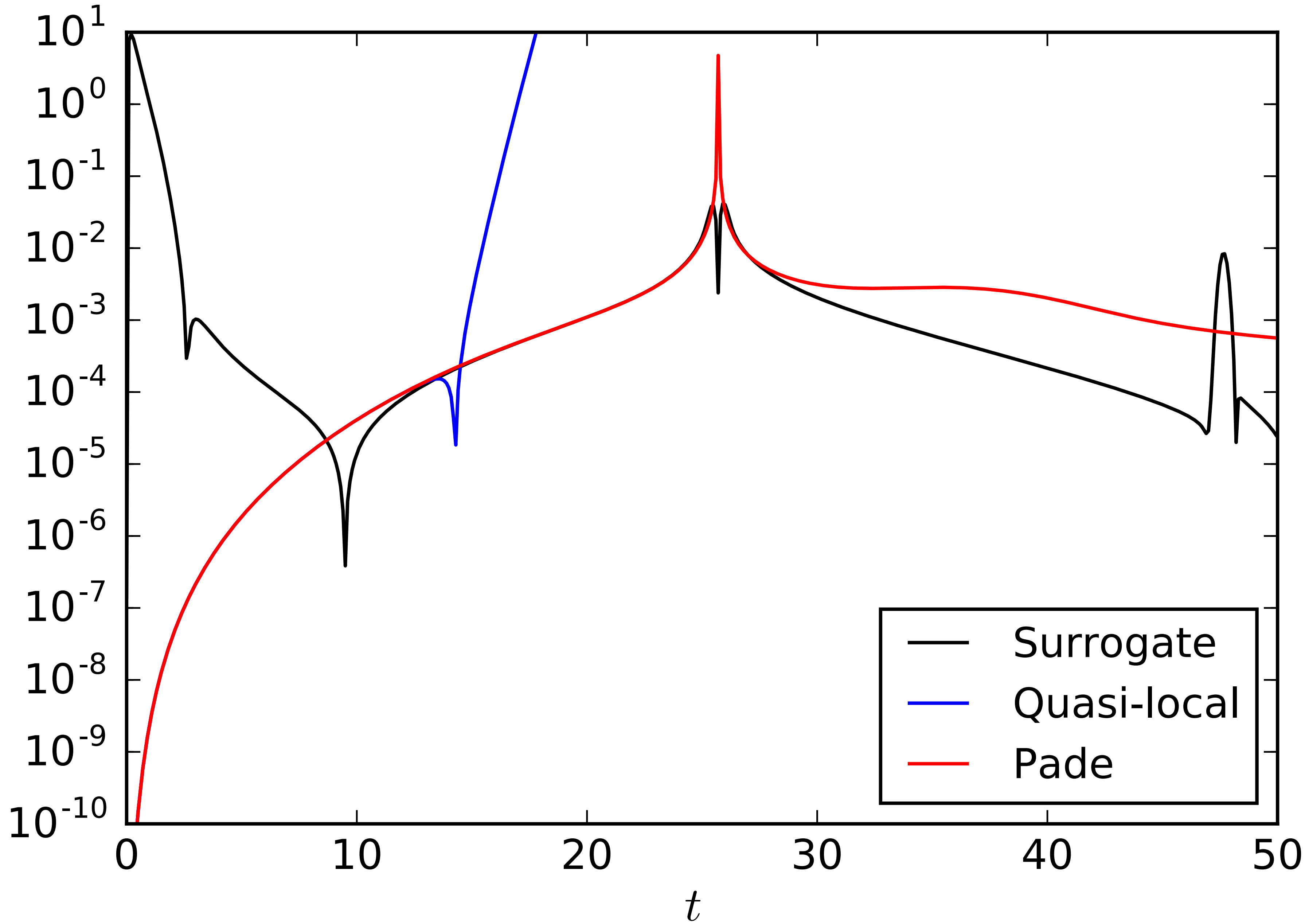
PDE	~380s
Surrogate	~25s
Speed-up*	~15x

**But not quite an apples-to-apples comparison.*

Physical memory: 30GB reduced to 2GB

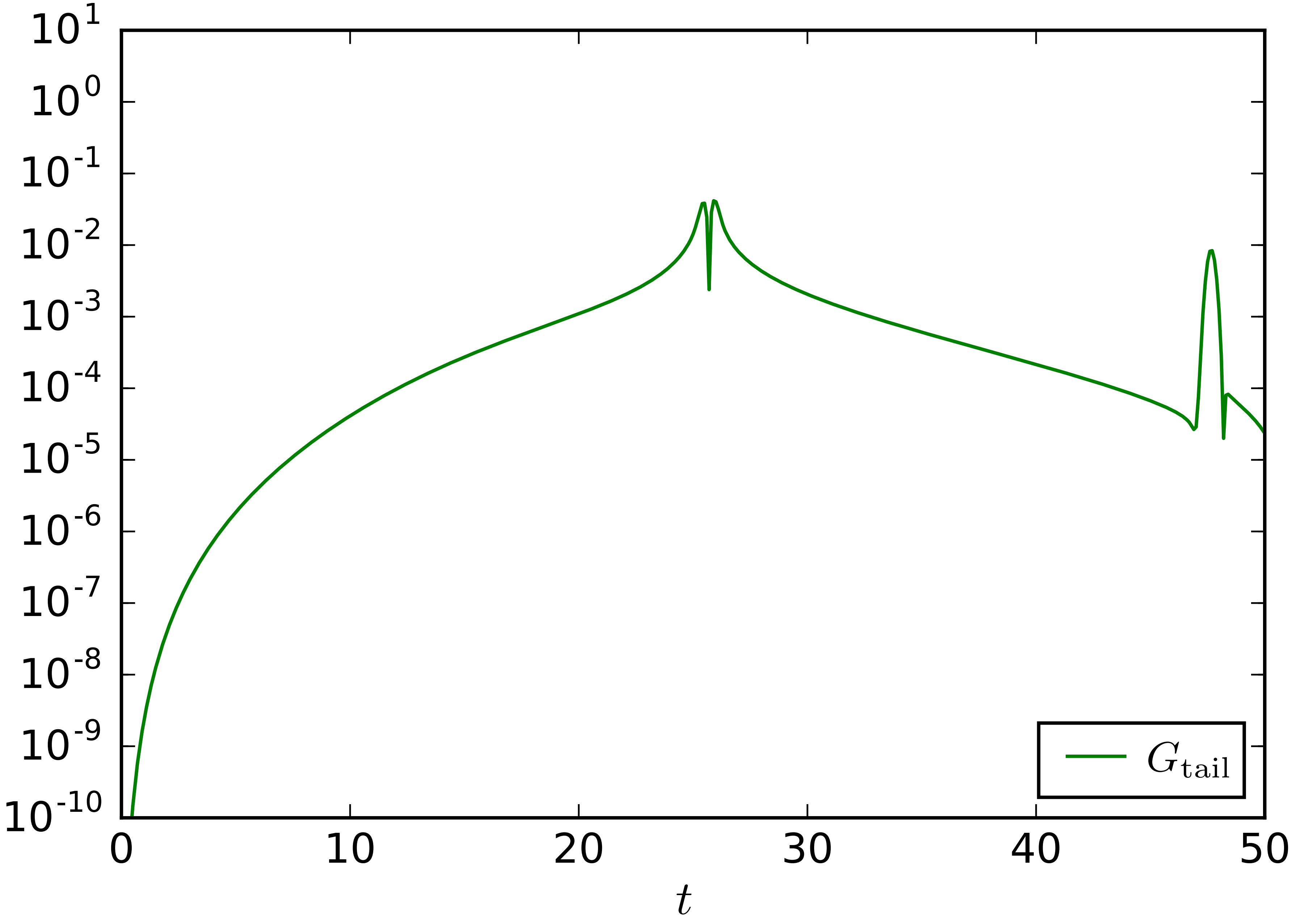
Using the surrogate predictions

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



Using the surrogate predictions

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



Surrogate self-force evaluation

History-dependent part of first-order scalar self-force in MiSaTaQuWa form *Quinn (00)*

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

Surrogate self-force evaluation

History-dependent part of first-order scalar self-force in MiSaTaQuWa form *Quinn (00)*

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

$$\approx q^2 P^{\mu\nu} \left\{ \int_{\tau_{\rm ql}}^{\tau} d\tau' \, {\rm Pade}\big[\nabla_\nu V_{\rm ql}(z^\mu, z^{\mu'})\big] + \int_{\tau_{\rm bc}}^{\tau_{\rm ql}} d\tau' \, \nabla_\nu G_{\rm surr}(z^\mu, z^{\mu'}) + \int_{-\infty}^{\tau_{\rm bc}} d\tau' \, \nabla_\nu G_{\rm branch}(z^\mu, z^{\mu'}) \right\}$$

Surrogate self-force evaluation

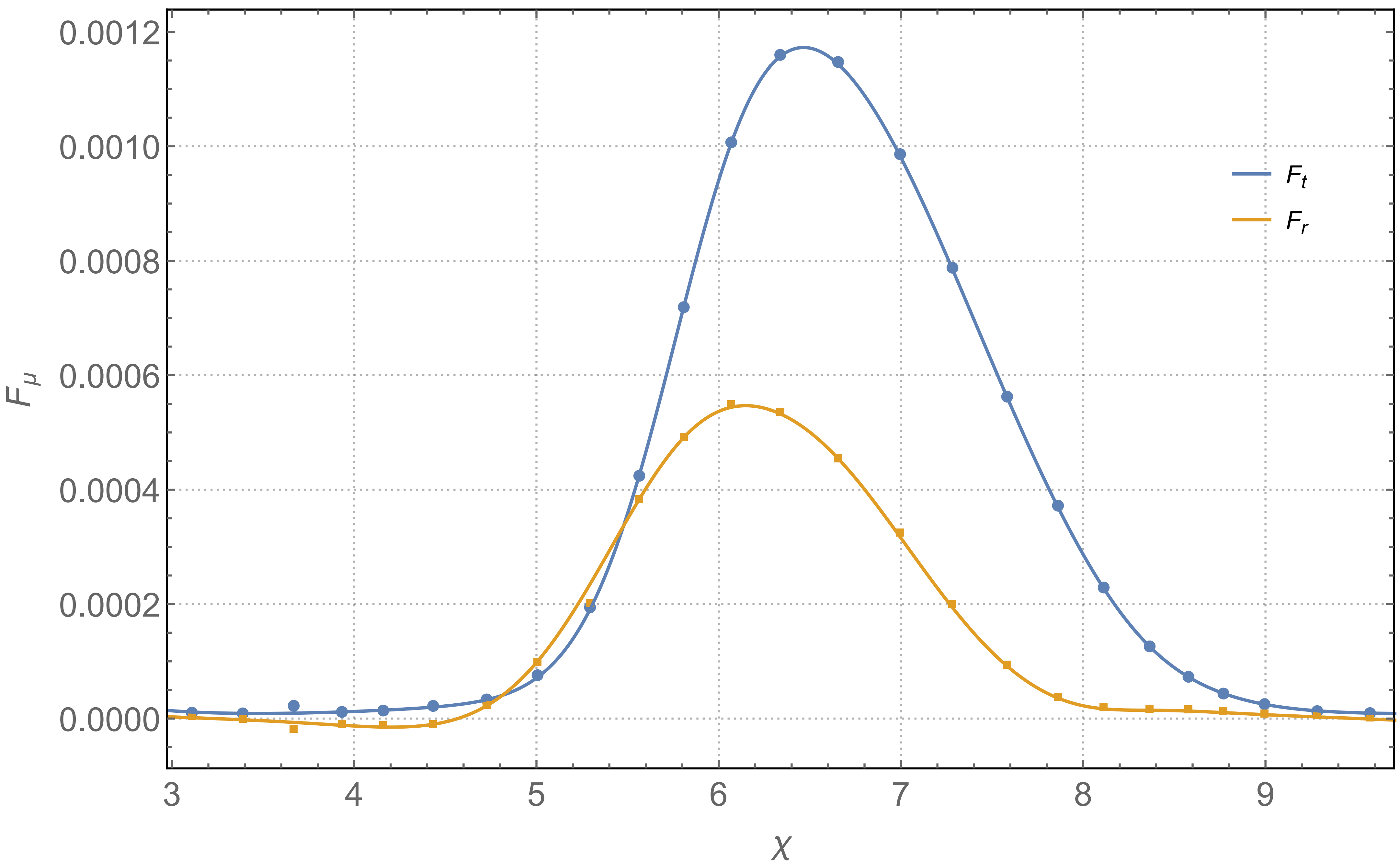
History-dependent part of first-order scalar self-force in MiSaTaQuWa form *Quinn (00)*

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

$$\approx q^2 P^{\mu\nu} \left\{ \int_{\tau_{\text{ql}}}^{\tau} d\tau' \, \text{Pade}\left[\nabla_\nu V_{\text{ql}}(z^\mu, z^{\mu'})\right] + \int_{\tau_{\text{bc}}}^{\tau_{\text{ql}}} d\tau' \, \nabla_\nu G_{\text{surr}}(z^\mu, z^{\mu'}) + \int_{-\infty}^{\tau_{\text{bc}}} d\tau' \, \nabla_\nu G_{\text{branch}}(z^\mu, z^{\mu'}) \right\}$$

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

$$\frac{d^2 z^\mu}{d\tau^2} + \Gamma^\mu_{\alpha\beta}\frac{dz^\alpha}{d\tau}\frac{dz^\beta}{d\tau}$$
$$\approx q^2 P^{\mu\nu}\left\{\int_{\tau_{\rm ql}}^{\tau} d\tau'\,\mathrm{Pade}\left[\nabla_\nu V_{\rm ql}(z^\mu, z^{\mu'})\right] + \int_{\tau_{\rm bc}}^{\tau_{\rm ql}} d\tau'\,\nabla_\nu G_{\rm surr}(z^\mu, z^{\mu'})\right\}$$
$$+\ \text{local terms}$$

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

$$\frac{d^2 z^\mu}{d\tau^2} + \Gamma^\mu_{\alpha\beta}\frac{dz^\alpha}{d\tau}\frac{dz^\beta}{d\tau}$$
$$\approx q^2 P^{\mu\nu}\left\{\int_{\tau_{\rm ql}}^{\tau} d\tau'\, \mathrm{Pade}\left[\nabla_\nu V_{\rm ql}(z^\mu, z^{\mu'})\right] + \int_{\tau_{\rm bc}}^{\tau_{\rm ql}} d\tau'\, \nabla_\nu G_{\rm surr}(z^\mu, z^{\mu'})\right\}$$
$$+ \text{local terms}$$

- Higher-order, self-consistent evolutions

- Self-consistent field/waveform and at higher orders

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

- Self-consistent field/waveform and at higher orders

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

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

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

- Self-consistent field/waveform and at higher orders

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

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

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

- Studying basic wave propagation in black hole spacetimes

- Self-consistent field/waveform and at higher orders

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

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

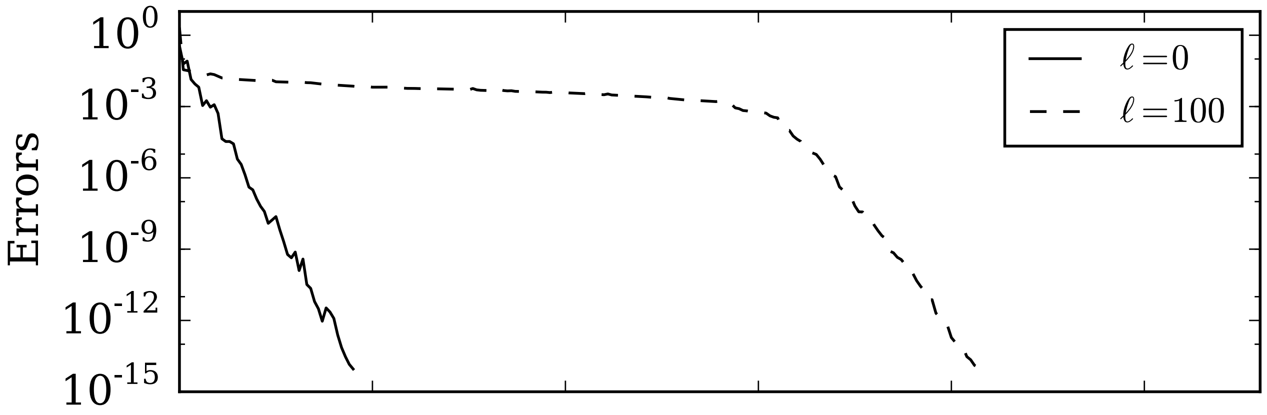
$$F^\mu_{\rm hist}(\tau) = q^2 P^{\mu\nu} \lim_{\epsilon\to 0^+} \int_{-\infty}^{\tau-\epsilon} d\tau'\, \nabla_\nu G_{\rm ret}(z^\mu, z^{\mu'})$$

- Studying basic wave propagation in black hole spacetimes

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

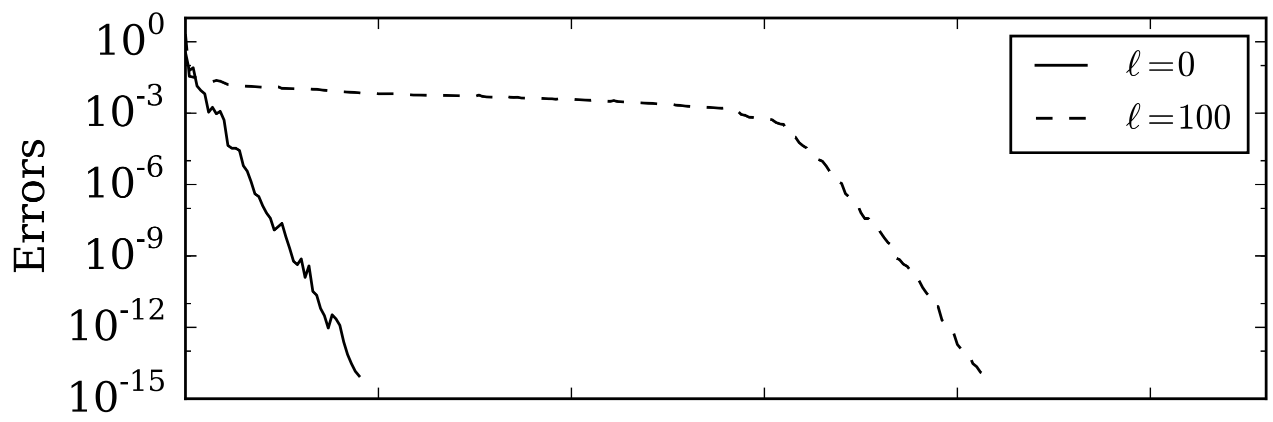
Improving the surrogate building strategy

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

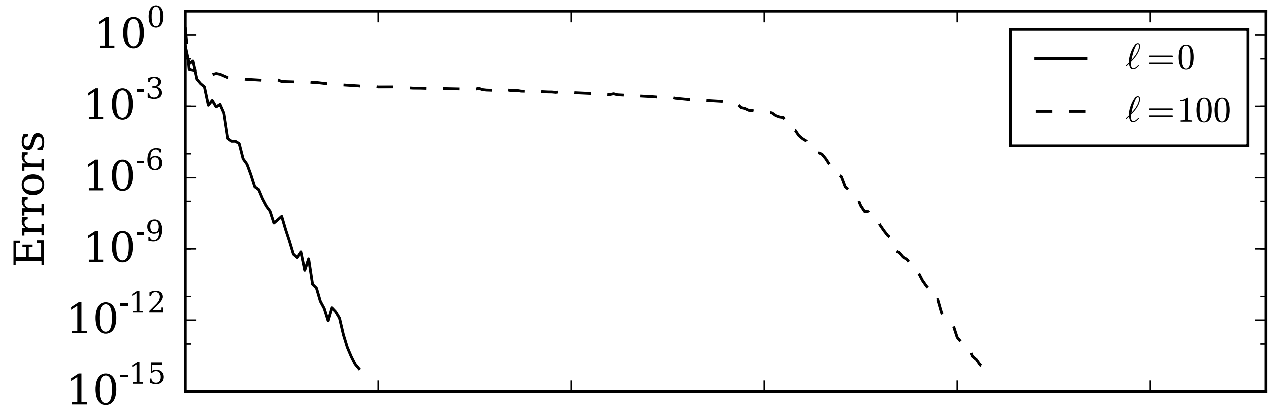
The plateau in the max projection errors often hints that a different representation of the data may generate a more compact basis



- 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

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

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

Different and useful ways to parametrize the data?

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

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

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

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

Different and useful ways to parametrize the data?

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

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

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

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

- 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

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

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

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

- 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

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 given worldline, the surrogate is more than 15x faster to evaluate than solving the wave equation, with little loss of accuracy
- Other choices in the surrogate modeling strategy may (should!) improve both the speed and size of the Green's function surrogate
- 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?