Title: Highly accurate EMRI self-force calculations

Date: Jun 20, 2010 03:20 PM

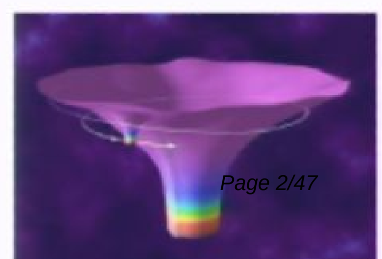
URL: http://pirsa.org/10060041

Abstract: TBA

Highly accurate and efficient self-force computation using time-domain methods: Error estimates, validation, and optimization

Jonathan Thornburg

Department of Astronomy
Indiana University
Bloomington, Indiana, USA



Why High Accuracy: LISA (1)

Matched filtering of the entire LISA data stream is impractically expensive, but once sources have been detected by hierarchical and/or time-frequency searches, matched filtering will be used to estimate parameters for source modelling, cosmology, GR tests, ...

Why High Accuracy: LISA (2)

High LISA S/N after matched filtering

$\Rightarrow$ Need highly accurate calculation of GW phase over the full LISA timespan ($\sim 10^5$ EMRI orbits) to prevent template-phase errors from dominating the overall parameter-estimation error budget

Vhy High Accuracy: LISA (1)

Matched filtering of the entire LISA data stream is impractically expensive, but once sources have been detected by hierarchical and/or time-frequency searches, matched filtering will be used to estimate parameters for source modelling, cosmology, GR tests, . . .

The **strongest** LISA EMRI sources will have quite large S/N after matched filtering:

a	$\mathcal{R}_{\rm MW}^{\rm BH}$ (Gyr^{-1})	LISA Performance 5yr,2chan	LISA Performance 2yr,1chan
0	4	140	16*
	40	460	180
	400	1300	490
0.5	4	160	20*
	40	560	210
	400	1500	590
0.9	4	240	46*
	40	780	330
	400	2100	860

Why High Accuracy: LISA (2)

High LISA S/N after matched filtering

$\Rightarrow$ Need highly accurate calculation of GW phase over the full LISA timespan ($\sim 10^5$ EMRI orbits) to prevent template-phase errors from dominating the overall parameter-estimation error budget

Why High Accuracy: LISA (2)

High LISA S/N after matched filtering

$\Rightarrow$ Need highly accurate calculation of GW phase over the full LISA timespan ($\sim 10^5$ EMRI orbits) to prevent template-phase errors from dominating the overall parameter-estimation error budget

$\Rightarrow$ Need highly accurate calculation of self-force: Huerta & Gair, PRD **79** (2009), 084021 find that a 10^{-6} fractional change in the EMRI self-force changes the cumulative GW phase by ~ 2 radians. Hence:

Self-Force Relative Error Tolerance

	$C = 1$	$C = 30$	$C = 1000$
$\rho_{\max} = 30$	2×10^{-8}	5×10^{-7}	2×10^{-5}
$\rho_{\max} = 300$	2×10^{-9}	5×10^{-8}	2×10^{-6}
$\rho_{\max} = 2000$	3×10^{-10}	8×10^{-9}	3×10^{-7}

[C measures the degeneracy (ill-conditioning) of the parameter estimation.]

Why High Accuracy: Orbit Correction

We can use the orbit-correction scheme of Gralla & Wald, CQG **25** (2008), 205009 to explore $\mathcal{O}(\mu^2)$ effects, where $\mu = m/M \sim 10^{-5}$ is the EMRI mass ratio.

Why High Accuracy: Orbit Correction

We can use the orbit-correction scheme of Gralla & Wald, CQG **25** (2008), 205009 to explore $\mathcal{O}(\mu^2)$ effects, where $\mu = m/M \sim 10^{-5}$ is the EMRI mass ratio.

To reliably distinguish true $\mathcal{O}(\mu^2)$ effects due to the orbit correction from numerical errors in the $\mathcal{O}(\mu)$ self force, the self force needs to be calculated with a relative error $\ll \mu$.

Other uses of highly accurate self-force calculations:

- calibrate/constraint post-Newtonian expansions (e.g., Blanchet *et al.*, arXiv:0910.0207)
- test correctness of self-force calculations & theory by comparisons between calculations using different methods (e.g., Sago, Barack, & Detweiler, PRD **78** (2008), 124024)

Why High Efficiency: Orbit Correction

The orbit-correction scheme of Gralla & Wald, CQG **25** (2008), 205009 requires time-integrating a set of coupled ODEs for the orbital-parameter evolution on radiation-reaction and longer timescales, with the ODEs' right-hand-side functions being given by a self-force computation.

Even the most efficient ODE-integration schemes will require evaluating the right-hand-side functions (i.e., computing the self-force for some specified intermediate orbit) hundreds of times in the course of a single low-accuracy orbit-correction calculation, and many more for higher accuracies.

$\Rightarrow$ Self-force calculation should be as efficient as possible!

Why High Efficiency: Orbit Correction

The orbit-correction scheme of Gralla & Wald, CQG **25** (2008), 205009 requires time-integrating a set of coupled ODEs for the orbital-parameter evolution on radiation-reaction and longer timescales, with the ODEs' right-hand-side functions being given by a self-force computation.

Even the most efficient ODE-integration schemes will require evaluating the right-hand-side functions (i.e., computing the self-force for some specified intermediate orbit) hundreds of times in the course of a single low-accuracy orbit-correction calculation, and many more for higher accuracies.

$\Rightarrow$ Self-force calculation should be as efficient as possible!

[For comparison, Barack & Sago, PRD 81 (2010), 084021 report that their gravitational self-force calculation takes 4–8 cpu days at a 10^{-4} relative accuracy level.]

Barack-Ori mode-sum regularization [**conceptual**]

Lorenz gauge $\Rightarrow$ metric perturbation from point particle is relatively simple (locally isotropic).

Work in **time domain** $\Rightarrow$ allows arbitrary particle orbits.

Decompose metric perturbation into spherical harmonics (ℓ, m), compute each (ℓ, m) mode via time-domain **numerical integration** of the complex wave equation

$$\Box\phi_{\ell m} + V_\ell(r)\phi_{\ell m} = S_{\ell m}(t)\,\delta\big(r(t) - r_{\text{particle}}(t)\big)$$

Compute regularized self-force modes for each ℓ:

$$F^{(\pm)}_{\ell,\text{reg}} \sim \left(\sum_m (\nabla\phi_{\ell m})\Big|_{r=r_p^\pm} \right) \mp (\ell + \tfrac{1}{2})A - B$$

(known coefficients)

Compute self-force:

$$F_{\text{self}} = \sum^{\infty} F^{(\pm)}$$

Tail Force

The tail force $F_{\text{self,tail}}$ can be estimated via a "tail fit" of some subset of the $F_{\ell,\text{reg}}$ to the large-ℓ series

$$F_{\ell,\text{reg}} = \sum_{\substack{p \text{ even} \\ p \geq 2}} c_p f_p(\ell)$$

where the basis functions $f_p(\ell) = \mathcal{O}(\ell^{-p})$ are

$$\begin{aligned}
f_2(\ell) &= 1/((\ell - \tfrac{1}{2})(\ell + \tfrac{3}{2})) \\
f_4(\ell) &= 1/((\ell - \tfrac{3}{2})(\ell - \tfrac{1}{2})(\ell + \tfrac{3}{2})(\ell + \tfrac{5}{2})) \\
f_6(\ell) &= 1/((\ell - \tfrac{5}{2})(\ell - \tfrac{3}{2})(\ell - \tfrac{1}{2})(\ell + \tfrac{3}{2})(\ell + \tfrac{5}{2})(\ell + \tfrac{7}{2})) \\
&\cdots
\end{aligned}$$

Then $F_{\text{self,tail}} = \sum_p c_p \Gamma_p$ where $\Gamma_p = \sum_{\ell=K+1}^{\infty} f_p(\ell)$ can be computed analytically.

Adaptive Mesh Refinement (AMR)

I use **characteristic AMR** with ($\geq$) **4th order finite differencing** to numerically integrate the complex wave equation.

- Typically **speeds up computation by a factor of $\sim$ 30–40** over an equivalent-resolution (time-domain) unigrid computation.

Adaptive Mesh Refinement (AMR)

I use **characteristic AMR** with ($\geq$) **4th order finite differencing** to numerically integrate the complex wave equation.

- Typically **speeds up computation by a factor of $\sim$ 30–40** over an equivalent-resolution (time-domain) unigrid computation.
- CPU time now scales $\sim$ length of particle world-line (integration is cheap outside neighborhood of worldline).
  $\Rightarrow$ CPU time scales **linearly** with domain size for 1+1-D code
  $\Rightarrow$ relatively cheap to use very large domains for high accuracy (including waiting for low-ℓ initial-data transients to decay).

Adaptive Mesh Refinement (AMR)

I use **characteristic AMR** with ($\geq$) **4th order finite differencing** to numerically integrate the complex wave equation.

- Typically **speeds up computation by a factor of $\sim$ 30–40** over an equivalent-resolution (time-domain) unigrid computation.
- CPU time now scales $\sim$ length of particle world-line (integration is cheap outside neighborhood of worldline).
  $\Rightarrow$ CPU time scales **linearly** with domain size for 1+1-D code
  $\Rightarrow$ relatively cheap to use very large domains for high accuracy (including waiting for low-ℓ initial-data transients to decay).
- Turn on AMR **gradually** $\Rightarrow$ don't waste high resolution on resolving the initial-data transient.

See Thornburg, arXiv:0909.0036 for details & **source code (C++).**

Floating-Point Arithmetic

In some cases, my highest-accuracy results are limited in accuracy by floating-point rounding errors in solving the complex wave equation.

To investigate these effects, I compare results computed using standard IEEE "double" (64-bit) floating-point arithmetic versus IEEE "double-extended" (80-bit) floating-point arithmetic. Double-extended has a factor of $2^{11} = 2048$ smaller rounding errors, but is a factor of ~ 2 slower.

Test Parameters

Scalar-field particle in circular geodesic orbit in Schwarzschild spacetime at $r_{\text{areal}} = 10M$ (orbital period $\approx 199M$).

Double precision:

AMR error tolerance 10^{-14}, 10^{-15}, 10^{-16},
FMR may be none or $\times 2$, $\times 3$, $\times 4$, $\times 6$, or $\times 8$

Double-extended precision:

AMR error tolerance 10^{-14} down to 10^{-19},
FMR may be none or $\times 2$

Complex wave equation solved for $0 \leq \ell \leq 30$, $\ell = 35$, and $\ell = 40$.

Numerical force sums modes for $0 \leq \ell \leq K{=}30$.

Tail fit coefficients $\{c_4, c_6\}$ fitted to modes $20 \leq \ell \leq 30$.
[For a circular particle orbit c_2 is given analytically by Detweiler, Messaritaki, & Whiting, PRD **67** (2003), 104016.]

Generic-orbit case ($\{c_2, c_4, c_6\}$ fitted) omitted here for brevity.

Error Estimates for Individual Regularized Modes

Difference between $F_{\ell,\mathrm{reg}}^{(+)}$ and $F_{\ell,\mathrm{reg}}^{(-)}$ $\Rightarrow$ "internal" error estimate:

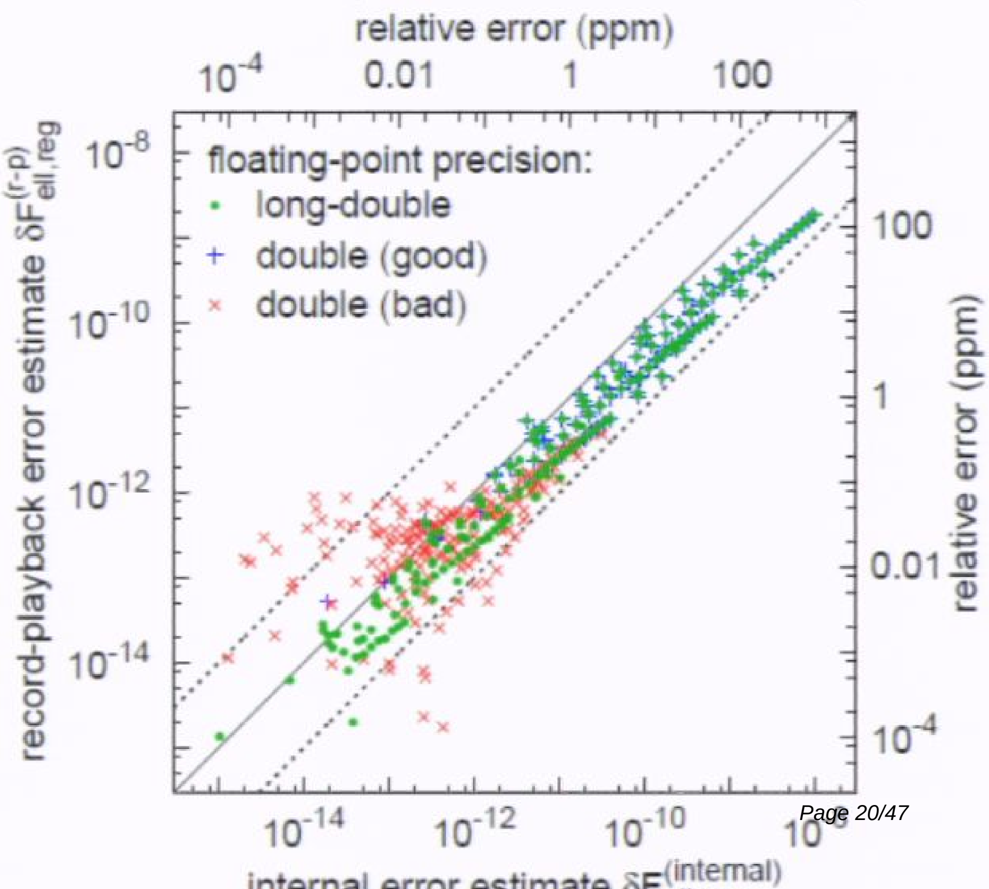
$$F_{\ell,\mathrm{reg}} \pm \delta F_{\ell,\mathrm{reg}}^{(\mathrm{internal})} \equiv \frac{1}{2}\left(F_{\ell,\mathrm{reg}}^{(+)} + F_{\ell,\mathrm{reg}}^{(-)}\right) \pm \frac{1}{2}\left|F_{\ell,\mathrm{reg}}^{(+)} - F_{\ell,\mathrm{reg}}^{(-)}\right|$$

Error Estimates for Individual Regularized Modes

Difference between $F^{(+)}_{\ell,\text{reg}}$ and $F^{(-)}_{\ell,\text{reg}}$ $\Rightarrow$ "internal" error estimate:

$$F_{\ell,\text{reg}} \pm \delta F^{(\text{internal})}_{\ell,\text{reg}} \equiv \tfrac{1}{2}\left(F^{(+)}_{\ell,\text{reg}} + F^{(-)}_{\ell,\text{reg}}\right) \pm \tfrac{1}{2}\left|F^{(+)}_{\ell,\text{reg}} - F^{(-)}_{\ell,\text{reg}}\right|$$

Validate by comparing with convergence-test ("record-playback") error estimates:



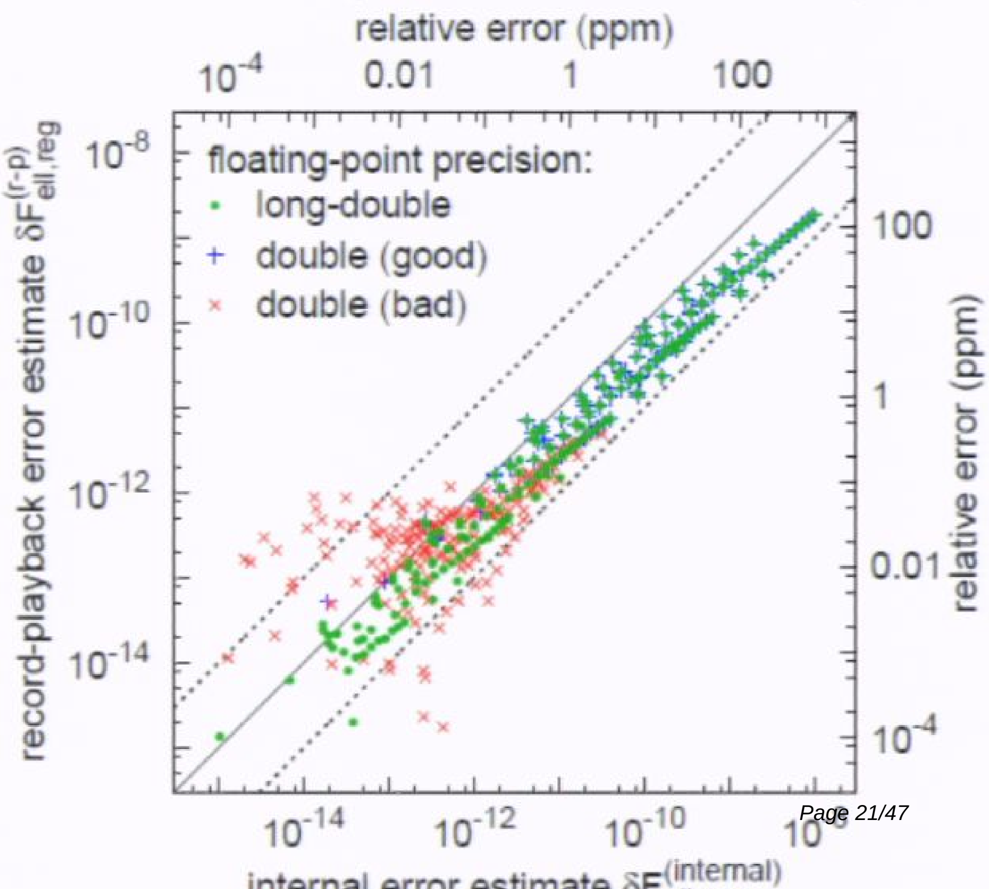
Error Estimates for Individual Regularized Modes

Difference between $F^{(+)}_{\ell,\mathrm{reg}}$ and $F^{(-)}_{\ell,\mathrm{reg}}$ $\Rightarrow$ "internal" error estimate:

$$F_{\ell,\mathrm{reg}} \pm \delta F^{(\mathrm{internal})}_{\ell,\mathrm{reg}} \equiv \tfrac{1}{2}\left(F^{(+)}_{\ell,\mathrm{reg}} + F^{(-)}_{\ell,\mathrm{reg}}\right) \pm \tfrac{1}{2}\left|F^{(+)}_{\ell,\mathrm{reg}} - F^{(-)}_{\ell,\mathrm{reg}}\right|$$

Validate by comparing with convergence-test ("record-playback") error estimates:

Using double-extended floating point arithmetic, the internal error estimate is reliable & somewhat conservative, (pessimistic), overestimating the convergence-test errors by ~ 1–$10\times$.



Error Estimates for Individual Regularized Modes

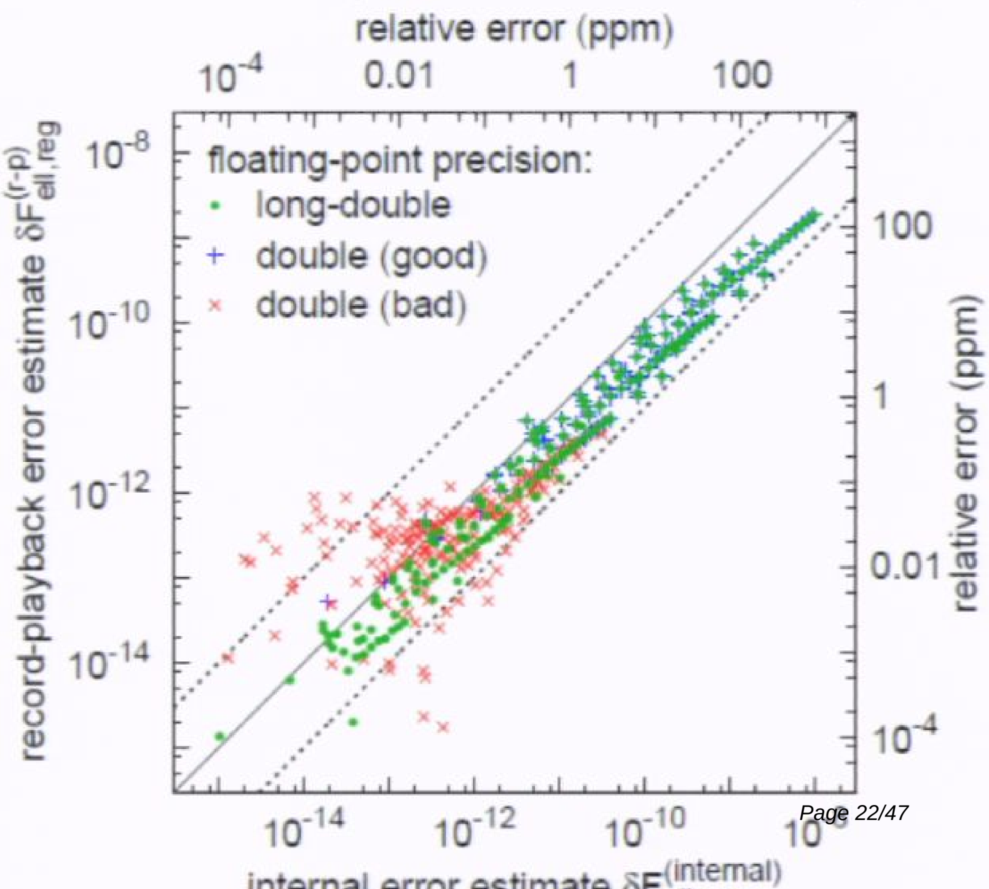
Difference between $F^{(+)}_{\ell,\text{reg}}$ and $F^{(-)}_{\ell,\text{reg}}$ $\Rightarrow$ "internal" error estimate:

$$F_{\ell,\text{reg}} \pm \delta F^{(\text{internal})}_{\ell,\text{reg}} \equiv \tfrac{1}{2}\left(F^{(+)}_{\ell,\text{reg}} + F^{(-)}_{\ell,\text{reg}}\right) \pm \tfrac{1}{2}\left|F^{(+)}_{\ell,\text{reg}} - F^{(-)}_{\ell,\text{reg}}\right|$$

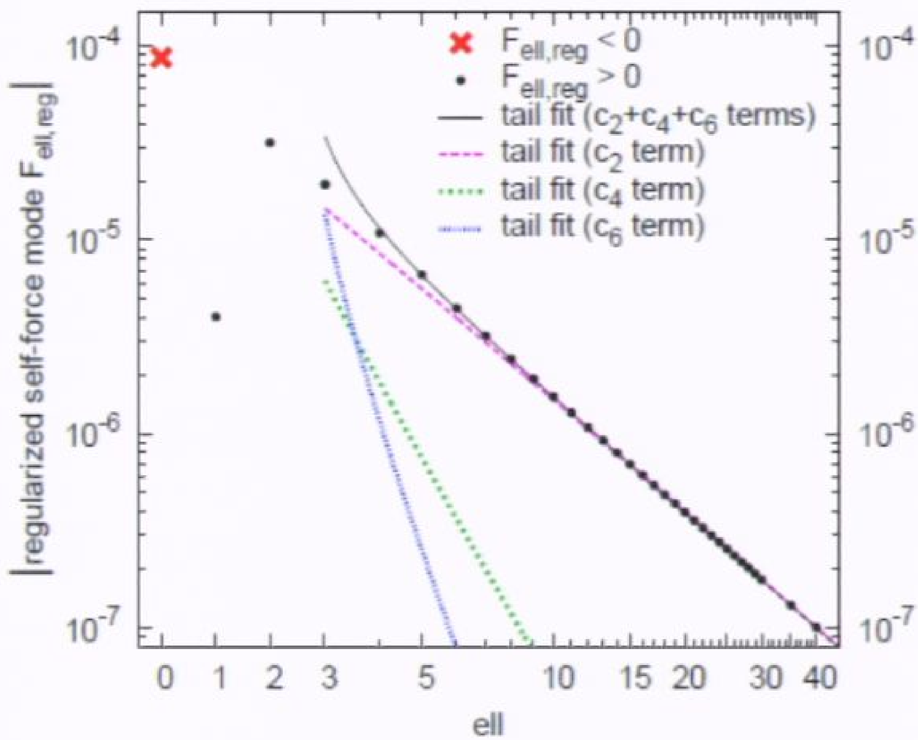
Validate by comparing with convergence-test ("record-playback") error estimates:

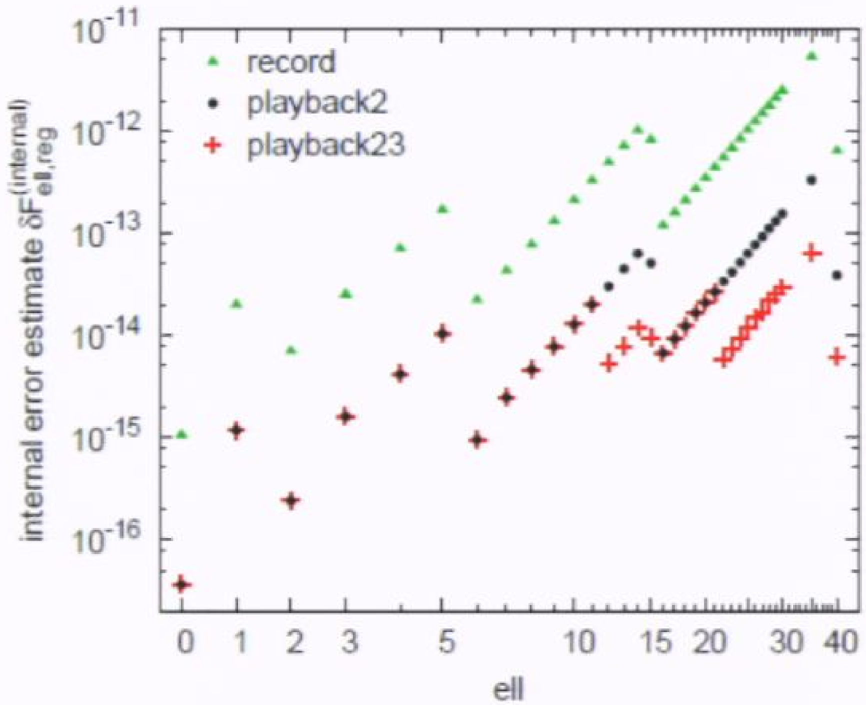
Using double-extended floating point arithmetic, the internal error estimate is reliable & somewhat conservative, (pessimistic), overestimating the convergence-test errors by ~ 1–$10\times$.

Using standard "double" arithmetic, the internal error estimates is ok for $\delta F^{(\text{internal})}_{\ell,\text{reg}} \gtrsim 0.1$ ppm, but unreliable for $\delta F^{(\text{internal})}_{\ell,\text{reg}} \lesssim 0.1$ ppm due to floating-point rounding errors in solving the complex wave equation.



Overview of Modes and Error Estimates





Error Estimates for the Numerical Force (1)

We don't *a priori* know whether or to what extent the actual errors in the individual modes $F_{\ell,\mathrm{reg}}$ for different ℓ might be correlated, so it's not obvious how to sum these to obtain an internal error estimate for the numerical force:

Error Estimates for the Numerical Force (1)

We don't *a priori* know whether or to what extent the actual errors in the individual modes $F_{\ell,\mathrm{reg}}$ for different ℓ might be correlated, so it's not obvious how to sum these to obtain an internal error estimate for the numerical force:

Worst case: Actual numerical errors in different modes are perfectly correlated $\Rightarrow$ should add individual modes' internal error estimates arithmetically [Barack & Sago, PRD 81 (2010), 084021 use this method].

Error Estimates for the Numerical Force (1)

We don't *a priori* know whether or to what extent the actual errors in the individual modes $F_{\ell,\mathrm{reg}}$ for different ℓ might be correlated, so it's not obvious how to sum these to obtain an internal error estimate for the numerical force:

Worst case: Actual numerical errors in different modes are perfectly correlated $\Rightarrow$ should add individual modes' internal error estimates arithmetically [Barack & Sago, PRD 81 (2010), 084021 use this method].

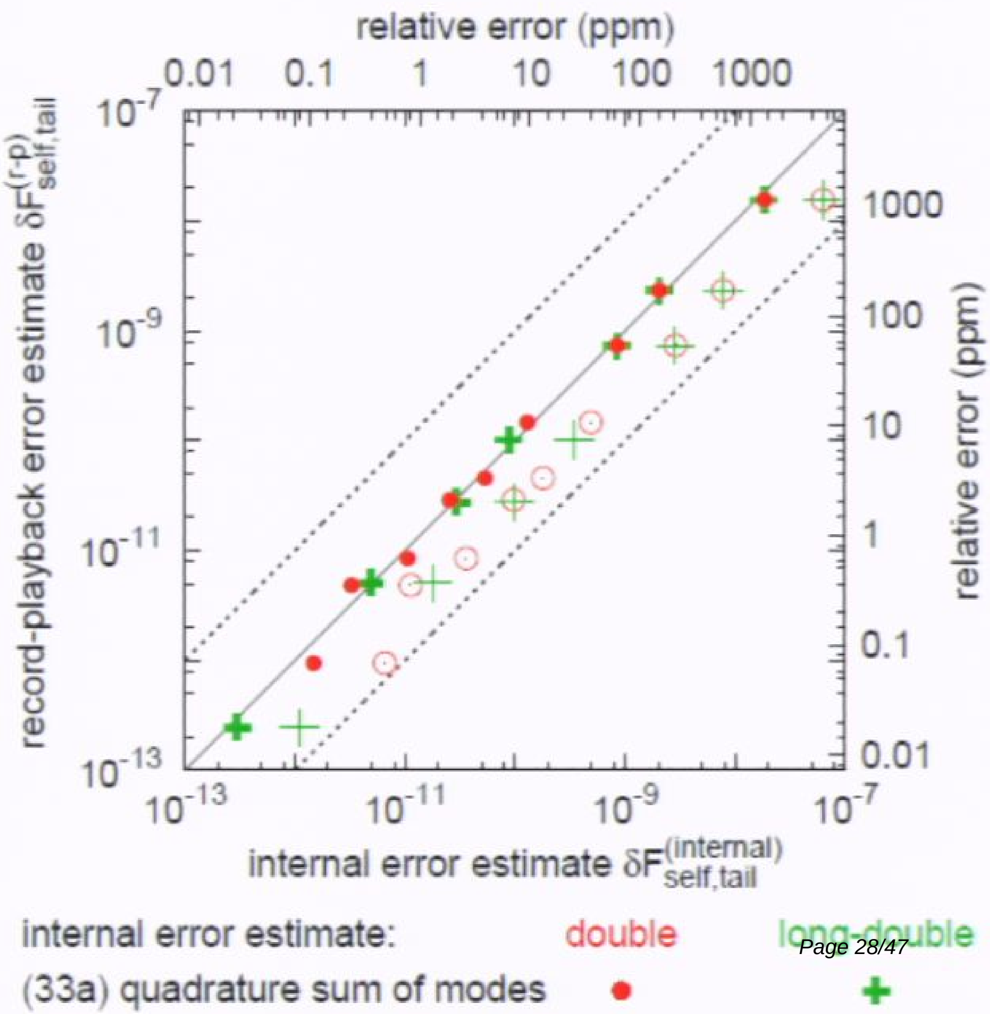
Best case: Actual numerical errors in different modes are (close enough to) statistically independent $\Rightarrow$ should add individual modes' internal error estimates in quadrature.

Error Estimates for the Numerical Force (2)

Compute numerical-force internal error estimates $\delta F^{(\text{internal})}_{\text{self,num}}$ both ways (arithmetic & quadrature sums) and compare with convergence-test error estimates to see which works better:

Error Estimates for the Numerical Force (2)

Compute numerical-force internal error estimates $\delta F^{(\text{internal})}_{\text{self,num}}$ both ways (arithmetic & quadrature sums) and compare with convergence-test error estimates to see which works better:

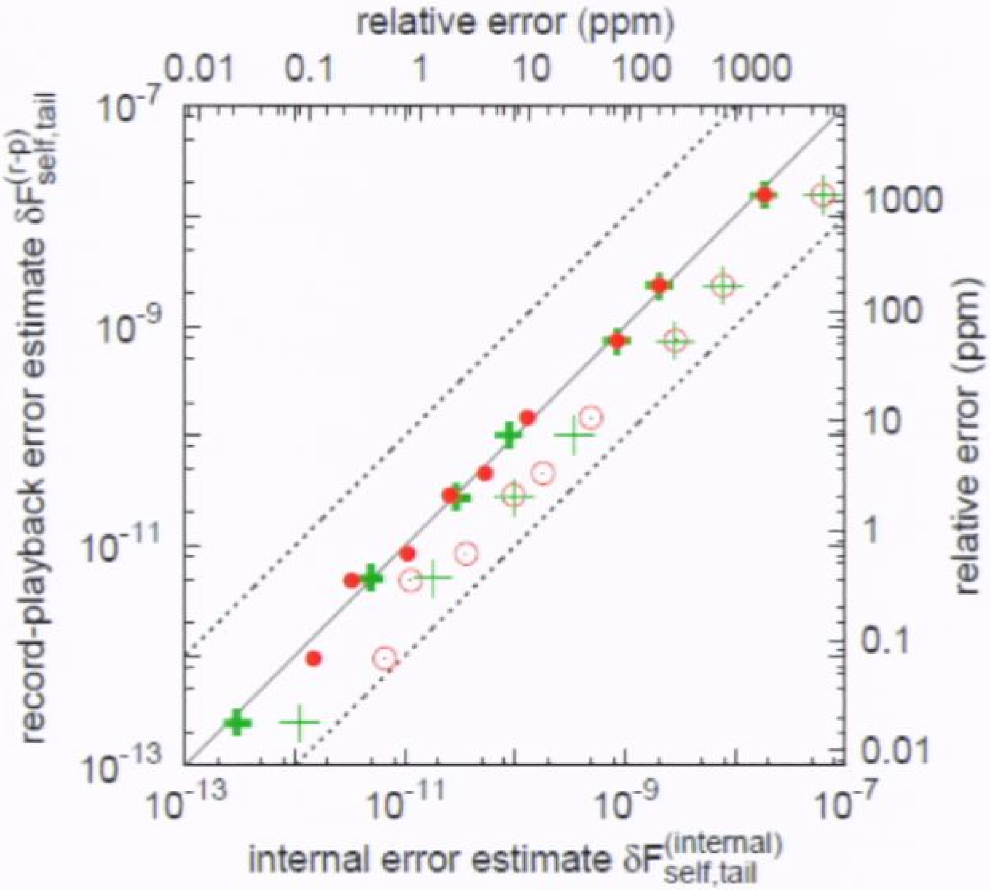


Error Estimates for the Numerical Force (2)

Compute numerical-force internal error estimates $\delta F^{(\mathrm{internal})}_{\mathrm{self,num}}$ both ways (arithmetic & quadrature sums) and compare with convergence-test error estimates to see which works better:

The arithmetic sum systematically overestimates the convergence-test error by $\sim 3\times$, while the quadrature sum gives an excellent estimate of the convergence-test error.

This is true for both double and double-extended arithmetic, *even for errors so small that the individual modes' internal error estimates are unreliable!* (This is apparently due to the averaging inherent in summing



internal error estimate: double long-double

(33a) quadrature sum of modes

Ill-Conditioning of the Tail Fit

The tail-fit basis functions $\{f_p\}$ are nearly linearly dependent, so the tail fit is numerically ill-conditioned. That is, there are linear combinations $\sum_p b_p f_p$ where the linear-combination coefficients $\{b_p\}$ have unit 2-norm, yet the linear combination $\sum_p b_p f_p$ is much smaller than the largest f_p.

Coefficients Being Fitted	Condition Number κ: Basis is $\{f_p\}$	Condition Number κ: Basis is $\{\overline{f}_p\}$
$\{c_4, c_6\}$	2.6×10^3	11
$\{c_4, c_6, c_8\}$	5.9×10^6	100
$\{c_2, c_4\}$	2.2×10^3	8.3
$\{c_2, c_4, c_6\}$	4.4×10^6	66
$\{c_2, c_4, c_6, c_8\}$	8.0×10^9	460

The ill-conditioning can be greatly reduced by renormalizing the basis functions to have similar magnitudes over the range of ℓ used

Error Estimates for the Tail Fit (1)

Again, we don't *a priori* know whether or to what extent the actual errors in the individual modes $F_{\ell,\text{reg}}$ for different ℓ might be correlated, so it's not obvious that we can apply the standard "statistics-101" formulae for error propagation in least-squares fitting:

rror Estimates for the Tail Fit (1)

Again, we don't *a priori* know whether or to what extent the actual errors in the individual modes $F_{\ell,\text{reg}}$ for different ℓ might be correlated, so it's not obvious that we can apply the standard "statistics-101" formulae for error propagation in least-squares fitting:

Best case: Actual numerical errors in different modes are (close enough to) statistically independent that we can use the standard statistical error estimate.

Error Estimates for the Tail Fit (1)

Again, we don't *a priori* know whether or to what extent the actual errors in the individual modes $F_{\ell,\mathrm{reg}}$ for different ℓ might be correlated, so it's not obvious that we can apply the standard "statistics-101" formulae for error propagation in least-squares fitting:

Best case: Actual numerical errors in different modes are (close enough to) statistically independent that we can use the standard statistical error estimate.

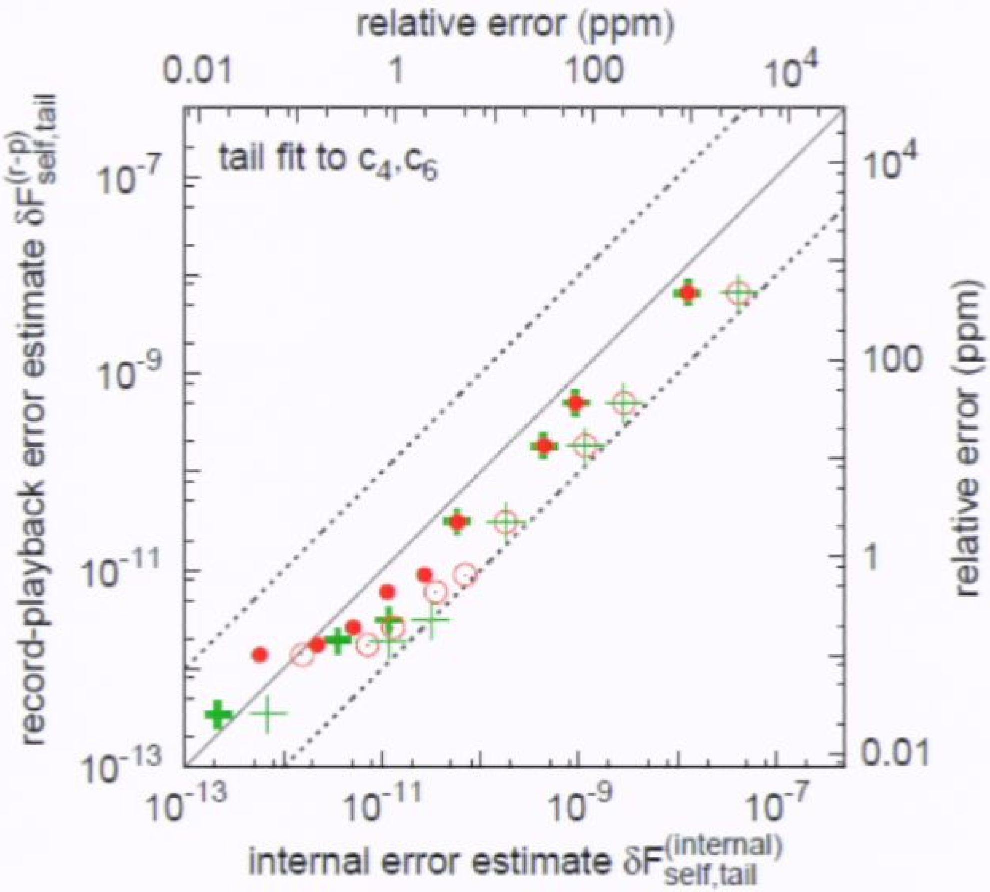
Worst case: Estimate self-force errors by trying a trial fit for each possible combination of $\{-, 0, +\}$ signs of the different modes' internal error estimates. (For Q modes in the fit, there are 3^Q combinations, and hence 3^Q trial fits are needed.)

Error Estimates for the Tail Fit (2)

Compute tail-fit internal error estimates $\delta F^{(\text{internal})}_{\text{self,tail}}$ both ways (statistical and worst-case-of-3^Q-combinations-of-error-signs) and compare with convergence-test error estimates to see which works better:

Error Estimates for the Tail Fit (2)

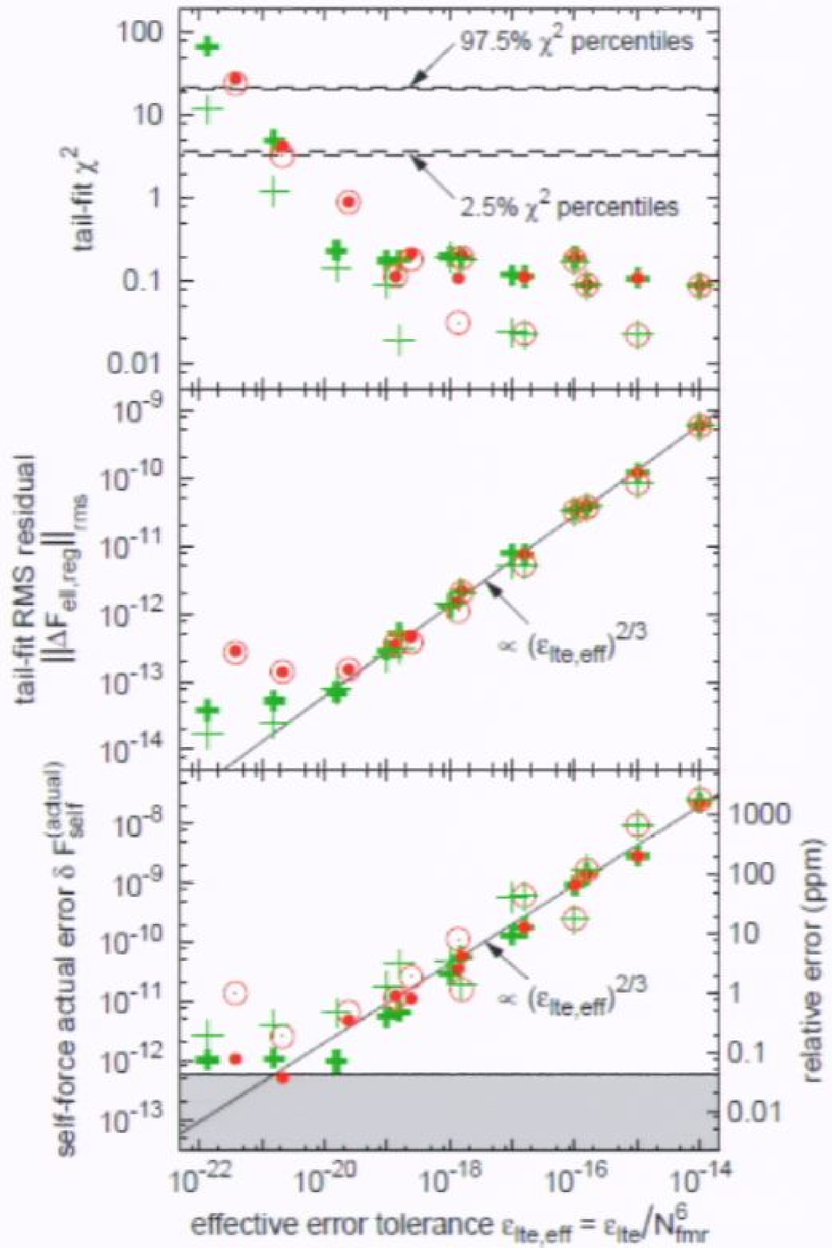
Compute tail-fit internal error estimates $\delta F^{(\text{internal})}_{\text{self,tail}}$ both ways (statistical and worst-case-of-3^Q-combinations-of-error-signs) and compare with convergence-test error estimates to see which works better:



internal error estimate: double long-double

(34a) statistical error in tail fit

Tail-Fit Results

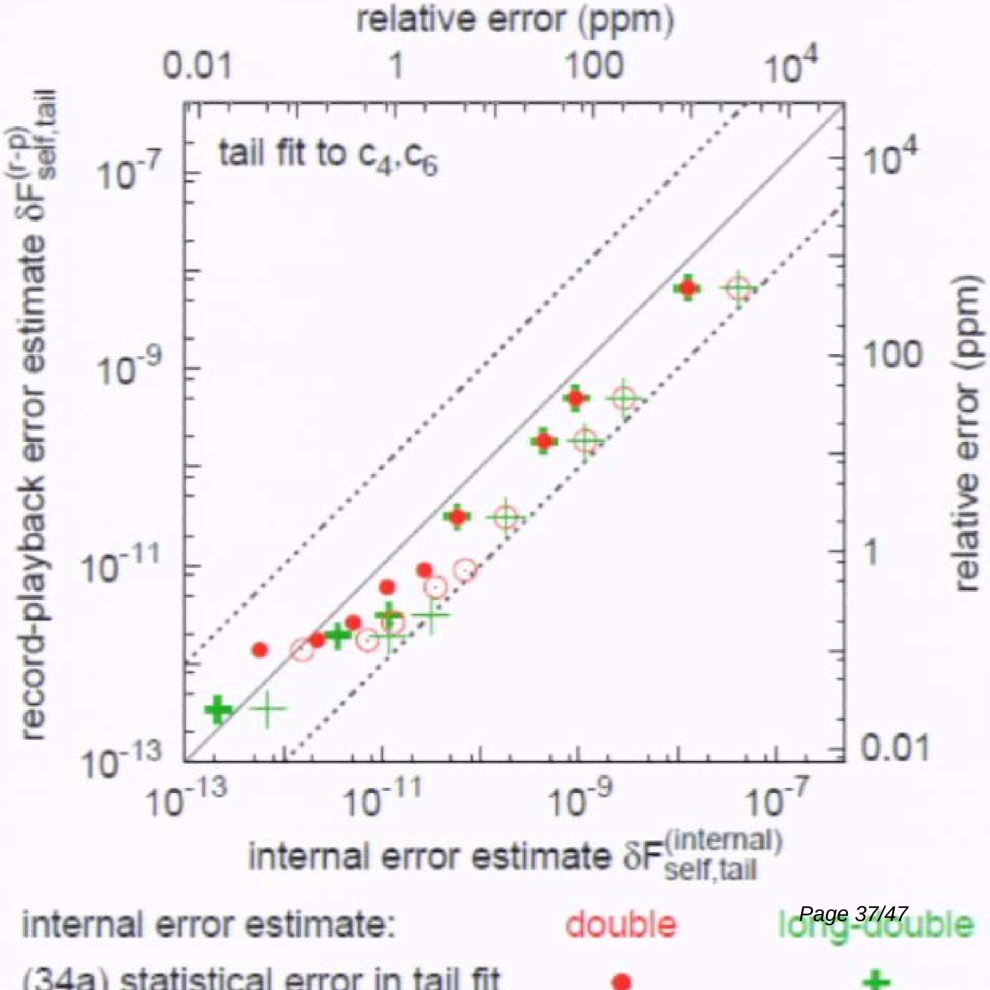


tail fit coefficients: double long-double

Error Estimates for the Tail Fit (2)

Compute tail-fit internal error estimates $\delta F^{(\text{internal})}_{\text{self,tail}}$ both ways (statistical and worst-case-of-3^Q-combinations-of-error-signs) and compare with convergence-test error estimates to see which works better:

The worst-case error estimate systematically overestimates the convergence-test error by $\sim 6\times$, while the statistical error estimate is a fairly good (conservative by $\sim 2\times$) estimate of the convergence-test error.



Ill-Conditioning of the Tail Fit

The tail-fit basis functions $\{f_p\}$ are nearly linearly dependent, so the tail fit is numerically ill-conditioned. That is, there are linear combinations $\sum_p b_p f_p$ where the linear-combination coefficients $\{b_p\}$ have unit 2-norm, yet the linear combination $\sum_p b_p f_p$ is much smaller than the largest f_p.

Coefficients	Condition Number κ	
Being Fitted	Basis is $\{f_p\}$	Basis is $\{\overline{f}_p\}$
$\{c_4, c_6\}$	2.6×10^3	11
$\{c_4, c_6, c_8\}$	5.9×10^6	100
$\{c_2, c_4\}$	2.2×10^3	8.3
$\{c_2, c_4, c_6\}$	4.4×10^6	66
$\{c_2, c_4, c_6, c_8\}$	8.0×10^9	460

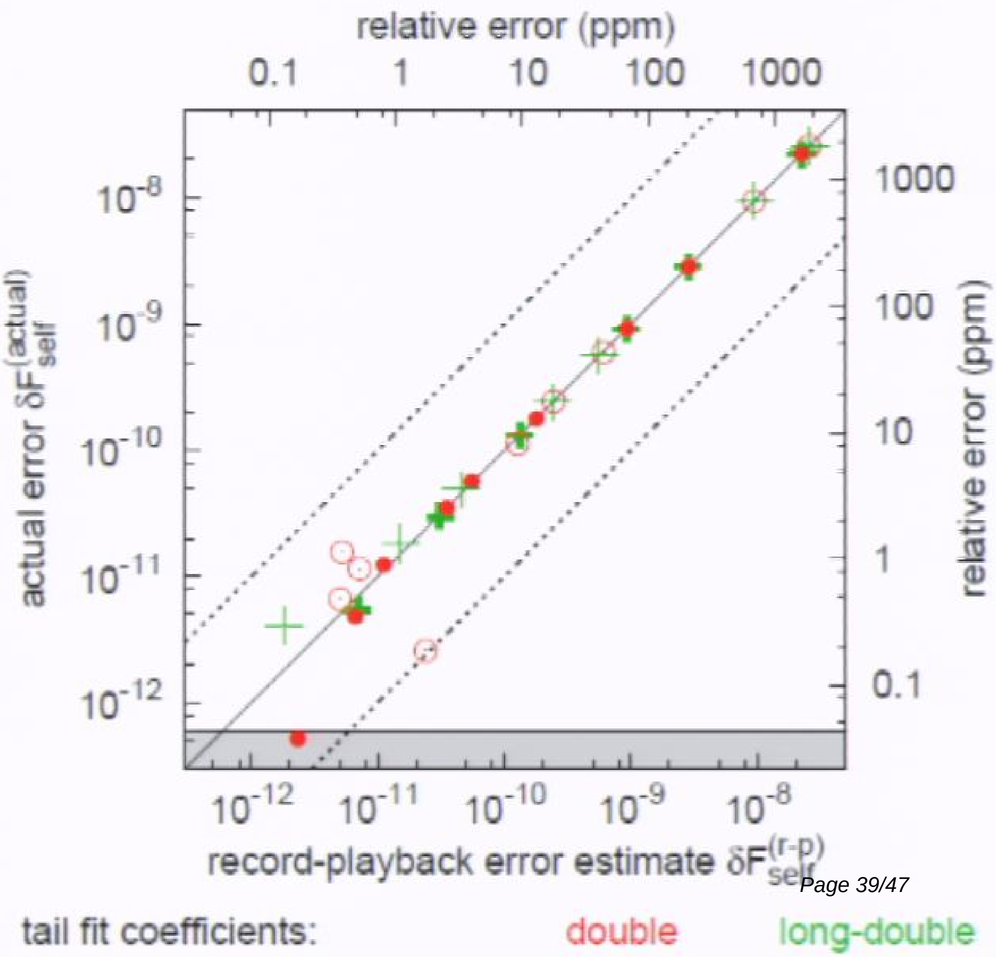
The ill-conditioning can be greatly reduced by renormalizing the basis functions to have similar magnitudes over the range of ℓ used

/alidate Overall Self-Force Error Estimates

Validate the convergence-test error estimates for the overall self-force [quadrature sum of numerical-force and tail-force error; details omitted for brevity] by comparing with highly-accurate frequency-domain results from Detweiler, Messaritaki, & Whiting, PRD **67** (2003), 104016:

Very good agreement except for high-accuracy results with double floating-point precision.

[Shaded region shows error $\le 3\times$ Detweiler, Messaritaki, & Whiting error estimate]



Error Estimates for the Self-Force

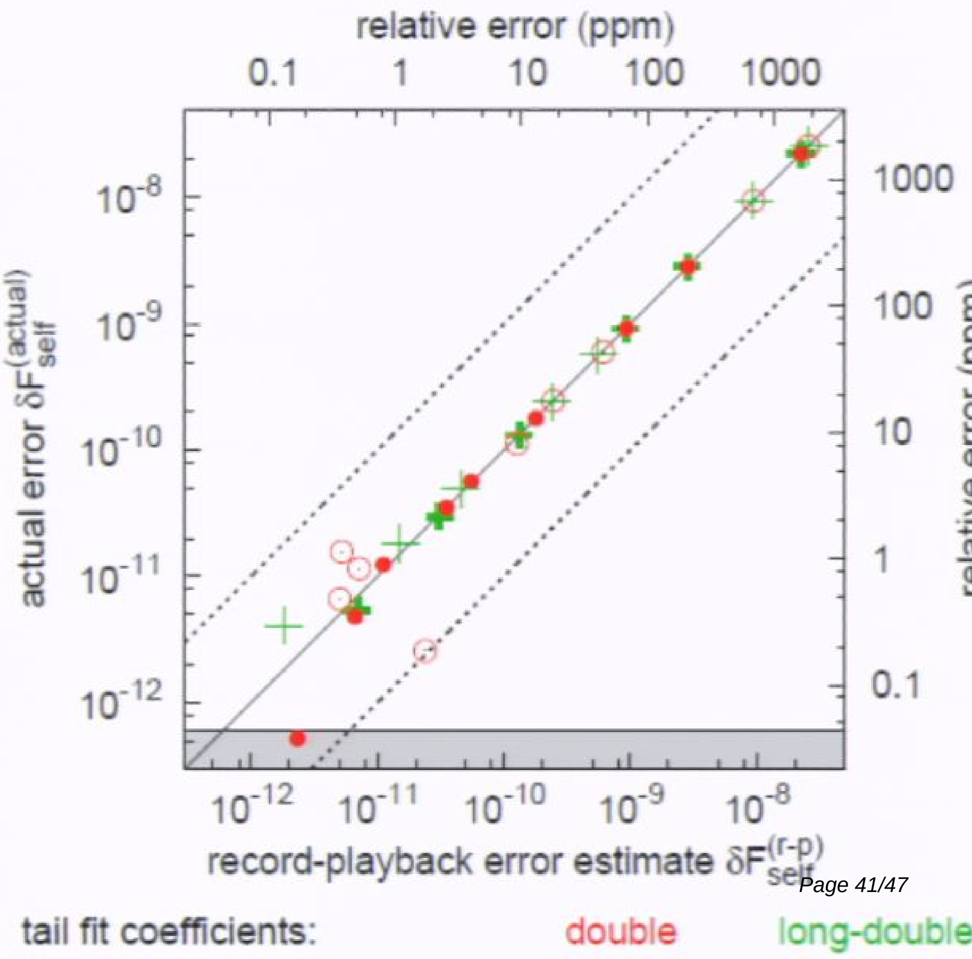
Compute this error estimate as quadrature sum of error estimates for numerical force & tail force [details of validation of quadrature vs. arithmetic sum omitted for brevity].

Validate Overall Self-Force Error Estimates

Validate the convergence-test error estimates for the overall self-force [quadrature sum of numerical-force and tail-force error; details omitted for brevity] by comparing with highly-accurate frequency-domain results from Detweiler, Messaritaki, & Whiting, PRD **67** (2003), 104016:

Very good agreement except for high-accuracy results with double floating-point precision.

[Shaded region shows error $\leq 3\times$ Detweiler, Messaritaki, & Whiting error estimate]



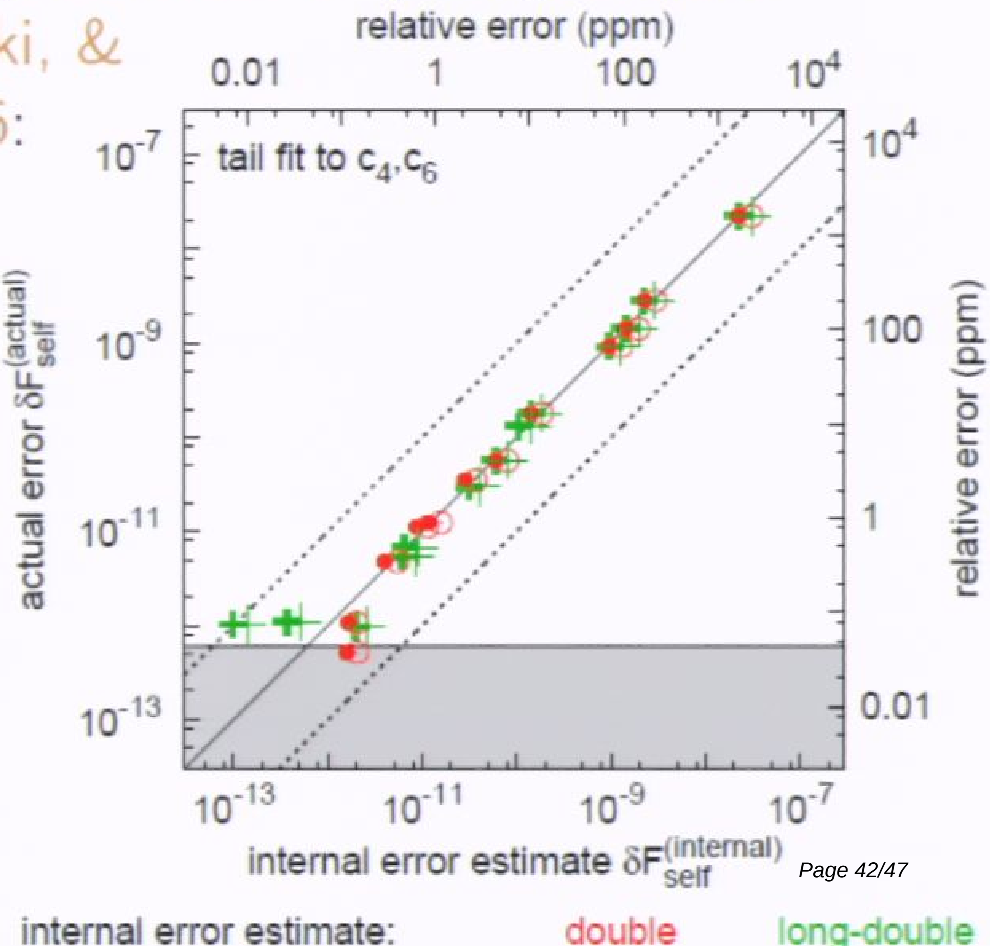
Error Estimates for the Self-Force

Compute this error estimate as quadrature sum of error estimates for numerical force & tail force [details of validation of quadrature vs. arithmetic sum omitted for brevity].

Validate by comparing with highly-accurate frequency-domain results from Detweiler, Messaritaki, & Whiting, PRD **67** (2003), 104016:

Very good agreement except for errors $\lesssim 0.1$ ppm.

[Shaded region shows error $\leq 3\times$ Detweiler, Messaritaki, & Whiting error estimate]



Sample Results

AMR error tolerance 10^{-19} (double-extended floating-point)

method	F_{self} (q^2/M)	$\delta F_{\text{self}}^{\text{(internal)}}$ (ppm)	$\delta F_{\text{self}}^{\text{(actual)}}$ (ppm)
record	$1.378\,448\,82 \times 10^{-5}$	0.44	0.39
playback2	$1.378\,448\,169\,8 \times 10^{-5}$	0.027	0.080
D., M., & W.	$1.378\,448\,28 \times 10^{-5}$	0.015	

CPU time is currently quite slow at more modest accuracy, because numerical parameters are still tuned for the very highest-accuracy test runs:

- Mostly same AMR error tolerance for each ℓ (more efficient would be less-accurate tolerance for modes with small $F_{\ell,\text{reg}}$)
- $\ell = 0$ integrations use a domain size of $30\,000M$ or $100\,000M$.
- Complex wave equation solved for $0 \leq \ell \leq 30$, $\ell = 35$, and

Conclusions

- AMR is **very** useful for self-force calculations.

simultaneously; renormalizing basis functions mostly solves this problem.

Conclusions

- AMR is **very** useful for self-force calculations.
- Internal error estimate from $|\text{outside} - \text{inside}|$.
- It's safe to treat different modes' internal error estimates as statistically independent for purposes of computing error estimates for the numerical force and tail force.
- Tail fit is highly ill-conditionined when fitting many orders simultaneously; renormalizing basis functions mostly solves this problem.

Conclusions

- AMR is **very** useful for self-force calculations.
- Internal error estimate from $|\text{outside} - \text{inside}|$.
- It's safe to treat different modes' internal error estimates as statistically independent for purposes of computing error estimates for the numerical force and tail force.
- Tail fit is highly ill-conditionined when fitting many orders simultaneously; renormalizing basis functions mostly solves this problem.
- Floating-point roundoff limits my AMR code to ~ 0.1 ppm.
- My results agree beautifully with highly-accurate frequency-domain results of Detweiler, Messaritaki, & Whiting, PRD **67** (2003), 104016.
  $\Rightarrow$ confirms correctness of both codes & theories

Conclusions

- AMR is **very** useful for self-force calculations.
- Internal error estimate from $|\text{outside} - \text{inside}|$.
- It's safe to treat different modes' internal error estimates as statistically independent for purposes of computing error estimates for the numerical force and tail force.
- Tail fit is highly ill-conditionined when fitting many orders simultaneously; renormalizing basis functions mostly solves this problem.
- Floating-point roundoff limits my AMR code to ~ 0.1 ppm.
- My results agree beautifully with highly-accurate frequency-domain results of Detweiler, Messaritaki, & Whiting, PRD **67** (2003), 104016.

  $\Rightarrow$ confirms correctness of both codes & theories

See Thornburg, arXiv:1006.xxxx for details (submitted to gr-qc