SEOBNRv3_opt: A Case Study in Code Optimization for the Benefit of LIGO Science

D.A. Buch[1], T.D. Knowles[1], Z.B. Etienne[1], S.T. McWilliams[2]

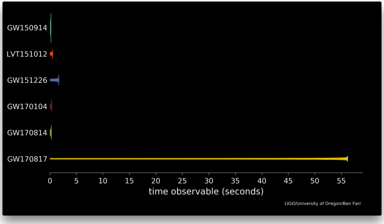
[1]Department of Mathematics
West Virginia University

[2]Department of Physics and Astronomy
West Virginia University

28th Midwest Relativity Meeting

Work Supported by NSF Grants PHY-1607405, EPSCoR 1458952

Motivation: From detection to Inference



- Learning from GWs involves model selection and parameter estimation
- Bayesian inference techniques sample candidate waveforms
- Numerical Relativity far too slow
- IMR PhenomP is a suitable approximate solution

Motivation: Why SEOBNRv3 and why direct optimization?

- SEOBNRv3 is an 8 dimensional alternative to IMR PhenomP
- Unlike Phenom, SEOBNRv3 is notoriously *slow*
- Parameter estimation would take centuries
- ROMs not feasible in 8 dimensions
- T.D. Knowles, C. Devine, D.A. Buch, S.A. Bilgili, T.R. Adams, Z.B. Etienne, S.T. McWilliams. Improving performance of SEOBNRv3 by ∼300x. *Class. Quantum Grav.*, 35 155003 (2018). arXiv: 1803.06346

Anatomy of SEOBNRv3

Four Components of SEOBNRv3 Evaluation:

- Solve Hamilton's Equations of Motion

$$\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\frac{\partial \mathcal{H}_{\mathrm{SEOBNRv3}}}{\partial \boldsymbol{q}}, \quad \frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} = \frac{\partial \mathcal{H}_{\mathrm{SEOBNRv3}}}{\partial \boldsymbol{p}}.$$

- Interpolate to even sampling
- Attach Ringdown
- Fourier Transform (FFT)

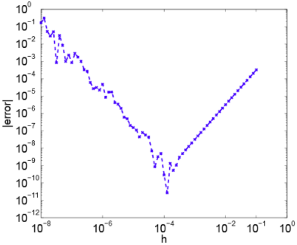
Analytic Derivatives

The Woes of Finite Differences:

$$\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\frac{\partial \mathcal{H}_{\mathrm{Sv3}}}{\partial \boldsymbol{q}}$$

$$\frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} = \frac{\partial \mathcal{H}_{\mathrm{Sv3}}}{\partial \boldsymbol{p}}$$

Error vs. Grid Size



- Finite Difference derivatives $\rightarrow$ huge roundoff error!
- Migration to analytical derivatives
- Larger time steps (x6 faster). Transition from 4th order Runge-Kutta to 8th order Runge-Kutta (x2 faster)

Guided Automatic Differentiation (GAD)

Hamiltonian calculations still accounted for $\sim$80% of total runtime.
Mathematica-generated derivatives suffered from:

- Many repeated calculations, common expressions not shared between derivatives
- A far cry from being human-readable

Solution: *Guided Automatic Differentiation* (GAD)

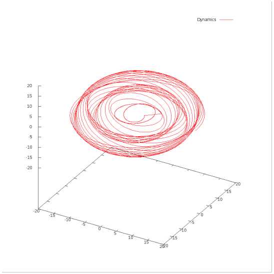
Derivative scheme	Space derivative (FLOPs)	Momentum derivative (FLOPs)	Spin derivatives (FLOPs)	Total (FLOPs)
Mathematica-based	3 x 5073	3 x 2319	6 x 4333	48174
GAD	3 x 1418	3 x 527	6 x 1264	13419

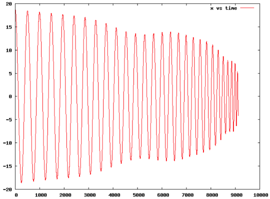
On-the-Fly Interpolation

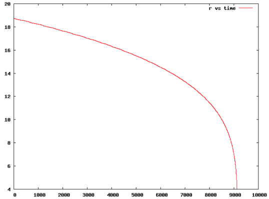
- Replace cubic spline interpolation with "Dense Output" technique
- Carries out cubic Hermite interpolation on-the-fly during the ODE-solve using derivative and function values available within that routine.
- Increased faithfulness to SEOBNRv3

Physical Optimizations

Coordinate singularities discouraged use of spherical coordinates in original code base, despite natural fit to the problem.







Conclusions

- Centuries $\rightarrow$ Months
- Code optimization can be key to unlocking new science!
- Approach optimization strategically - Profile.
- Rudimentary computer-science optimizations can be helpful
- Creative applications of domain knowledge can sometimes be *more* helpful