Black Hole Perturbation Toolkit

Barry Wardell
University College Dublin

Advances in Computational Relativity,
ICERM, 16th September 2020

bhptoolkit.org

Workshop outline

- Overview of the toolkit (~30 minutes)
- Tutorial: Getting started with the Black Hole Perturbation Toolkit (~45 minutes)
- Tutorial: Black hole scattering and absorption (~30 minutes)
- Tutorial: Gravitational wave fluxes from extreme mass-ratio inspirals (~60 minutes)

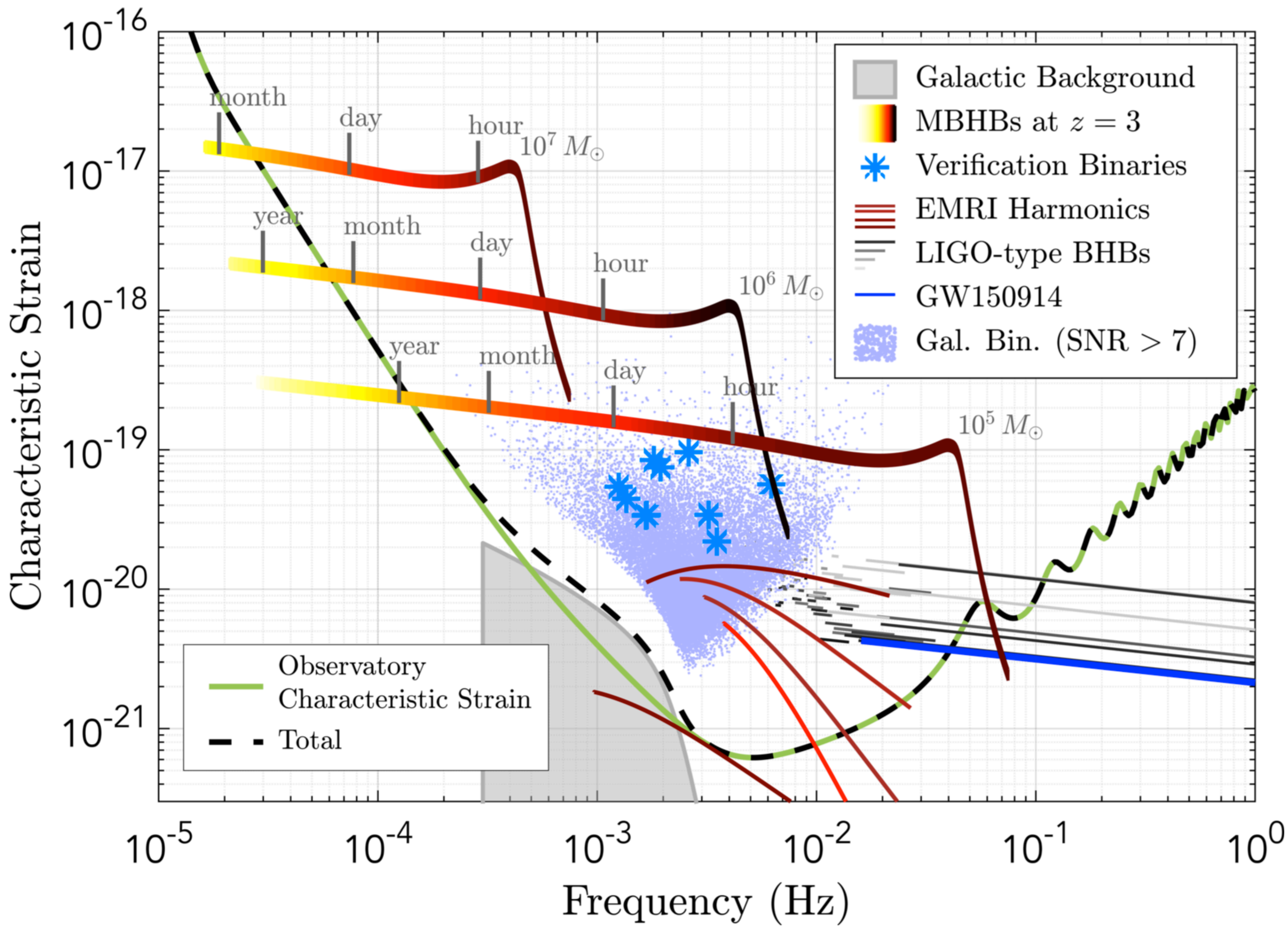
Please ask questions during this workshop on Zoom chat and/or in the workshop Slack channel

https://join.slack.com/t/icermfall2020/shared_invite/zt-h5qv0ij6-OAVgEOrz7S3ihXpEz6JJnA

If you are interested in the long term, please feel free to join the the Black Hole Perturbation Toolkit Slack channel

https://join.slack.com/t/bhptoolkitworkspace/shared_invite/zt-hejyvhpx-6r_rjQk9wwLca34Eg~Ev9g

Black Hole Perturbation Theory: why do we need it?



Detect and estimate parameters for extreme mass-ratio inspirals (EMRIs) using LISA

Black Hole Perturbation Theory: what is it?

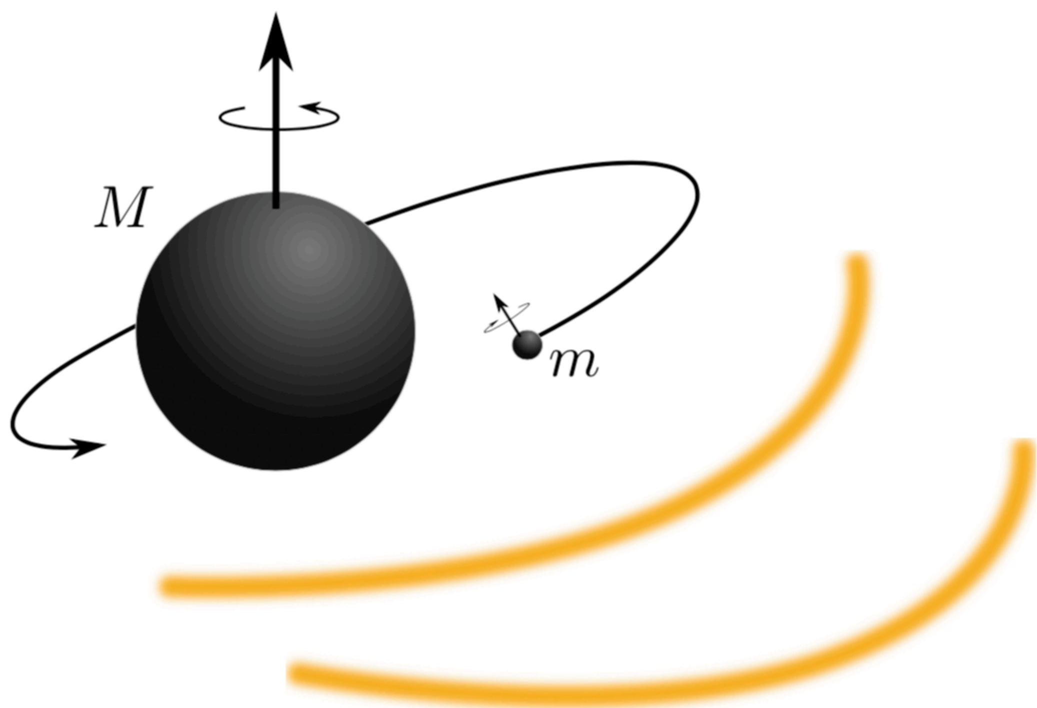


Image credit: A. Pound

- Both bodies spinning with spins not aligned, high eccentricity
- Highly relativistic, strong fields: cannot use post-Newtonian theory
- Wide separation of length- and time-scales: cannot use numerical relativity
- Use the mass ratio, ε = m/M, as a small parameter in perturbation theory

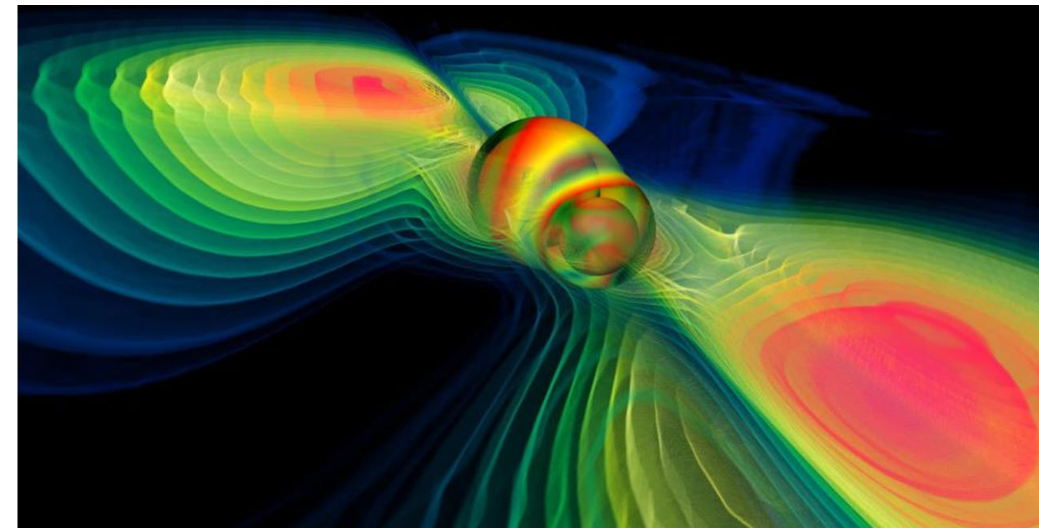
$$g_{\alpha\beta} = \bar{g}_{\alpha\beta} + \epsilon h^{(1)}_{\alpha\beta} + \epsilon^2 h^{(2)}_{\alpha\beta} + \mathcal{O}(\epsilon^3)$$

$$\Phi(t) = \epsilon^{-1}\Phi_0\left[\langle h^1_{\textbf{diss}}\rangle\right] + \Phi_1\left[h^1_{\textbf{diss,osc}} + h^1_{\textbf{cons}} + \langle h^2_{\textbf{diss}}\rangle\right] + \mathcal{O}(\epsilon)$$

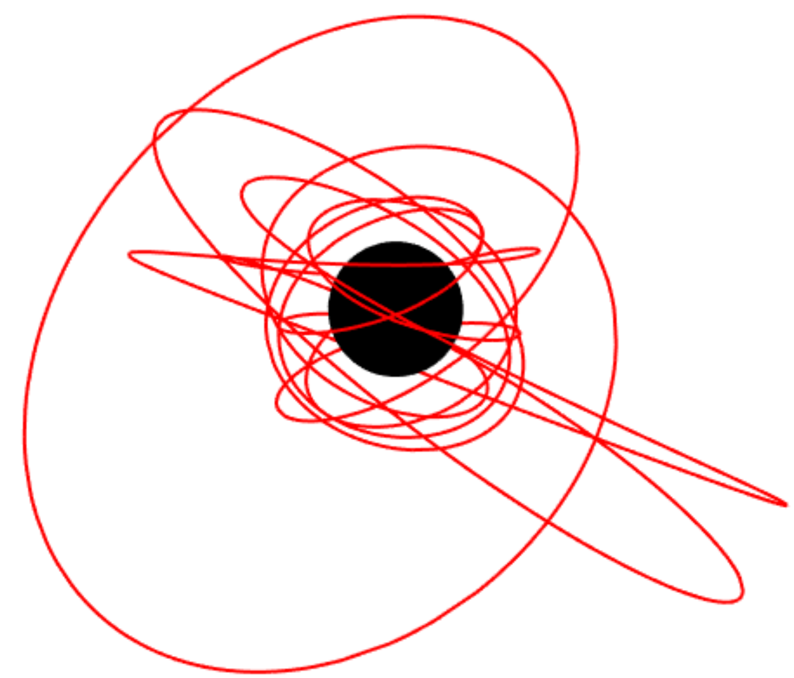
Key observation: writing a first-order generic Kerr Teukolsky code is already well beyond the scope of a PhD student project

Black Hole Perturbation Theory: analogy with NR

NR:
- 2005 was the breakthrough year
- first detection was 10 years later
- models just ready in time
- key: successful software collaborations



BHPTheory:
- first post-adiabatic waveform in 2020
- LISA launch in ~12-14 years
- need to get models ready
- key: successful software collaborations



Introducing the Black Hole Perturbation Toolkit

http://bhptoolkit.org

"Our goal is for less researcher time to be spent writing code and more time spent doing physics. Currently there exist multiple scattered black hole perturbation theory codes developed by a wide array of individuals or groups over a number of decades. This project aims to bring together some of the core elements of these codes into a Toolkit that can be used by all.

Additionally, we want to provide easy, open access to data from black hole perturbation codes and calculations."

Community driven, led by

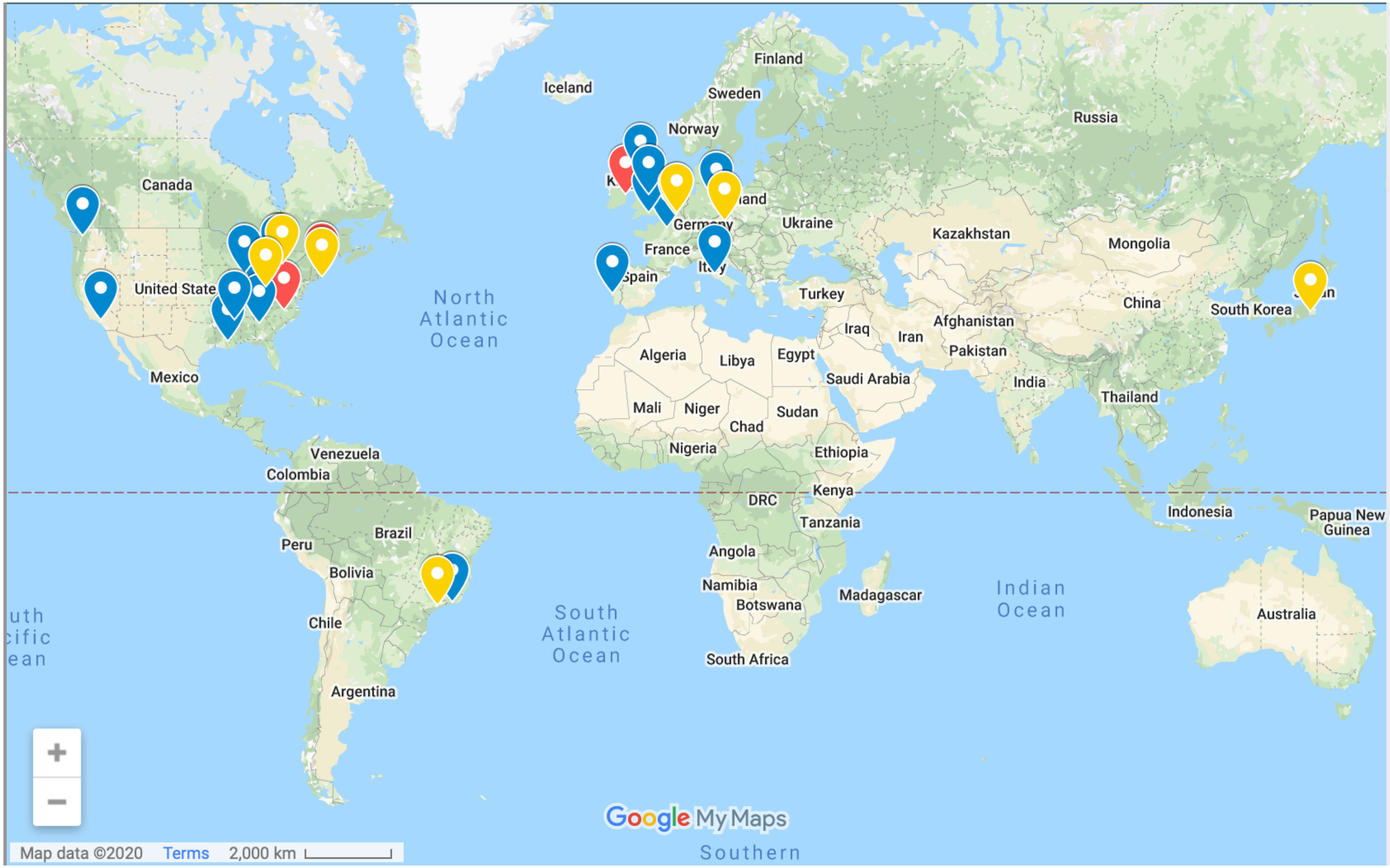
UCD DUBLIN

THE UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

MIT KAVLI INSTITUTE

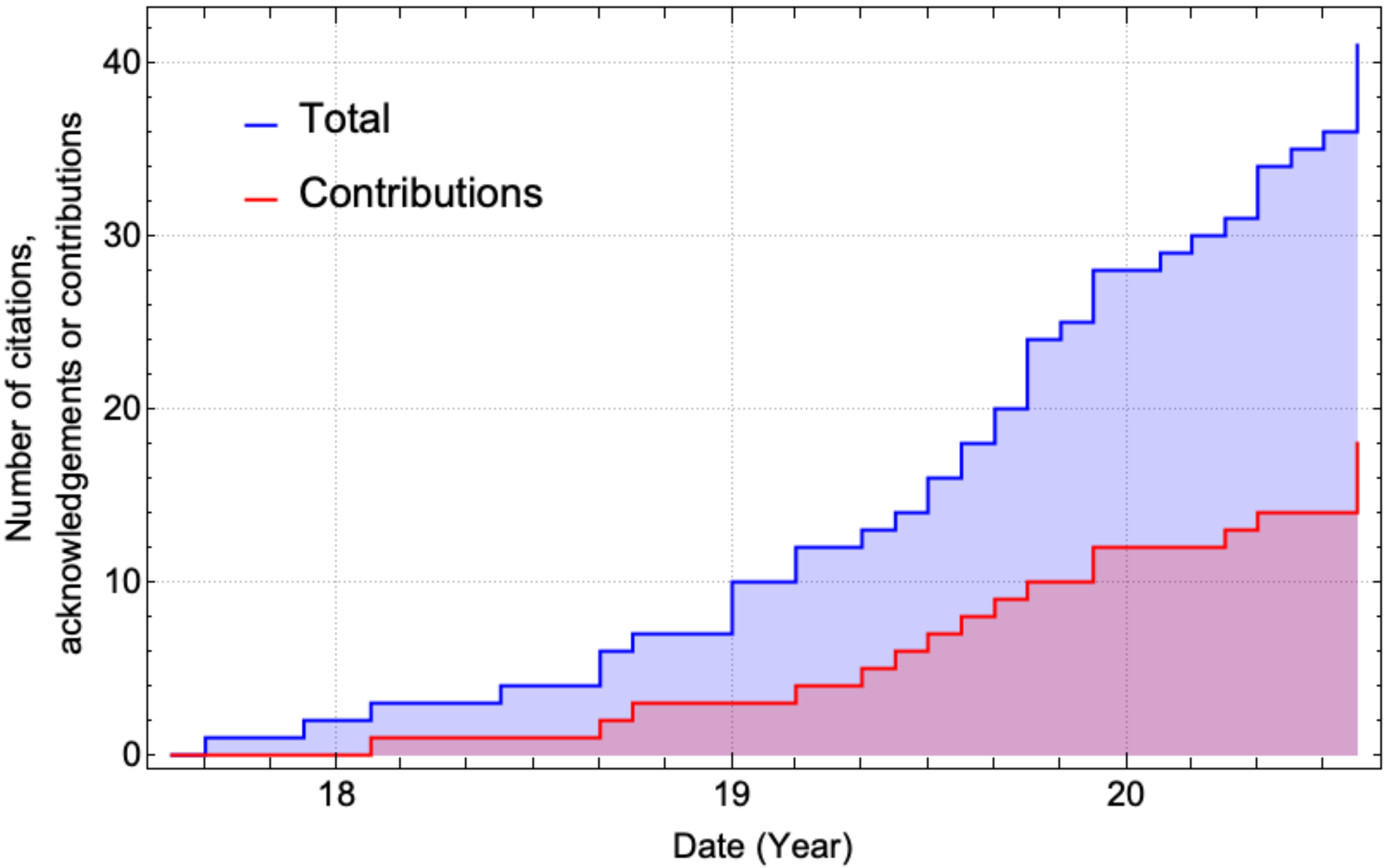
but many other individuals and groups have also contributed…

Contributors and users



http://bhptoolkit.org/users.html

Contributors and users



Since August 2017: 41 papers cite the Toolkit and 18 have contributed code or data

Current components

Code

Target 3 main languages:
- Mathematica
- Low-level (C/C++/Fortran)
- Python

but happy to include high-quality, documented code in any language.

Most code released under MIT or GPL licences.

Data

- Fluxes
- Local invariants
- High-order PN series
- Regularisation parameters

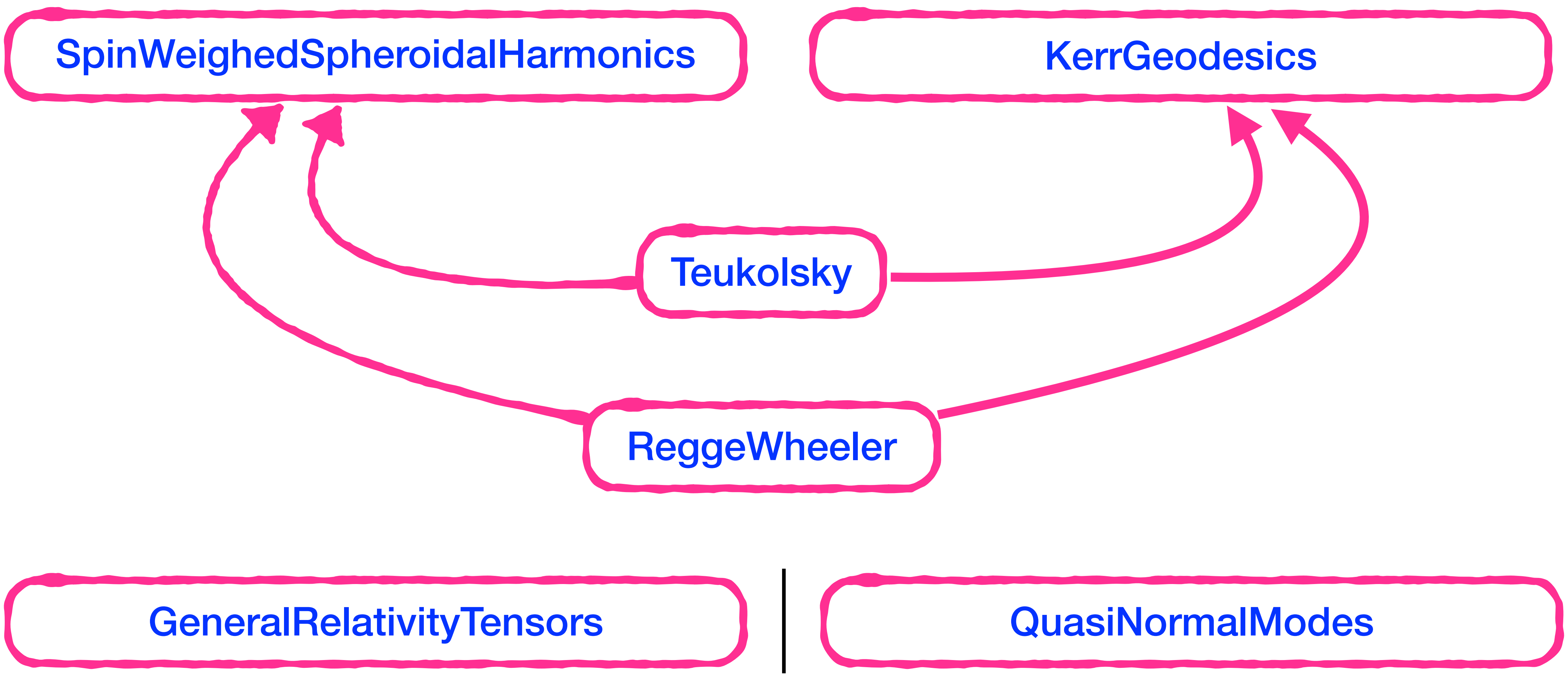
Currently most of the data is for circular, equatorial orbits

But eccentric, generic data coming online

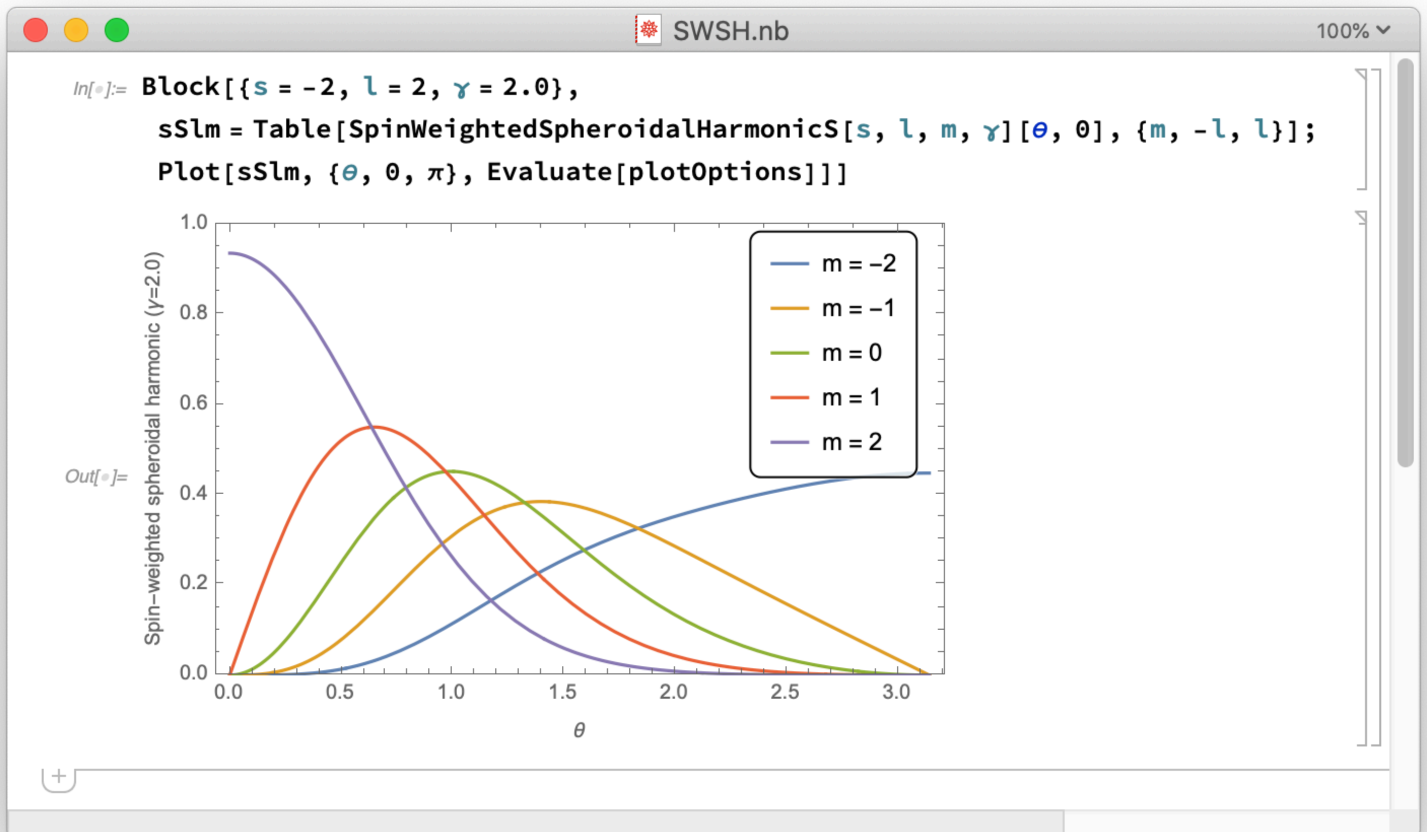
Store large datasets in Google Drive initially. Later transition to Zenodo for finalised dataset

Current components

Code: Mathematica



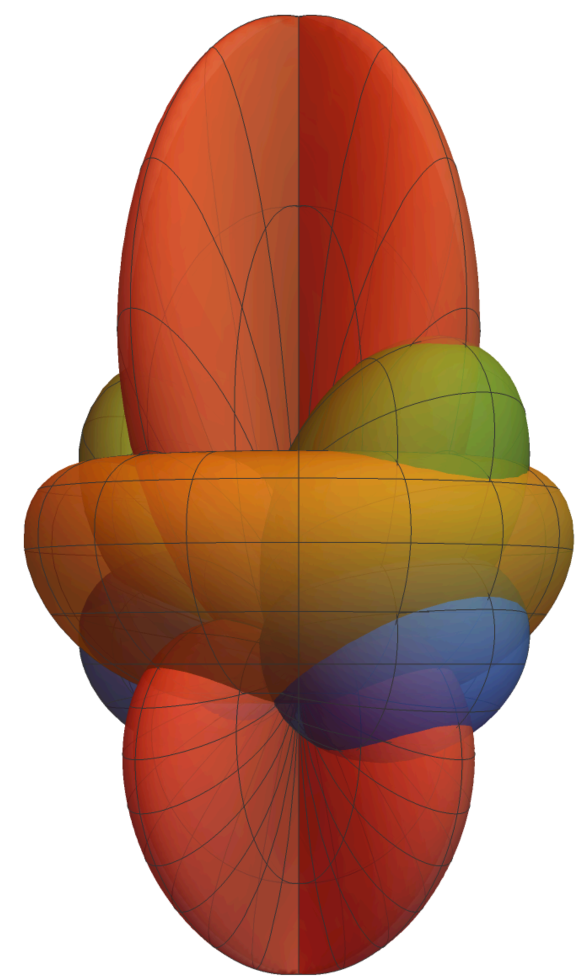
Mathematica: Spin-weighted Spheroidal Harmonics



Arbitrary precision and analytics functions for:

- eigenvalues
- harmonics

Spin−weighted spheroidal harmonic (γ=2.0)

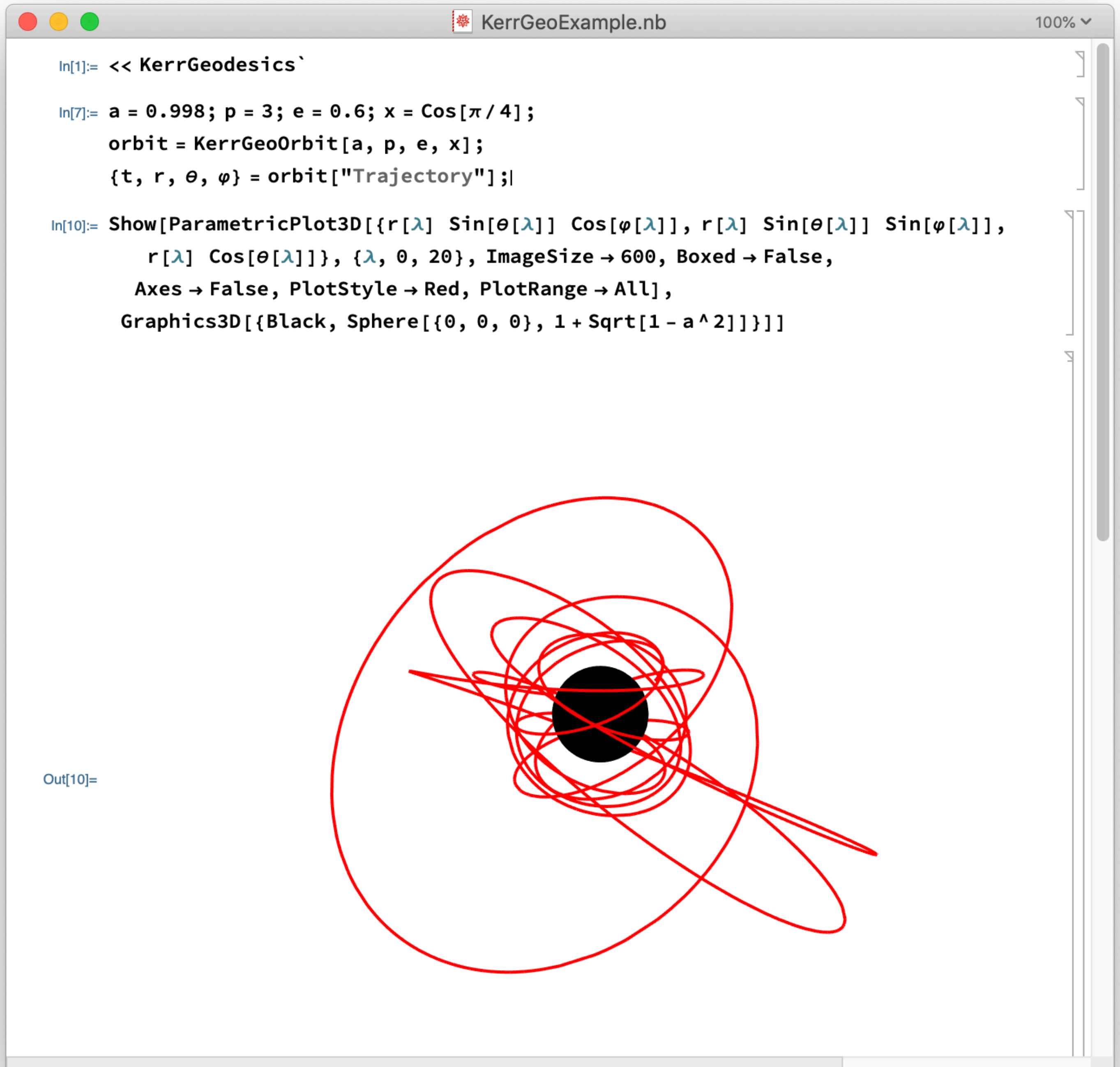


Series[SpinWeightedSpheroidalEigenvalue[s, l, m, γ], {γ, 0, 1}]

$$\left(l^2 + l - s(s+1)\right) + \gamma\left(-\frac{2ms^2}{l(l+1)} - 2m\right) + O\left(\gamma^2\right)$$

Can also perform series expansion about $\gamma = \infty$

Mathematica: Kerr Geodesics



Compute properties of bound timelike geodesics of Kerr spacetime

- constants of motion
- orbital frequencies
- special orbits (ISCO, ISSO, separatrix)
- orbital trajectory

Also recently added sub-package for parallel transport calculations

Mathematica: Teukolsky equation

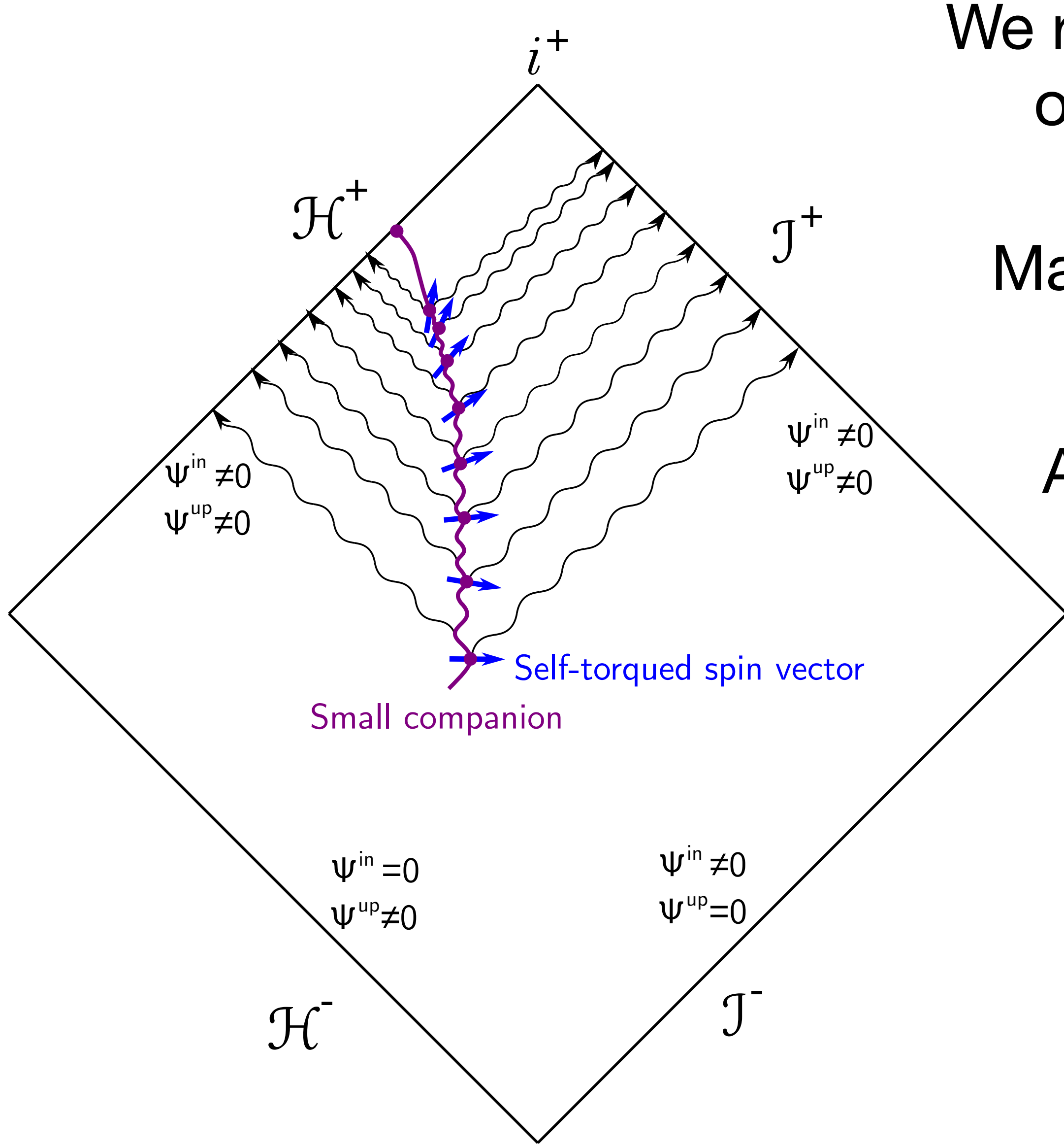
Fluxes.nb 100%

```
In[1]:= << Teukolsky`

In[2]:= With[{a = 0.9`16, p = 10.0`16, e = 0, x = 1},
         orbit = KerrGeoOrbit[a, p, e, x]]

Out[2]= KerrGeoOrbitFunction[ a: 0.9000000000000000  p: 10.00000000000000  e: 0  x: 1
                              Parametrization: Mino ]

In[3]:= With[{s = -2, l = 2, m = 2},
         ψ4[l, m] = TeukolskyPointParticleMode[s, l, m, 0, 0, orbit]]

Out[3]= TeukolskyMode[ s: -2  l: 2  m: 2  a: 0.9000000000000000  ω: 0.0614953644485709
                       Type: PointParticleCircular ]

In[4]:= ψ4[2, 2]["Fluxes"]

Out[4]= <|Energy → <|ℐ → 0.000022273, ℋ → -5.984×10^-8|>,
         AngularMomentum → <|ℐ → 0.0007244, ℋ → -1.9461×10^-6|>|>
```

Extremely easy to compute fluxes to arbitrary precision

Point particle source implemented for circular orbits for s={-2,-1,0}. Fully generic orbits coming soon.

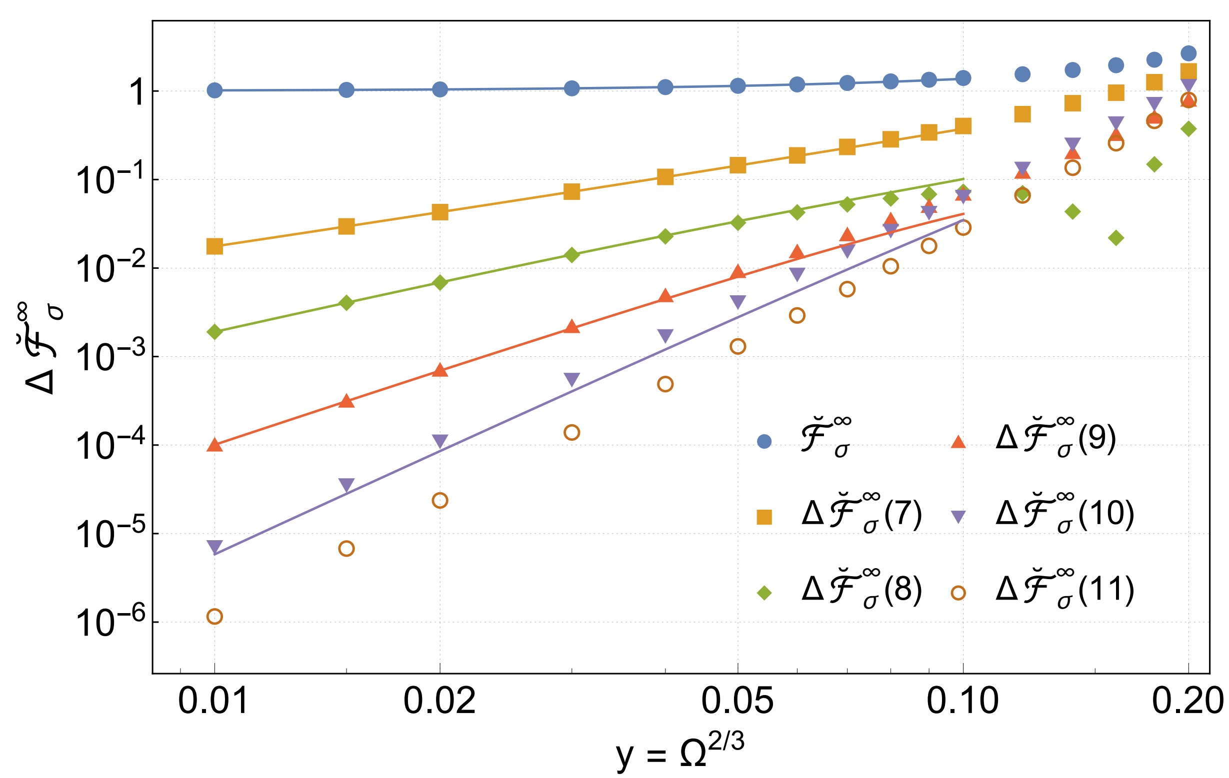
Mathematica: complete calculation



We recently computed the flux from a spinning body on a circular orbit in Schwarzschild spacetime

Many long hours spent writing and debugging code

At the end, the Toolkit was mature enough that we calculated the same flux in an afternoon



arXiv:1912.09461

Current components

Code: C/C++/Fortran

EMRI Kludge Suite	Kludge waveforms by Alvin Chua +
Gremlin	Teukolsky solver from Scott Hughes +
Fast Self-forced Inspirals	Self-force inspirals from Niels Warburton +
selfforce-1d	Time domain self-force

Code: Python/Sage Math

EMRISurrogates	Surrogate model for quasi-circular inspirals by Rifat +
kerrgeodesic_gw	GWs for circular orbits by Eric Gourgoulhon +
qnm	Quasi-normal modes from Leo Stein

Python Example: EMRISurrogate

Latest addition to the Toolkit following Rifat+, arXiv:1910.10473

This paper both used and then extended the Toolkit

Surrogate model for EMRI waveforms

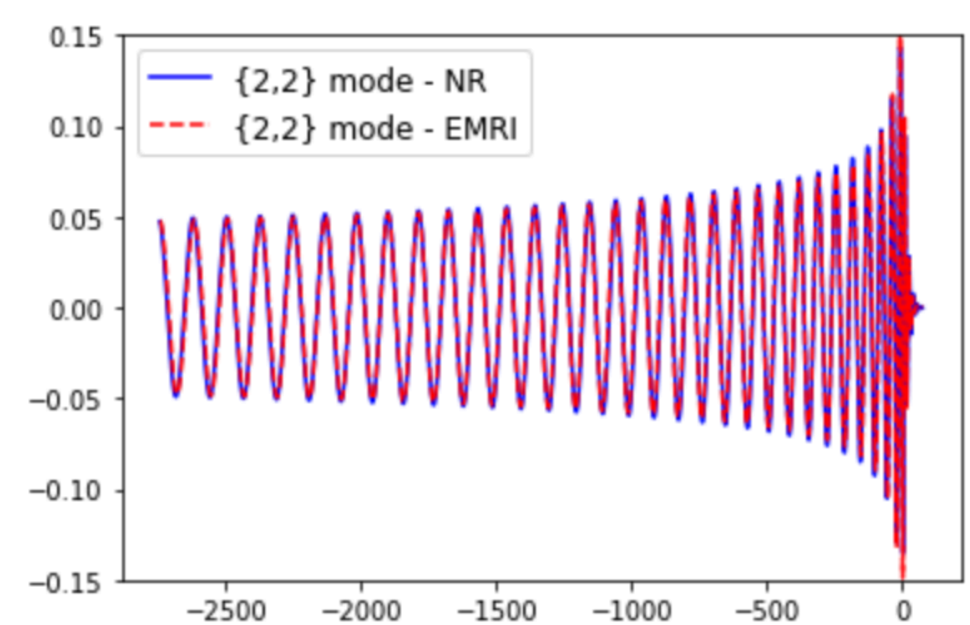
Implementation within Python, with example notebooks in the repository

Surrogate data stored in zenodo

```
# generate a q=8 EMRI and NR waveform
q=8
time_emri, h_emri = emriSur.generate_surrogat

modes_spec, times_spec, hp_spec, hc_spec = s
h_spec = hp_spec + 1.0j*hc_spec
h_spec = modes_list_to_dict(modes_spec,h_spe
```

```
# plot waveforms after minimizations
plt.figure(1)
plt.plot(common_times, np.real(h_nr_aligned),'b
plt.plot(common_times, np.real(h_emri_aligned),
plt.legend(fontsize=12)
plt.ylim(-0.15,0.15)

plt.figure(2)
plt.plot(common_times, np.imag(h_nr_aligned),'b
plt.plot(common_times, np.imag(h_emri_aligned),
plt.legend(fontsize=12)
plt.ylim(-0.15,0.15)
```

```
(-0.15, 0.15)
```



Current components

Data:

CircularOrbitData	Flux, local invariants, etc for circular orbits
PostNewtonianSelfForce	Mathematica module to load high-order Post-Newtonian series
Regularisation Parameters	Mathematica notebooks containing regularisation parameters
Mathematica Toolkit Examples	Mathematica notebooks showing example usage of various modules

These are all small datasets and thus stored in GitHub

Data examples: PostNewtonianSelfForce

Combining PN and self-force techniques leads to very high order series (e.g., 22PN)

Currently have 57 PN series in the Toolkit and a package to search through and manipulate them

PostNewtonianSelfForceExample.nb

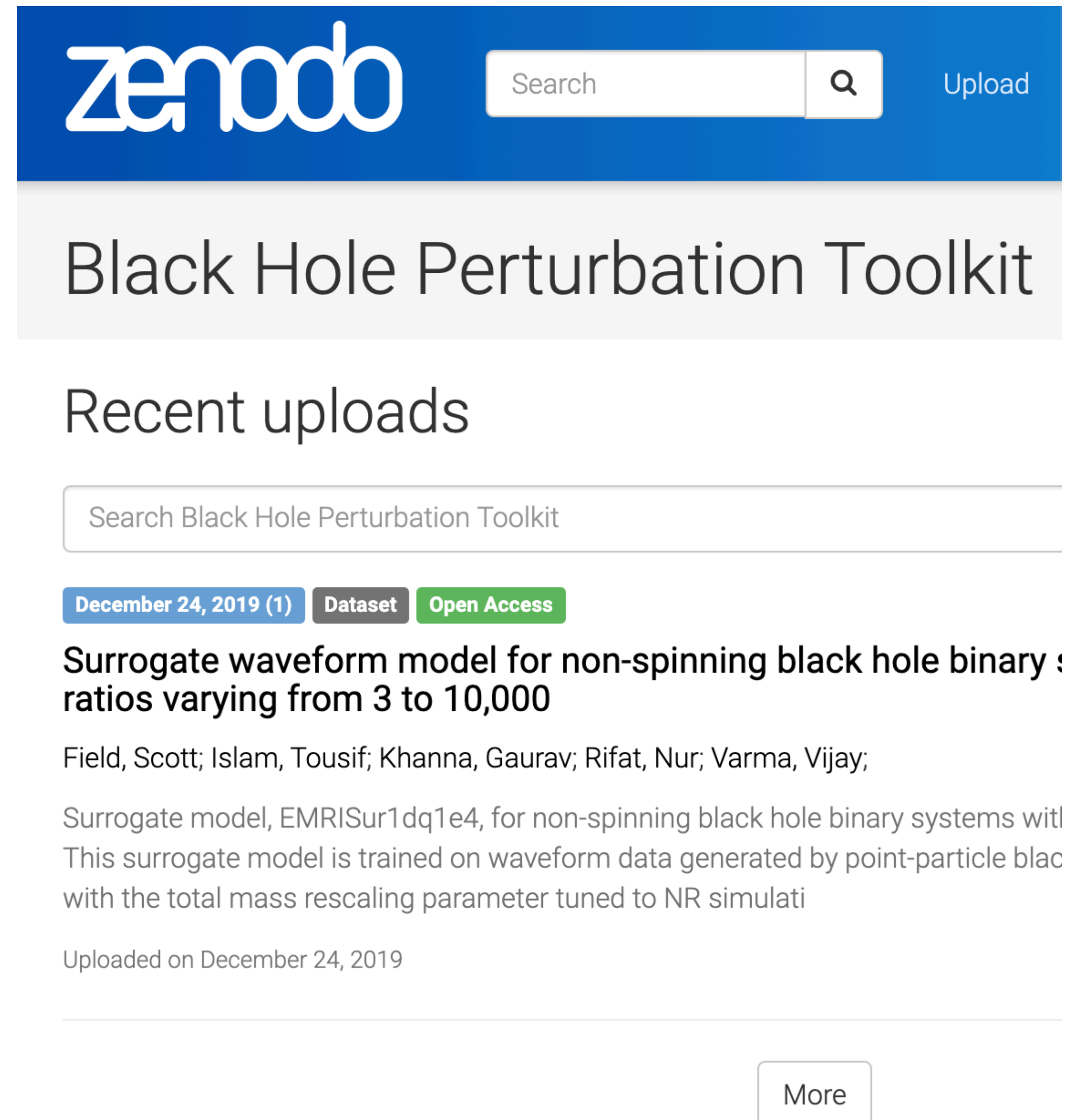
125%

```
In[17]:= << PostNewtonianSelfForce`

In[33]:= FluxPN22 = PostNewtonianExpansion[{"Flux", "l2m2", "!Spin"}]

Out[33]= PostNewtonianData[ Name: Schwarzschild Circular Orbit Infinity Flux for the l = 2, m = 2 mode
                            Starting order: 5     Maximum order: 27.5 ]

In[34]:= FluxPN22["Series"] + O[y]^8
```

Out[34]=
$$\frac{32\,y^5}{5} - \frac{3424\,y^6}{105} + \frac{128}{5}\pi\,y^{13/2} + \frac{153\,088\,y^7}{6615} - \frac{13\,696}{105}\pi\,y^{15/2} + O[y]^8$$

Data storage

Data testing and storage

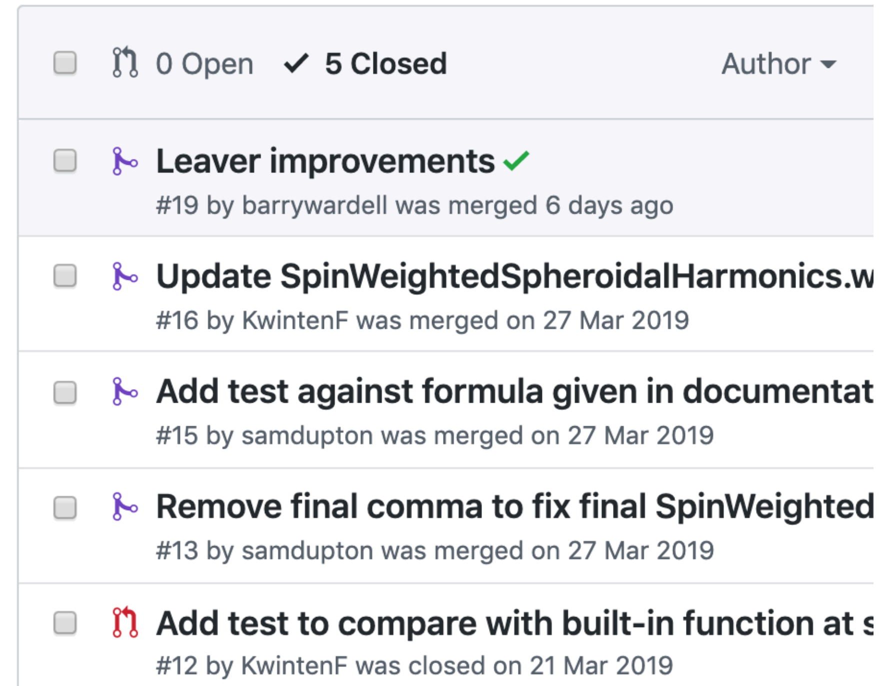


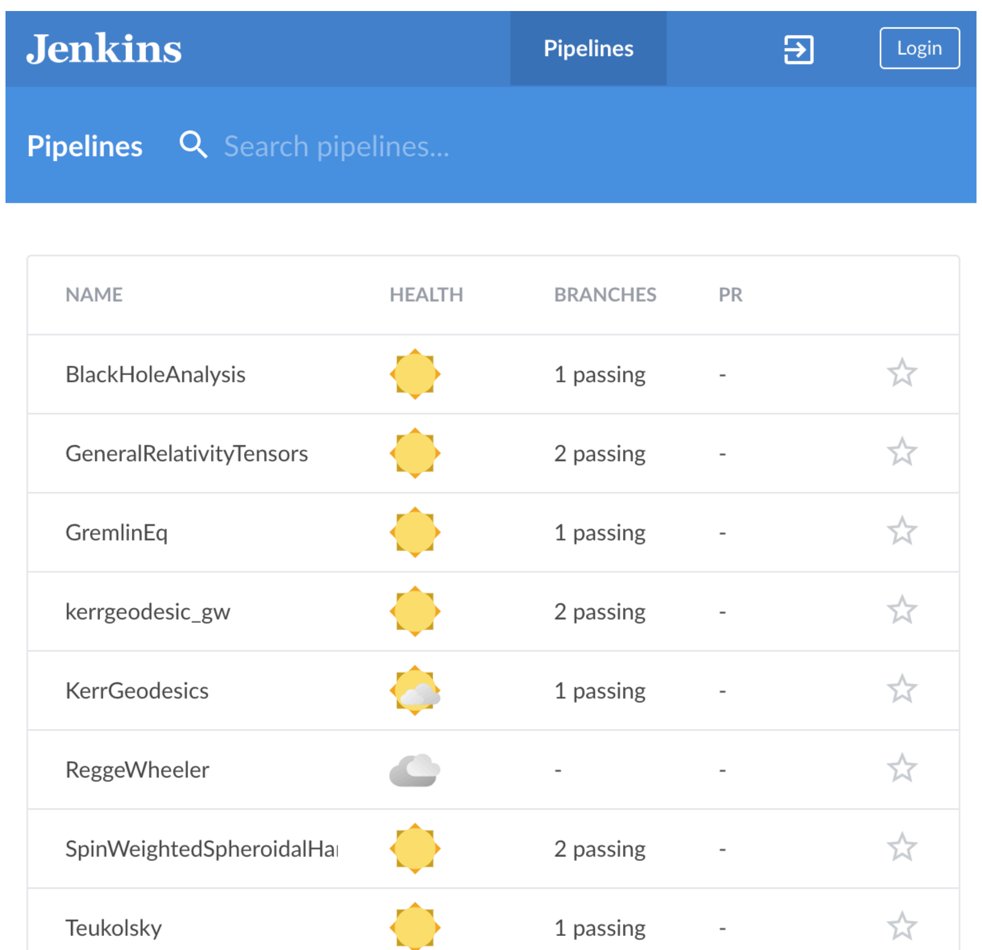
1. Data shared on a Google Drive with other Toolkit users but without warranty

2. Verified and/or published data stored on Zenodo for longevity. Data gets a unique DOI

Hosting and continuous integration testing

BHPToolkit is hosted on GitHub
- website designed in gh-pages
- built-in issue tracker
- continuous integration testing with Jenkins
- Jenkins integration with GitHub





Jenkins (jenkins.io) runs unit tests every time code is committed

http://build.bhptoolkit.org/blue
(password protected)

How to get involved

1. Download and use the code

Install this package!

Teukolsky

A Mathematica package for computing solutions to the Teukolsky equation. Note this package depends upon the SpinWeightedSpheroidalHarmonics and the KerrGeodesics package to run.

Explicitly the package computes solutions to:

$$\Delta^{-s}\frac{d}{dr}\left[\Delta^{s+1}\frac{dR}{dr}\right]+\left[\frac{K^2-2is(r-M)K}{\Delta}+4is\omega r-\lambda\right]R=\mathcal{T}$$

2. Submit issues to the issue tracker
   - Bugs
   - Enhancements

BlackHolePerturbationToolkit / **Teukolsky**

<> Code | Issues 3 | Pull requests 1 | Actions

3. Submit a pull request
   - bug fix
   - new unit tests
   - documentation
   - new functionality

How to get involved

Toolkit workshops

- First public workshop in ~~Prague~~ in March 2020 (funded by COST)
- Second workshop part of the ICERM meeting in Brown (this workshop)
- Workshops offer training for new users coming to the Toolkit, also an opportunity for developers to come together

Paper

- Currently preparing a draft of a BHPToolkit paper
- Aiming to have it out before the end of the year
- All contributors to the Toolkit will be invited to co-author

How to cite

Until the paper is published please acknowledge usage of the Toolkit via

Citation Guideline

If you make use of any of the Toolkit in your research please acknowledge using:

> This work makes use of the Black Hole Perturbation Toolkit.

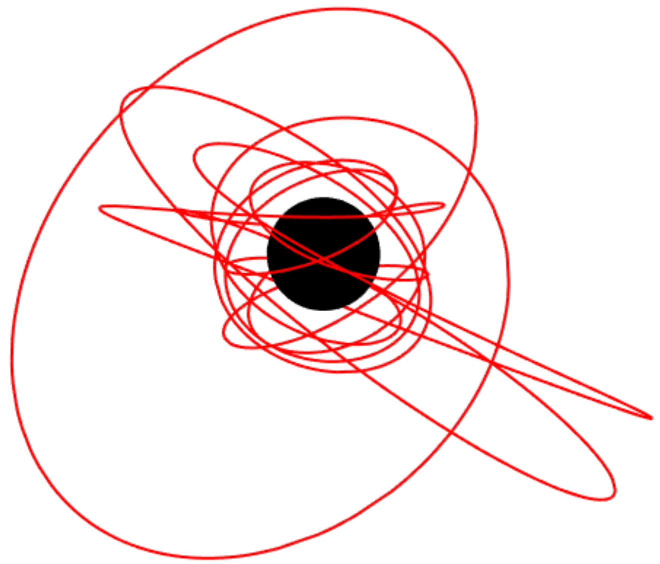
To cite the Toolkit please use this BibTeX entry (or similar). Some modules also request additional citations. Please check the documentation for individual modules.

Knowing who is using the Toolkit helps us prioritise work and helps us secure funding for workshops etc.

Black Hole Perturbation Toolkit: the future

Near term:

- Writing an introductory paper
- Effective communication (google groups, email, etc)
- Run Toolkit workshops
- Standardising on good data formats
- Formalize how to make Toolkit contributions



Longer term:

- Grow the community
- Encourage other researchers to tackle second order Self-force
- Compute fluxes for leading-order inspirals
- Make accurate waveforms for LISA

1. Add the Black Hole Perturbation Toolkit server to Mathematica's list of paclet servers:

```
If[$VersionNumber >= 12.1,
  PacletSiteRegister["https://pacletserver.bhptoolkit.org",
    "Black Hole Perturbation Toolkit Paclet Server”],
  PacletSiteAdd["http://pacletserver.bhptoolkit.org",
    "Black Hole Perturbation Toolkit Paclet Server"]
]
```

2. Get an updated list of packages available on the server:

```
If[$VersionNumber >= 12.1,
  PacletSiteUpdate[“https://pacletserver.bhptoolkit.org"],
  PacletSiteUpdate["http://pacletserver.bhptoolkit.org"]
]
```

3. Install the paclets:

```
PacletInstall[“GeneralRelativityTensors"];
PacletInstall["KerrGeodesics"];
PacletInstall["SpinWeightedSpheroidalHarmonics"];
PacletInstall[“ReggeWheeler”];
PacletInstall["Teukolsky"];
PacletInstall["PostNewtonianSelfForce"];
```

If a new version of a package is released and you would like to update, just run steps 2 and 3 above again and the latest available version will be installed.

https://bhptoolkit.org/mathematica-install