HUN REN | Wigner

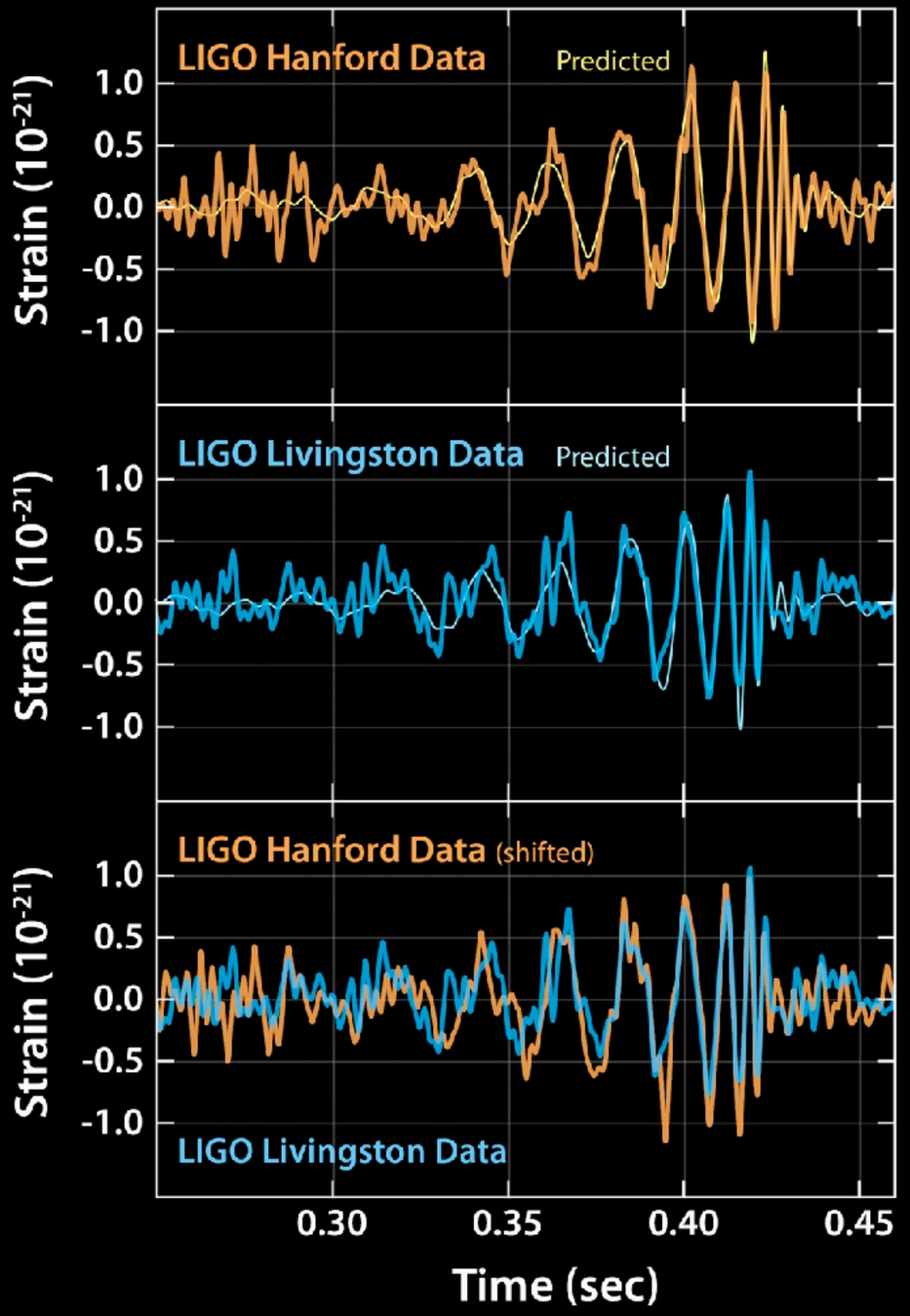
Numerical Analysis of Mismatch Between Eccentric Gravitational Wave Modeling Numerical Codes

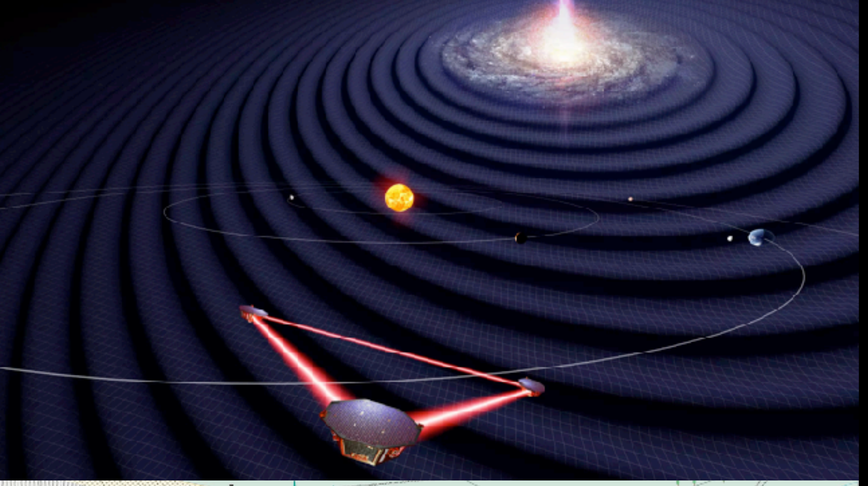
on a HTCondor cluster

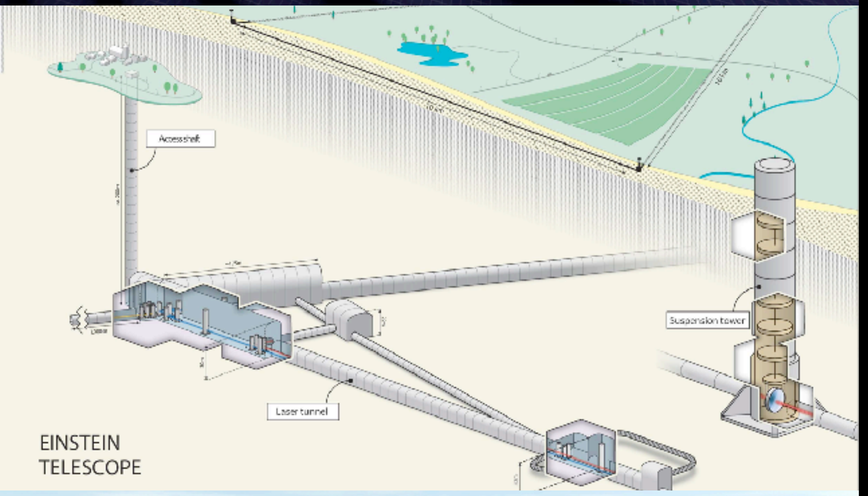
Balázs Kacskovics

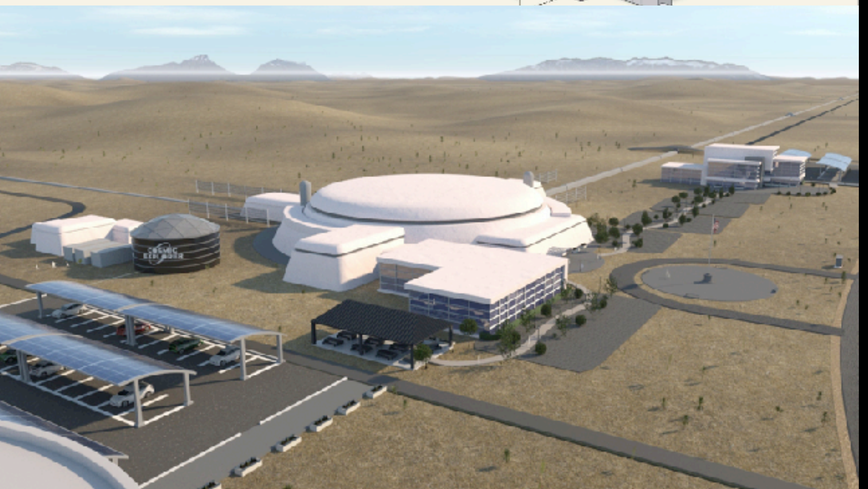
Algorithms Used in GW Search

- In the LAL library (used by the mainstream): Numerical, EOB, Taylor waveforms
- Excellently modeling the currently detectable waveforms
- Problems:
  - ➡ Only short waveforms
  - ➡ Only specific waveforms
  - ➡ No eccentric waveforms
  - ➡ Mostly spin-aligned









New GW Detectors on the Horizon

eLISA, Einstein Telescope, Cosmic Explorer

- Targeting new sources like NS-NS binaries, merging galactic nuclei, supernovae, stochastic background
- Significantly longer observational times $\Longrightarrow$ longer waveforms (up to 6 months — eLISA)
- Research of the inspiral phase
- Eccentricity and spin effects will be important in the orbital evolution of compact binaries
- eLISA got the green light this year

Gravitational Waves

Linearized theory

- Starting from the Einstein equation

$$R_{ab} - \frac{1}{2} g_{ab} R = \frac{8\pi G}{c^4} T^{ab}$$

- Take a small perturbation of the Einstein eq. around a flat spacetime (gauge symmetry of GR)

$$g_{ab} = \eta_{ab} + h_{ab}, \quad h_{ab} << 1$$

- The Riemann-tensor expressed in h_{ab} linear order

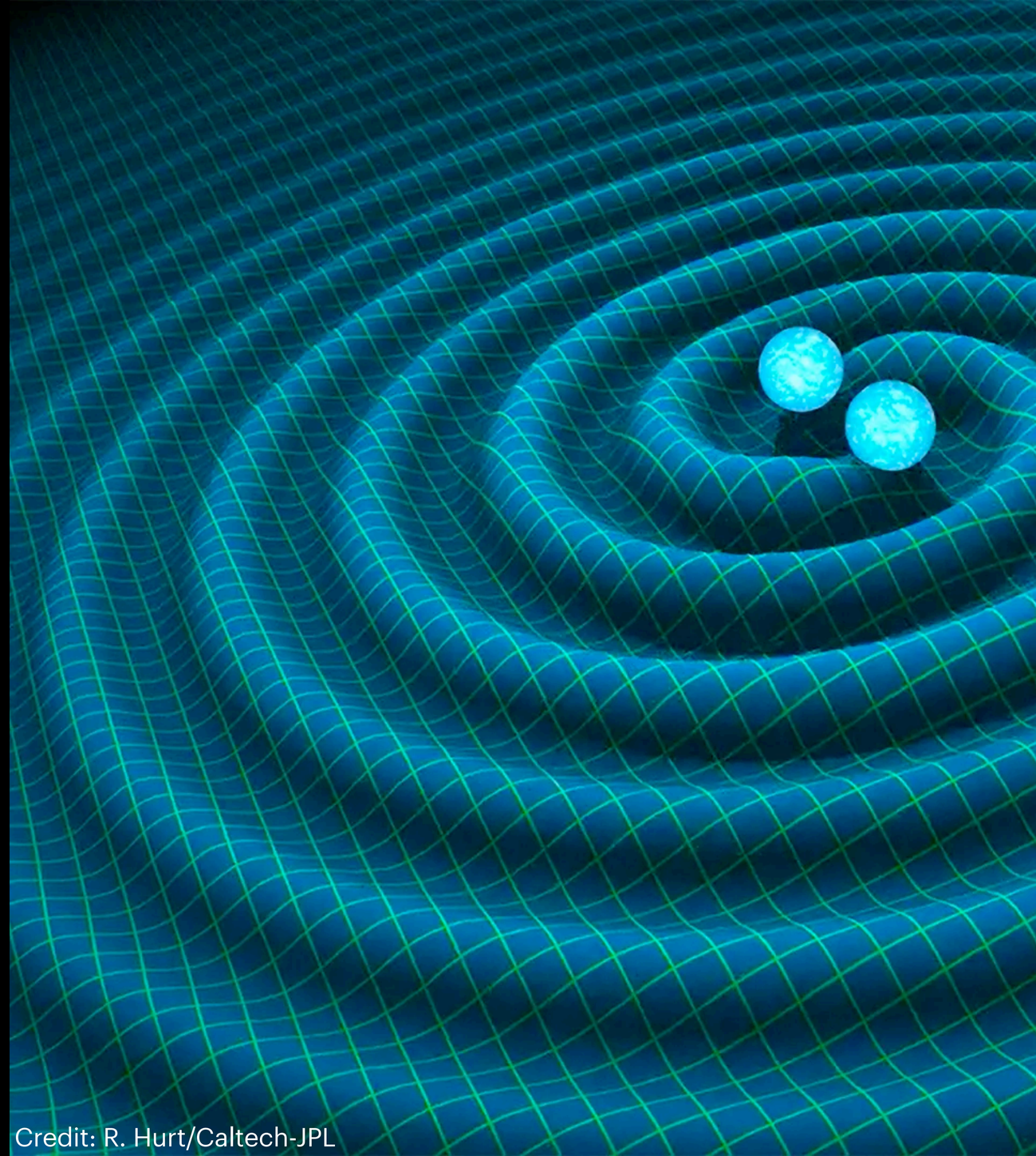
$$R_{abcd} = \frac{1}{2}(\partial_b \partial_c h_{ad} + \partial_a \partial_d h_{bc} - \partial_a \partial_c h_{bd} - \partial_b \partial_d h_{ac})$$

- The linearized Einstein equation

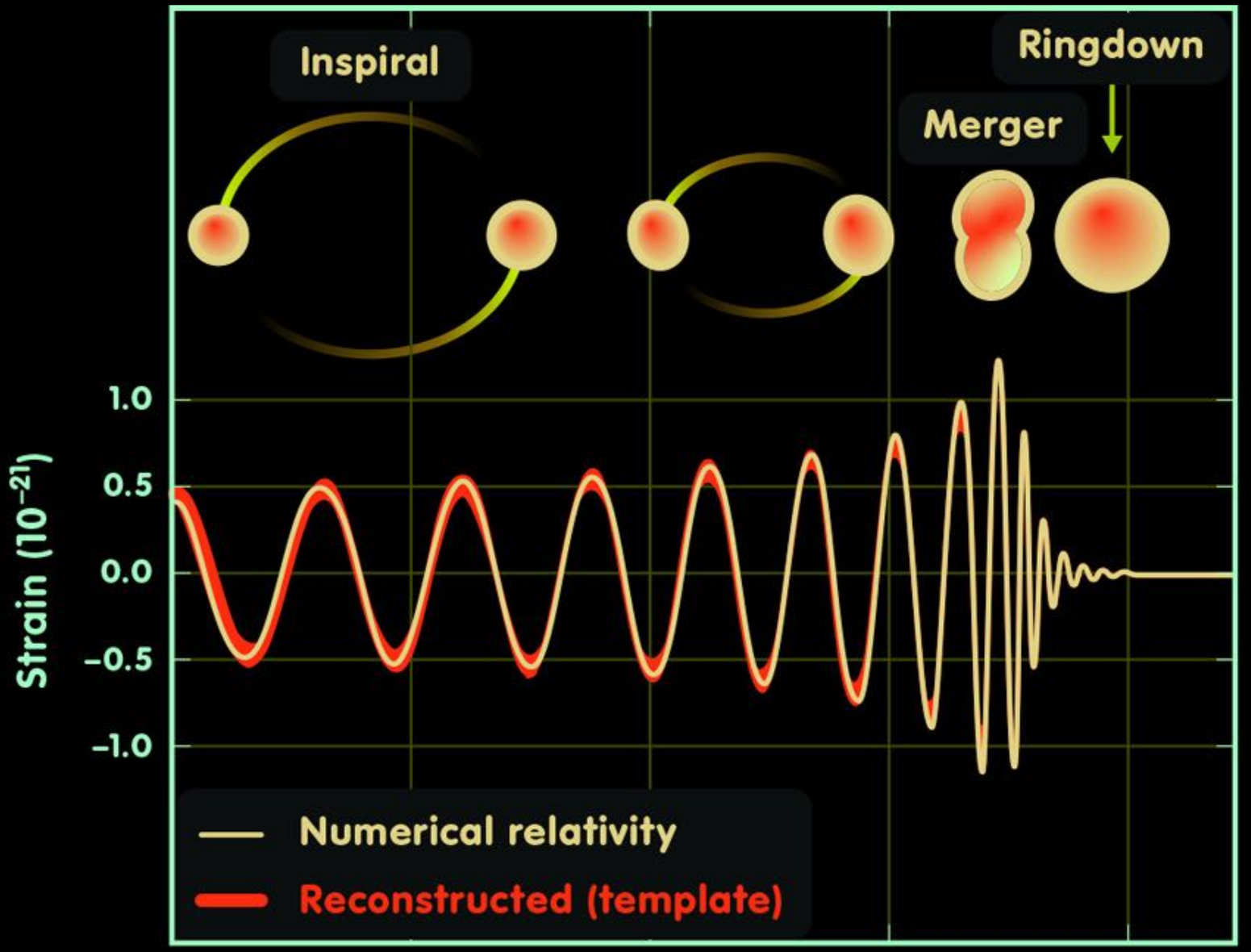
$$\Box \bar{h}_{ab} + \eta_{ab} \partial^c \partial^d \bar{h}_{cd} - \partial^c \partial_b \bar{h}_{ac} - \partial^c \partial_a \bar{h}_{bc} = -\frac{16\pi G}{c^4} T_{ab}$$

- Using the gauge freedom of GR and choosing the De Donder gauge, $\partial^b \bar{h}_{ab} = 0$

$$\Box \bar{h}_{ab} = -\frac{16\pi G}{c^4} T_{ab}$$



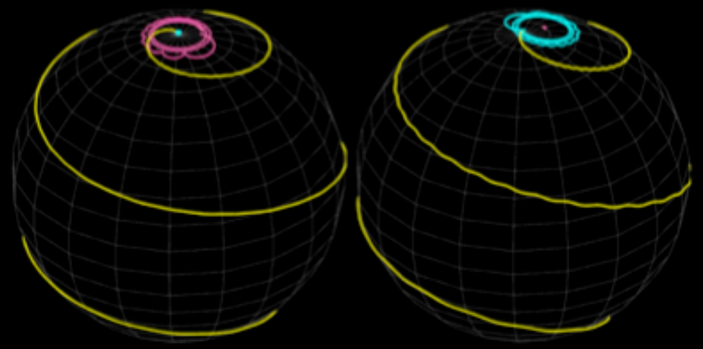
Credit: R. Hurt/Caltech-JPL



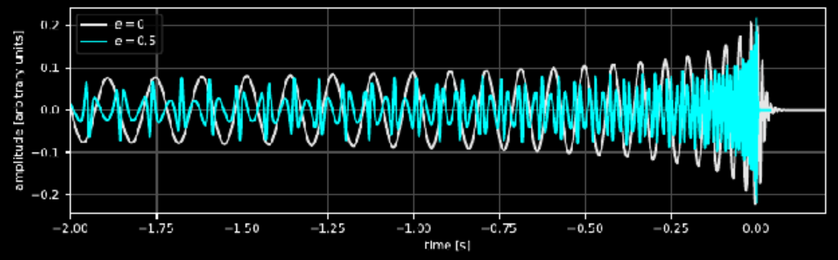
LIGO / Redesign: Daniela Leitner

Motivation

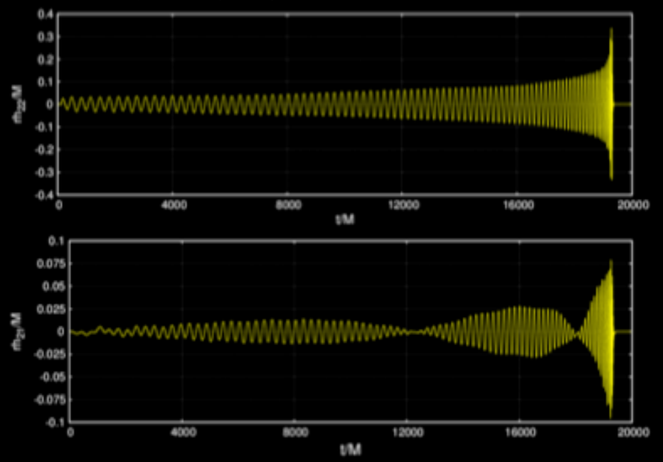
PRL 114, 141101 (2015)



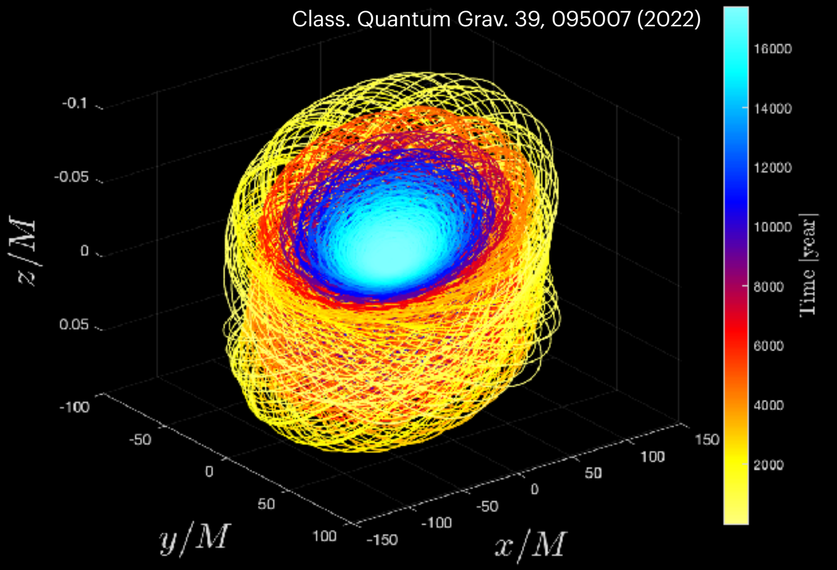
ApJ 883, 149 (2019)



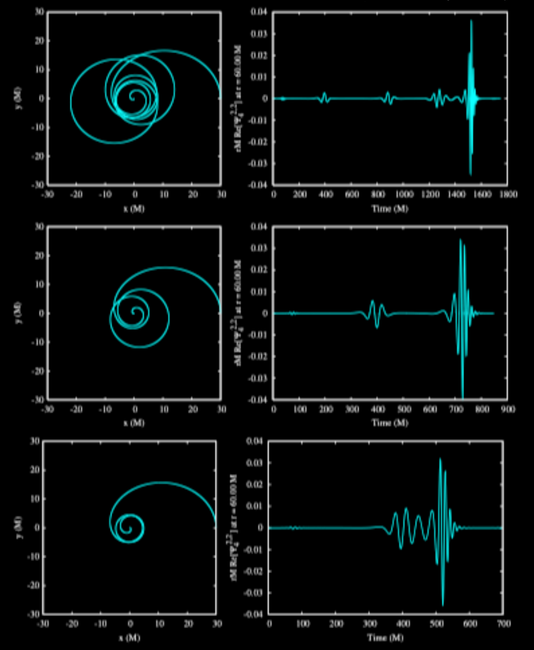
PRL 114, 141101 (2015)



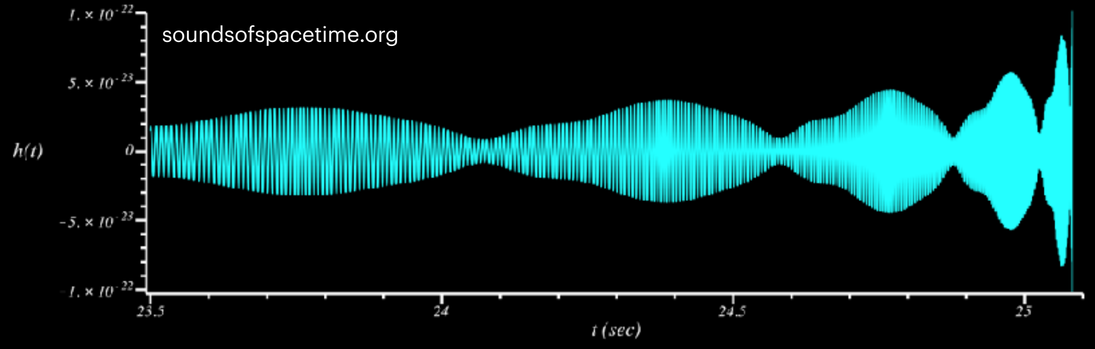
Class. Quantum Grav. 39, 095007 (2022)



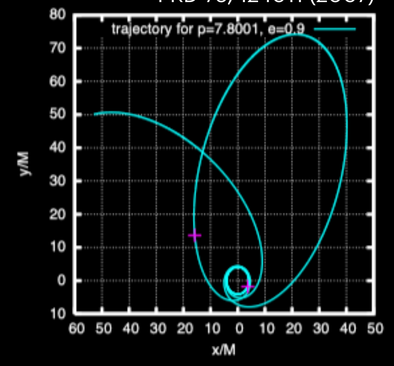
PRL 103, 131101 (2009)



soundsofspacetime.org

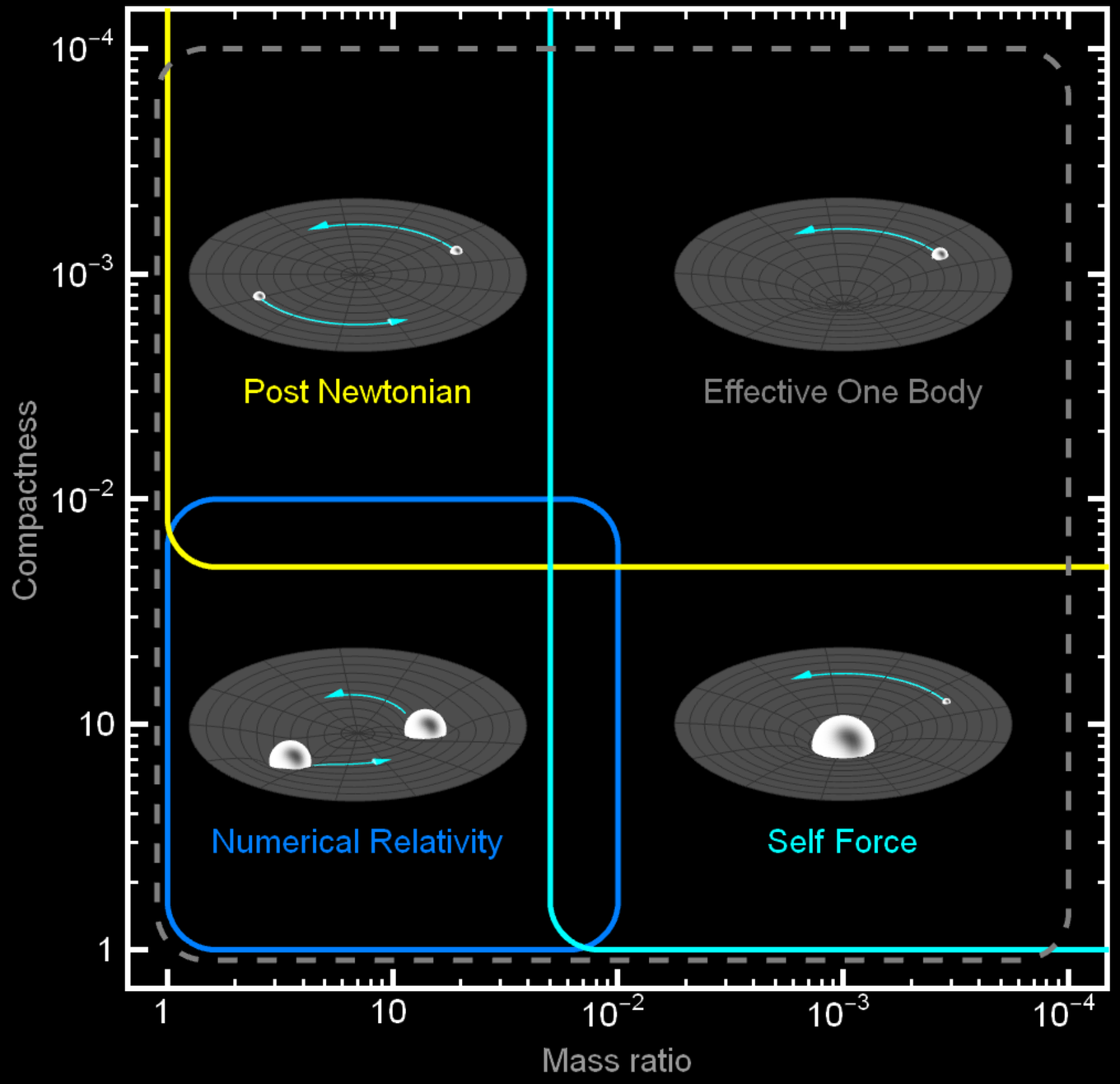


PRD 75, 124011 (2007)



Two-body problem of General Relativity

Maarten van de Meent - Own work



Post-Newtonian Expansion

- Built upon two assumptions:
  1. gravity inside the source is weak like in the post-Minkowsikian expansion
  2. the motion of the components of the source is slow
- The equation of motion

$$\mathbf{a} = \mathbf{a}_{\mathrm{N}} + \mathbf{a}_{\mathrm{PN}} + \mathbf{a}_{\mathrm{2PN}} + \mathbf{a}_{\mathrm{3PN}} + \mathbf{a}_{\mathrm{4PN}} + \underset{\mathrm{SO}}{\mathbf{a}}_{\mathrm{1.5PN}} + \underset{\mathrm{SS}}{\mathbf{a}}_{\mathrm{2PN}} + \underset{\mathrm{BT}}{\mathbf{a}}^{\mathrm{RR}}_{\mathrm{2.5PN}} + \underset{\mathrm{SO}}{\mathbf{a}}_{\mathrm{2.5PN}}$$

$$+ \underset{\mathrm{SO}}{\mathbf{a}}_{\mathrm{3.5PN}} + \underset{\mathrm{BT}}{\mathbf{a}}^{\mathrm{RR}}_{\mathrm{3.5PN}} + \underset{\mathrm{SS}}{\mathbf{a}}^{\mathrm{RR}}_{\mathrm{3.5PN}} + \underset{\mathrm{SO}}{\mathbf{a}}^{\mathrm{RR}}_{\mathrm{3.5PN}}$$

- The radiation field equation

$$h_{ij} = \frac{2G\mu}{c^4 D}[Q_{ij} + P^{0.5}Q_{ij} + PQ_{ij} + P^{1.5}Q_{ij} + P^2 Q_{ij} + PQ_{ij}^{SO}$$

$$+P^{1.5}Q_{ij}^{SO} + P^2 Q_{ij}^{SO} + PQ_{ij}^{SS} + P^{1.5}Q_{ij}^{tail}]$$

Effective One-Body Approach

- reduce the conservative dynamics of the general relativistic two-body problem
- Mathisson–Papapetrou–Dixon equation is taken on a deformed Kerr black hole
- Hamiltonian of the Mathisson–Papapetrou–Dixon equations:

$$H_{\mathrm{eff}} = M\eta\left(\beta^i p_i + \alpha\sqrt{1+\gamma^{ij}p_i p_j + Q_4(p)} + H_{\mathrm{S}}\right) + H_{SC}$$

$$H = M\sqrt{1+2\eta\left(\frac{H_{\mathrm{eff}}}{M\eta}-1\right)}$$

- In the EOBNR framework, the quasicircular part of the radiation field is divided into two:
  - the inspiral-plunge
  - post-merger phase

$$h_{lm}^{(C)} = h_{lm}^{(N,\epsilon)}\hat{S}_{\mathrm{eff}}^{(\epsilon)}T_{lm}e^{i\delta_{lm}}(\rho_{lm})^l N_{lm}$$

$$h_{lm}^{(N,\epsilon)} = \frac{M\eta}{D}n_{lm}^{(\epsilon)}c_{l+\epsilon}V_{\Phi}^{l}Y^{l-\epsilon,-m}\left(\frac{\pi}{2},\Phi\right)$$

- For the eccentric part, in the radiation field terms up to the second post-Newtonian order are considered

Numerical Results

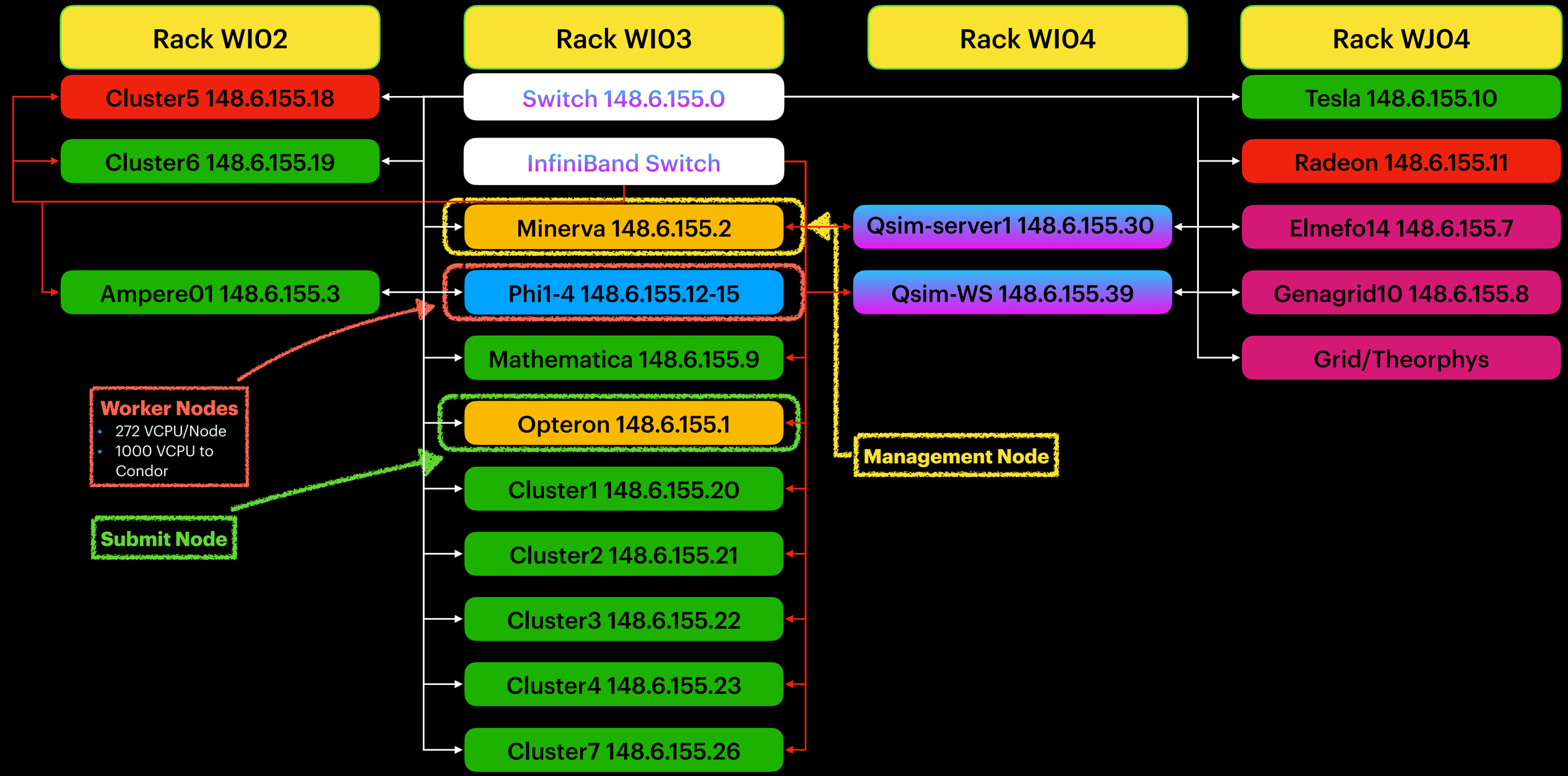
- 2 codes were used; one based on the PN, **CBwaves**; and one based on EOB, **SEOBNRE**
- both codes use a 4th-order Runge—Kutta integrator
- on an identical initial parameter space

Initial Parameters

m_1 [$M_\odot$]	10 ... 100
m_2 [$M_\odot$]	10 ... 100
R [M_{tot}]	30
R_{min} [M_{tot}]	6
e_0	0.003
dt [sec]	1/4096

- **SEOBNRE** uses the initial orbital frequency: $$f_{\rm init} = \frac{c^3}{\pi G(m_1+m_2)M_\odot\sqrt{\mathbf{r}_0^3}}$$

Technical Background


Rack WI02
Rack WI03
Rack WI04
Rack WJ04
Cluster5 148.6.155.18
Cluster6 148.6.155.19
Ampere01 148.6.155.3
Switch 148.6.155.0
InfiniBand Switch
Minerva 148.6.155.2
Phi1-4 148.6.155.12-15
Mathematica 148.6.155.9
Opteron 148.6.155.1
Cluster1 148.6.155.20
Cluster2 148.6.155.21
Cluster3 148.6.155.22
Cluster4 148.6.155.23
Cluster7 148.6.155.26
Qsim-server1 148.6.155.30
Qsim-WS 148.6.155.39
Management Node
Tesla 148.6.155.10
Radeon 148.6.155.11
Elmefo14 148.6.155.7
Genagrid10 148.6.155.8
Grid/Theorphys
Worker Nodes
272 VCPU/Node
1000 VCPU to Condor
Submit Node

Setting Up a Dynamic HTCondor Cluster

- 3 schemes for slot distribution
  1. Static
  2. Partionable
  3. Dynamic
- One partionable slot
- Dynamic slots are created from it
- Dynamic slots merge back to the partionable slot upon finishing
- The partionable slots split on: the CPU, Memory, Disk
- How to configure:

```
NUM_SLOTS = 1
NUM_SLOTS_TYPE_1 = 1
SLOT_TYPE_1 = cpus=100%
SLOT_TYPE_1_PARTITIONABLE = true
```

- Documentation to install HTCondor: https://htcondor.readthedocs.io/en/latest/admin-manual/index.html

Edit it to change the number of CPUs used

Name	OpSys	Arch	State	Activity	LoadAv	Mem	ActvtyTime
slot1@ph1.gpu.wigner.hun-ren.hu	LINUX	X86_64	Unclaimed	Idle	0.000	64244	54+17:27:35
slot1@phi2.gpu.wigner.hun-ren.hu	LINUX	X86_64	Unclaimed	Idle	0.000	64226	186+18:10:59
slot1@phi3.gpu.wigner.hun-ren.hu	LINUX	X86_64	Unclaimed	Idle	0.000	64226	186+18:08:56
slot1@phi4.gpu.wigner.hun-ren.hu	LINUX	X86_64	Unclaimed	Idle	0.000	64226	186+18:09:48

	Total	Owner	Claimed	Unclaimed	Matched	Preempting	Drain	Backfill	BkIdle
X86_64/LINUX	4	0	0	4	0	0	0	0	0
Total	4	0	0	4	0	0	0	0	0

Name	OpSys	Arch	State	Activity	LoadAv	Mem	ActvtyTime
slot1@ph1.wigner.hun-ren.hu	LINUX	X86_64	Unclaimed	Idle	0.000	63220	54+18:10:11
slot1_1@ph1.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:01
slot1_2@ph1.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:01
slot1_3@ph1.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:00
slot1_4@ph1.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:00
slot1_5@ph1.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:01
slot1_6@ph1.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:00
slot1_7@ph1.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:01
slot1_8@ph1.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:00
slot1@phi2.gpu.wigner.hun-ren.hu	LINUX	X86_64	Unclaimed	Idle	0.000	63202	186+18:53:00
slot1_1@phi2.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:01
slot1_2@phi2.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:01
slot1_3@phi2.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:00
slot1_4@phi2.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:00
slot1_5@phi2.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:00
slot1_6@phi2.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:00
slot1_7@phi2.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:01
slot1_8@phi2.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:00
slot1@phi3.gpu.wigner.hun-ren.hu	LINUX	X86_64	Unclaimed	Idle	0.000	63202	186+18:53:13
slot1_1@phi3.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:01
slot1_2@phi3.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:01
slot1_3@phi3.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:00
slot1_4@phi3.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:00
slot1_5@phi3.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:00
slot1_6@phi3.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:00
slot1_7@phi3.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:01
slot1_8@phi3.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:00
slot1@phi4.gpu.wigner.hun-ren.hu	LINUX	X86_64	Unclaimed	Idle	0.000	63202	186+18:53:08
slot1_1@phi4.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:01
slot1_2@phi4.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:01
slot1_3@phi4.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:00
slot1_4@phi4.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:01
slot1_5@phi4.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:00
slot1_6@phi4.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:00
slot1_7@phi4.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:01
slot1_8@phi4.gpu.wigner.hun-ren.hu	LINUX	X86_64	Claimed	Busy	0.000	128	0+00:00:00

	Total	Owner	Claimed	Unclaimed	Matched	Preempting	Drain	Backfill	BkIdle
X86_64/LINUX	36	0	32	4	0	0	0	0	0
Total	36	0	32	4	0	0	0	0	0

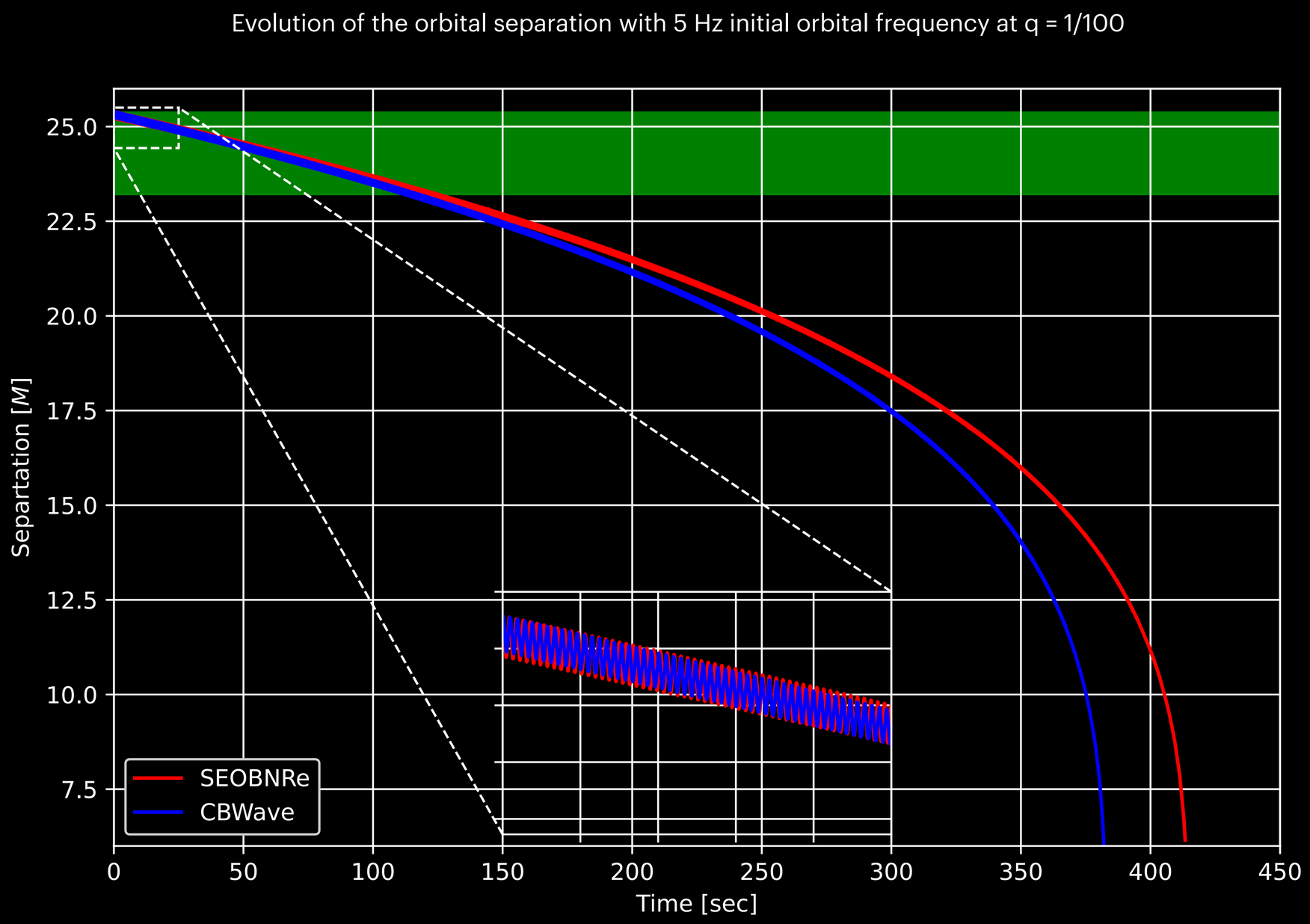
Submitting Jobs With Python

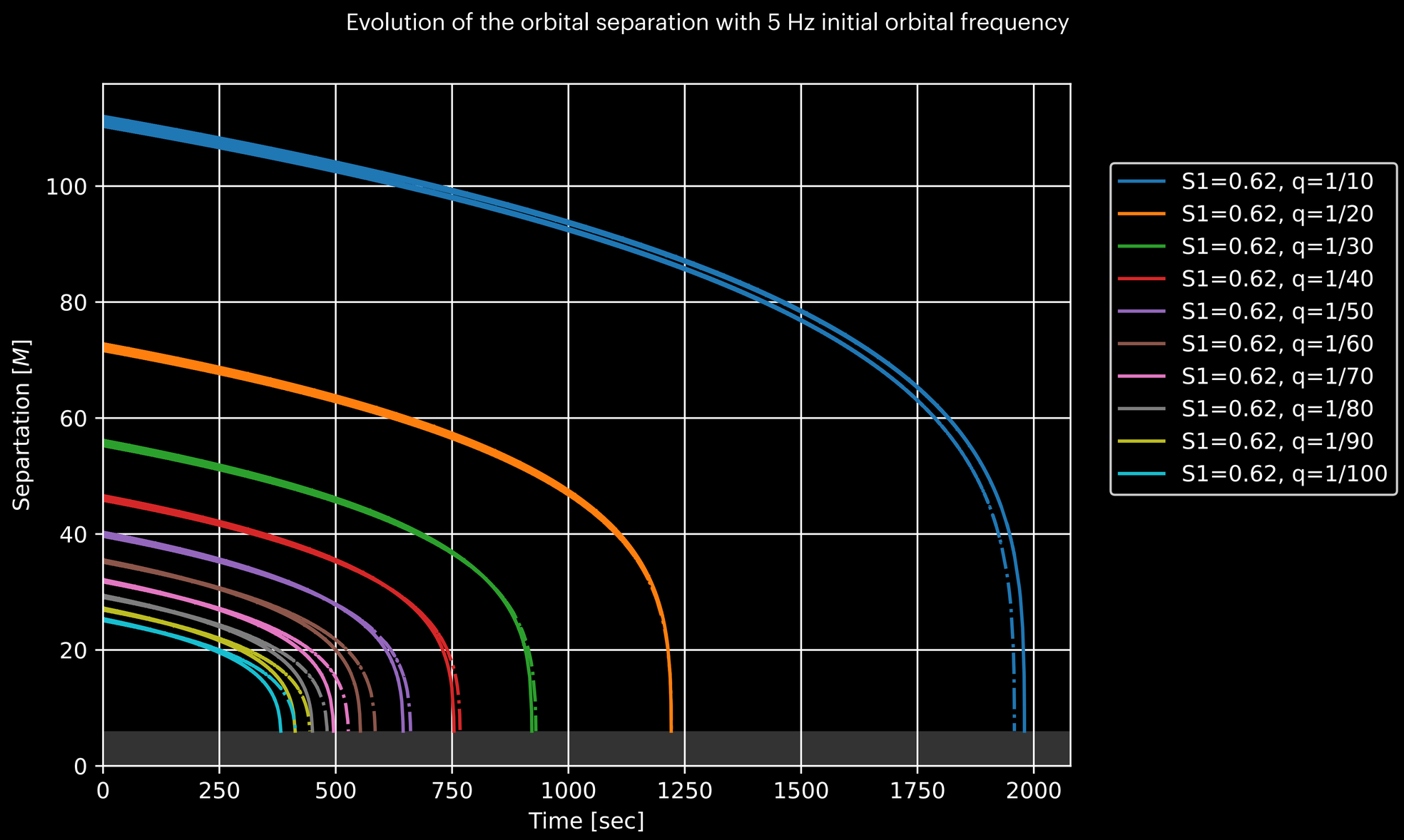
```
for j in range(m):
    for j in range(m):
        schedd = htc.Schedd()

        prefix = str(int(M1*np.power(10,acc))) + "_" + str(int(e0[j]*np.power(10,acc)))
        condor_cbw_out = "out/cbwaves_" + prefix + ".out"
        condor_seo_out = "out/seobnre_" + prefix + ".out"
        condor_cbw_err = "error/cbwaves_" + prefix + ".err"
        condor_seo_err = "error/seobnre_" + prefix + ".err"

        #** HTCondor submit for SEOBNRE **
        cmd_seo = path + "/seobnre.py --m1 " + str(M1) + " --m2 " + str(M2) + " --e0 " + str(e0[j]) + " --sampleRate " + str(int(dt)) + " --spin1x “ + str(S1x) + " --spin1y " + str(S1y) + " --spin1z " + str(S1z)
        + " --spin2x " + str(S2x) + " --spin2y " + str(S2y) + " --spin2z " + str(S2z) + " --fmin " + str(Fmin) + " --printstep " + str(printstep) + " --fnameprefix " + prefix + " --dir " + dirpath + " --seobin " +
        str(seobin)


        seobnre_job = htc.Submit({
            "initialdir": inidir,              # sets the location of the initial working directory
            "executable": "/usr/bin/python3.10",    # the program to run on the execute node
            "arguments": cmd_seo,              # the arguments of the program, that will run on the execute node
            "getenv": "True",
            "output": condor_seo_out,          # anything the job prints to standard output will end up in this file
            "error": condor_seo_err,           # anything the job prints to standard error will end up in this file
            "log": "seobnre.log",              # this file will contain a record of what happened to the job
            "request_cpus": "1",               # how many CPU cores we want
        })
        submit_seo = schedd.submit(seobnre_job)     # submit the job
```


Evolution of the orbital separation with 5 Hz initial orbital frequency at q = 1/100
Separtation [M]
Time [sec]
25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
0
50
100
150
200
250
300
350
400
450
SEOBNRe
CBWave


Evolution of the orbital separation with 5 Hz initial orbital frequency
Separtation [M]
Time [sec]
0
20
40
60
80
100
0
250
500
750
1000
1250
1500
1750
2000
S1=0.62, q=1/10
S1=0.62, q=1/20
S1=0.62, q=1/30
S1=0.62, q=1/40
S1=0.62, q=1/50
S1=0.62, q=1/60
S1=0.62, q=1/70
S1=0.62, q=1/80
S1=0.62, q=1/90
S1=0.62, q=1/100

Mismatch/Unfaithfulness

- To calculate the mismatch, one first has to calculate the Overlap:

$$\mathcal{O} = \frac{\left\langle h_1, h_2 \right\rangle}{\sqrt{\left\langle h_1, h_1 \right\rangle \left\langle h_2, h_2 \right\rangle}}$$

  where

$$\left\langle h_1, h_2 \right\rangle = 4\Re \int_{f_{\min}}^{f_{\max}} \frac{\tilde{h}_1 \tilde{h}_2^{\,*}}{S_n(f)} \mathrm{d}f$$

- The mismatch (or unfaithfulness) is the marginalized overlap over some quantities

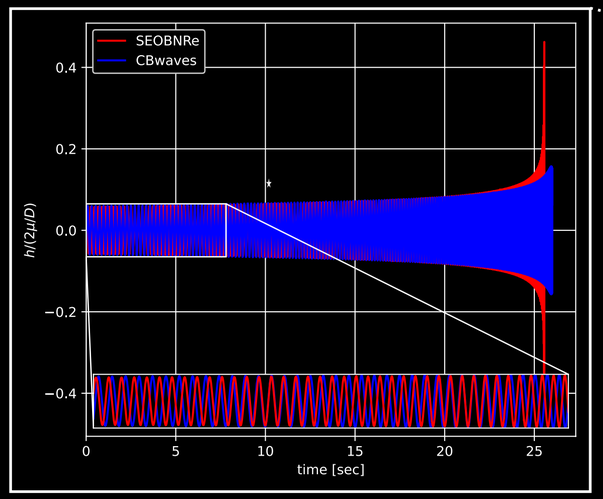
$$\mathcal{M} = \max_{t,\phi,\psi} \mathcal{O}(h_1, h_2)$$

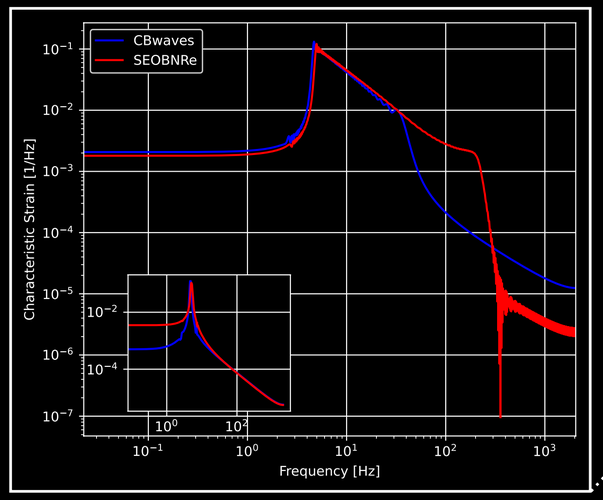
  where the $\max$ was taken over timeshifts, polarization angles, and phase

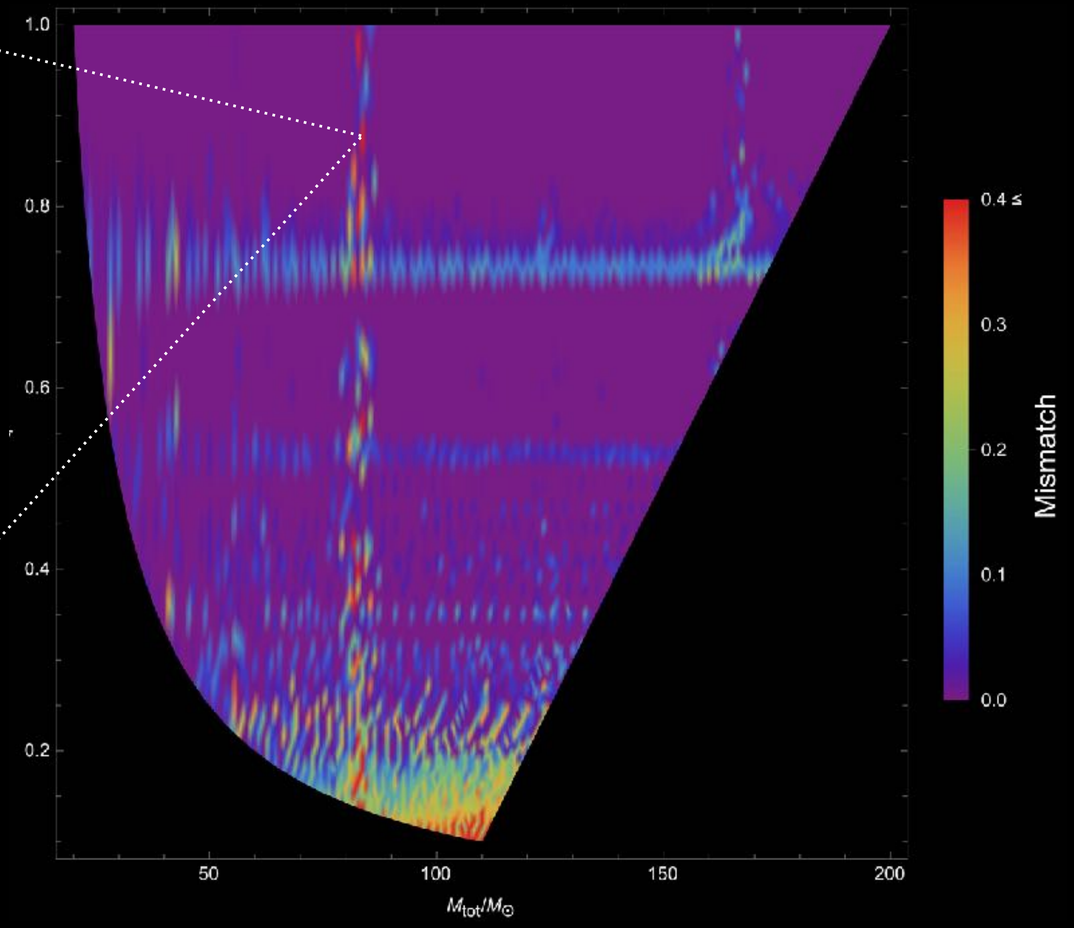
- The `kuibit` Python library was used.

Mismatch map for the not-spinning configurations

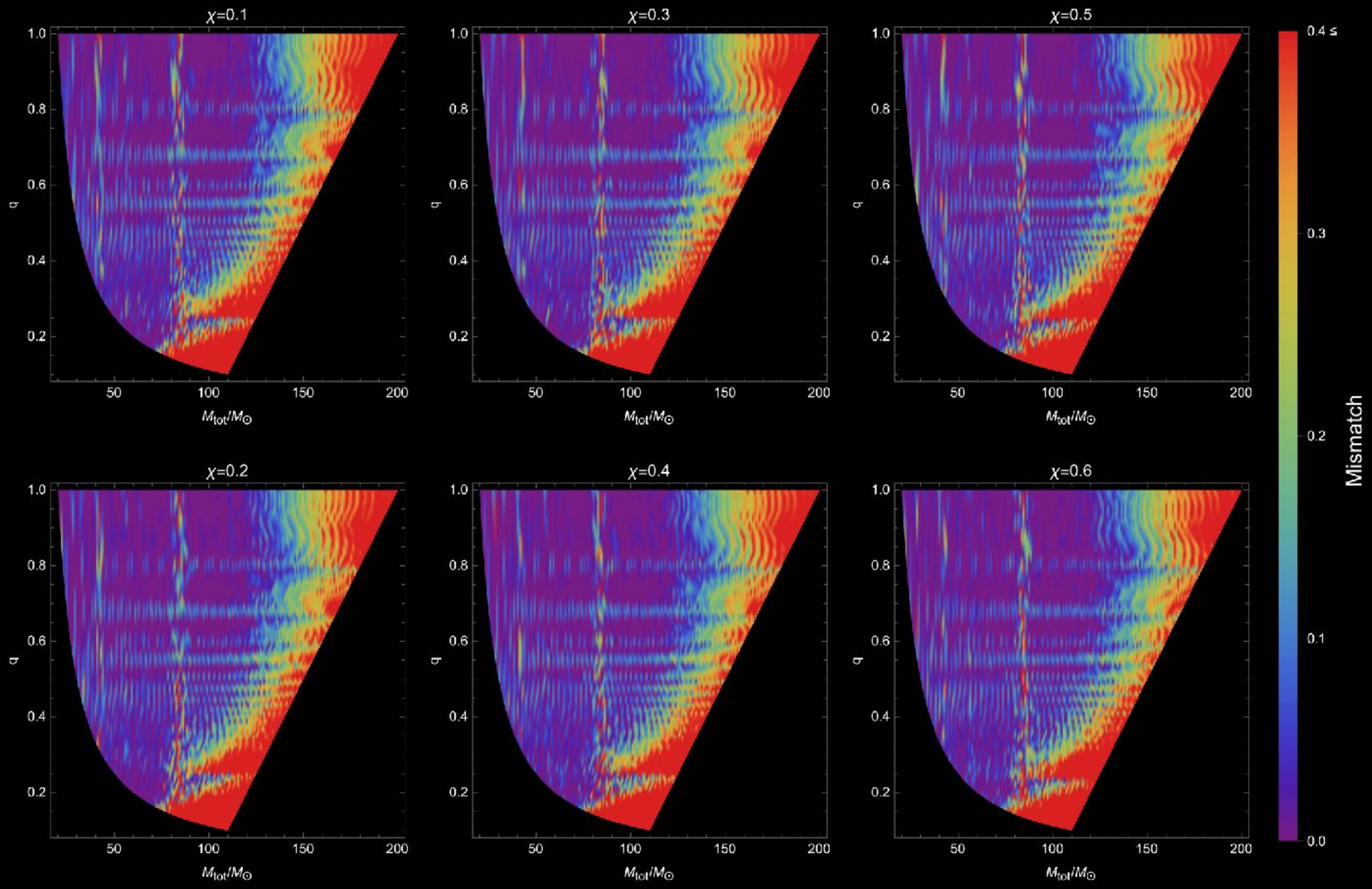
m_1 = 44.5455 $M_\odot$, m_2 = 39.0909 $M_\odot$ and q = 0.8775499209



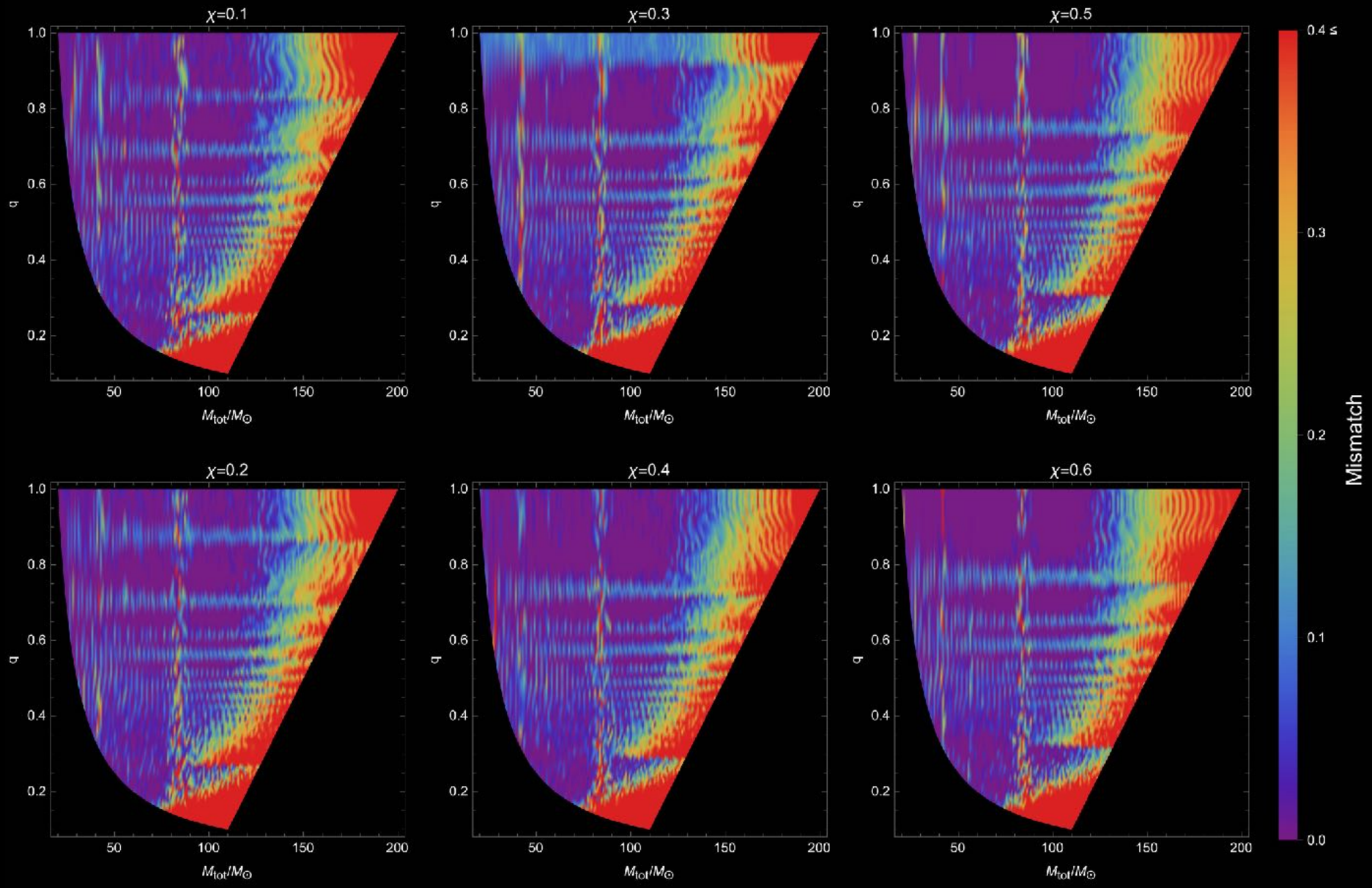




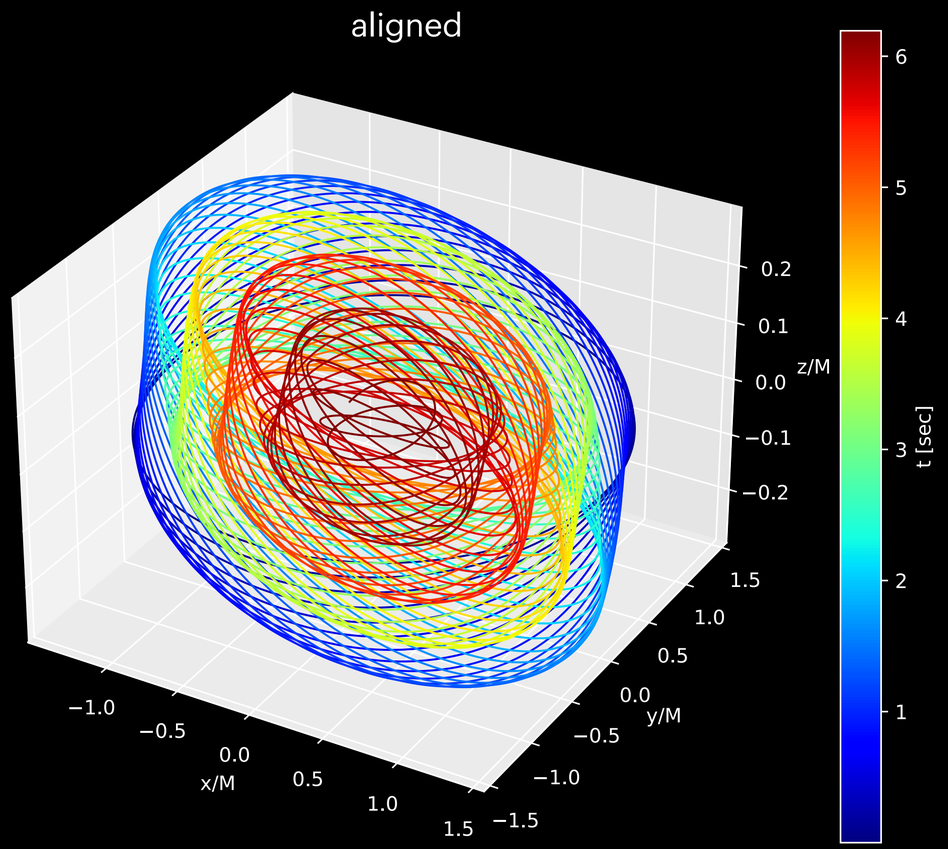
Mismatch map for the spin-aligned configurations

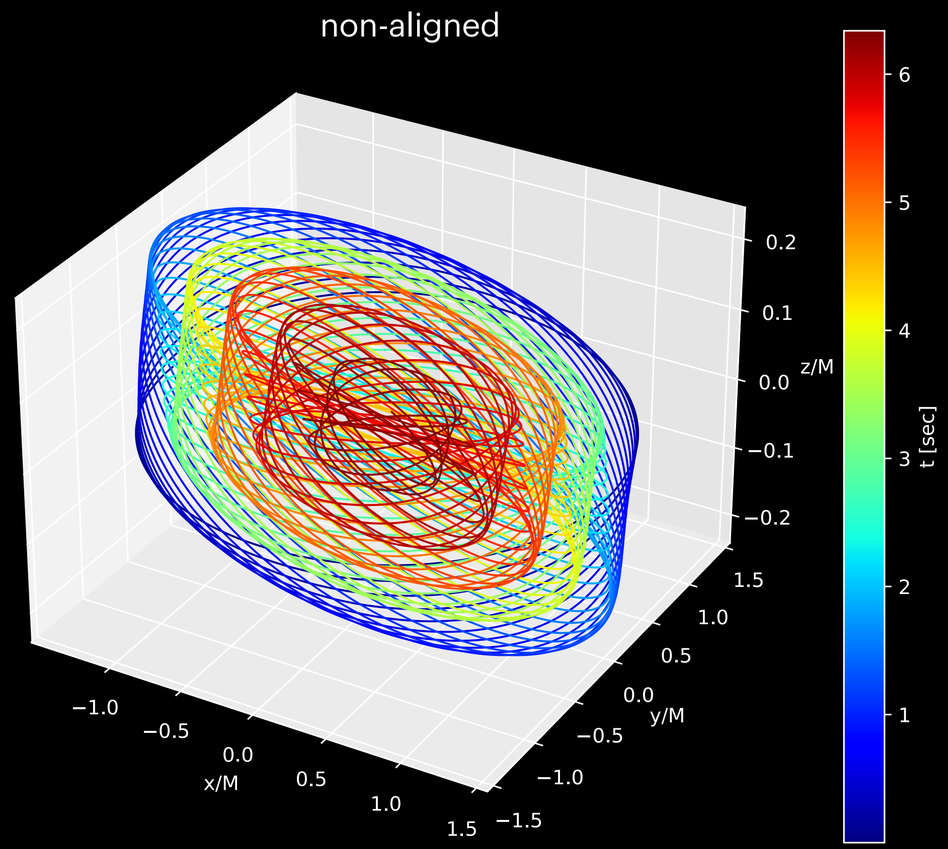


Mismatch map for the non-aligned spin configurations

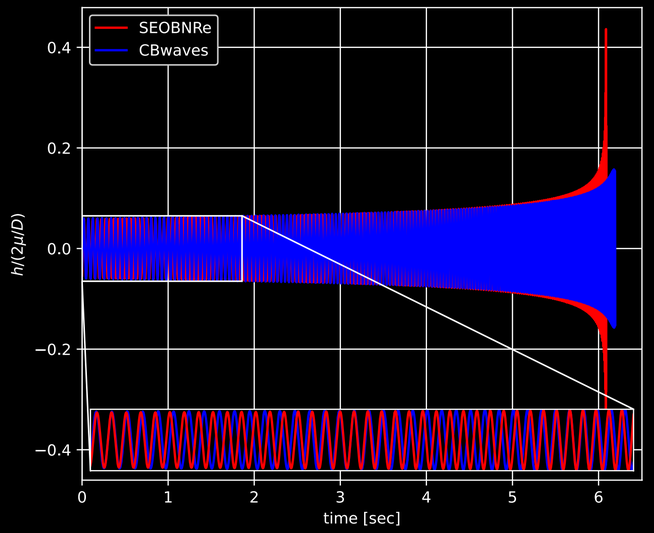


Orbital evolution of the binary at $\chi_1 = 0.6$, $m_1 = m_2 = 10\ \mathrm{M}_\odot$

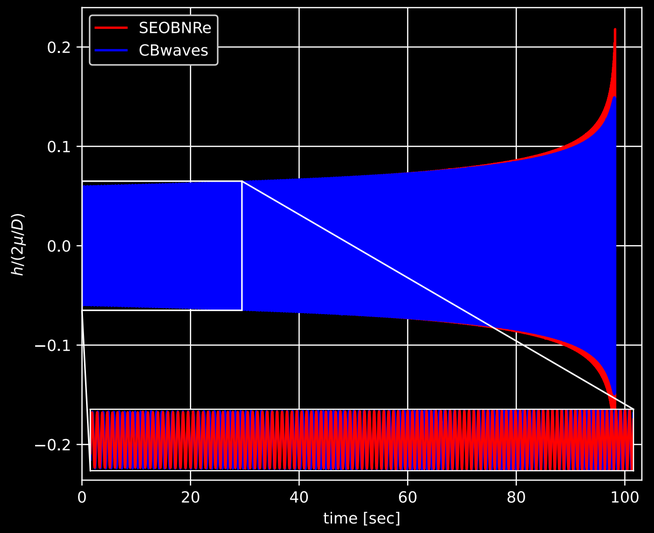




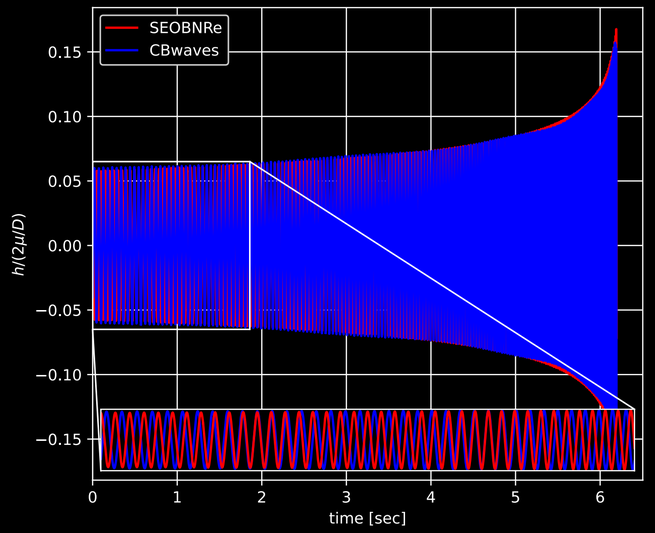
not-spinning, $m_2 = 10\ M_\odot$, $m_2 = 10\ M_\odot$



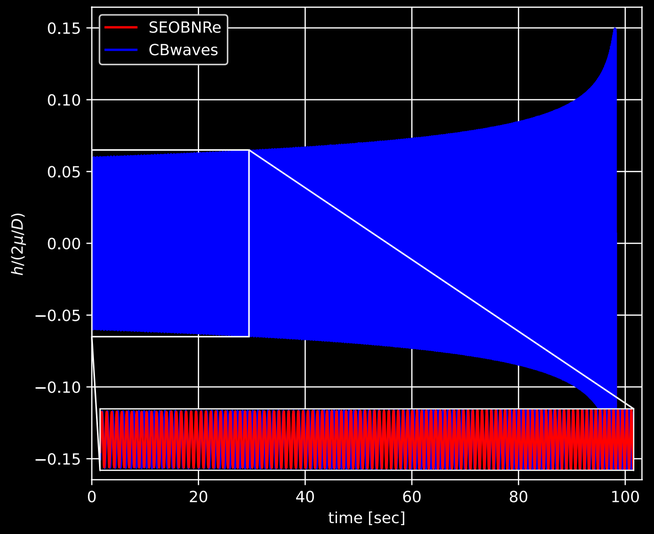
not-spinning, $m_2 = 100\ M_\odot$, $m_2 = 10\ M_\odot$



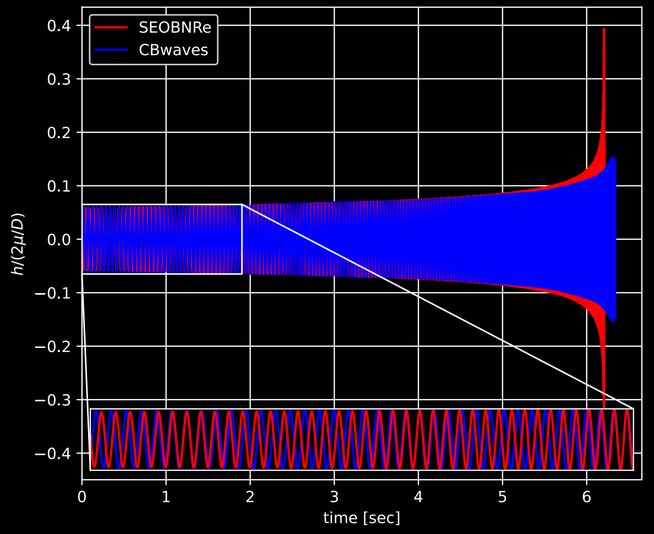
$\chi_1 = 0.6$, aligned, $m_2 = 10\ M_\odot$, $m_2 = 10\ M_\odot$



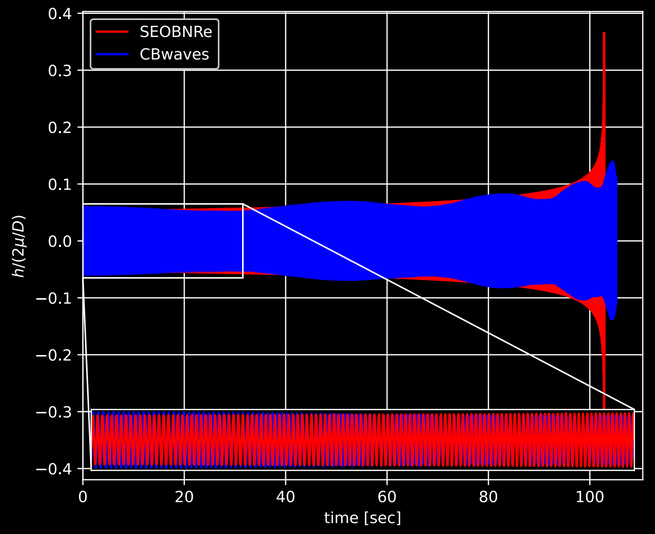
$\chi_1 = 0.6$, aligned, $m_2 = 100\ M_\odot$, $m_2 = 10\ M_\odot$



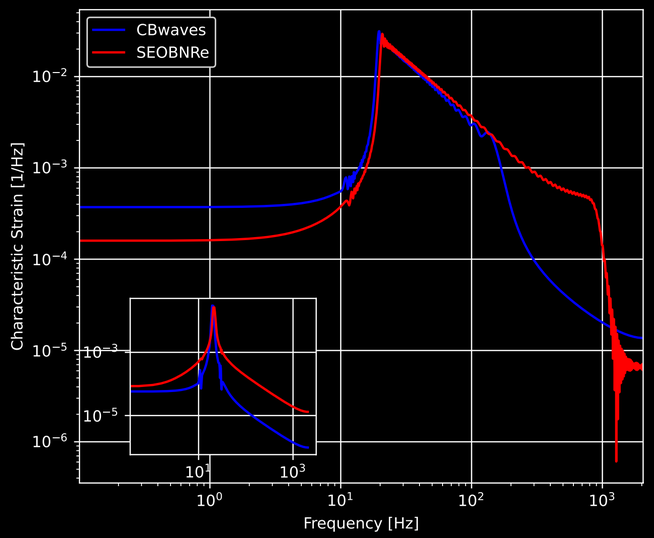
$\chi_1 = 0.6$, ~~aligned~~, $m_2 = 10\ M_\odot$, $m_2 = 10\ M_\odot$



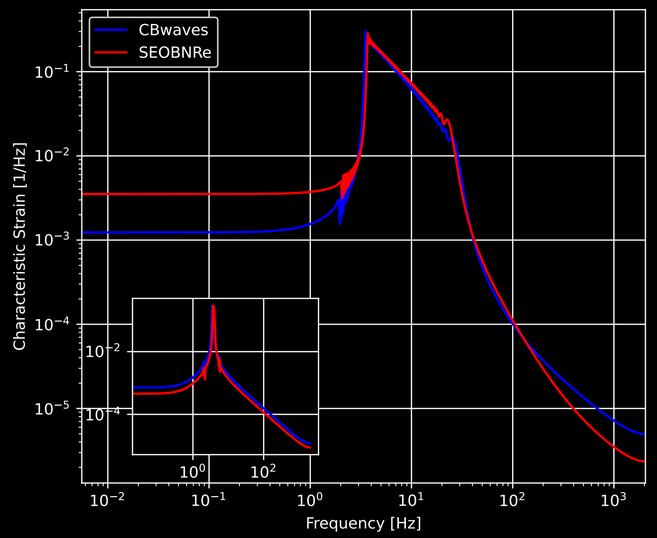
$\chi_1 = 0.6$, ~~aligned~~, $m_2 = 100\ M_\odot$, $m_2 = 10\ M_\odot$



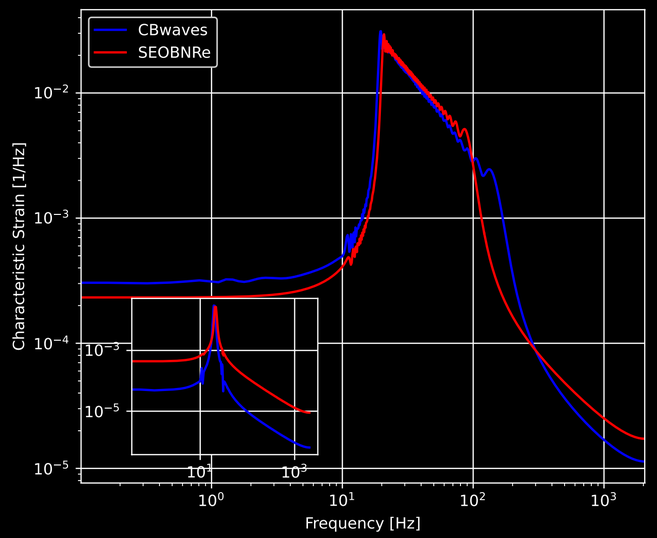
not-spinning, $m_2 = 10\ M_\odot$, $m_2 = 10\ M_\odot$



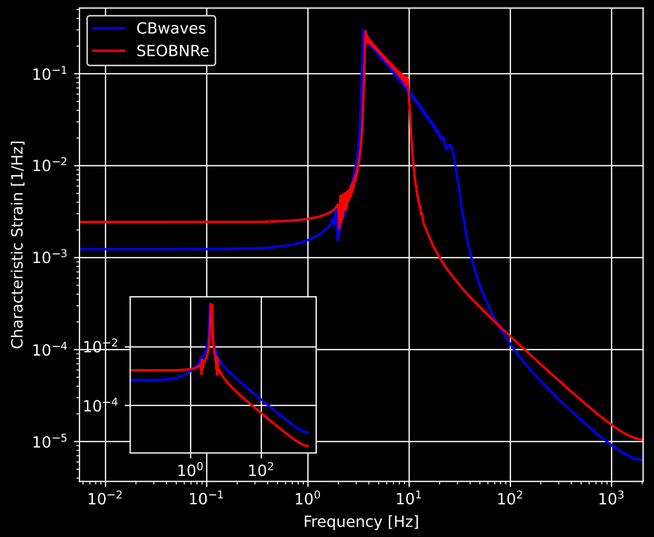
not-spinning, $m_2 = 100\ M_\odot$, $m_2 = 10\ M_\odot$



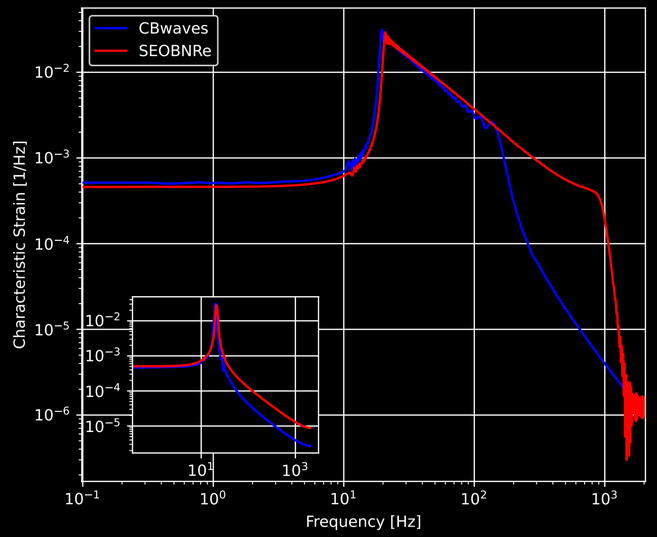
$\chi_1 = 0.6$, aligned, $m_2 = 10\ M_\odot$, $m_2 = 10\ M_\odot$



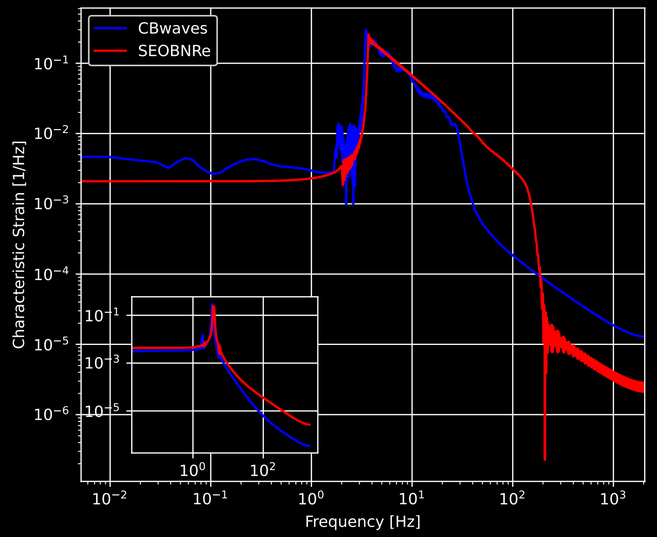
$\chi_1 = 0.6$, aligned, $m_2 = 100\ M_\odot$, $m_2 = 10\ M_\odot$



$\chi_1 = 0.6$, ~~aligned~~, $m_2 = 10\ M_\odot$, $m_2 = 10\ M_\odot$



$\chi_1 = 0.6$, ~~aligned~~, $m_2 = 100\ M_\odot$, $m_2 = 10\ M_\odot$



Discussion and Conclusion

- At $1:20$ mass-ratio, the separation computed by both codes is in close agreement
- For configurations with $q < 1/20$ the $6\ \mathrm{M}$ limit reached earlier by `SEBNRE`
- For configurations with $q > 1/20$ the $6\ \mathrm{M}$ limit reached earlier by `CBWaves`
- Made detailed contour maps for the mismatch (or unfaithfulness) of various spin configurations
- As the mass-ratio is closing $1:10$, the mismatch between the two models grows larger
- A similar behavior is exhibited toward larger total masses with spins, but irrespective of the spin alignment
- the spins did not retain the initial alignment set in CBwaves
- However, the effects of the spin are unnoticeable on the aligned waveforms

Acknowledgment

- Deeply grateful to **WSCLab.** for computational resources provided
- This work was made in collaboration with **Dániel Barta**
- Thanks for the insight and pieces of advice given by **László Á. Gergely**

Thanks for your attention!