This talk (Google Slides):
https://tinyurl.com/icermzach

BlackHoles@Home Status Report

Zach Etienne

Center for Gravitational Waves And Cosmology

NRPy+: Python-based C code generation framework for NR

Tensorial expressions in Einstein-like notation ⇒ Highly optimized C-code kernels (with FDs)

"Nerpy", the NRPy+ mascot. Photo CC2.0 Pacific Environment (modified).

https://nrpyplus.net

https://github.com/zachetienne/nrpytutorial

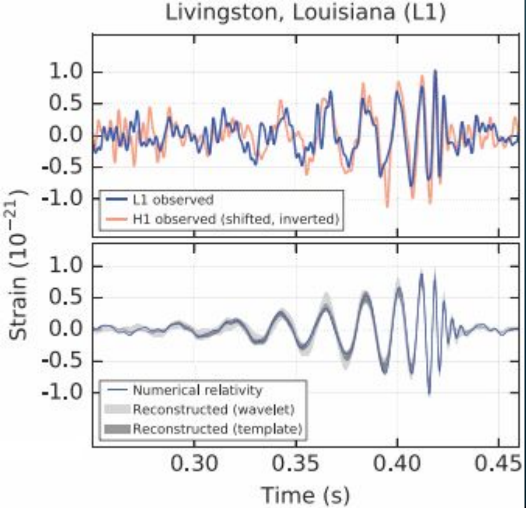
BlackHoles@Home

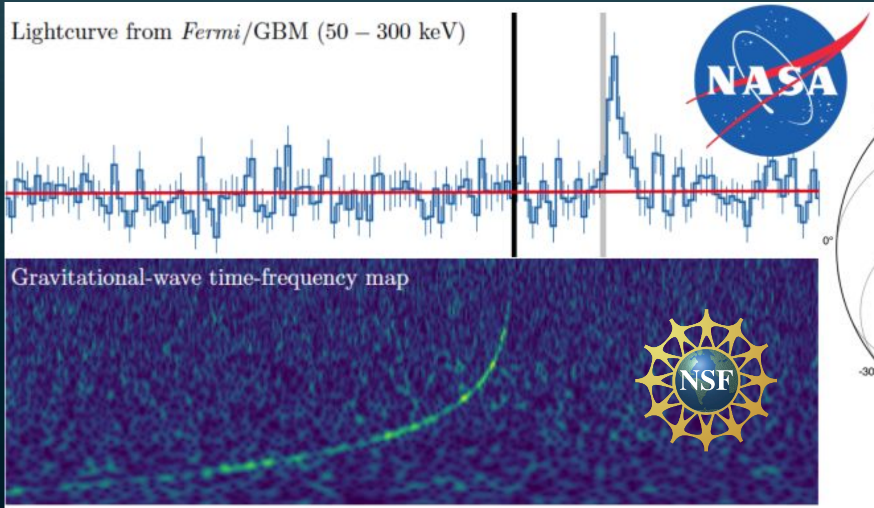
https://blackholesathome.net

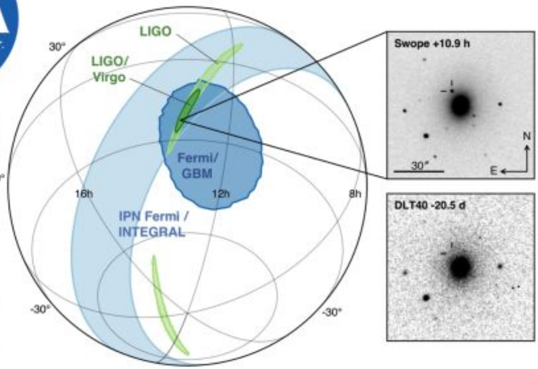
BlackHoles@Home aims to fit numerical-relativity-based binary black hole (BBH) calculations on a consumer-grade desktop computer, enabling gravitational waveform follow-ups and catalogs at unprecedentedly large scales using volunteer computers.

Importance of modeling gravitational wave and multimessenger sources

- Example: LIGO detects a gravitational wave from a black hole or neutron star binary

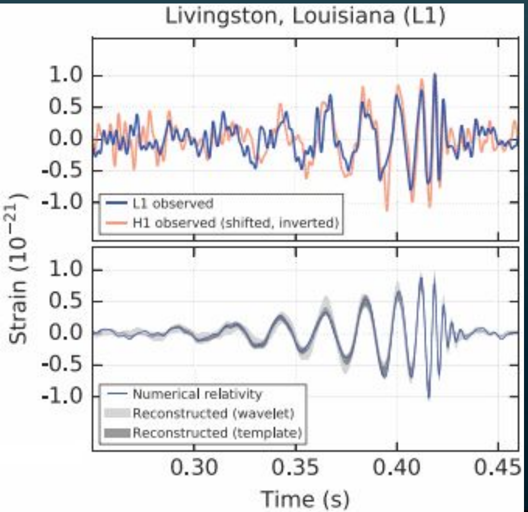


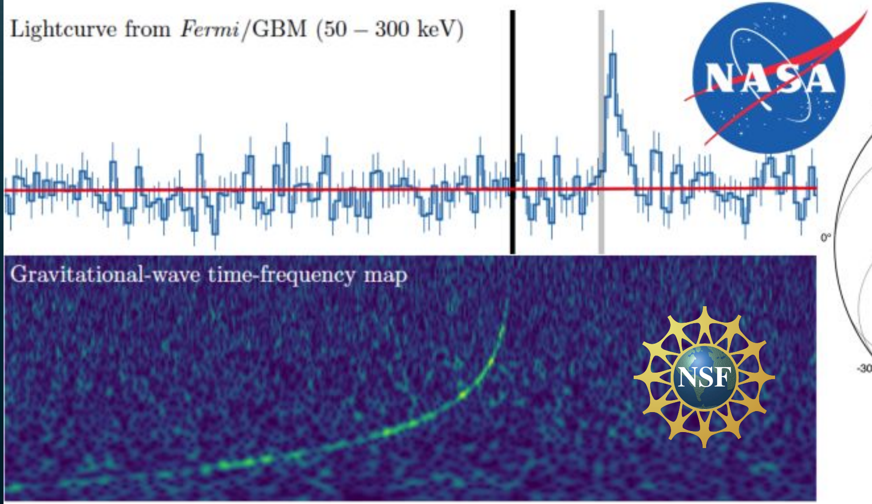


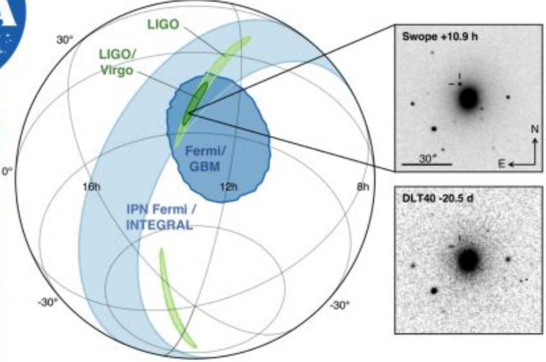


Importance of modeling gravitational wave and multimessenger sources

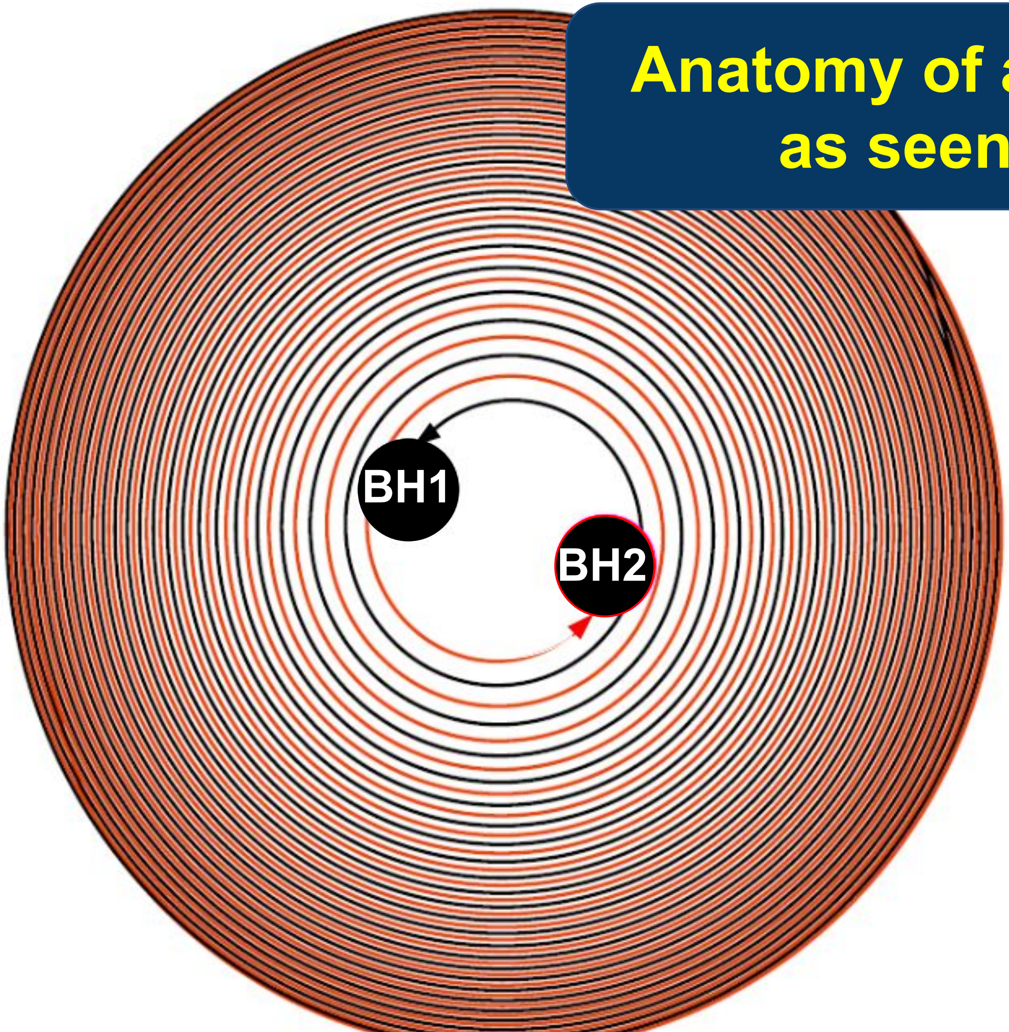
- $1B+ Question: What *exactly* caused this and *how*?
  - Answer can provide deep insights into extreme gravity & extreme matter, testing theories beyond current limits
  - To advance science, must compare observations with theoretical predictions
    - Theoretical predictions need to span observ. & theor. uncertainties







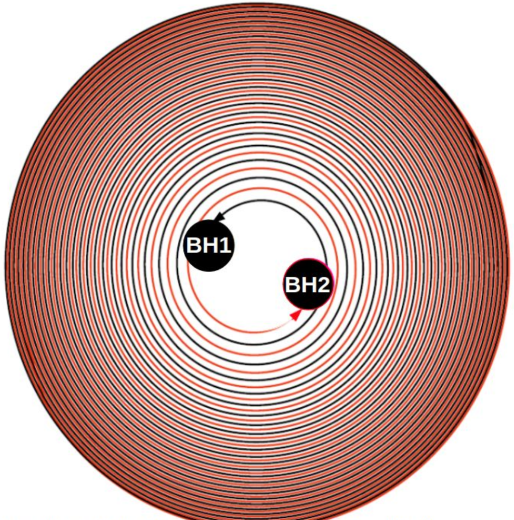
Anatomy of a Binary Black Hole Merger, as seen in gravitational waves



Lovelace et al., CQG 29, **045003** (2012) (modified)

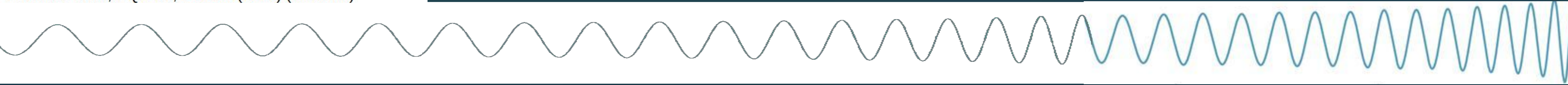
- Gravitational-wave driven "Relativistic death spiral"

Anatomy of a Binary Black Hole Merger, as seen in gravitational waves



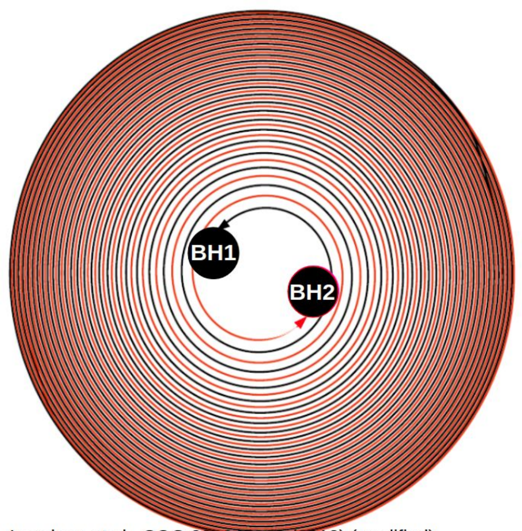
Lovelace et al., CQG 29, 045003 (2012) (modified)

- Gravitational-wave driven "Relativistic death spiral"



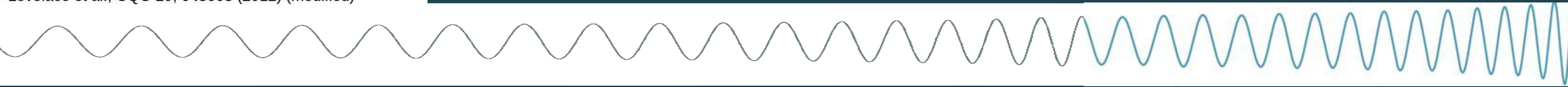
Time axis ⇨
(spans ~200ms)
Wave amplitude ⇧
(wave strain, arb. units)

Anatomy of a Binary Black Hole Merger, as seen in gravitational waves



Lovelace et al., CQG 29, 045003 (2012) (modified)

- Gravitational-wave driven <u>“Relativistic death spiral”</u>

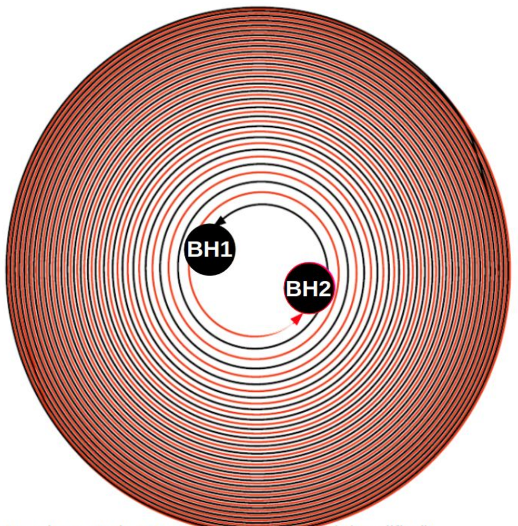


These waves encode info about masses, spins, and eccentricity of orbiting black holes

Time axis ⇨
(spans ~200ms)

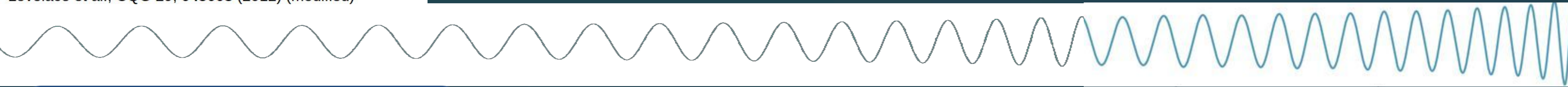
Wave amplitude ⇧
(wave strain, arb. units)

Anatomy of a Binary Black Hole Merger, as seen in gravitational waves



Lovelace et al., CQG 29, 045003 (2012) (modified)

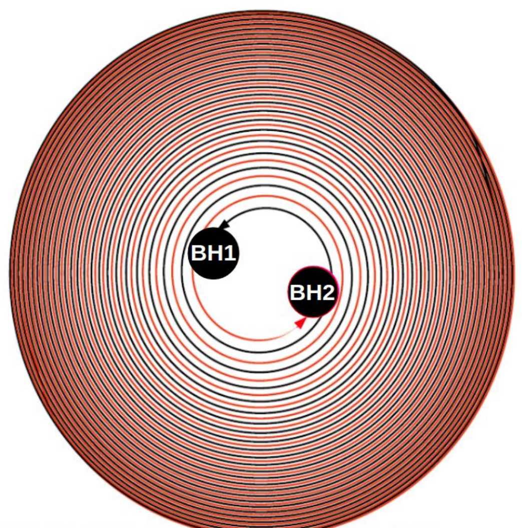
- Gravitational-wave driven <u>"Relativistic death spiral"</u>



⇦ (Very) early inspiral: Perturbative solutions to Einstein gravity (GR)

Time axis ⇨
(spans ~200ms)

Wave amplitude ⇧
(wave strain, arb. units)

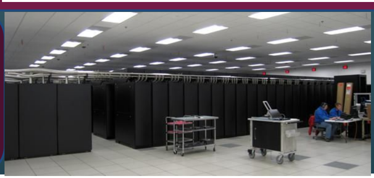


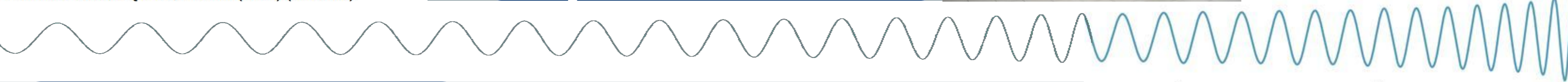
Lovelace et al., CQG 29, 045003 (2012) (modified)

Anatomy of a Binary Black Hole Merger, as seen in gravitational waves

- Gravitational-wave driven <u>“Relativistic death spiral”</u>

Late inspiral: Perturb. theory breaks down; **Only full GR solutions**





⇦ (Very) early inspiral: Perturbative solutions to Einstein gravity (GR)

Time axis ⇨
(spans ~200ms)

Wave amplitude ⇧
(wave strain, arb. units)

Modeling Challenges

Reformulate Einstein's theory of gravity for the computer

1. **Stability**, even when simulating BHs
2. **Reliability**: numerical errors small and well-understood

Modeling Challenges

Reformulate Einstein's theory of gravity for the computer

1. **Stability**, even when simulating BHs
2. **Reliability**: numerical errors small and well-understood

GR Equations are complex;
Solving the 2-body problem (two orbiting point masses) in GR took <u>90 years</u> (1915-2005)

Modeling Challenges

Reformulate Einstein's theory of gravity for the computer

1. **Stability**, even when simulating BHs
2. **Reliability**: numerical errors small and well-understood

GR Equations are complex;
Solving the 2-body problem (two orbiting point masses) in GR took 90 years (1915-2005)

$$G_{\mu\upsilon} = 8\pi T_{\mu\upsilon}$$

Modeling Challenges

Reformulate Einstein's theory of gravity for the computer

1. **Stability**, even when simulating BHs
2. **Reliability**: numerical errors small and well-understood

GR Equations are complex;
Solving the 2-body problem (two orbiting point masses) in GR took <u>90 years</u> (1915-2005)

$$G_{\mu\upsilon} = 8\pi T_{\mu\upsilon}$$

$$\partial_t \bar{\gamma}_{ij} = \left[\beta^k \partial_k \bar{\gamma}_{ij} + \partial_i \beta^k \bar{\gamma}_{kj} + \partial_j \beta^k \bar{\gamma}_{ik}\right] + \frac{2}{3}\bar{\gamma}_{ij}\left(\alpha \bar{A}^k_k - \bar{D}_k \beta^k\right) - 2\alpha \bar{A}_{ij} \, ,$$

$$\partial_t \bar{A}_{ij} = \left[\beta^k \partial_k \bar{A}_{ij} + \partial_i \beta^k \bar{A}_{kj} + \partial_j \beta^k \bar{A}_{ik}\right] - \frac{2}{3}\bar{A}_{ij} \bar{D}_k \beta^k - 2\alpha \bar{A}_{ik} \bar{A}^k{}_j + \alpha \bar{A}_{ij} K$$
$$+ e^{-4\phi}\left\{-2\alpha \bar{D}_i \bar{D}_j \phi + 4\alpha \bar{D}_i \phi \bar{D}_j \phi + 4 \bar{D}_{(i} \alpha \bar{D}_{j)} \phi - \bar{D}_i \bar{D}_j \alpha + \alpha \bar{R}_{ij}\right\}^{\mathrm{TF}} \, ,$$

$$\partial_t \phi = \left[\beta^k \partial_k \phi\right] + \frac{1}{6}\left(\bar{D}_k \beta^k - \alpha K\right) \, ,$$

$$\partial_t K = \left[\beta^k \partial_k K\right] + \frac{1}{3}\alpha K^2 + \alpha \bar{A}_{ij} \bar{A}^{ij} - e^{-4\phi}\left(\bar{D}_i \bar{D}^i \alpha + 2 \bar{D}^i \alpha \bar{D}_i \phi\right) \, ,$$

$$\partial_t \bar{\Lambda}^i = \left[\beta^k \partial_k \bar{\Lambda}^i - \partial_k \beta^i \bar{\Lambda}^k\right] + \bar{\gamma}^{jk} \hat{D}_j \hat{D}_k \beta^i + \frac{2}{3}\Delta^i \bar{D}_j \beta^j + \frac{1}{3}\bar{D}^i \bar{D}_j \beta^j$$
$$- 2\bar{A}^{ij}\left(\partial_j \alpha - 6\partial_j \phi\right) + 2\alpha \bar{A}^{jk} \Delta^i_{jk} - \frac{4}{3}\alpha \bar{\gamma}^{ij} \partial_j K$$

$$\partial_t \alpha = \left[\beta^i \partial_i \alpha\right] - 2\alpha K$$

$$\partial_t \beta^i = \left[\beta^j \partial_j \beta^i\right] + B^i$$

$$\partial_t B^i = \left[\beta^j \partial_j B^i\right] + \frac{3}{4}\partial_0 \bar{\Lambda}^i - \eta B^i$$

Most popular formulation

NRPy+: Automated code generation for numerical relativity

$$
\begin{aligned}
\partial_t \bar{\gamma}_{ij} &= \left[\beta^k \partial_k \bar{\gamma}_{ij} + \partial_i \beta^k \bar{\gamma}_{kj} + \partial_j \beta^k \bar{\gamma}_{ik}\right] + \frac{2}{3}\bar{\gamma}_{ij}\left(\alpha \bar{A}^k_k - \bar{D}_k \beta^k\right) - 2\alpha \bar{A}_{ij} \ , \\
\partial_t \bar{A}_{ij} &= \left[\beta^k \partial_k \bar{A}_{ij} + \partial_i \beta^k \bar{A}_{kj} + \partial_j \beta^k \bar{A}_{ik}\right] - \frac{2}{3}\bar{A}_{ij}\bar{D}_k \beta^k - 2\alpha \bar{A}_{ik}\bar{A}^k{}_j + \alpha \bar{A}_{ij} K \\
&\quad + e^{-4\phi}\left\{-2\alpha \bar{D}_i \bar{D}_j \phi + 4\alpha \bar{D}_i \phi \bar{D}_j \phi + 4\bar{D}_{(i}\alpha \bar{D}_{j)}\phi - \bar{D}_i \bar{D}_j \alpha + \alpha \bar{R}_{ij}\right\}^{\rm TF} \ , \\
\partial_t \phi &= \left[\beta^k \partial_k \phi\right] + \frac{1}{6}\left(\bar{D}_k \beta^k - \alpha K\right) \ , \\
\partial_t K &= \left[\beta^k \partial_k K\right] + \frac{1}{3}\alpha K^2 + \alpha \bar{A}_{ij}\bar{A}^{ij} - e^{-4\phi}\left(\bar{D}_i \bar{D}^i \alpha + 2\bar{D}^i \alpha \bar{D}_i \phi\right) \ , \\
\partial_t \bar{\Lambda}^i &= \left[\beta^k \partial_k \bar{\Lambda}^i - \partial_k \beta^i \bar{\Lambda}^k\right] + \bar{\gamma}^{jk}\hat{D}_j \hat{D}_k \beta^i + \frac{2}{3}\Delta^i \bar{D}_j \beta^j + \frac{1}{3}\bar{D}^i \bar{D}_j \beta^j \\
&\quad - 2\bar{A}^{ij}\left(\partial_j \alpha - 6\partial_j \phi\right) + 2\alpha \bar{A}^{jk}\Delta^i_{jk} - \frac{4}{3}\alpha \bar{\gamma}^{ij}\partial_j K \\
\partial_t \alpha &= \left[\beta^i \partial_i \alpha\right] - 2\alpha K \\
\partial_t \beta^i &= \left[\beta^j \partial_j \beta^i\right] + B^i \\
\partial_t B^i &= \left[\beta^j \partial_j B^i\right] + \frac{3}{4}\partial_0 \bar{\Lambda}^i - \eta B^i , \ .
\end{aligned}
$$

Most popular formulation

NRPy+: Automated code generation for numerical relativity

$$\partial_t \bar{\gamma}_{ij} = \left[\beta^k \partial_k \bar{\gamma}_{ij} + \partial_i \beta^k \bar{\gamma}_{kj} + \partial_j \beta^k \bar{\gamma}_{ik}\right] + \frac{2}{3}\bar{\gamma}_{ij}\left(\alpha \bar{A}^k_k - \bar{D}_k \beta^k\right) - 2\alpha \bar{A}_{ij} ,$$

$$\partial_t \bar{A}_{ij} = \left[\beta^k \partial_k \bar{A}_{ij} + \partial_i \beta^k \bar{A}_{kj} + \partial_j \beta^k \bar{A}_{ik}\right] - \frac{2}{3}\bar{A}_{ij}\bar{D}_k \beta^k - 2\alpha \bar{A}_{ik}\bar{A}^k{}_j + \alpha \bar{A}_{ij} K$$
$$+ e^{-4\phi}\left\{-2\alpha \bar{D}_i \bar{D}_j \phi + 4\alpha \bar{D}_i \phi \bar{D}_j \phi + 4\bar{D}_{(i}\alpha \bar{D}_{j)}\phi - \bar{D}_i \bar{D}_j \alpha + \alpha \bar{R}_{ij}\right\}^{\rm TF} ,$$

$$\partial_t \phi = \left[\beta^k \partial_k \phi\right] + \frac{1}{6}\left(\bar{D}_k \beta^k - \alpha K\right) ,$$

$$\partial_t K = \left[\beta^k \partial_k K\right] + \frac{1}{3}\alpha K^2 + \alpha \bar{A}_{ij}\bar{A}^{ij} - e^{-4\phi}\left(\bar{D}_i \bar{D}^i \alpha + 2\bar{D}^i \alpha \bar{D}_i \phi\right) ,$$

$$\partial_t \bar{\Lambda}^i = \left[\beta^k \partial_k \bar{\Lambda}^i - \partial_k \beta^i \bar{\Lambda}^k\right] + \bar{\gamma}^{jk}\hat{D}_j \hat{D}_k \beta^i + \frac{2}{3}\Delta^i \bar{D}_j \beta^j + \frac{1}{3}\bar{D}^i \bar{D}_j \beta^j$$
$$- 2\bar{A}^{ij}\left(\partial_j \alpha - 6\partial_j \phi\right) + 2\alpha \bar{A}^{jk}\Delta^i_{jk} - \frac{4}{3}\alpha \bar{\gamma}^{ij}\partial_j K$$

$$\partial_t \alpha = \left[\beta^i \partial_i \alpha\right] - 2\alpha K$$

$$\partial_t \beta^i = \left[\beta^j \partial_j \beta^i\right] + B^i$$

$$\partial_t B^i = \left[\beta^j \partial_j B^i\right] + \frac{3}{4}\partial_0 \bar{\Lambda}^i - \eta B^i , .$$

How to code this up?!

NRPy+: Automated code generation for numerical relativity

$$\partial_t \bar{\gamma}_{ij} = \left[\beta^k \partial_k \bar{\gamma}_{ij} + \partial_i \beta^k \bar{\gamma}_{kj} + \partial_j \beta^k \bar{\gamma}_{ik}\right] + \frac{2}{3}\bar{\gamma}_{ij}\left(\alpha \bar{A}^k_k - \bar{D}_k \beta^k\right) - 2\alpha \bar{A}_{ij} \,,$$

$$\partial_t \bar{A}_{ij} = \left[\beta^k \partial_k \bar{A}_{ij} + \partial_i \beta^k \bar{A}_{kj} + \partial_j \beta^k \bar{A}_{ik}\right] - \frac{2}{3}\bar{A}_{ij}\bar{D}_k \beta^k - 2\alpha \bar{A}_{ik}\bar{A}^k{}_j + \alpha \bar{A}_{ij} K$$
$$+ e^{-4\phi}\left\{-2\alpha \bar{D}_i \bar{D}_j \phi + 4\alpha \bar{D}_i \phi \bar{D}_j \phi + 4\bar{D}_{(i}\alpha \bar{D}_{j)}\phi - \bar{D}_i \bar{D}_j \alpha + \alpha \bar{R}_{ij}\right\}^{\rm TF} ,$$

$$\partial_t \phi = \left[\beta^k \partial_k \phi\right] + \frac{1}{6}\left(\bar{D}_k \beta^k - \alpha K\right) \,,$$

$$\partial_t K = \left[\beta^k \partial_k K\right] + \frac{1}{3}\alpha K^2 + \alpha \bar{A}_{ij}\bar{A}^{ij} - e^{-4\phi}\left(\bar{D}_i \bar{D}^i \alpha + 2\bar{D}^i \alpha \bar{D}_i \phi\right) \,,$$

$$\partial_t \bar{\Lambda}^i = \left[\beta^k \partial_k \bar{\Lambda}^i - \partial_k \beta^i \bar{\Lambda}^k\right] + \bar{\gamma}^{jk}\hat{D}_j \hat{D}_k \beta^i + \frac{2}{3}\Delta^i \bar{D}_j \beta^j + \frac{1}{3}\bar{D}^i \bar{D}_j \beta^j$$
$$- 2\bar{A}^{ij}\left(\partial_j \alpha - 6\partial_j \phi\right) + 2\alpha \bar{A}^{jk}\Delta^i_{jk} - \frac{4}{3}\alpha \bar{\gamma}^{ij}\partial_j K$$

$$\partial_t \alpha = \left[\beta^i \partial_i \alpha\right] - 2\alpha K$$

$$\partial_t \beta^i = \left[\beta^j \partial_j \beta^i\right] + B^i$$

$$\partial_t B^i = \left[\beta^j \partial_j B^i\right] + \frac{3}{4}\partial_0 \bar{\Lambda}^i - \eta B^i \,, \,.$$

How to code this up?!

NRPy+: Python-based C code generation framework for NR

Tensorial expressions in Einstein-like notation ⇒ Highly optimized C-code kernels (with FDs)

"Nerpy", the NRPy+ mascot. Photo CC2.0 Pacific Environment (modified).

https://nrpyplus.net

https://github.com/zachetienne/nrpytutorial

NRPy+: Automated code generation for numerical relativity

$$
\begin{aligned}
\partial_t \bar{\gamma}_{ij} &= \left[\beta^k \partial_k \bar{\gamma}_{ij} + \partial_i \beta^k \bar{\gamma}_{kj} + \partial_j \beta^k \bar{\gamma}_{ik}\right] + \frac{2}{3}\bar{\gamma}_{ij}\left(\alpha \bar{A}^k_k - \bar{D}_k \beta^k\right) - 2\alpha \bar{A}_{ij}\ , \\
\partial_t \bar{A}_{ij} &= \left[\beta^k \partial_k \bar{A}_{ij} + \partial_i \beta^k \bar{A}_{kj} + \partial_j \beta^k \bar{A}_{ik}\right] - \frac{2}{3}\bar{A}_{ij}\bar{D}_k \beta^k - 2\alpha \bar{A}_{ik}\bar{A}^k{}_j + \alpha \bar{A}_{ij} K \\
&\quad + e^{-4\phi}\left\{-2\alpha \bar{D}_i \bar{D}_j \phi + 4\alpha \bar{D}_i \phi \bar{D}_j \phi + 4\bar{D}_{(i}\alpha \bar{D}_{j)}\phi - \bar{D}_i \bar{D}_j \alpha + \alpha \bar{R}_{ij}\right\}^{\text{TF}}\ , \\
\partial_t \phi &= \left[\beta^k \partial_k \phi\right] + \frac{1}{6}\left(\bar{D}_k \beta^k - \alpha K\right)\ , \\
\partial_t K &= \left[\beta^k \partial_k K\right] + \frac{1}{3}\alpha K^2 + \alpha \bar{A}_{ij}\bar{A}^{ij} - e^{-4\phi}\left(\bar{D}_i \bar{D}^i \alpha + 2\bar{D}^i \alpha \bar{D}_i \phi\right)\ , \\
\partial_t \bar{\Lambda}^i &= \left[\beta^k \partial_k \bar{\Lambda}^i - \partial_k \beta^i \bar{\Lambda}^k\right] + \bar{\gamma}^{jk}\hat{D}_j \hat{D}_k \beta^i + \frac{2}{3}\Delta^i \bar{D}_j \beta^j + \frac{1}{3}\bar{D}^i \bar{D}_j \beta^j \\
&\quad - 2\bar{A}^{ij}\left(\partial_j \alpha - 6\partial_j \phi\right) + 2\alpha \bar{A}^{jk}\Delta^i_{jk} - \frac{4}{3}\alpha \bar{\gamma}^{ij}\partial_j K \\
\partial_t \alpha &= \left[\beta^i \partial_i \alpha\right] - 2\alpha K \\
\partial_t \beta^i &= \left[\beta^j \partial_j \beta^i\right] + B^i \\
\partial_t B^i &= \left[\beta^j \partial_j B^i\right] + \frac{3}{4}\partial_0 \bar{\Lambda}^i - \eta B^i\ ,\ .
\end{aligned}
$$

How to code this up?!

NRPy+: Python-based C code generation framework for NR

Tensorial expressions in Einstein-like notation ⇒ Highly optimized C-code kernels (with FDs)

"Nerpy", the NRPy+ mascot. Photo CC2.0 Pacific Environment (modified).

https://nrpyplus.net

https://github.com/zachetienne/nrpytutorial

ICERM NRPy+ tutorial
Thursday, Oct 15; time TBD

Modeling Challenges

Reformulate Einstein's theory of gravity for the computer

1. **Stability**, even when simulating BHs
2. **Reliability**: numerical errors small and well-understood

Modeling Challenges

Reformulate Einstein’s theory of gravity for the computer

1. **Stability**, even when simulating BHs
2. **Reliability**: numerical errors small and well-understood

Address (~5 orders of mag) disparity in physical scales

1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

Modeling Challenges

Address (~5 orders of mag) disparity in physical scales

1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

Modeling Challenges

Address (~5 orders of mag) disparity in physical scales

1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

Modeling Challenges

Address (~5 orders of mag) disparity in physical scales

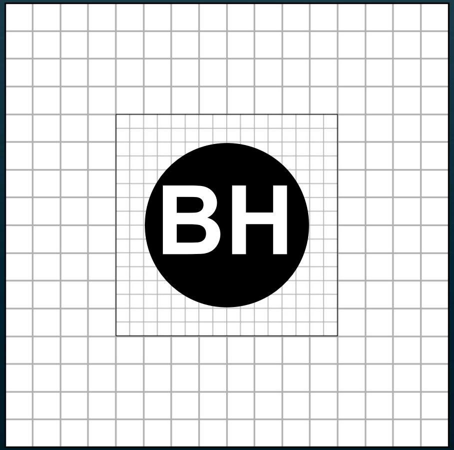
1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

Modeling Challenges

Address (~5 orders of mag) disparity in physical scales

1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

- Simulations performed on <u>numerical grids</u>
  - Numerical solution stored at each grid point
  - Need <u>denser</u> grids to model <u>sharper</u> features



Modeling Challenges

Address (~5 orders of mag) disparity in physical scales

1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

- Simulations performed on <u>numerical grids</u>
  - Numerical solution stored at each grid point
  - Need <u>denser</u> grids to model <u>sharper</u> features

Fewer grid points = ***lower computational cost***

Less computational cost unlocks

- More simulations, and/or
- More physical realism

Modeling Challenges

Address (~5 orders of mag) disparity in physical scales

1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

- Simulations performed on <u>numerical grids</u>
  - Numerical solution stored at each grid point
  - Need <u>denser</u> grids to model <u>sharper</u> features

Fewer grid points = ***lower computational cost***

Less computational cost unlocks

- More simulations, and/or
- More physical realism

How to address?

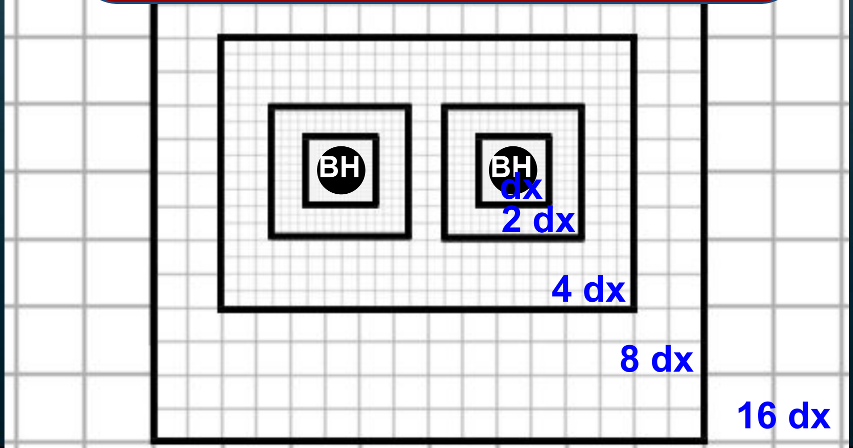
Modeling Challenges

Address (~5 orders of mag) disparity in physical scales

1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

AMR

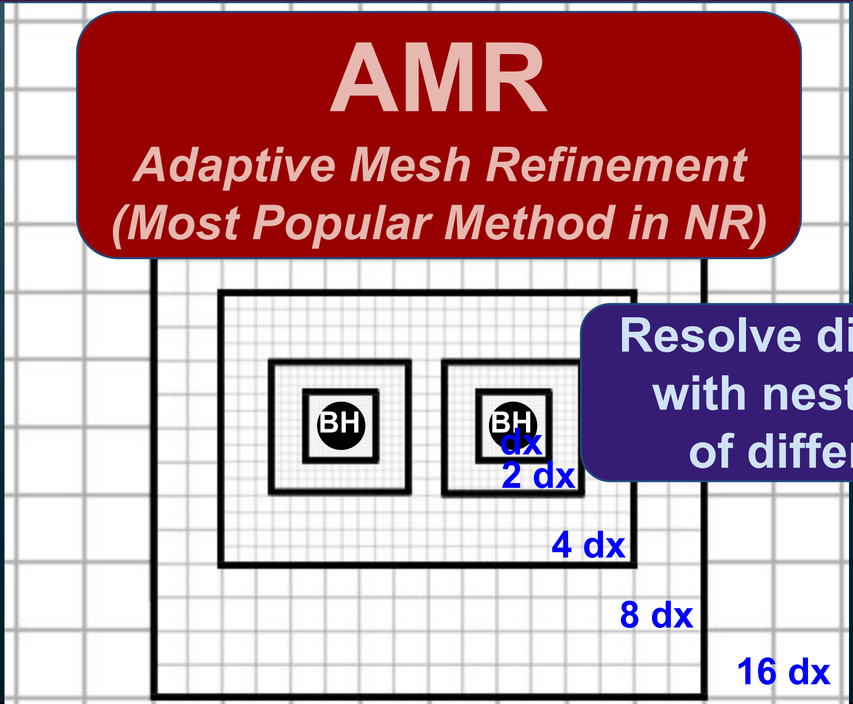
Adaptive Mesh Refinement
(Most Popular Method in NR)



Modeling Challenges

Address (~5 orders of mag) disparity in physical scales

1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)



AMR

Adaptive Mesh Refinement
(Most Popular Method in NR)

Resolve disparate lengthscales with nested Cartesian cubes of differing grid spacings

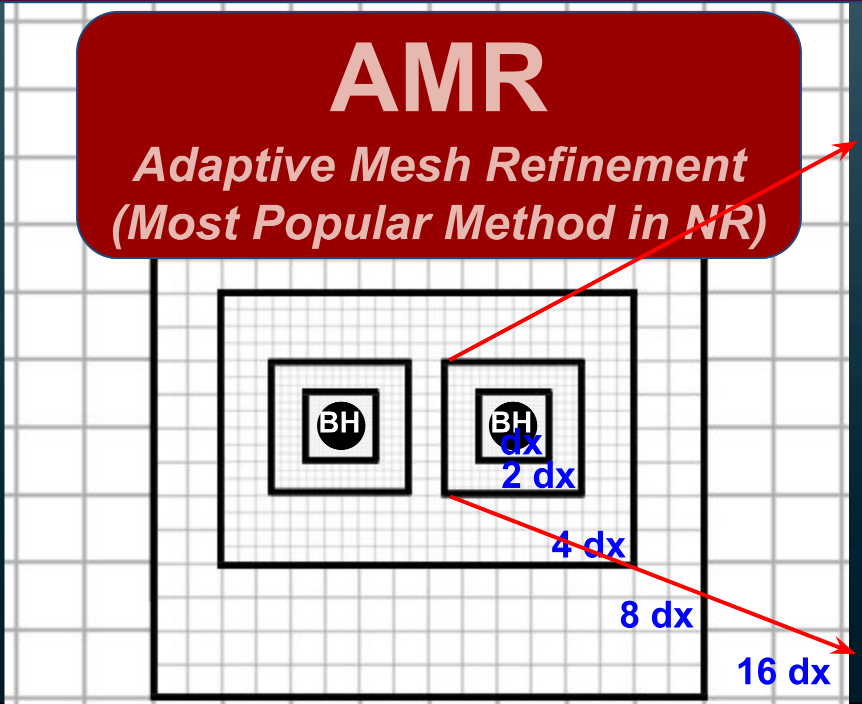
Modeling Challenges

Address (~5 orders of mag) disparity in physical scales

1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

AMR

Adaptive Mesh Refinement
(Most Popular Method in NR)



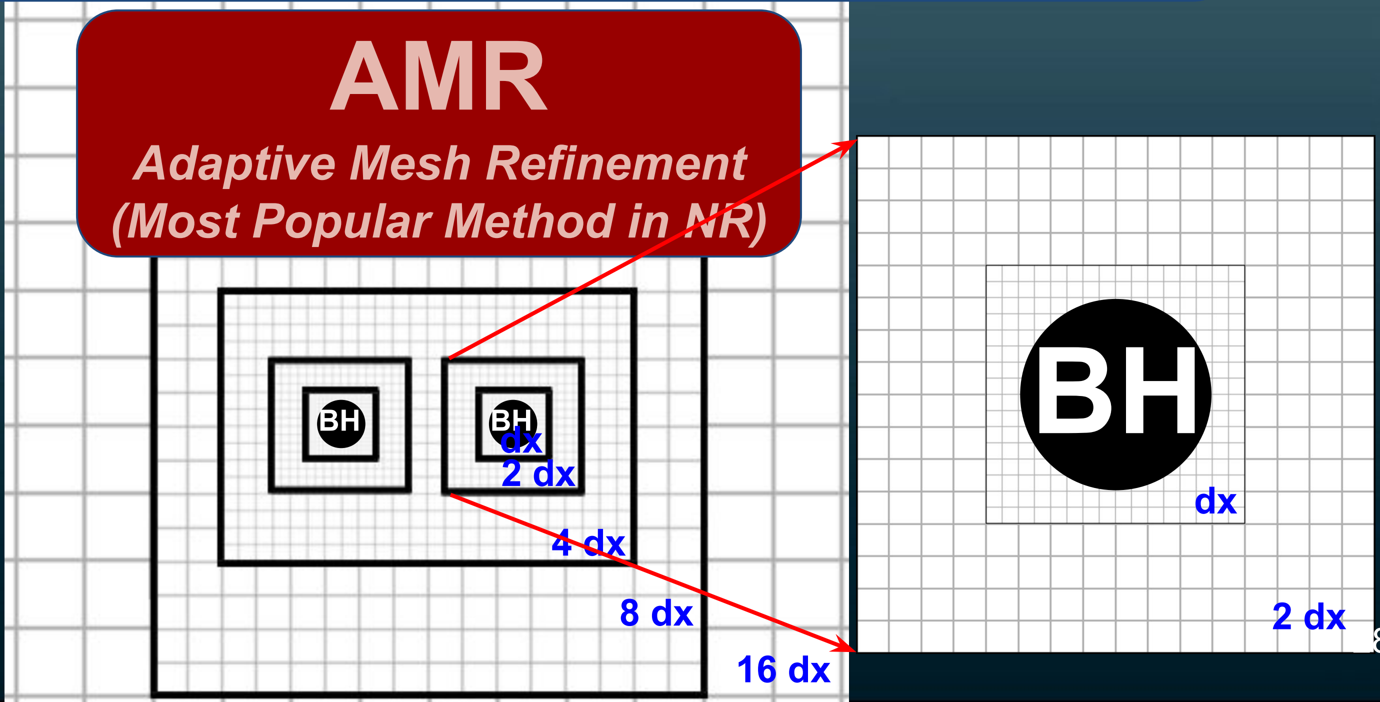
Modeling Challenges

Address (~5 orders of mag) disparity in physical scales

1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

AMR

Adaptive Mesh Refinement (Most Popular Method in NR)



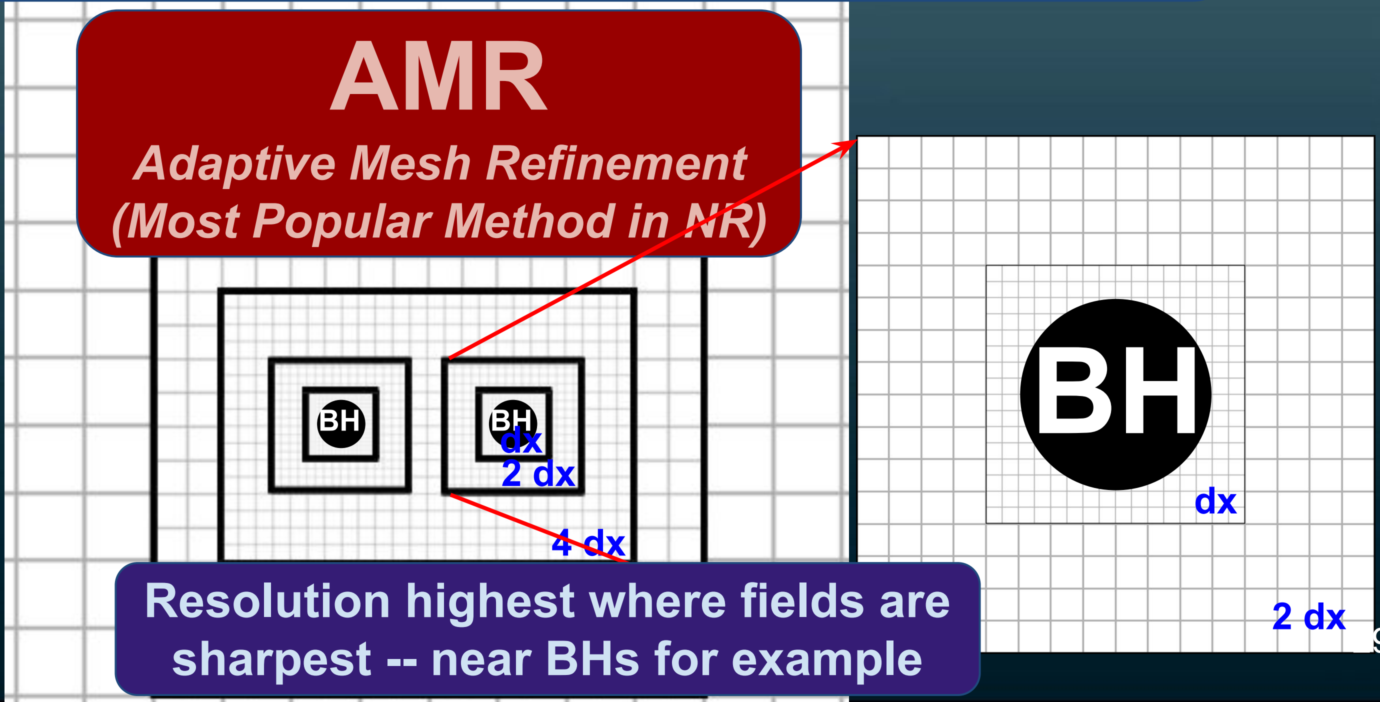
Modeling Challenges

Address (~5 orders of mag) disparity in physical scales

1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

AMR

Adaptive Mesh Refinement
(Most Popular Method in NR)



Resolution highest where fields are sharpest -- near BHs for example

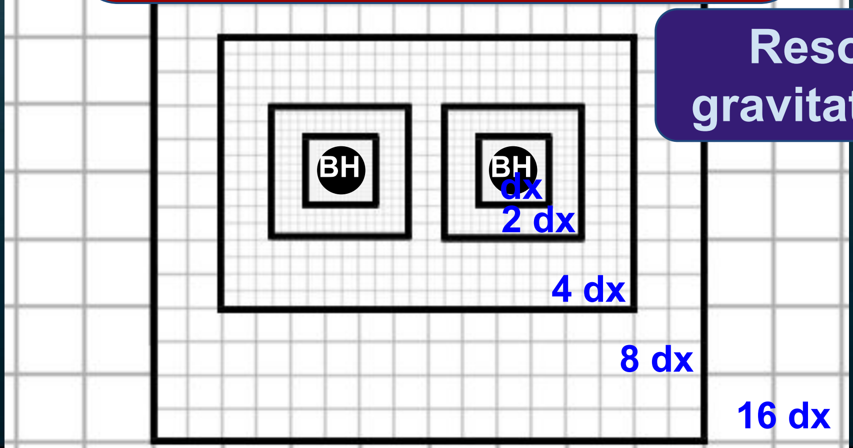
Modeling Challenges

Address (~5 orders of mag) disparity in physical scales

1. ✓ Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

AMR

Adaptive Mesh Refinement
(Most Popular Method in NR)



Resolution lower where gravitational waves modeled

Modeling Challenges

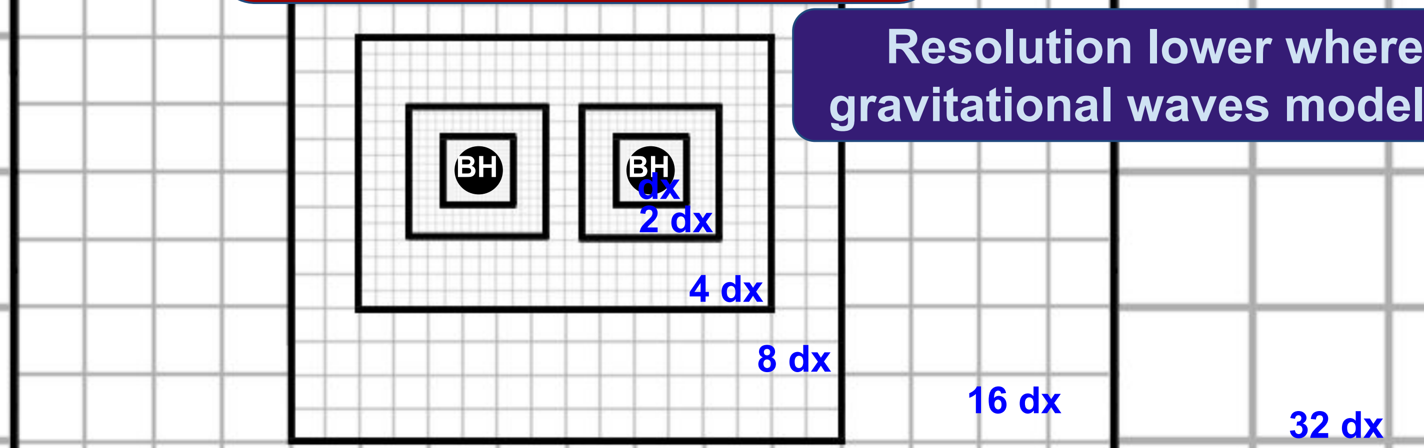
Address (~5 orders of mag) disparity in physical scales

- ✅ 1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
- 2. Model **long-wavelength** gravitational waves far away
- 3. Push outer boundary very far away (due to approx. BCs)

AMR

Adaptive Mesh Refinement
(Most Popular Method in NR)

Resolution lower where gravitational waves modeled



Modeling Challenges

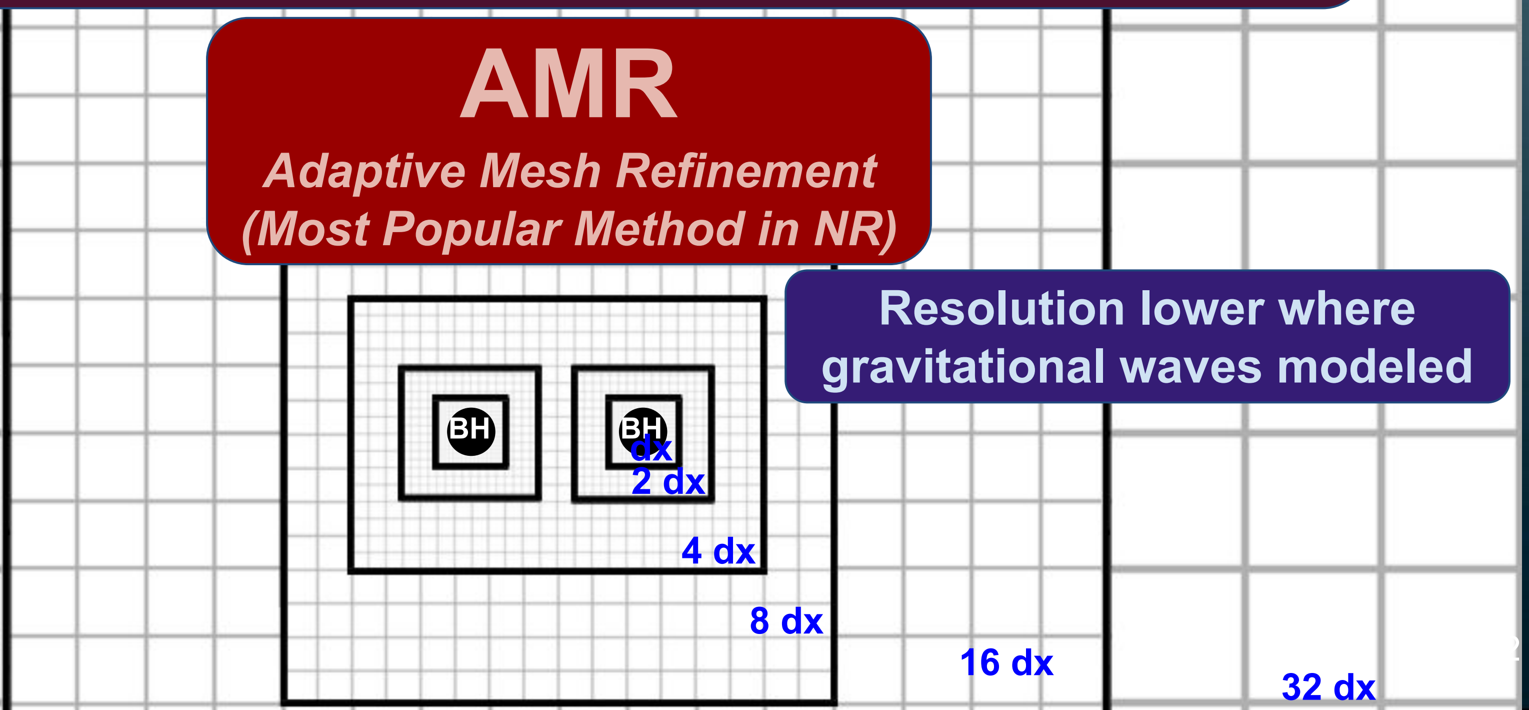
Address (~5 orders of mag) disparity in physical scales

- ✓ 1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
- ✓ 2. Model **long-wavelength** gravitational waves far away
- 3. Push outer boundary very far away (due to approx. BCs)

AMR

Adaptive Mesh Refinement
(Most Popular Method in NR)

Resolution lower where gravitational waves modeled



Modeling Challenges

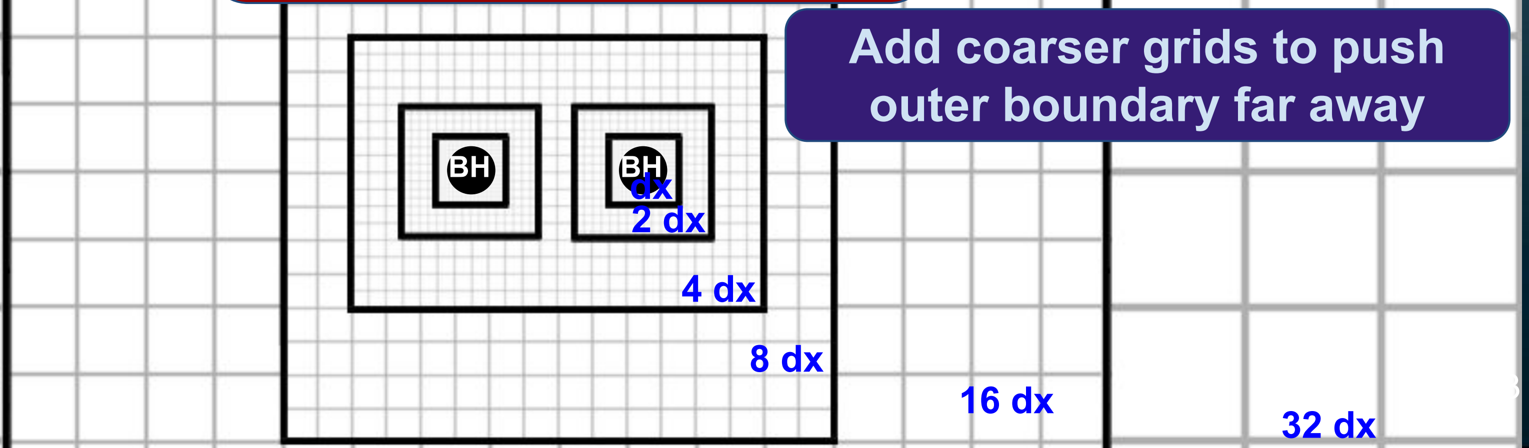
Address (~5 orders of mag) disparity in physical scales

- [x] 1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
- [x] 2. Model **long-wavelength** gravitational waves far away
- 3. Push outer boundary very far away (due to approx. BCs)

AMR

Adaptive Mesh Refinement
(Most Popular Method in NR)

Add coarser grids to push outer boundary far away



Modeling Challenges

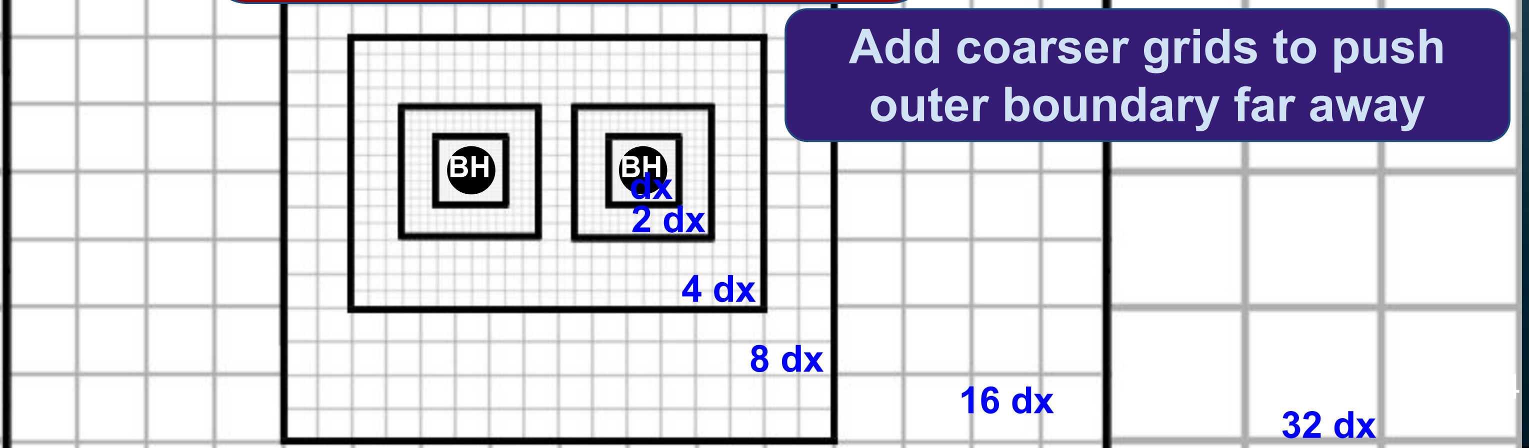
Address (~5 orders of mag) disparity in physical scales

1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

AMR

Adaptive Mesh Refinement
(Most Popular Method in NR)

Add coarser grids to push outer boundary far away



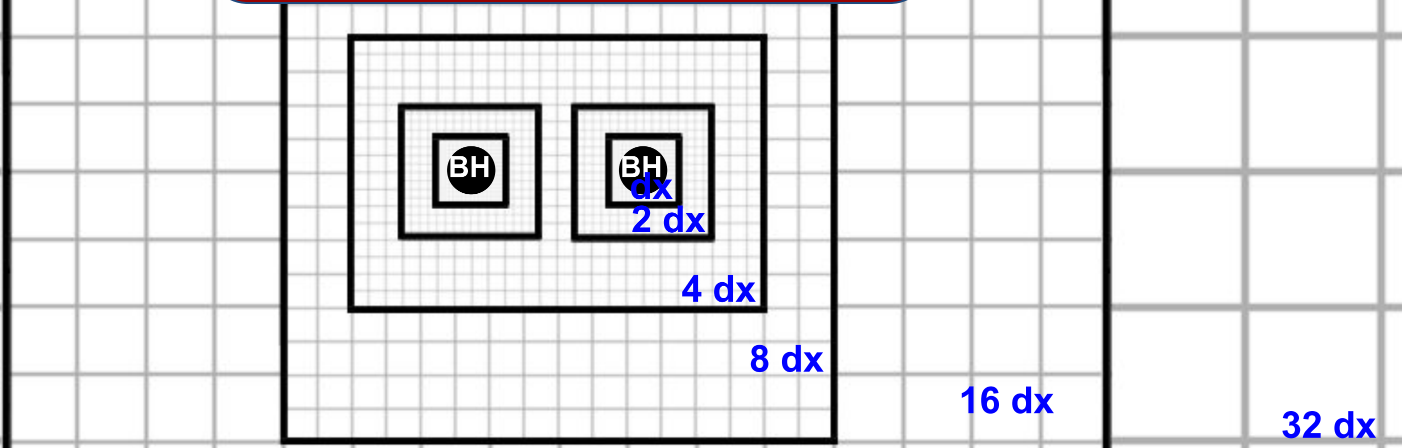
Modeling Challenges

Address (~5 orders of mag) disparity in physical scales

- ☑ 1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
- ☑ 2. Model **long-wavelength** gravitational waves far away
- ☑ 3. Push outer boundary very far away (due to approx. BCs)

AMR

Adaptive Mesh Refinement
(Most Popular Method in NR)



Modeling Challenges

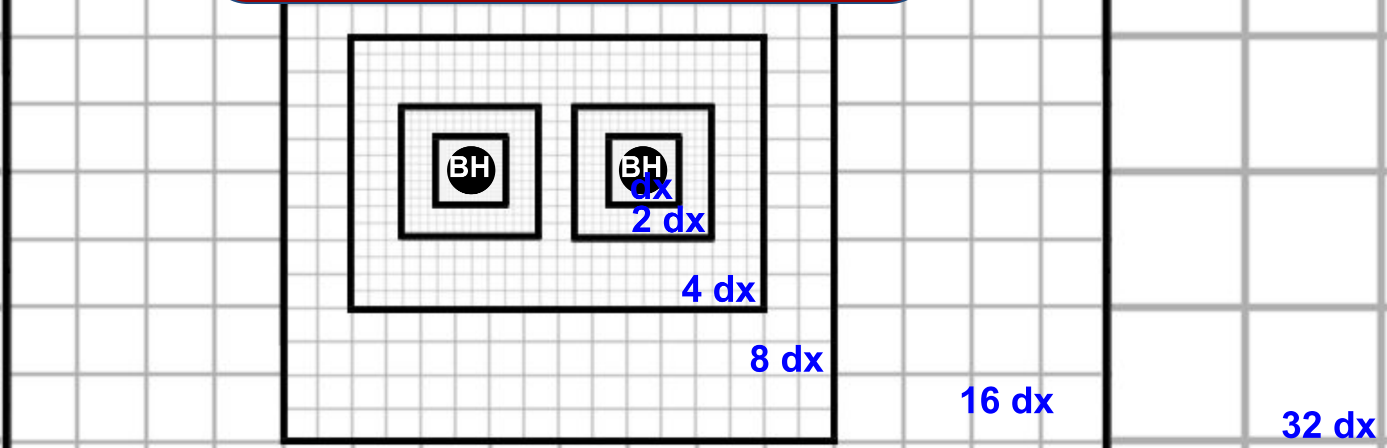
Address (~5 orders of mag) disparity in physical scales

1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

What's the problem?

AMR

Adaptive Mesh Refinement (Most Popular Method in NR)



Modeling Challenges

Address (~5 orders of mag) disparity in physical scales

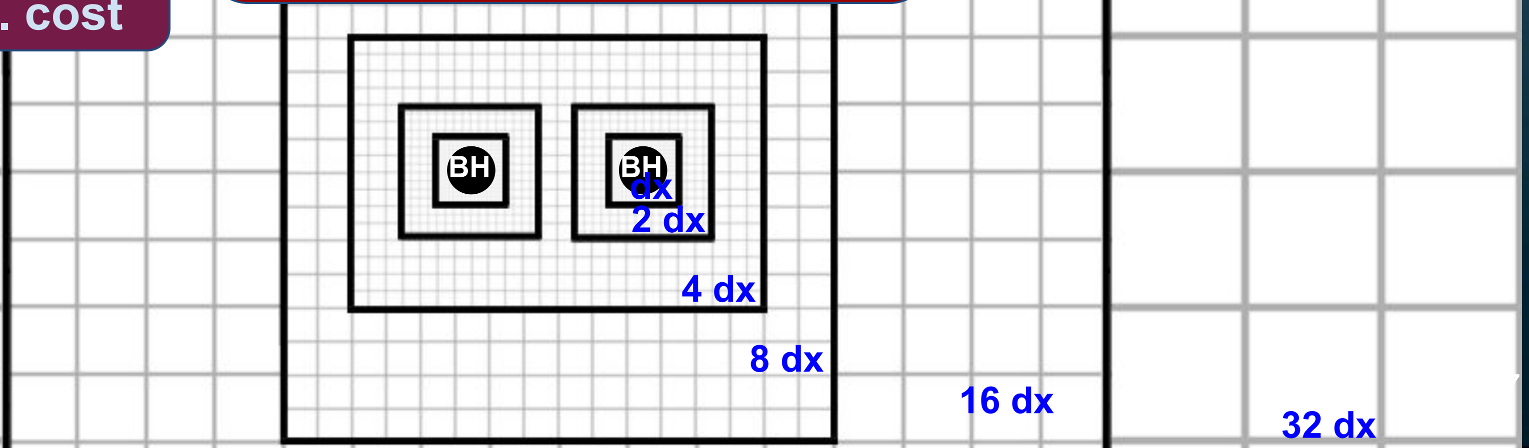
1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

What's the problem?

A: Inefficient!
⇨ greater comp. cost

AMR

Adaptive Mesh Refinement
(Most Popular Method in NR)

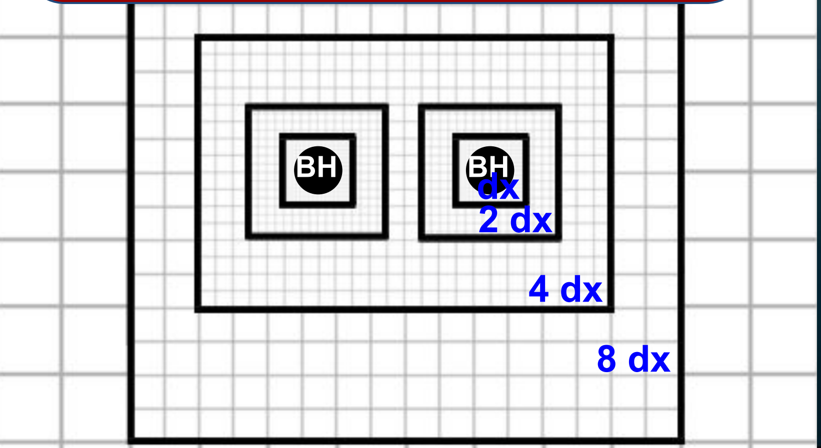


Modeling Challenges

Address (~5 orders of mag) disparity in physical scales

1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

AMR Grid *Inefficiencies*

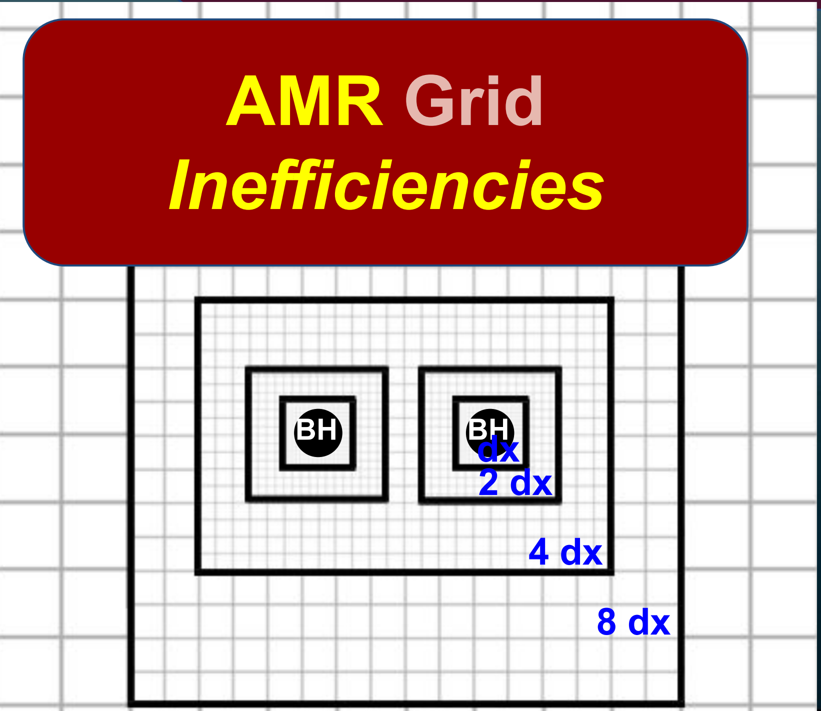


Modeling Challenges

Address (~5 orders of mag) disparity in physical scales

1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

AMR Grid *Inefficiencies*



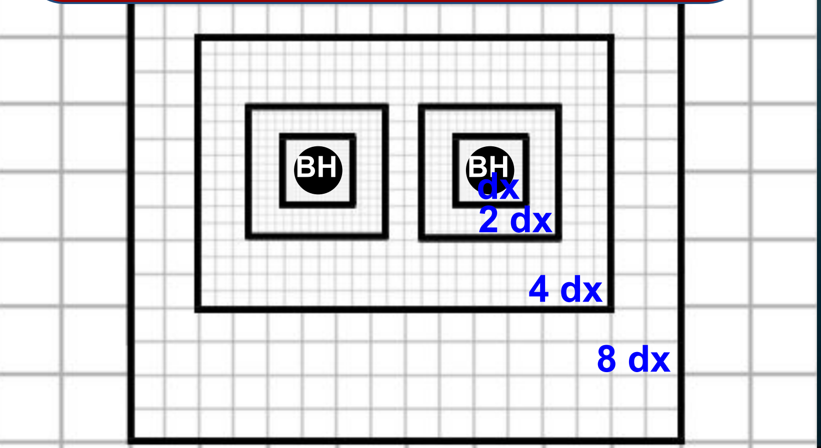
1. Black holes & gravitational waves: nearly spherical/axisymmetric
   - ⇨ grav fields vary most strongly in *radial* direction
     ⇨ <u>need high sampling in *only one* (*r*) direction</u>

Modeling Challenges

Address (~5 orders of mag) disparity in physical scales

1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

AMR Grid *Inefficiencies*



1. Black holes & gravitational waves: nearly spherical/axisymmetric
   - ⇨ grav fields vary most strongly in *radial* direction
     ⇨ <u>need high sampling in *only one* (*r*) direction</u>

2. <u>Gravitational fields are mostly smooth</u>
   - Cartesian **AMR** grids:
     - 2x jumps in resolution between boxes
     - Boxes have sharp corners

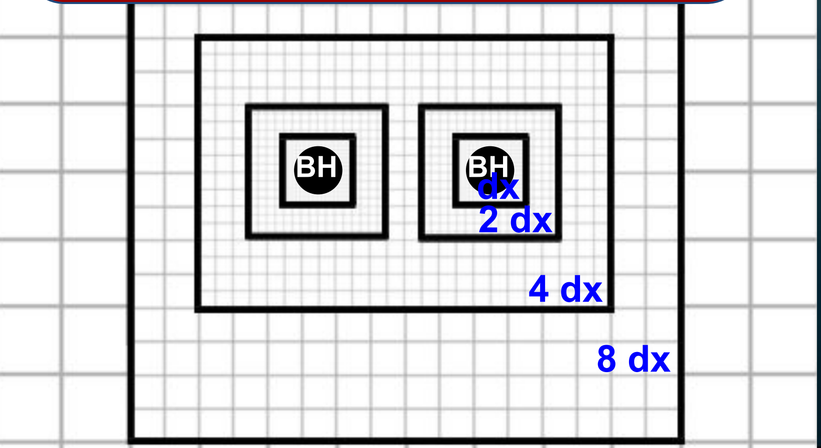
Modeling Challenges

Address (~5 orders of mag) disparity in physical scales

1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

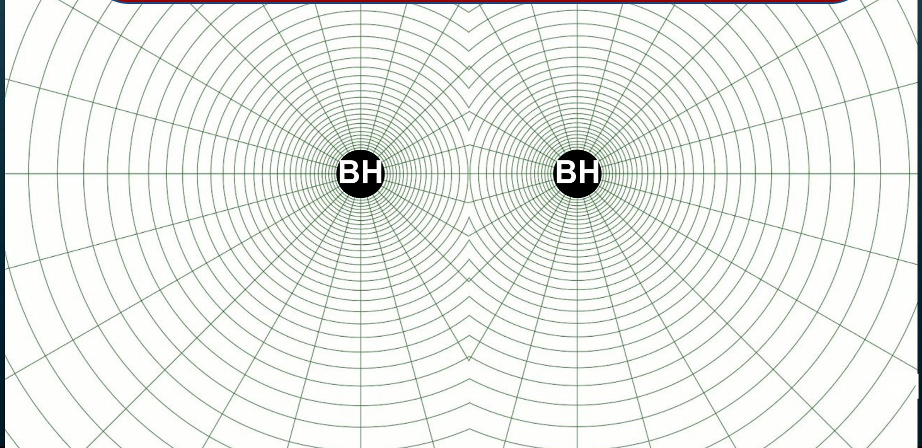
AMR Grids

Adaptive Mesh Refinement (Most Popular Method in NR)



New BiSphere Grids

~20x more efficient sampling for compact binary simulations



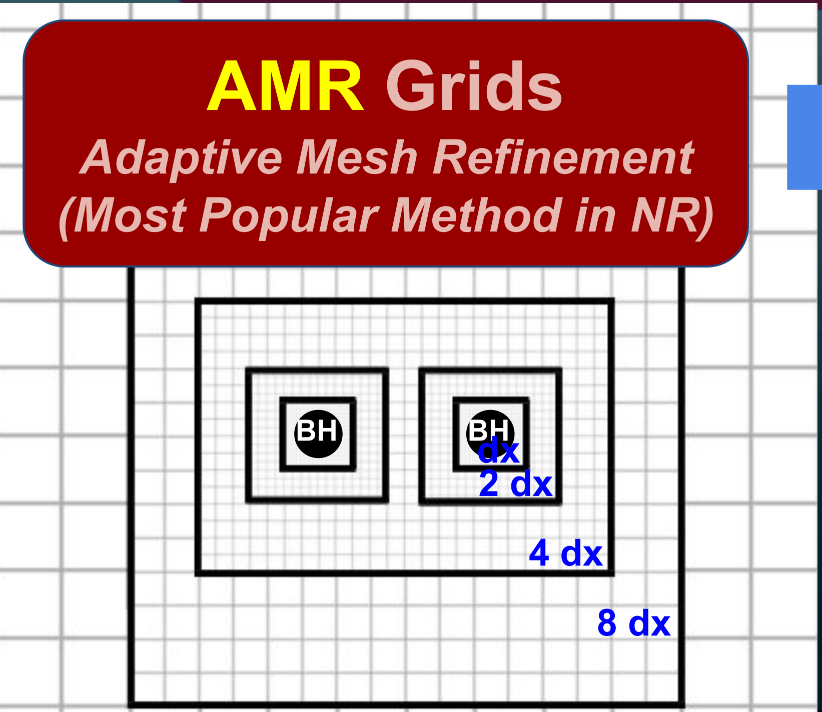
Modeling Challenges

Address (~5 orders of mag) disparity in physical scales

1. Resolve **sharp**, rapidly changing grav fields near BHs and NSs
2. Model **long-wavelength** gravitational waves far away
3. Push outer boundary very far away (due to approx. BCs)

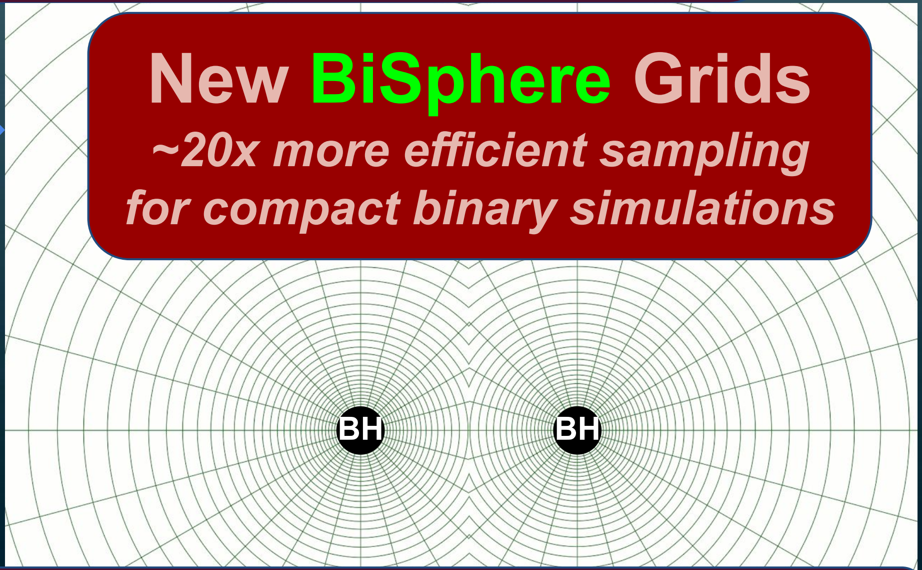
AMR Grids

Adaptive Mesh Refinement
(Most Popular Method in NR)



New BiSphere Grids

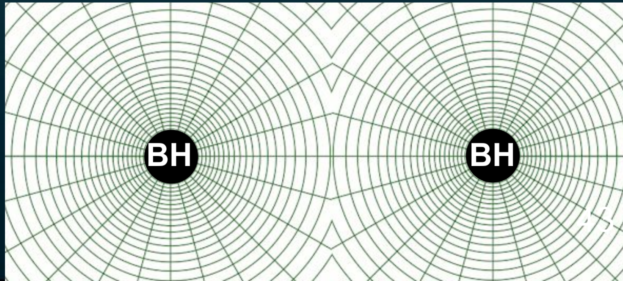
~20x more efficient sampling
for compact binary simulations



- **Exploits near-symmetries (~5x)**
- **Smooth transitions in resolution (~4x)**

BiSphere Challenges

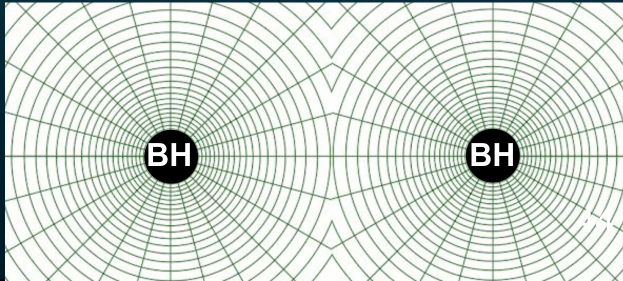
BiSphere grids: Two overlapping numerical grids in Spherical coordinates



BiSphere Challenges

BiSphere grids: Two overlapping numerical grids in Spherical coordinates

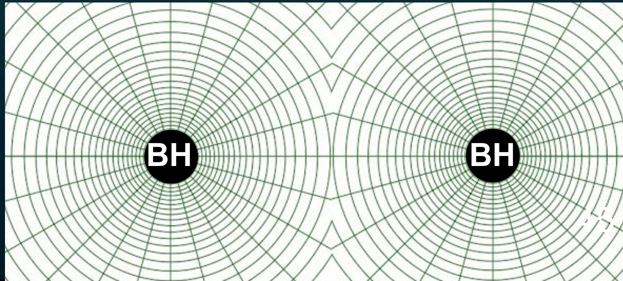
- Challenge #1: Spherical coordinates have **coordinate singularities**
  - Tensors and vectors diverge or go to zero -> numerically **unstable**



BiSphere Challenges

BiSphere grids: Two overlapping numerical grids in Spherical coordinates

- Challenge #1: Spherical coordinates have **coordinate singularities**
  - Tensors and vectors diverge or go to zero -> numerically **unstable**
- Idea: ***Scale out singular parts***, (treat singular & nonsingular separately)



BiSphere Challenges

BiSphere grids: Two overlapping numerical grids in Spherical coordinates

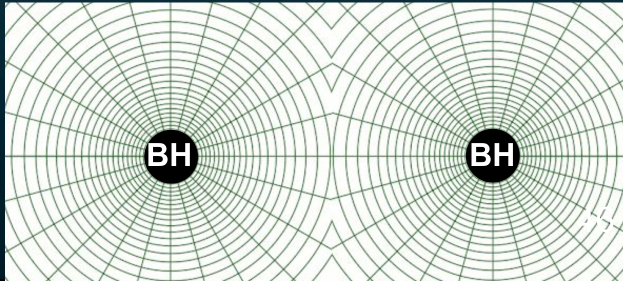
- <u>Challenge #1</u>: Spherical coordinates have **<u>coordinate singularities</u>**
  - Tensors and vectors diverge or go to zero -> numerically **<u>unstable</u>**
- <u>Idea: ***Scale out singular parts***</u>, (treat singular & nonsingular separately)
- Result: <u>Numerical stability & robustness *on par with Cartesian*</u>

a. **Ordinary spherical polar: done!**

Baumgarte, Montero, Cordero-Carrión, Müller (PRD 87, 044026, **2012**),
built upon covariant BSSN formulation of Brown (PRD 79, 104029, **2009**)

b. **Generic-radius spherical polar (incl. log-radial): done!**

Ruchlin, **Etienne**, Baumgarte (PRD 97, 064036, **2018**)



BiSphere Challenges

BiSphere grids: Two overlapping numerical grids in Spherical coordinates

- Challenge #1: Spherical coordinates have **coordinate singularities**
  - Tensors and vectors diverge or go to zero -> numerically **unstable**
- Idea: ***Scale out singular parts***, (treat singular & nonsingular separately)
- Result: Numerical stability & robustness *on par with Cartesian*

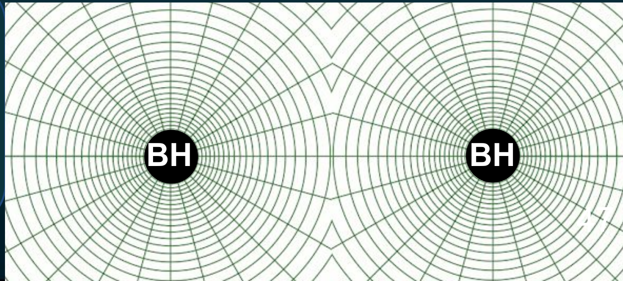
a. **Ordinary spherical polar: done!**

Baumgarte, Montero, Cordero-Carrión, Müller (PRD 87, 044026, **2012**), built upon covariant BSSN formulation of Brown (PRD 79, 104029, **2009**)

b. **Generic-radius spherical polar (incl. log-radial): done!**

Ruchlin, **Etienne**, Baumgarte (PRD 97, 064036, **2018**)

- Challenge #2: Develop new algorithm for
  - Interpolating between the two spherical grids; co-orbit grids with the binary system:
  - Changing basis between the two spherical grids:

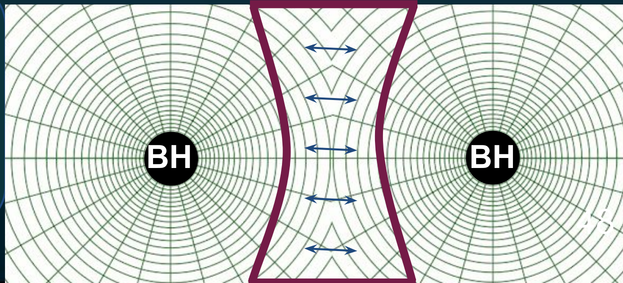


BiSphere Challenges

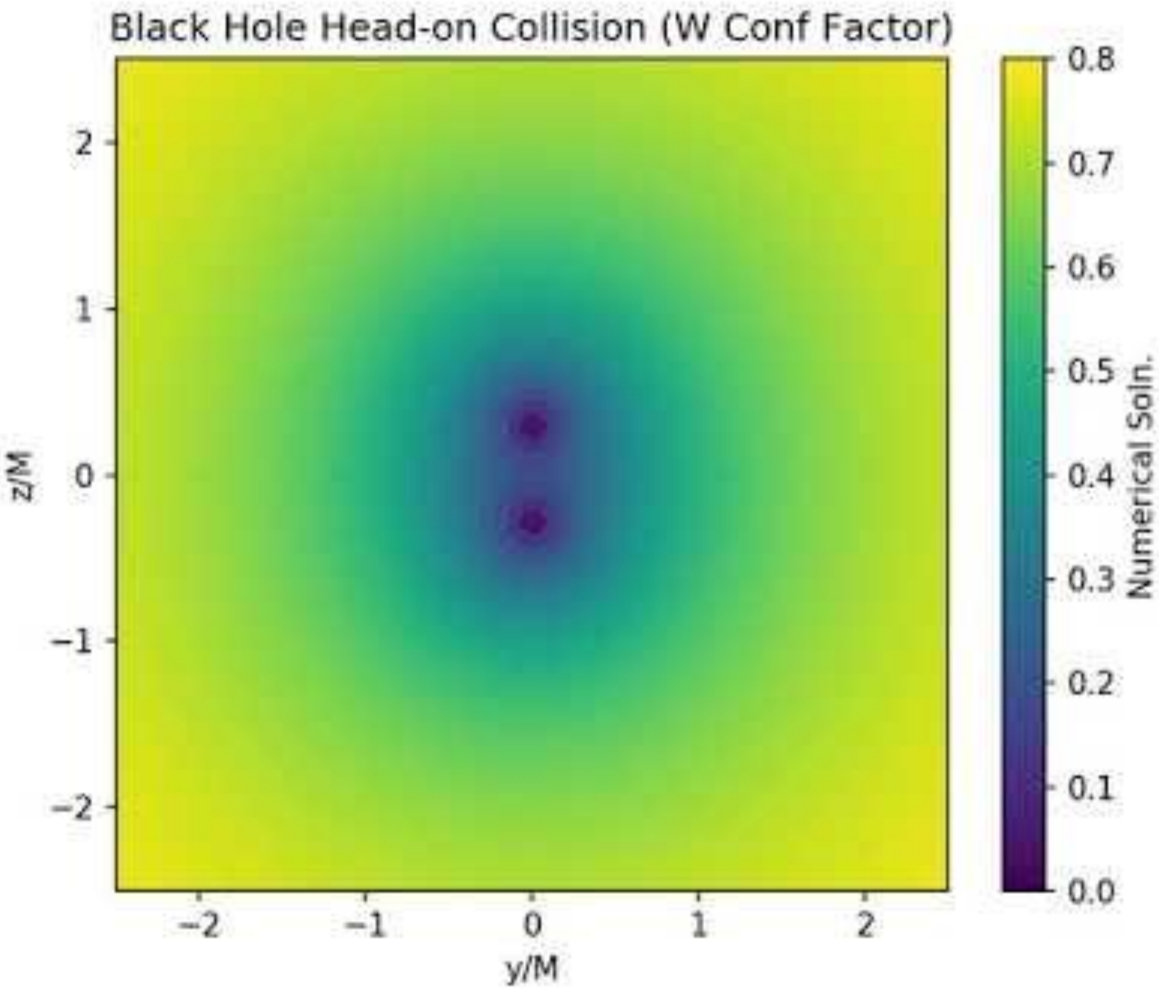
BiSphere grids: Two overlapping numerical grids in Spherical coordinates

- Challenge #1: Spherical coordinates have **coordinate singularities**
  - Tensors and vectors diverge or go to zero -> numerically **unstable**
- Idea: ***Scale out singular parts***, (treat singular & nonsingular separately)
- Result: Numerical stability & robustness *on par with Cartesian*
  a. **Ordinary spherical polar: done!**
     Baumgarte, Montero, Cordero-Carrión, Müller (PRD 87, 044026, **2012**), built upon covariant BSSN formulation of Brown (PRD 79, 104029, **2009**)
  b. **Generic-radius spherical polar (incl. log-radial): done!**
     Ruchlin, **Etienne**, Baumgarte (PRD 97, 064036, **2018**)

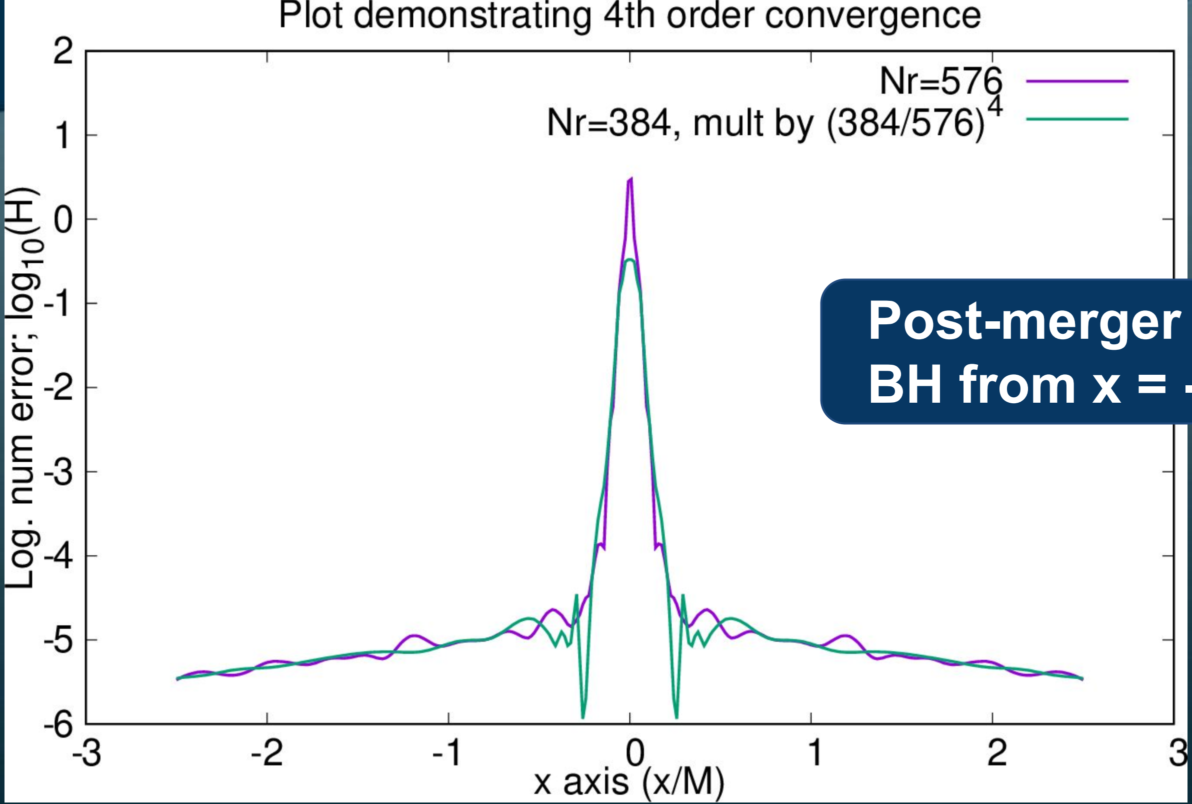
- Challenge #2: Develop new algorithm for
  - Interpolating between the two spherical grids; co-orbit grids with the binary system:
  - Changing basis between the two spherical grids:



2019 Results



2019 Results



Post-merger num error, BH from x = -0.5 to +0.5

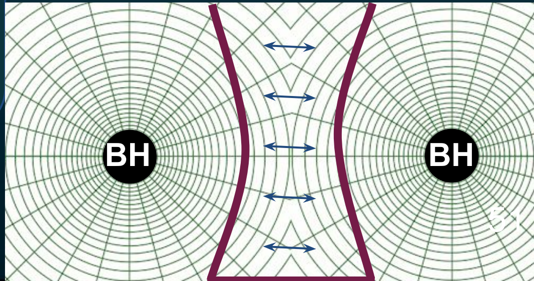
Finding from BH collision test: Numerical errors small and converge to zero at expected rate

BiSphere Challenges

BiSphere grids: Two overlapping numerical grids in Spherical coordinates

- Challenge #1: Spherical coordinates have **coordinate singularities**
  - Tensors and vectors diverge or go to zero -> numerically **unstable**
- Idea: ***Scale out singular parts***, (treat singular & nonsingular separately)
- Result: Numerical stability & robustness *on par with Cartesian*
  a. **Ordinary spherical polar: done!**
     Baumgarte, Montero, Cordero-Carrión, Müller (PRD 87, 044026, **2012**),
     built upon covariant BSSN formulation of Brown (PRD 79, 104029, **2009**)
  b. **Generic-radius spherical polar (incl. log-radial): done!**
     Ruchlin, **Etienne**, Baumgarte (PRD 97, 064036, **2018**)

- Challenge #2: Develop new algorithm for
  - Interpolating between the two spherical grids;
    co-orbit grids with the binary system: **done! (early 2019)**
  - Changing basis between the two spherical grids: **done!**



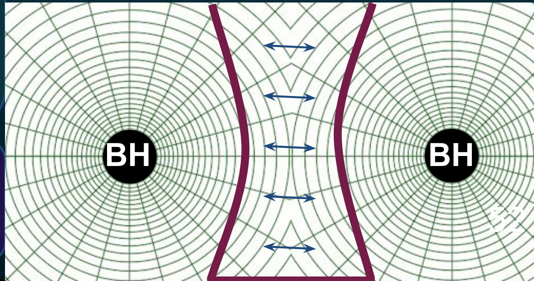
BiSphere Challenges

BiSphere grids: Two overlapping numerical grids in Spherical coordinates

- Challenge #1: Spherical coordinates have **coordinate singularities**
  - Tensors and vectors diverge or go to zero -> numerically **unstable**
- Idea: ***Scale out singular parts***, (treat singular & nonsingular separately)
- Result: Numerical stability & robustness *on par with Cartesian*
  a. **Ordinary spherical polar: done!**
     Baumgarte, Montero, Cordero-Carrión, Müller (PRD 87, 044026, **2012**),
     built upon covariant BSSN formulation of Brown (PRD 79, 104029, **2009**)
  b. **Generic-radius spherical polar (incl. log-radial): done!**
     Ruchlin, **Etienne**, Baumgarte (PRD 97, 064036, **2018**)

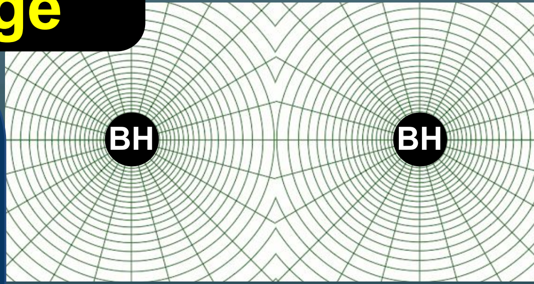
- Challenge #2: Develop new algorithm for
  - Interpolating between the two spherical grids;
    co-orbit grids with the binary system: **done! (early 2019)**
  - Changing basis between the two spherical grids: **done!**

- Challenge #3: Calculations *50x* too slow: *small timesteps!*



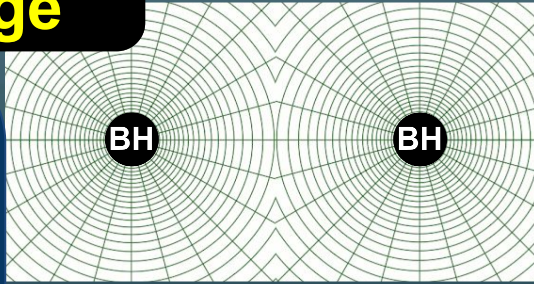
Small Timesteps,
The Last BiSphere Challenge

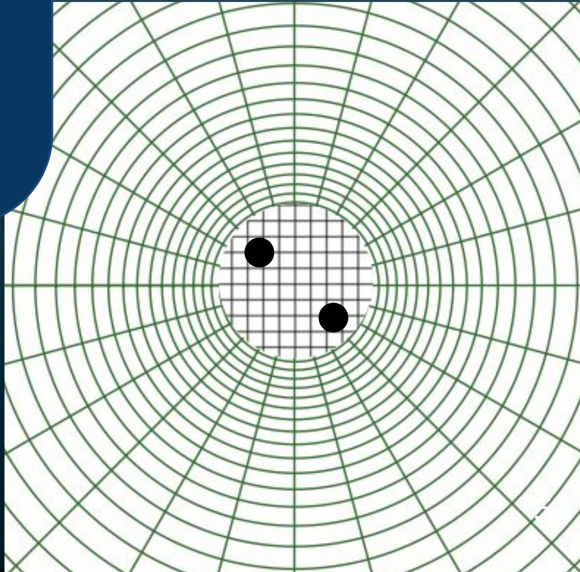
- BH binary fits on desktop now (6GB), but ~50x *too slow*
  - Recent: About 1.5x gain through software optimz!
- Problem:
  - <u>Simulation timestep ∝ min dist between gridpoints</u>
  - Spherical coords focus gridpoints at *r*=0, *z*-axis
- How to solve?



Small Timesteps, The Last BiSphere Challenge

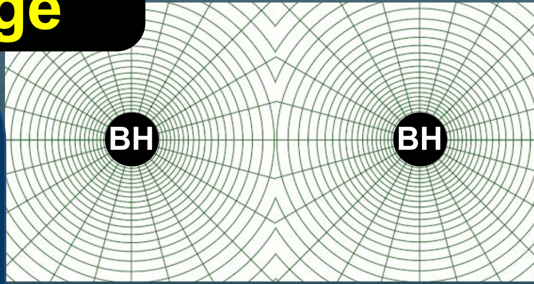
- BH binary fits on desktop now (6GB), but ~50x *too slow*
  - Recent: About 1.5x gain through software optimz!
- Problem:
  - Simulation timestep ∝ min dist between gridpoints
  - Spherical coords focus gridpoints at *r*=0, *z*-axis
- How to solve?
  - *Add a large Cartesian cube* (binary inside)
    - ~100x larger timesteps!
    - Memory usage?!

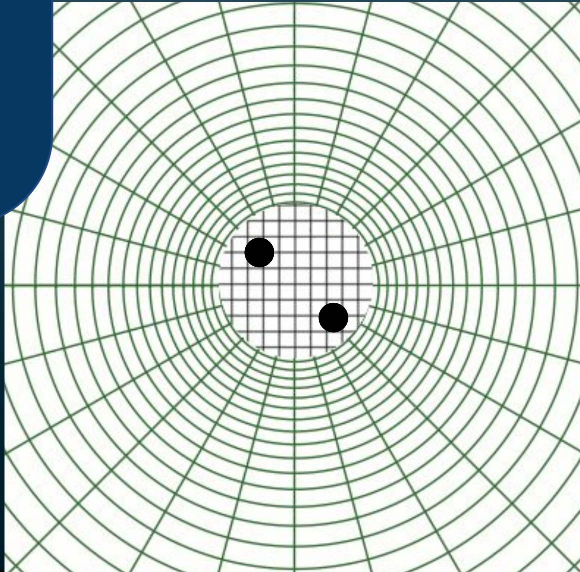




Small Timesteps, The Last BiSphere Challenge

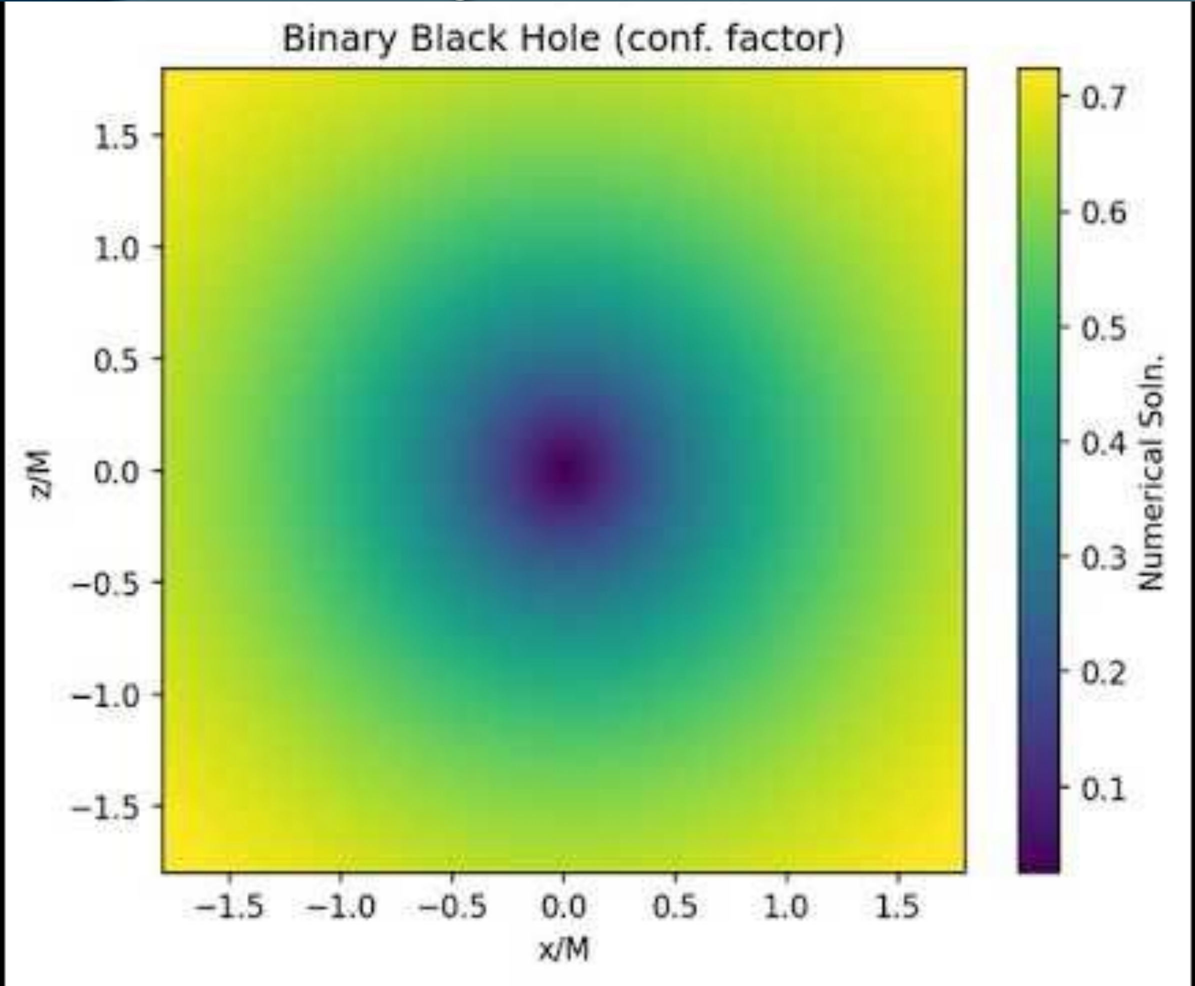
- BH binary fits on desktop now (6GB), but ~50x *too slow*
  - Recent: About 1.5x gain through software optimz!
- Problem:
  - Simulation timestep ∝ min dist between gridpoints
  - Spherical coords focus gridpoints at r=0, z-axis
- How to solve?
  - *Add a large Cartesian cube* (binary inside)
    - ~100x larger timesteps!
    - Memory usage?!





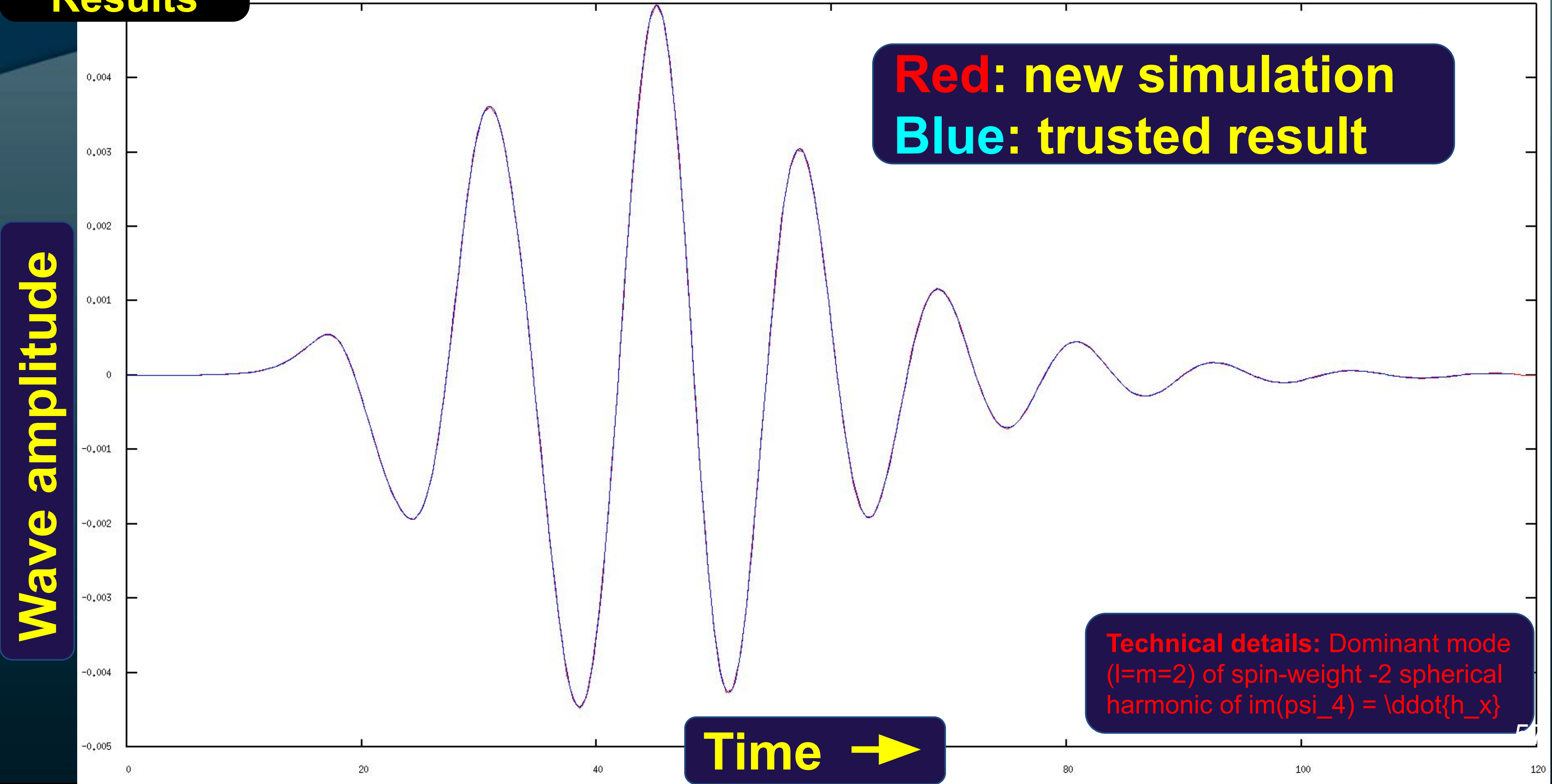
Will it work?!

Early 2020 Results



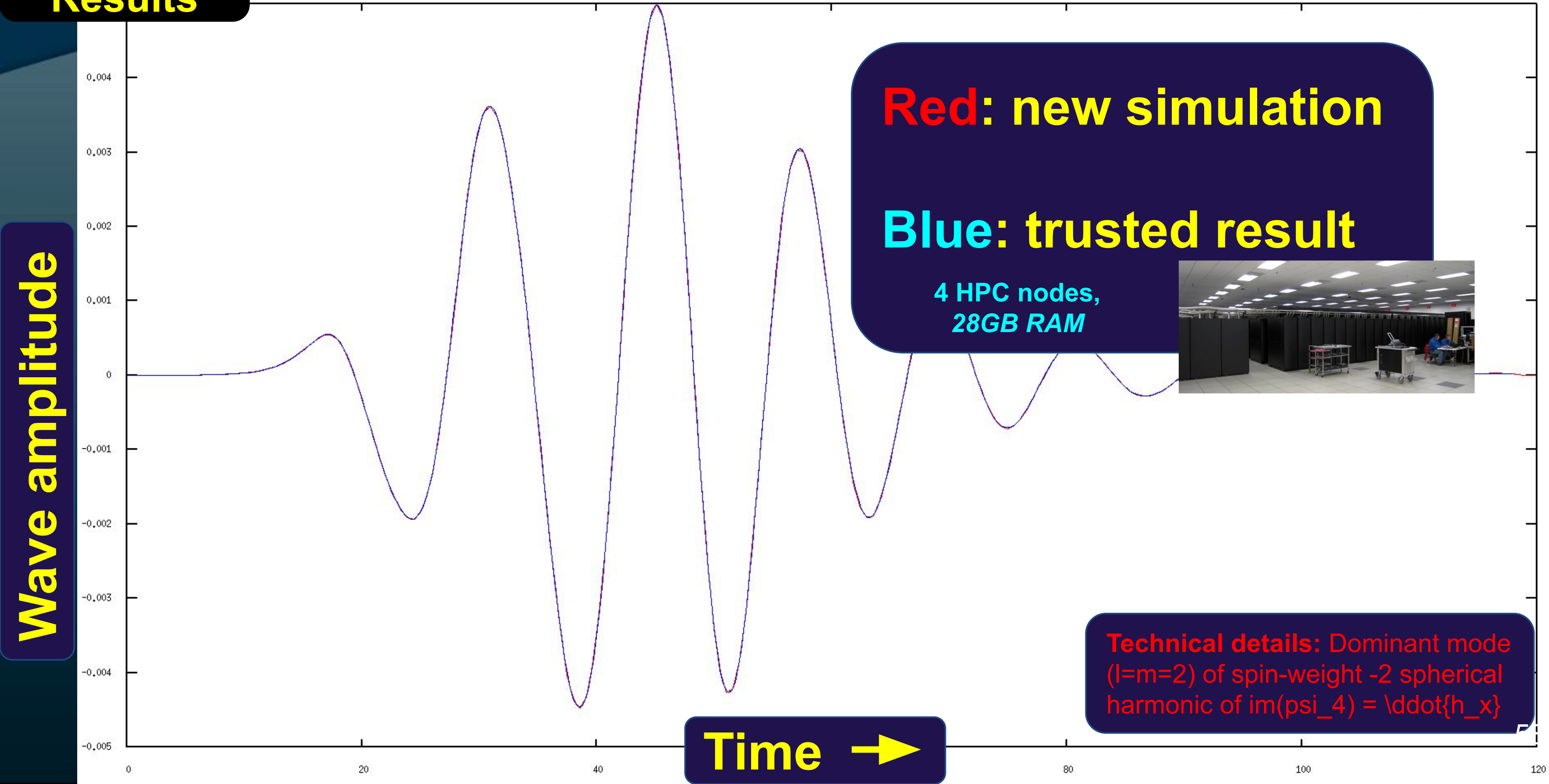
Early 2020 Results

Gravitational Wave Comparison



Early 2020 Results

Gravitational Wave Comparison



Red: new simulation

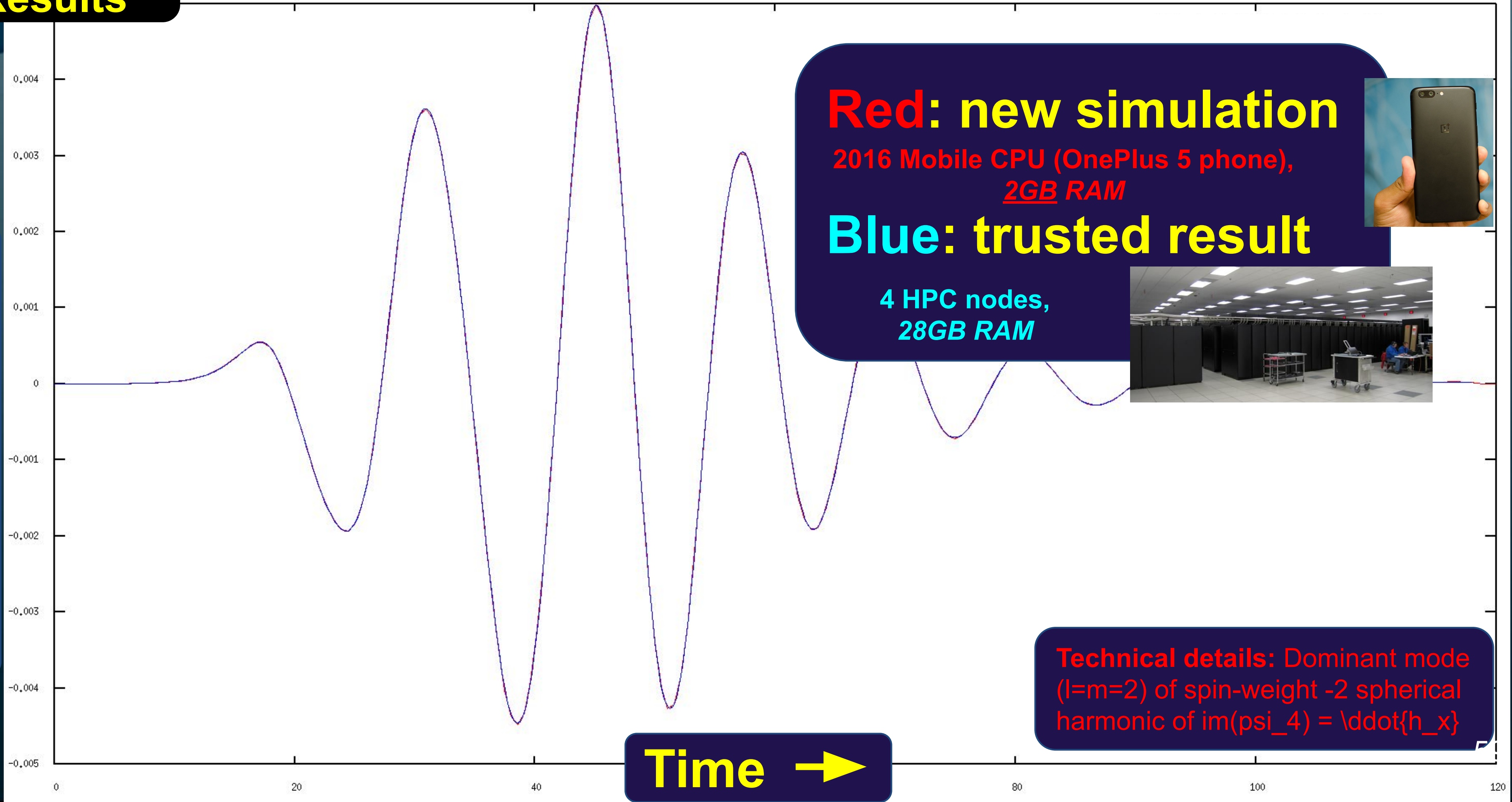
Blue: trusted result

4 HPC nodes,
28GB RAM

Technical details: Dominant mode (l=m=2) of spin-weight -2 spherical harmonic of im(psi_4) = \ddot{h_x}

Early 2020 Results

Gravitational Wave Comparison



Wave amplitude

Time →

Red: new simulation

2016 Mobile CPU (OnePlus 5 phone), *2GB RAM*

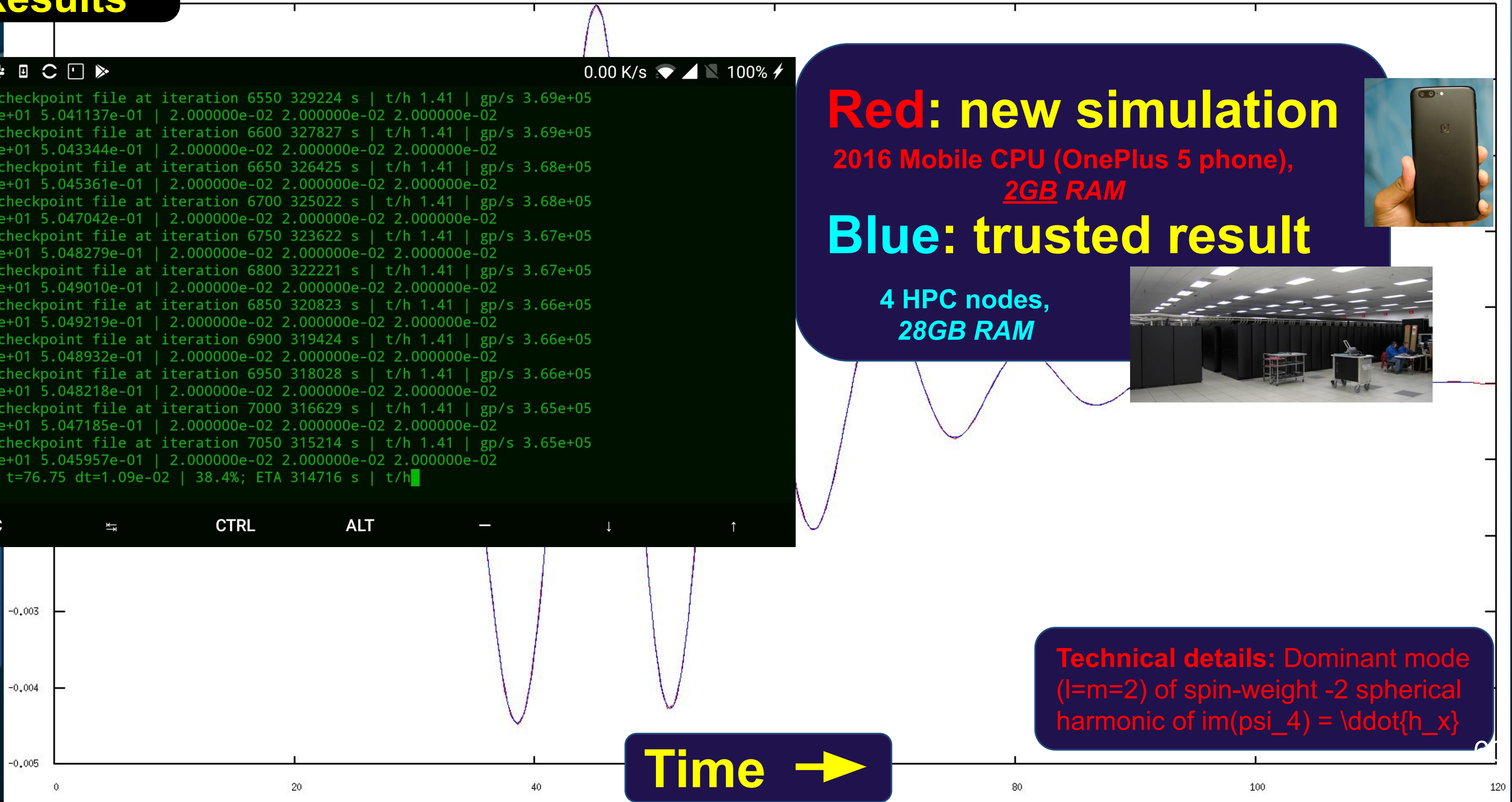
Blue: trusted result

4 HPC nodes, *28GB RAM*

Technical details: Dominant mode (l=m=2) of spin-weight -2 spherical harmonic of im(psi_4) = \ddot{h_x}

Early 2020 Results

Gravitational Wave Comparison




```
16:29                                   0.00 K/s   100%
Writing checkpoint file at iteration 6550 329224 s | t/h 1.41 | gp/s 3.69e+05
7.114185e+01 5.041137e-01 | 2.000000e-02 2.000000e-02 2.000000e-02
Writing checkpoint file at iteration 6600 327827 s | t/h 1.41 | gp/s 3.69e+05
7.168492e+01 5.043344e-01 | 2.000000e-02 2.000000e-02 2.000000e-02
Writing checkpoint file at iteration 6650 326425 s | t/h 1.41 | gp/s 3.68e+05
7.222799e+01 5.045361e-01 | 2.000000e-02 2.000000e-02 2.000000e-02
Writing checkpoint file at iteration 6700 325022 s | t/h 1.41 | gp/s 3.68e+05
7.277105e+01 5.047042e-01 | 2.000000e-02 2.000000e-02 2.000000e-02
Writing checkpoint file at iteration 6750 323622 s | t/h 1.41 | gp/s 3.67e+05
7.331412e+01 5.048279e-01 | 2.000000e-02 2.000000e-02 2.000000e-02
Writing checkpoint file at iteration 6800 322221 s | t/h 1.41 | gp/s 3.67e+05
7.385719e+01 5.049010e-01 | 2.000000e-02 2.000000e-02 2.000000e-02
Writing checkpoint file at iteration 6850 320823 s | t/h 1.41 | gp/s 3.66e+05
7.440026e+01 5.049219e-01 | 2.000000e-02 2.000000e-02 2.000000e-02
Writing checkpoint file at iteration 6900 319424 s | t/h 1.41 | gp/s 3.66e+05
7.494332e+01 5.048932e-01 | 2.000000e-02 2.000000e-02 2.000000e-02
Writing checkpoint file at iteration 6950 318028 s | t/h 1.41 | gp/s 3.66e+05
7.548639e+01 5.048218e-01 | 2.000000e-02 2.000000e-02 2.000000e-02
Writing checkpoint file at iteration 7000 316629 s | t/h 1.41 | gp/s 3.65e+05
7.602946e+01 5.047185e-01 | 2.000000e-02 2.000000e-02 2.000000e-02
Writing checkpoint file at iteration 7050 315214 s | t/h 1.41 | gp/s 3.65e+05
7.657253e+01 5.045957e-01 | 2.000000e-02 2.000000e-02 2.000000e-02
It: 7066 t=76.75 dt=1.09e-02 | 38.4%; ETA 314716 s | t/h
ESC   CTRL   ALT   —   ↓   ↑
```

Red: new simulation

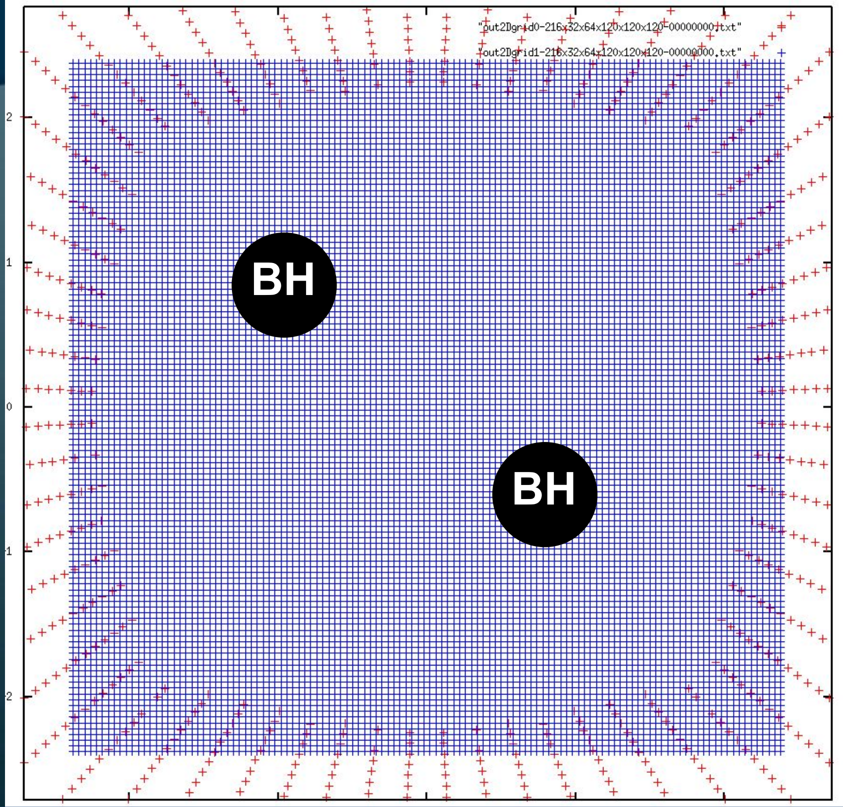
2016 Mobile CPU (OnePlus 5 phone), *2GB RAM*

Blue: trusted result

4 HPC nodes, *28GB RAM*

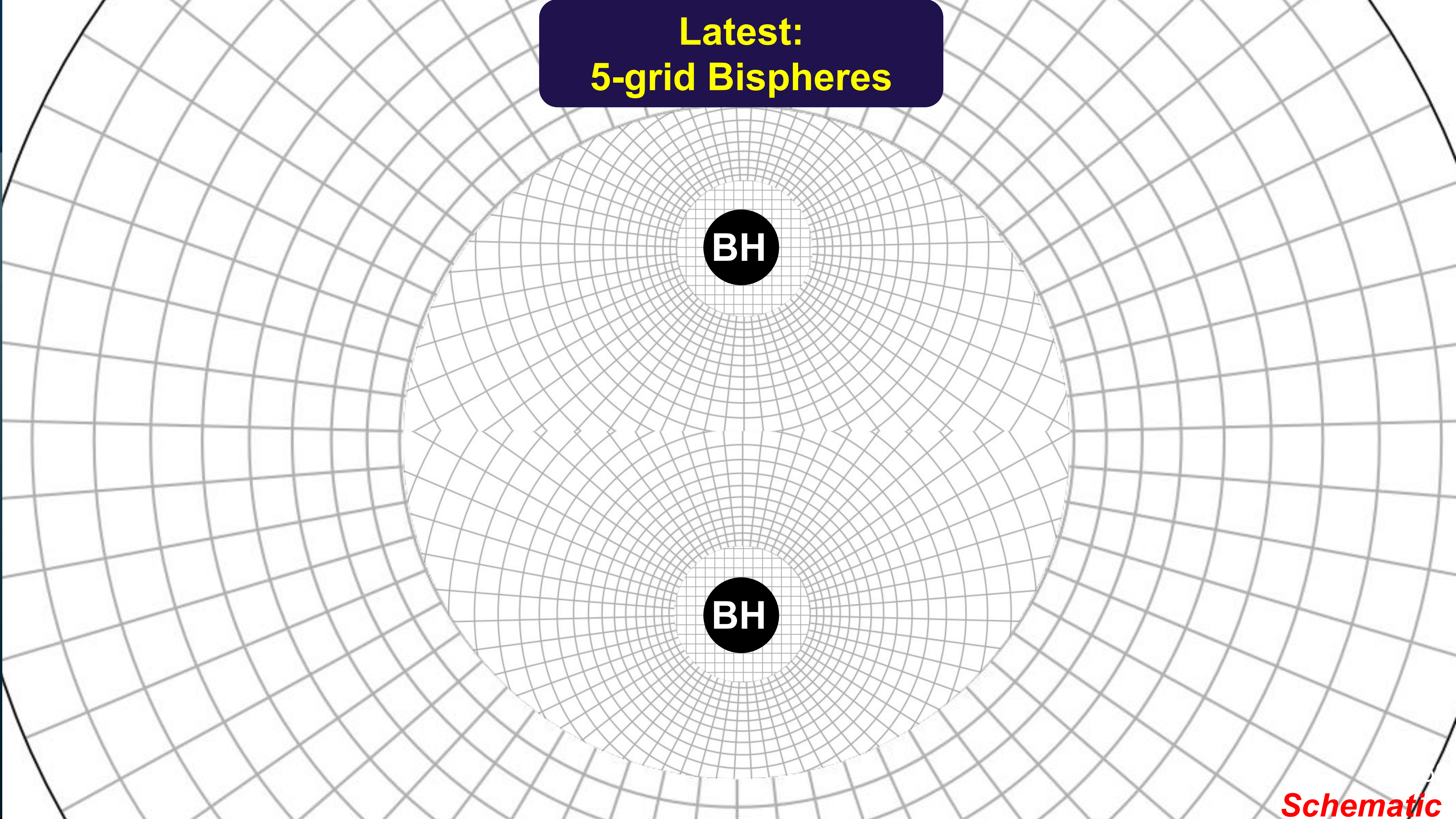
Technical details: Dominant mode (l=m=2) of spin-weight -2 spherical harmonic of im(psi_4) = \ddot{h_x}

Early 2020 Results

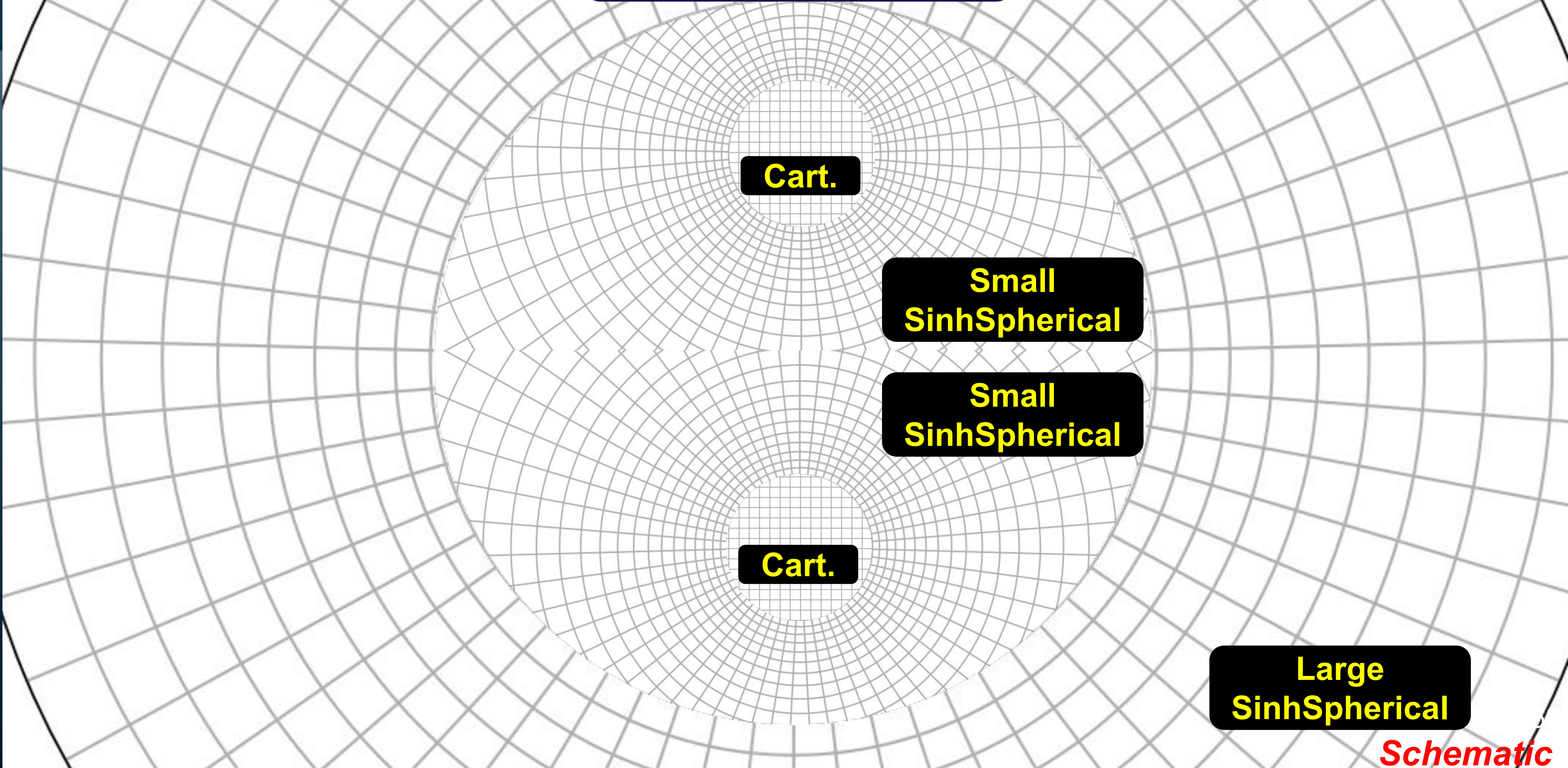


- Problem with this grid structure:
  - Only works well for two orbiting black holes ***very close* to merger**
  - Larger separations -- Cartesian grid too large -- too much memory!
  - Narrow Cartesian grid & rotate grids? Nope; resolution drop too large
- What to do?!

Latest:
5-grid Bispheres



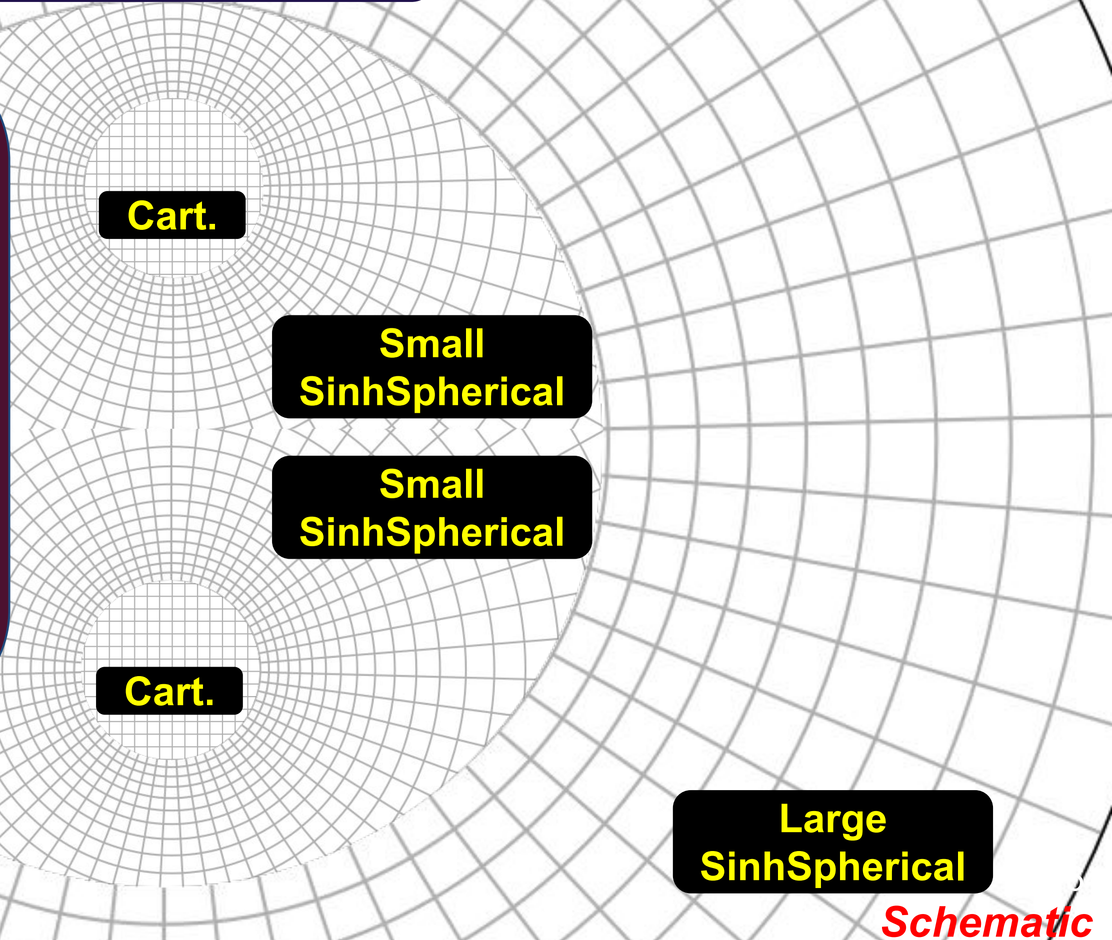
Latest: 5-grid Bispheres



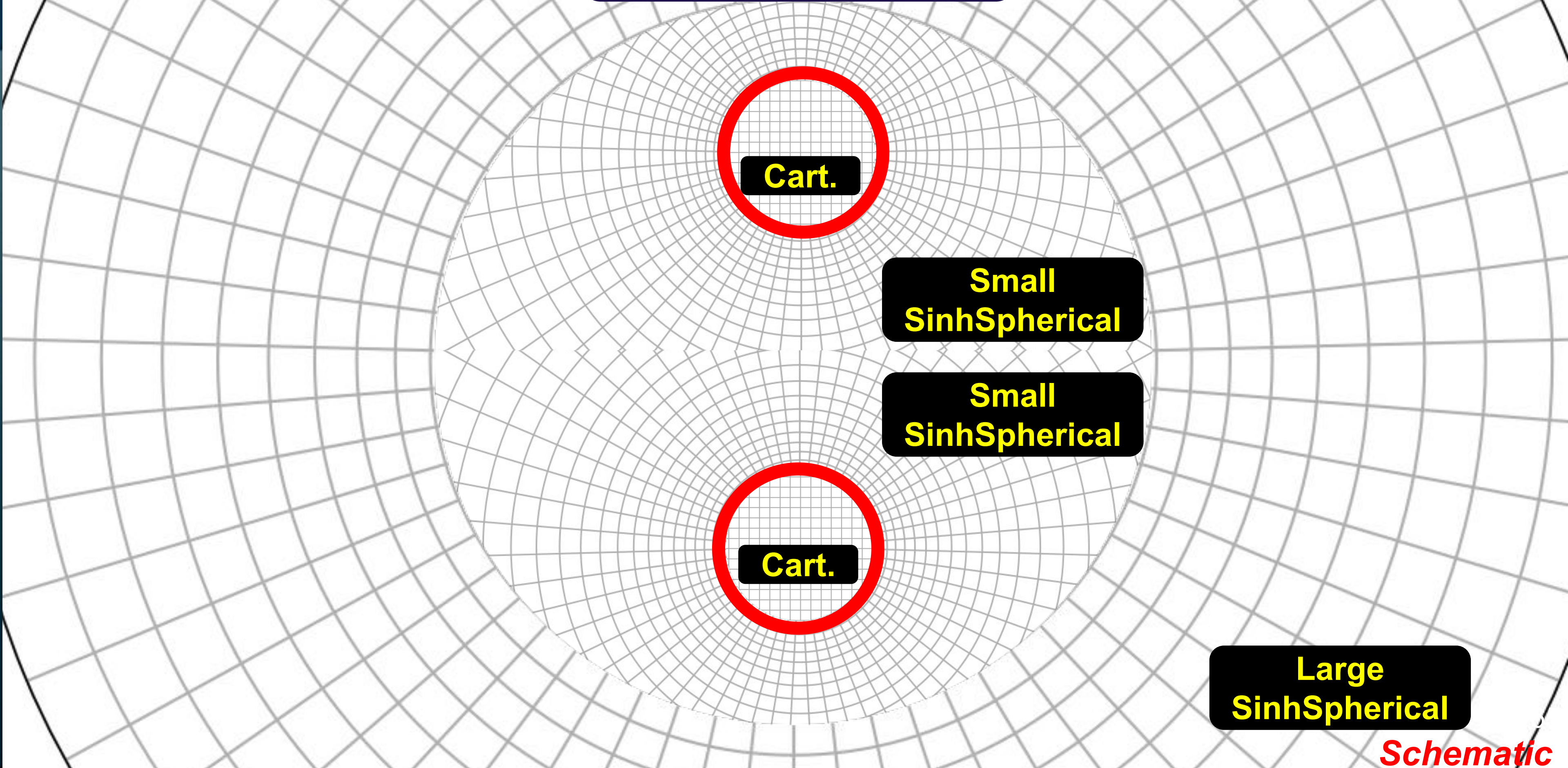
Latest: 5-grid Bispheres

Benefits

1. Cartesian grids maximize timestep where Sph grids would focus
2. *z*-axis collinear across all three SinhSpherical grids, Nphi fixed
   a. ⇒ 2D interpolations!
3. Large SinhSpherical *perfect* for GW extraction!
4. Low memory footprint
5. Intergrid surface area << Cartesian AMR ⇒ Could scale well!



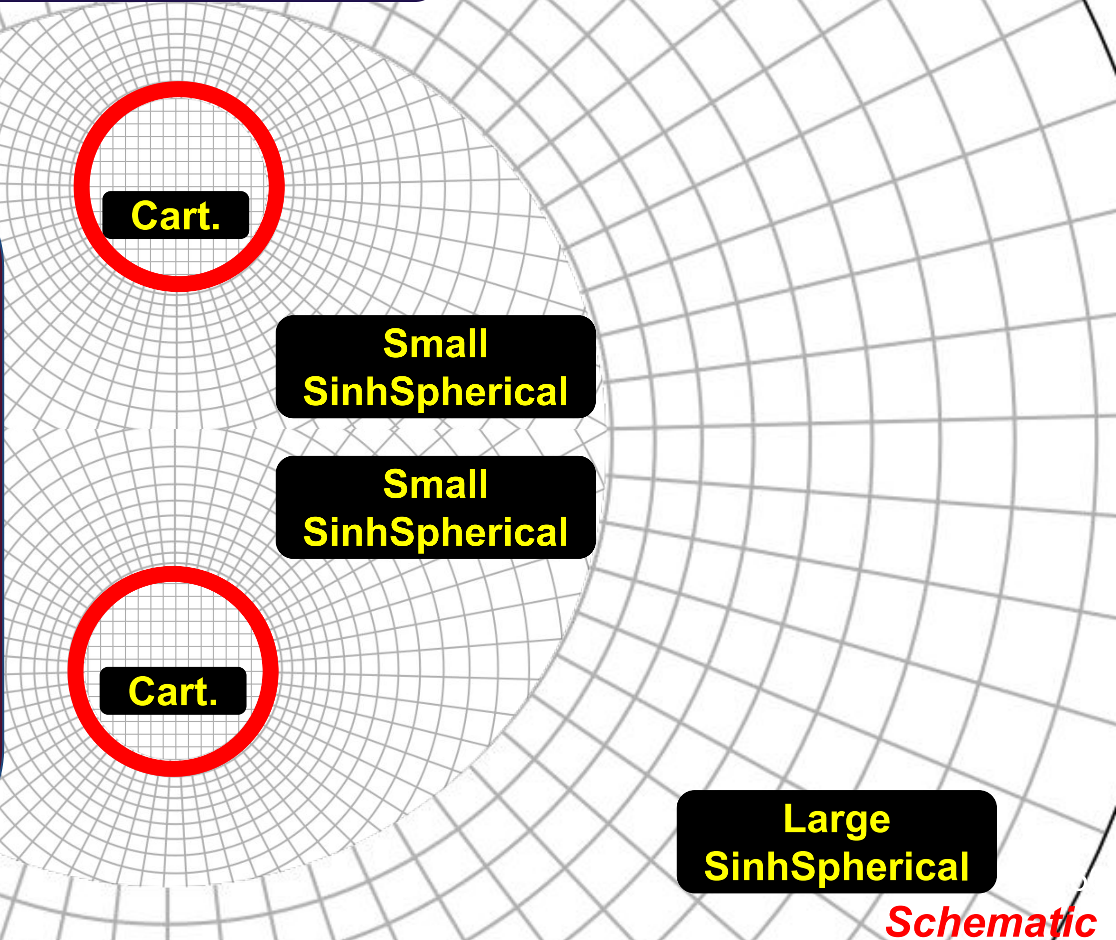
Latest: 5-grid Bispheres



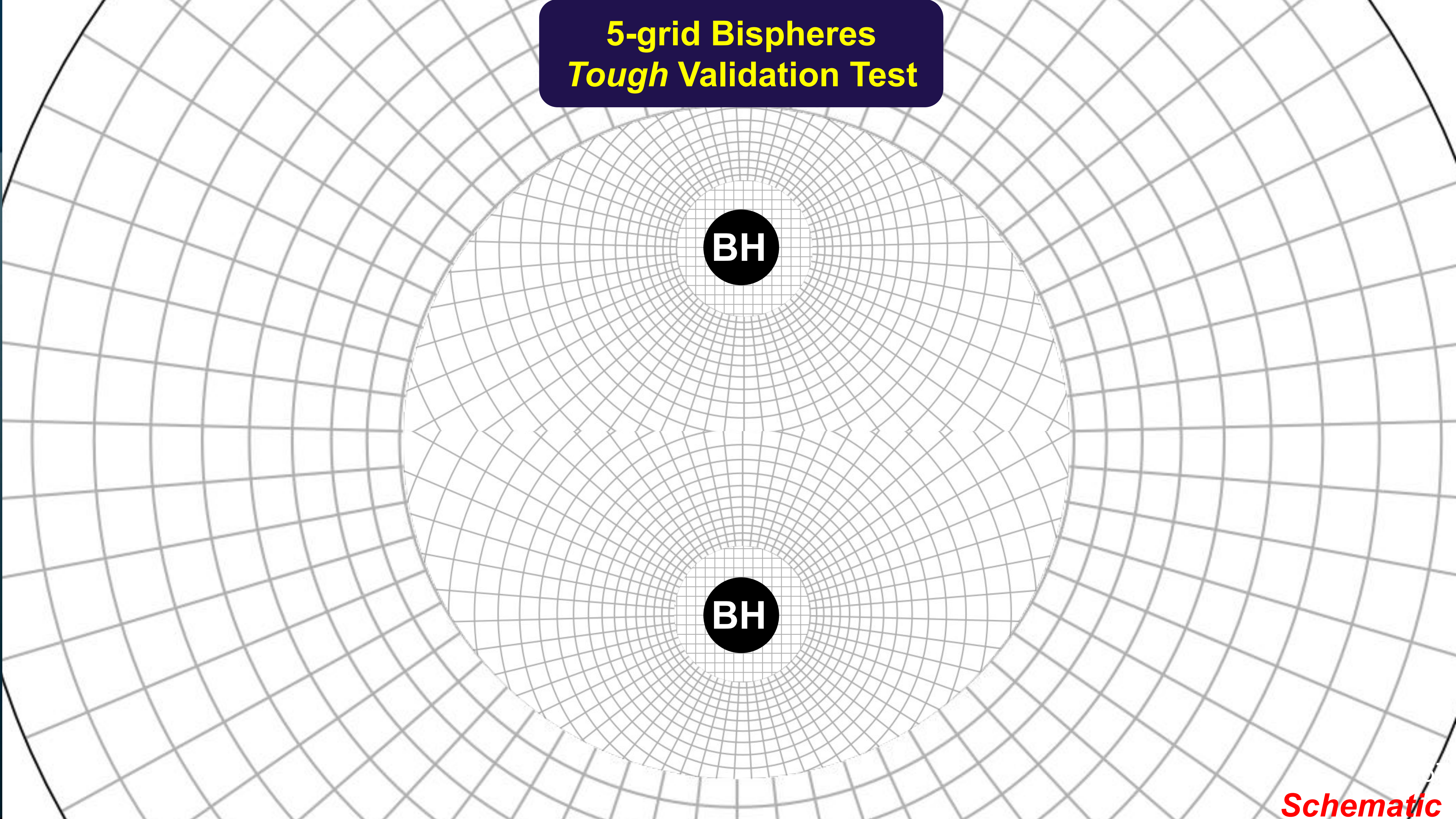
Latest:
5-grid Bispheres

<u>***Tough* Validation Test**</u>

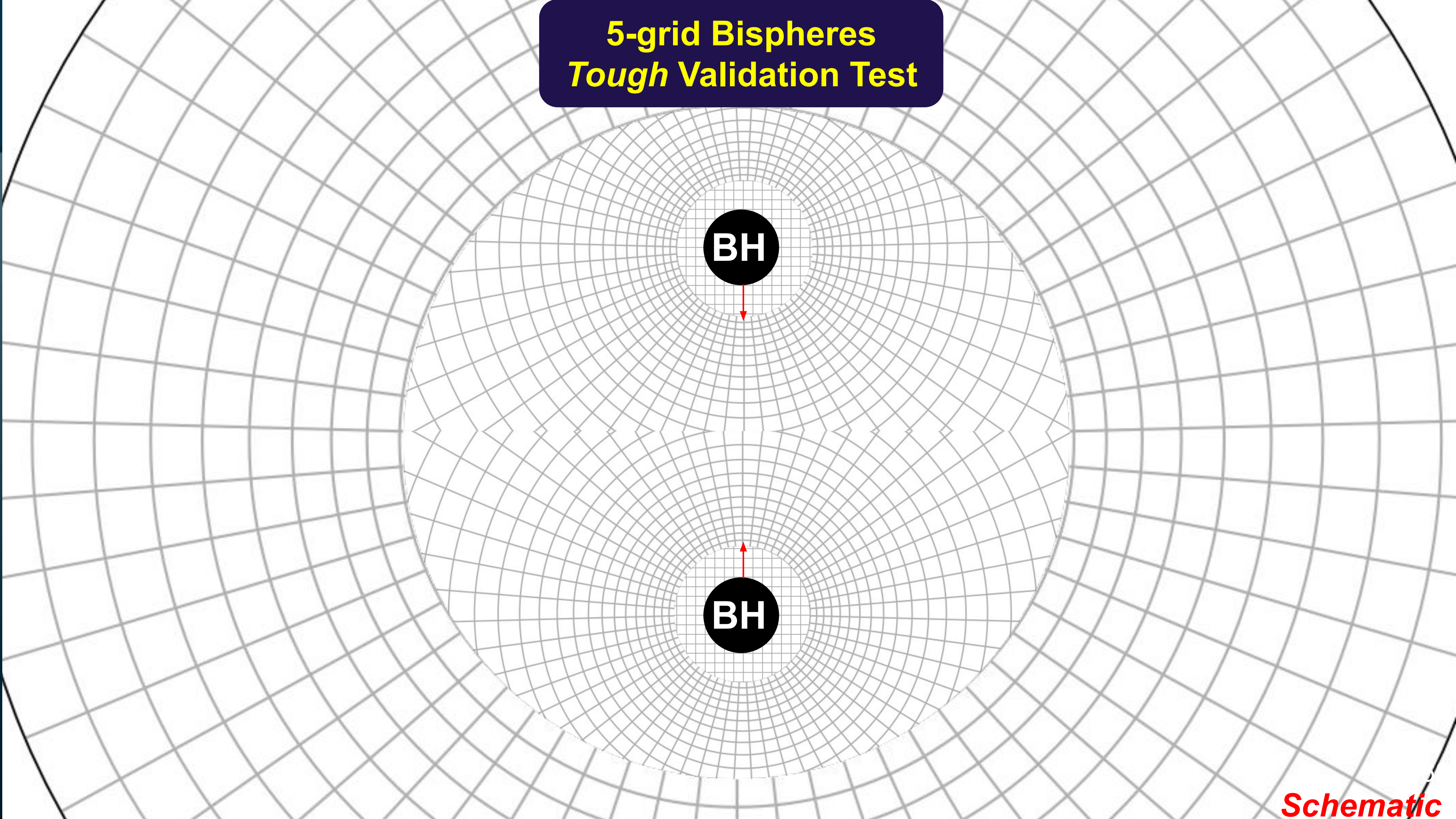
- Fix grids in place (static grids)
- Match resolutions at red circles
  - (SinhSpherical/Cartesian intergrid boundary)
- Increase resolution near CoM
- Release BHs from rest, allowing BHs to cross intergrid boundaries
- BH merger remnant sits at intergrid boundary (SinhSpherical/SinhSpherical)
- Gravitational waves?!



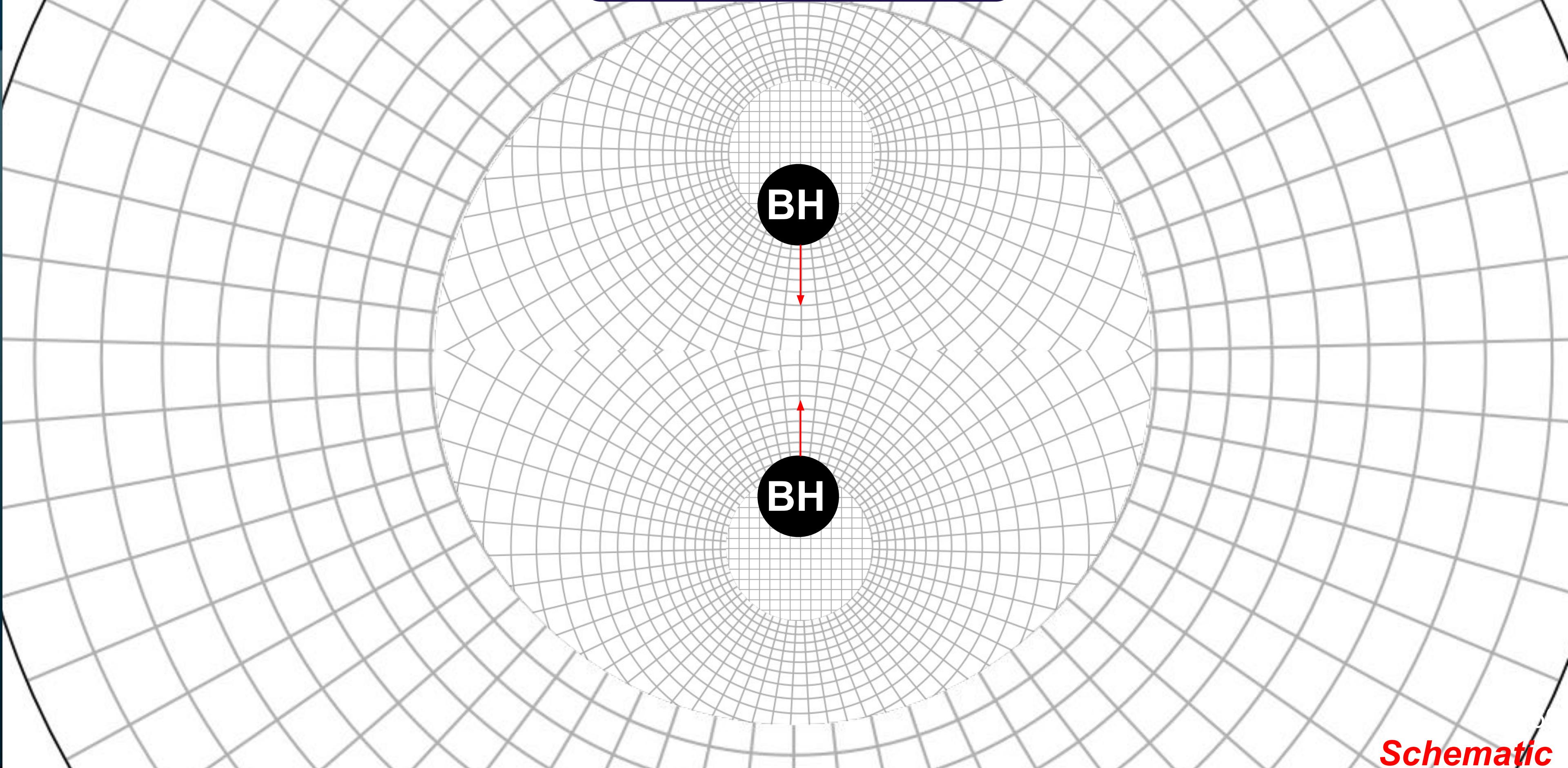
5-grid Bispheres
Tough Validation Test



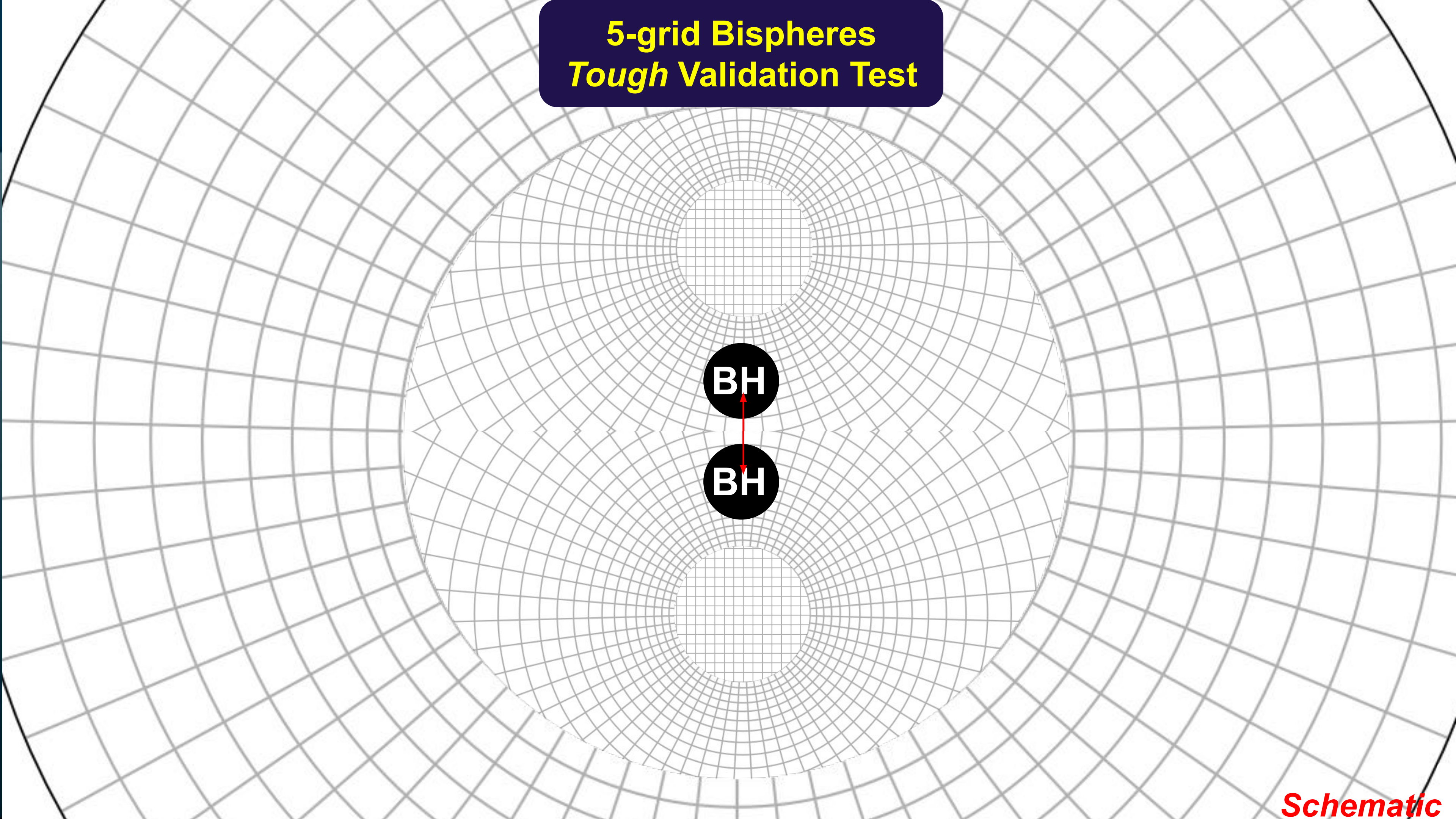
5-grid Bispheres
Tough Validation Test



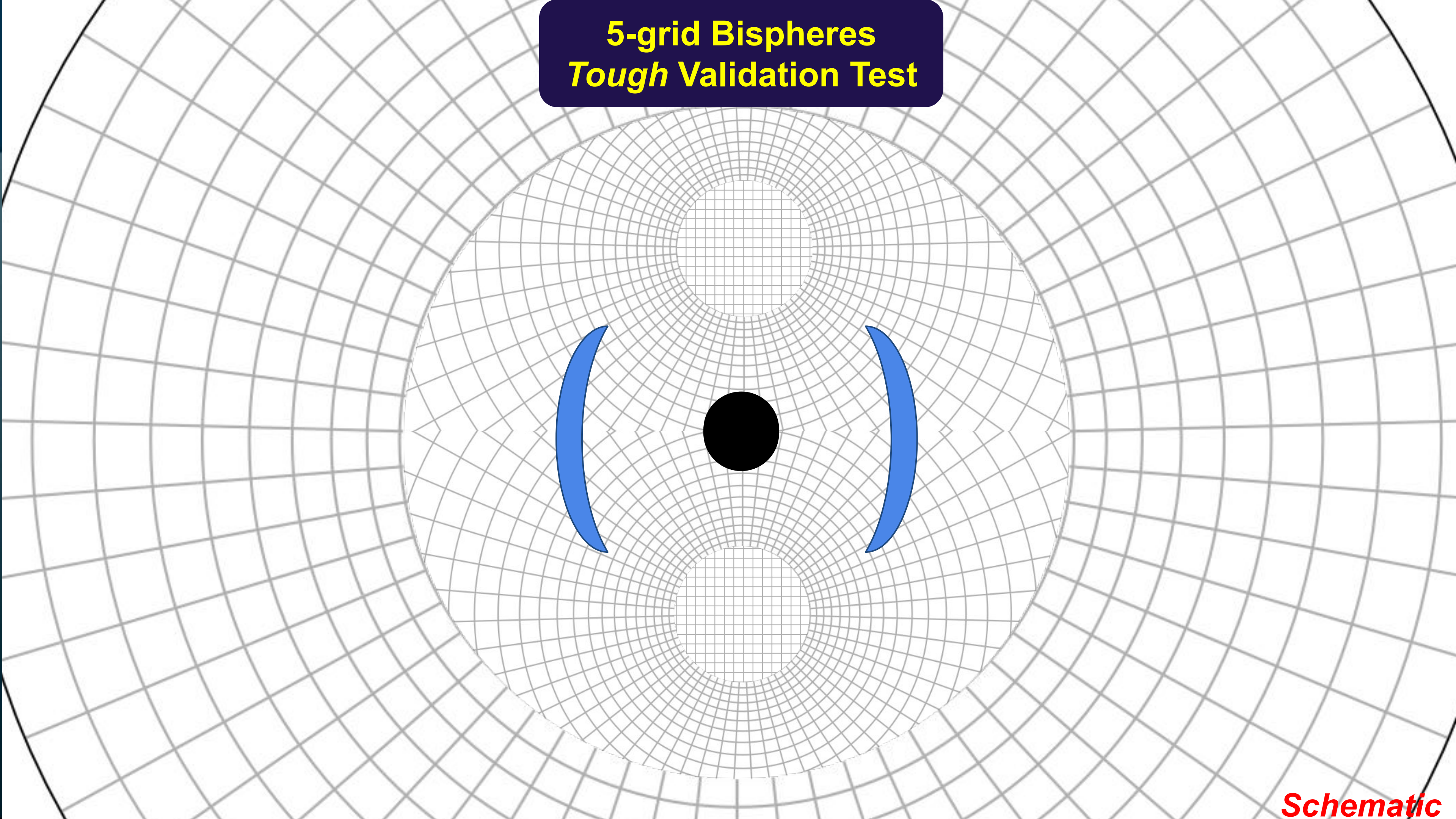
5-grid Bispheres
Tough Validation Test



5-grid Bispheres
Tough Validation Test

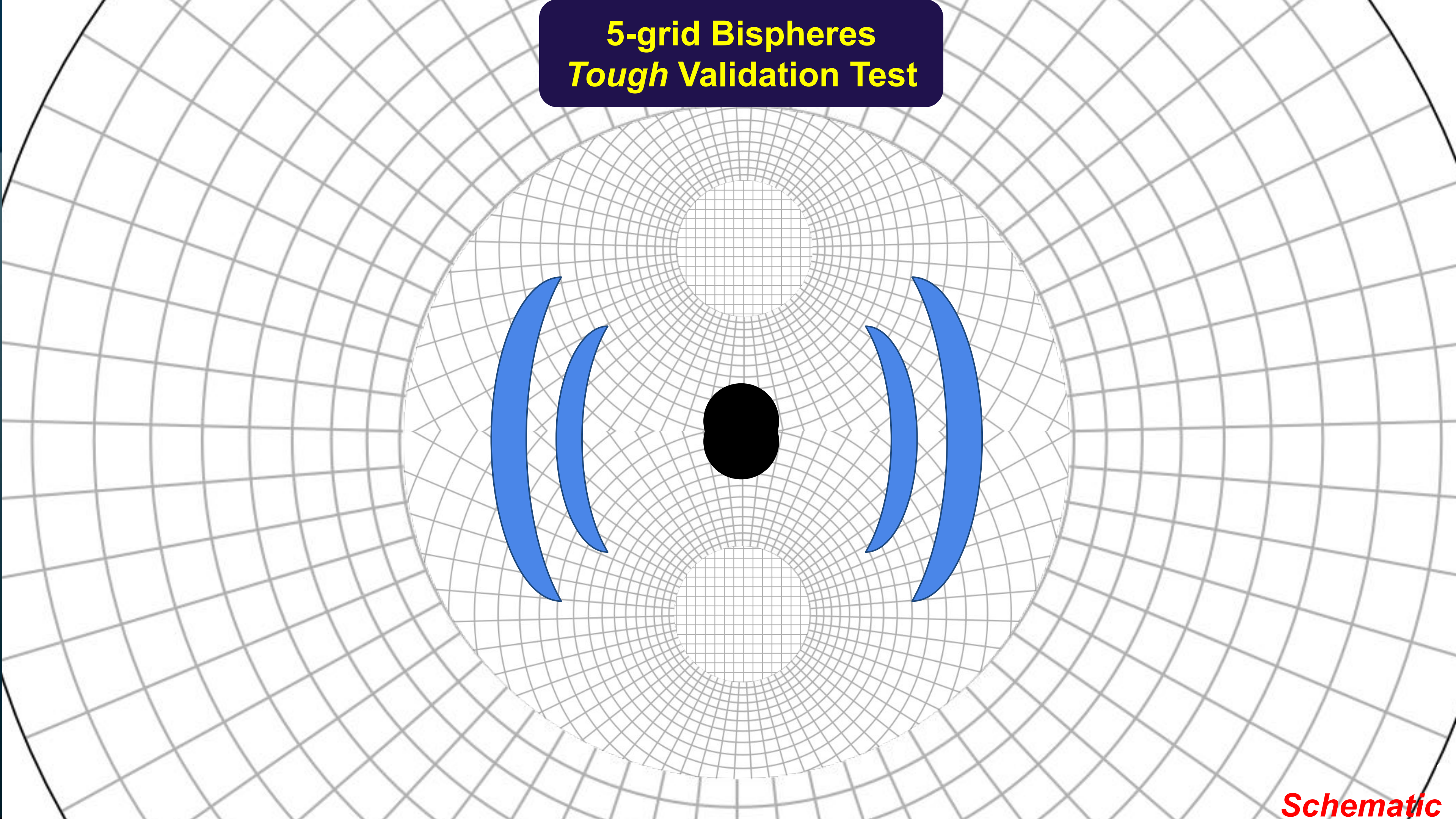


5-grid Bispheres
Tough Validation Test

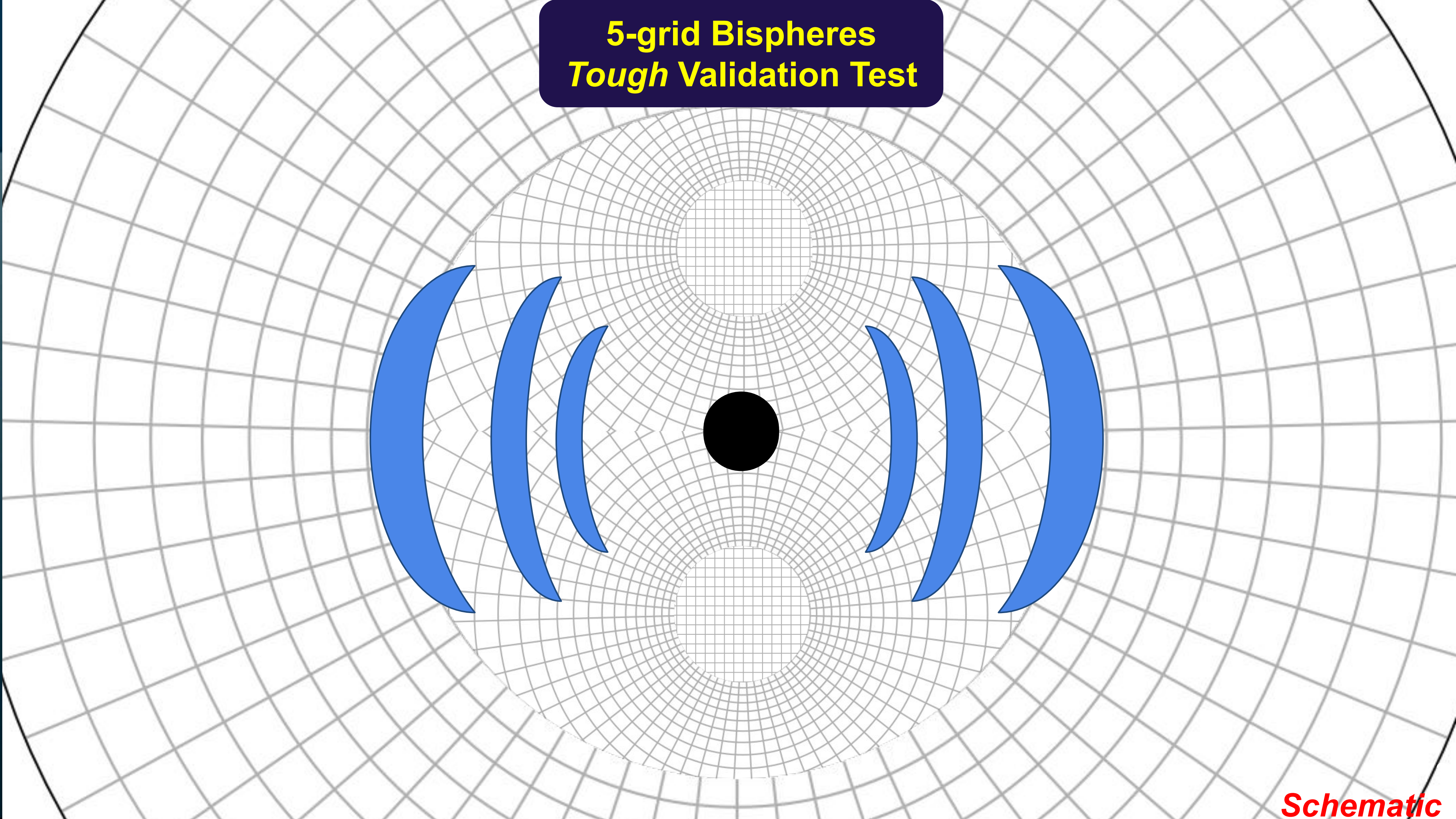


Schematic

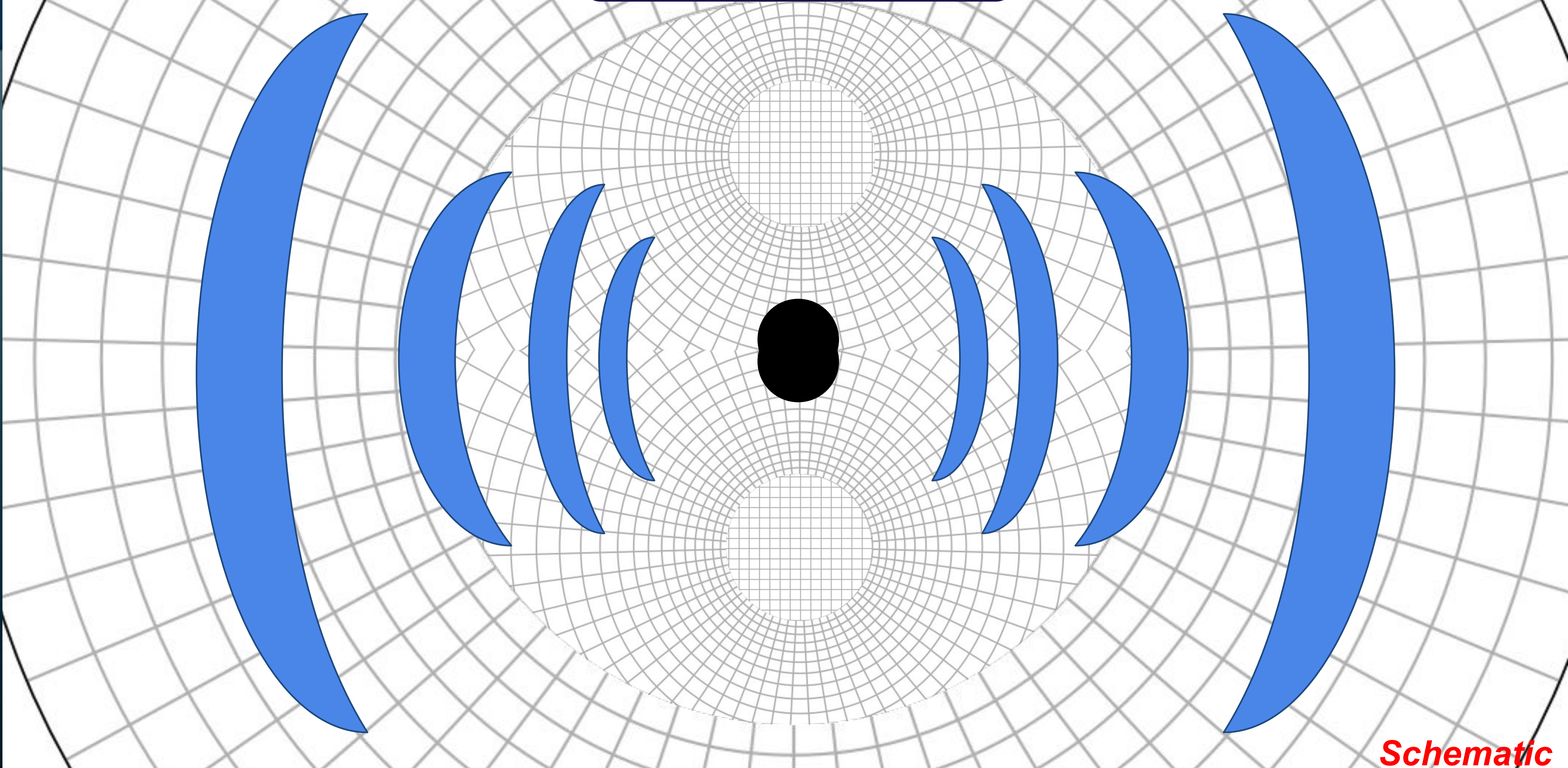
5-grid Bispheres *Tough* Validation Test



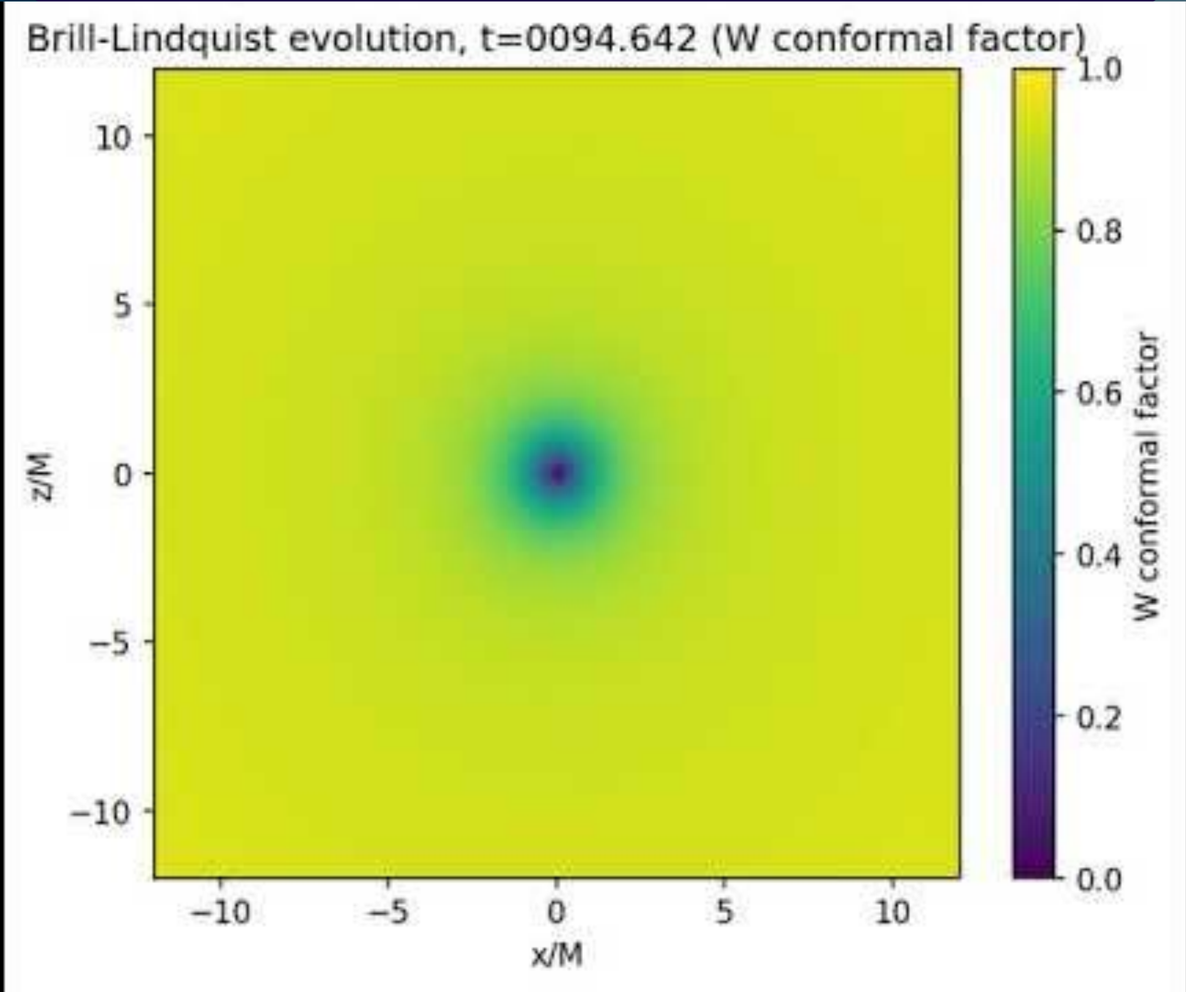
5-grid Bispheres
Tough Validation Test



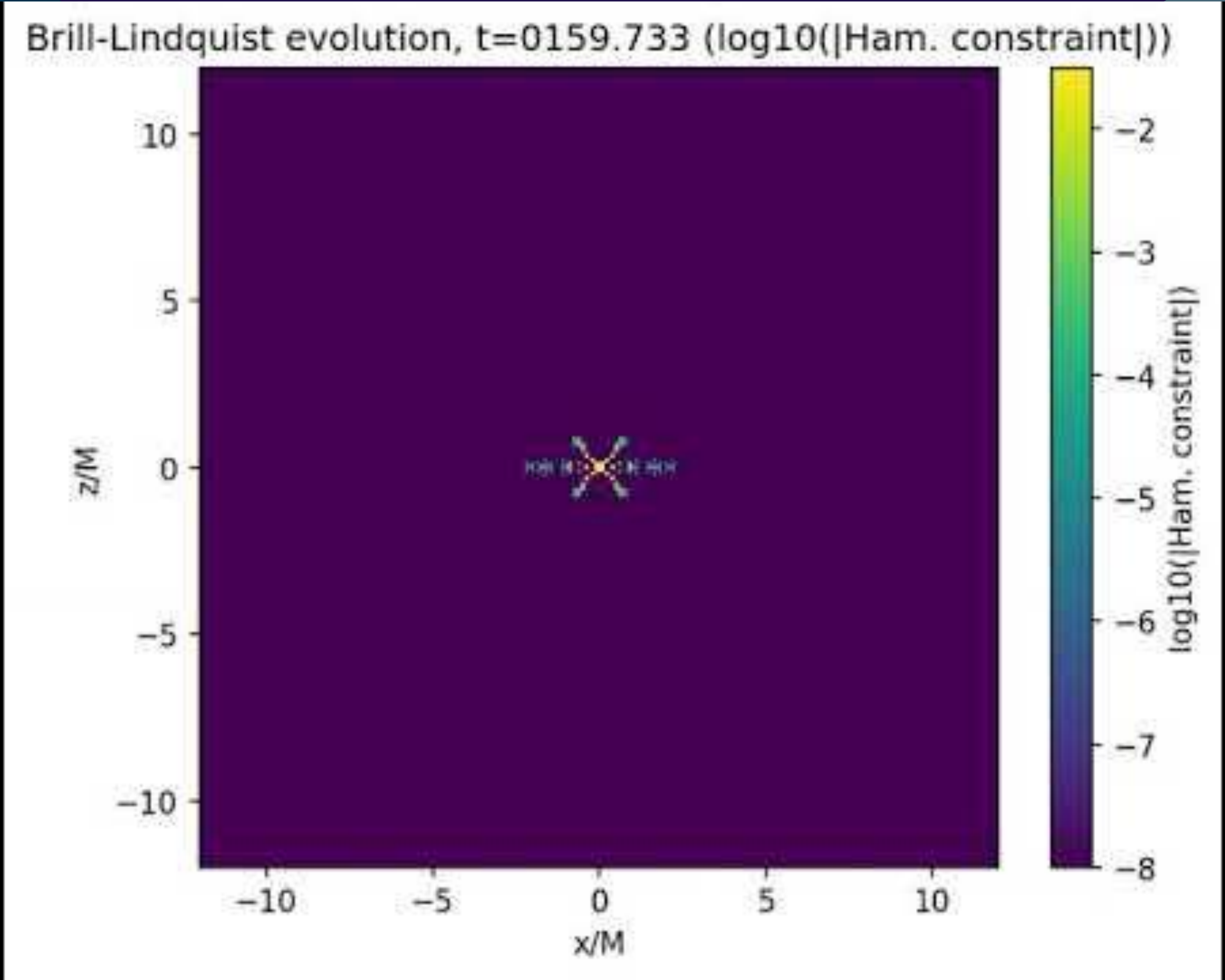
5-grid Bispheres
Tough Validation Test



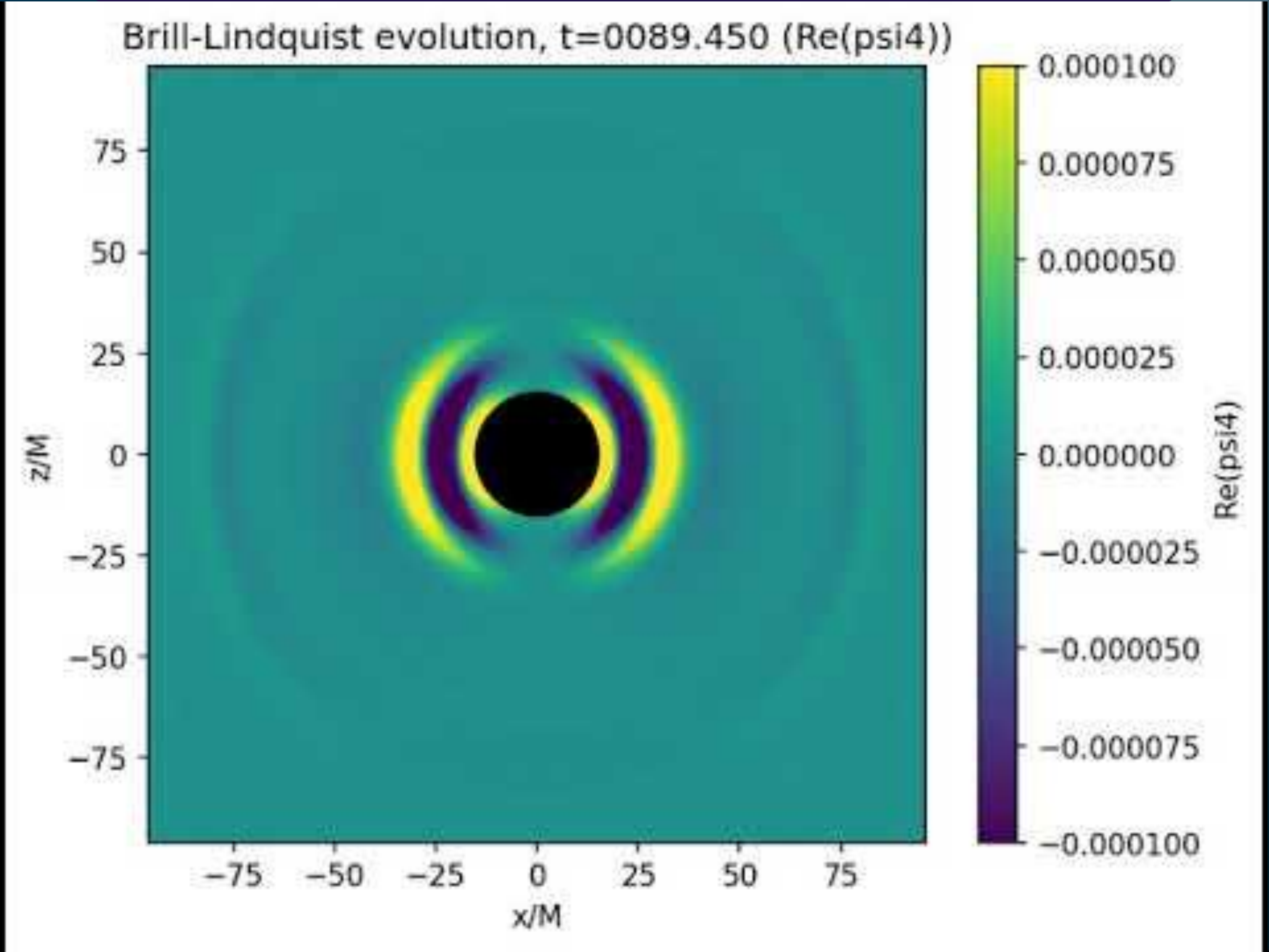
5-grid Bispheres: Two-BH, Head-on Collision



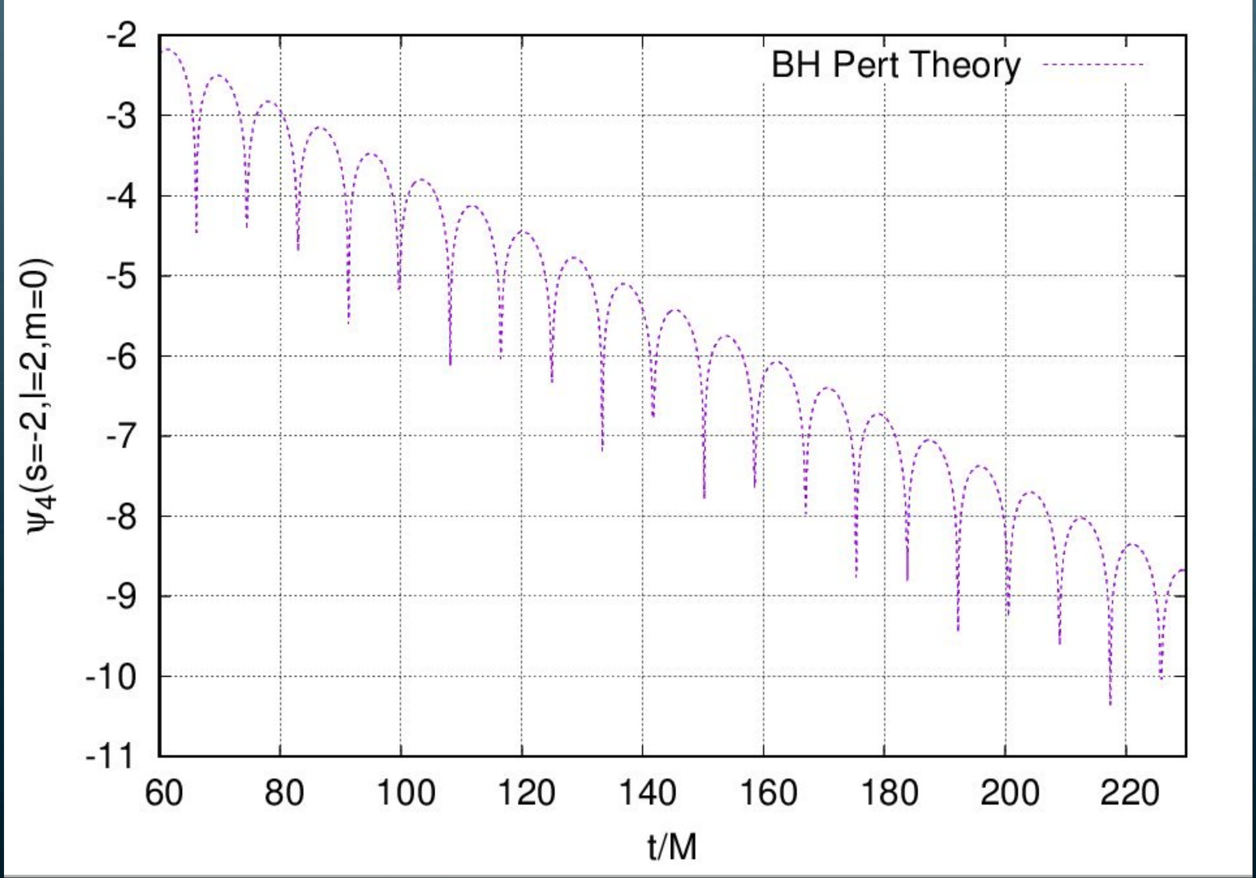
5-grid Bispheres: Two-BH, Head-on Collision



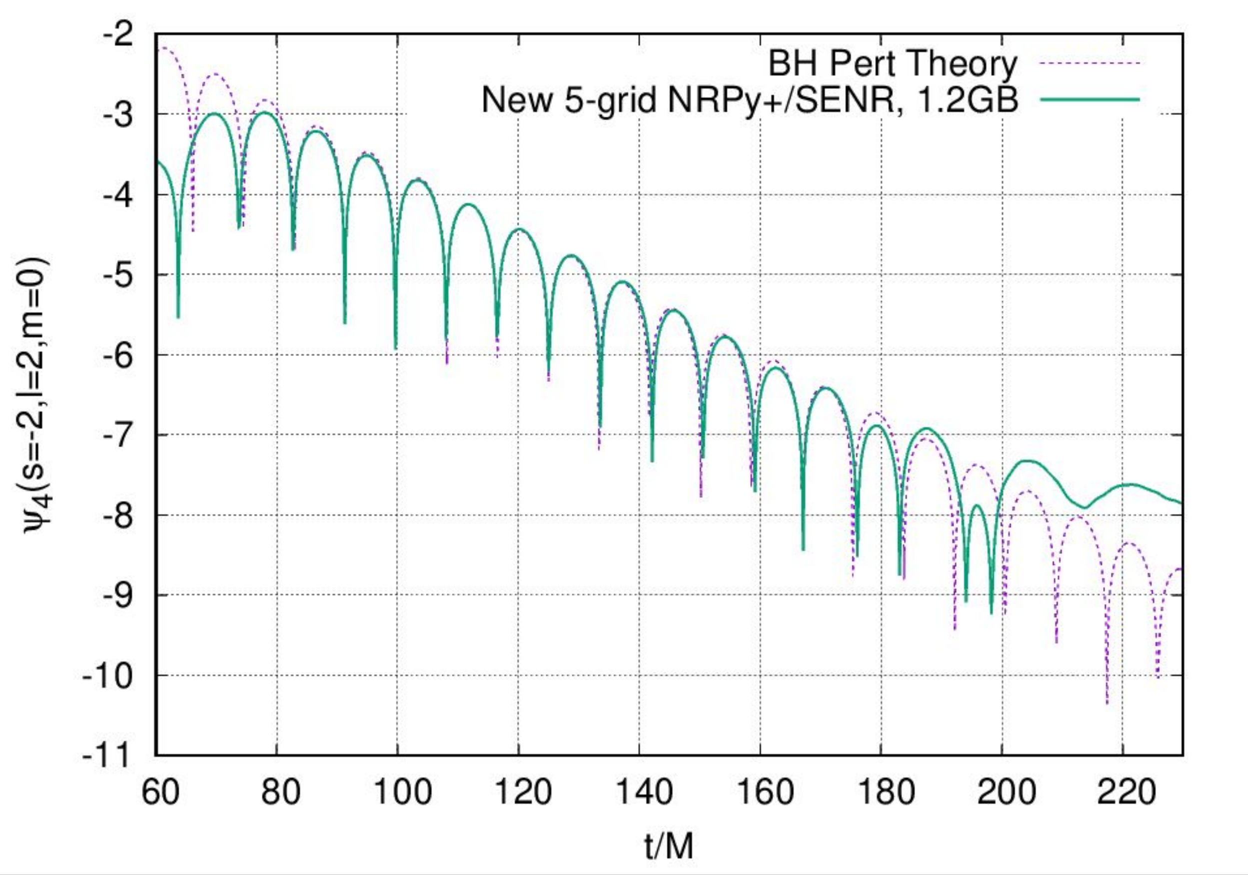
5-grid Bispheres: Two-BH, Head-on Collision



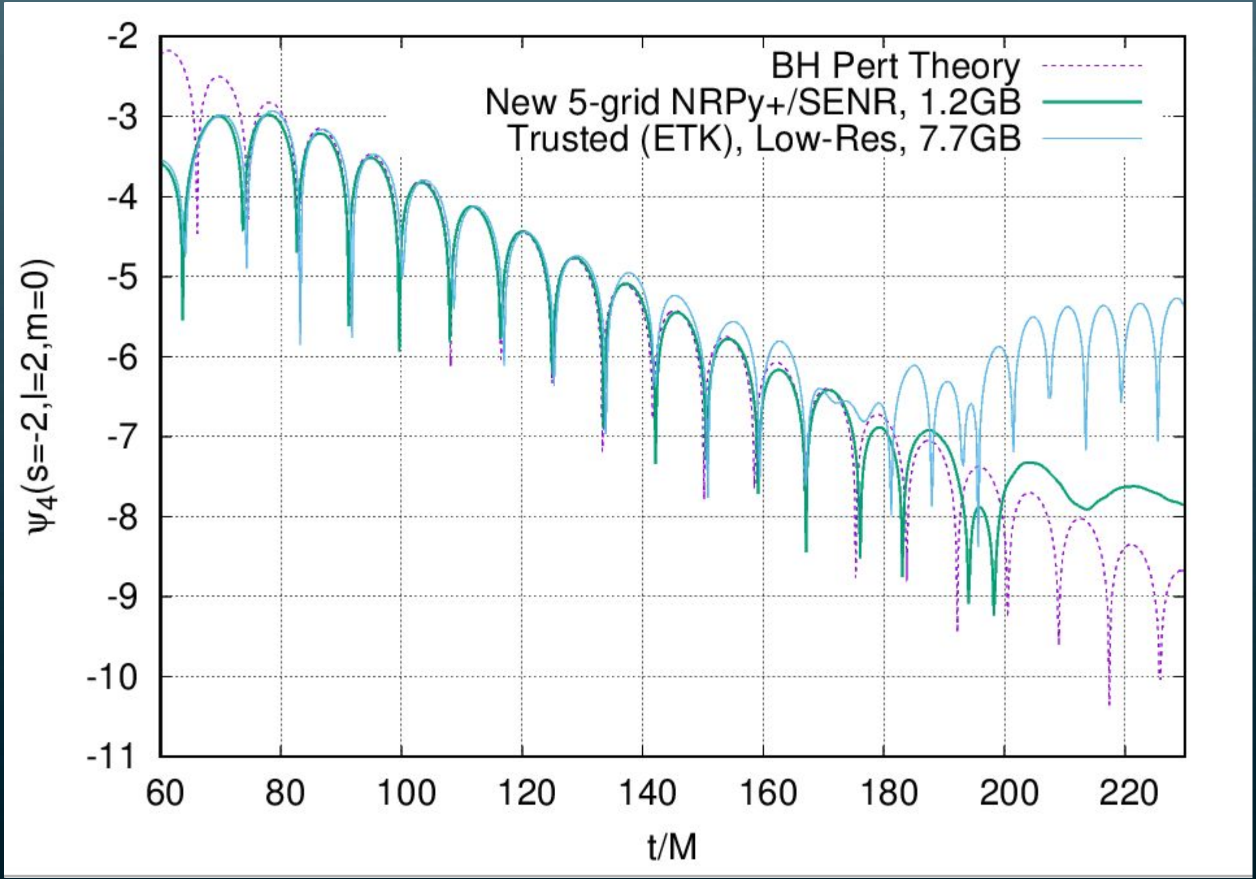
5-grid Bispheres Head-on Results



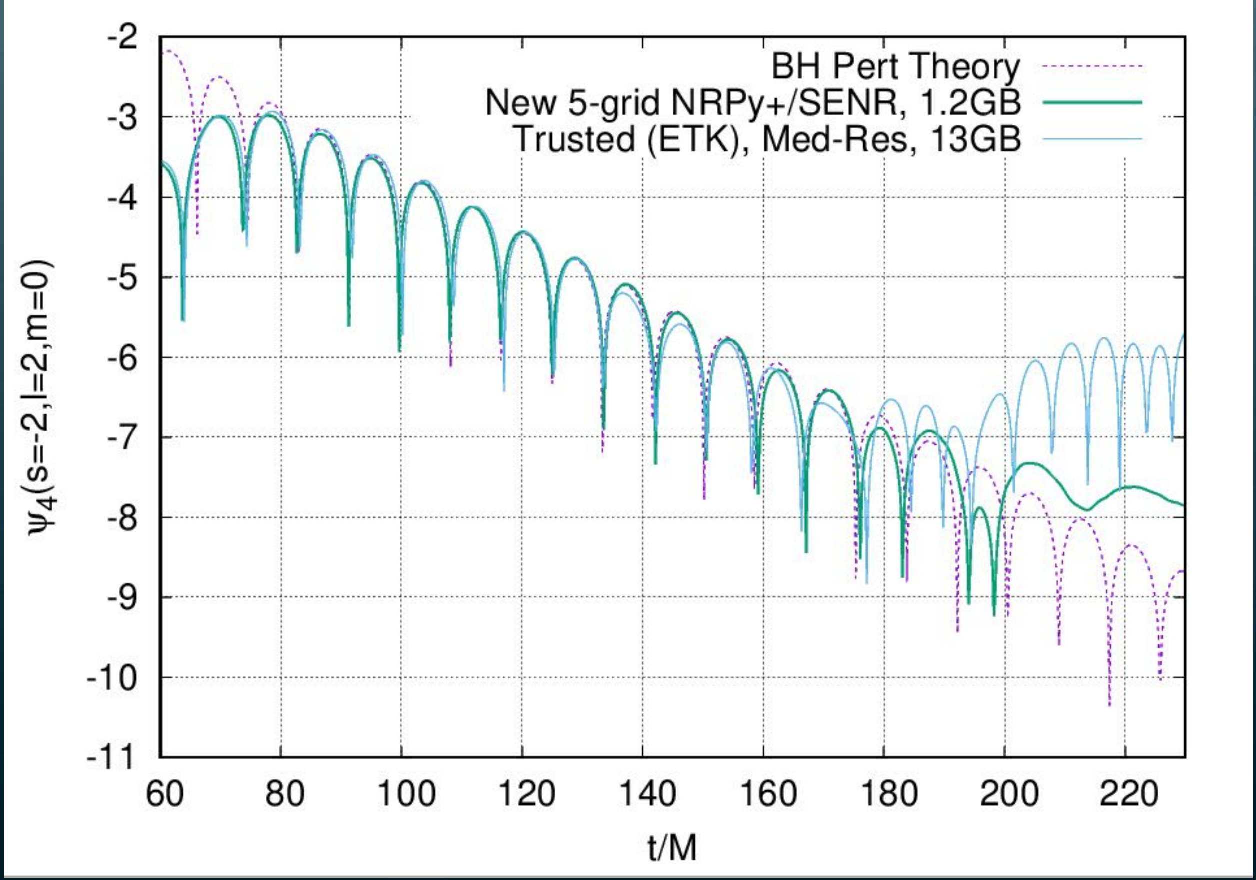
5-grid Bispheres Head-on Results



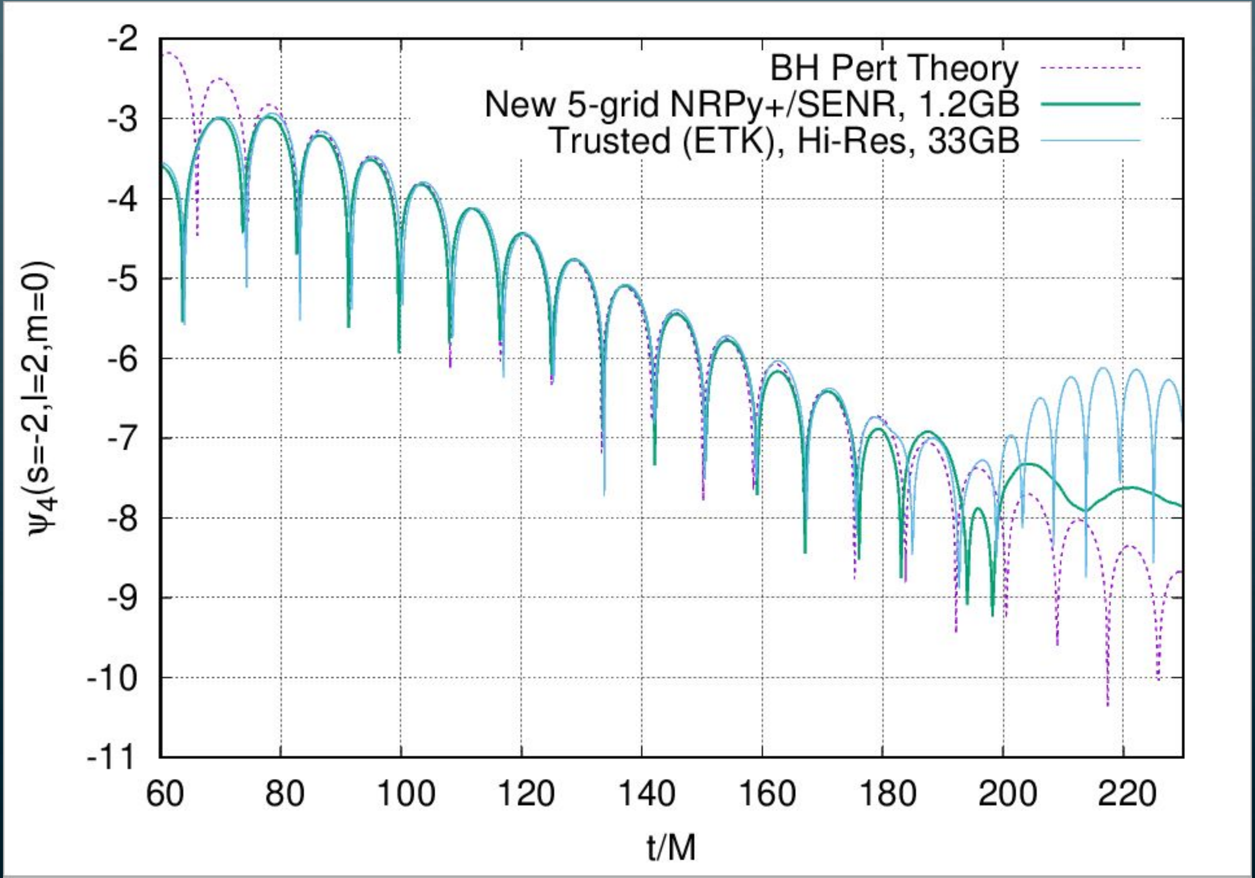
5-grid Bispheres Head-on Results



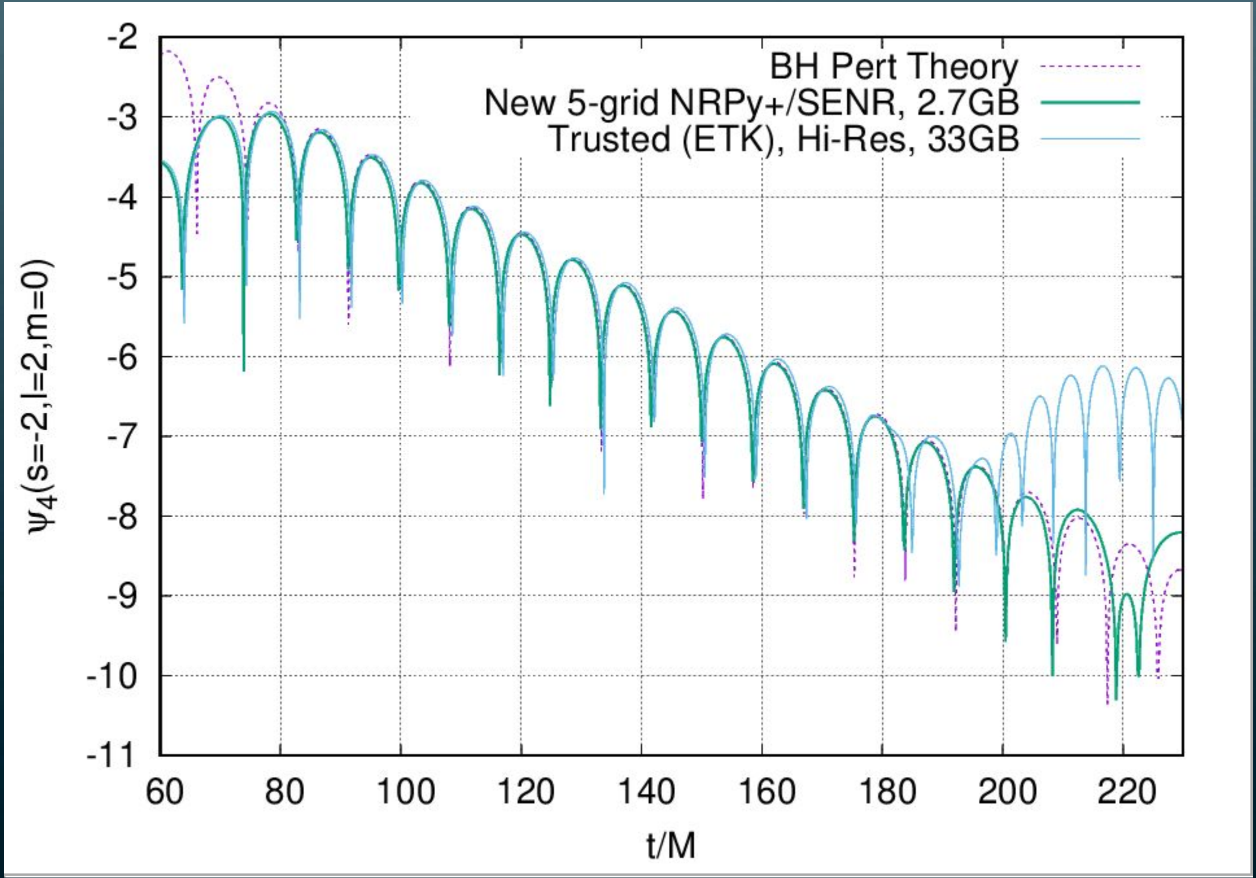
5-grid Bispheres Head-on Results



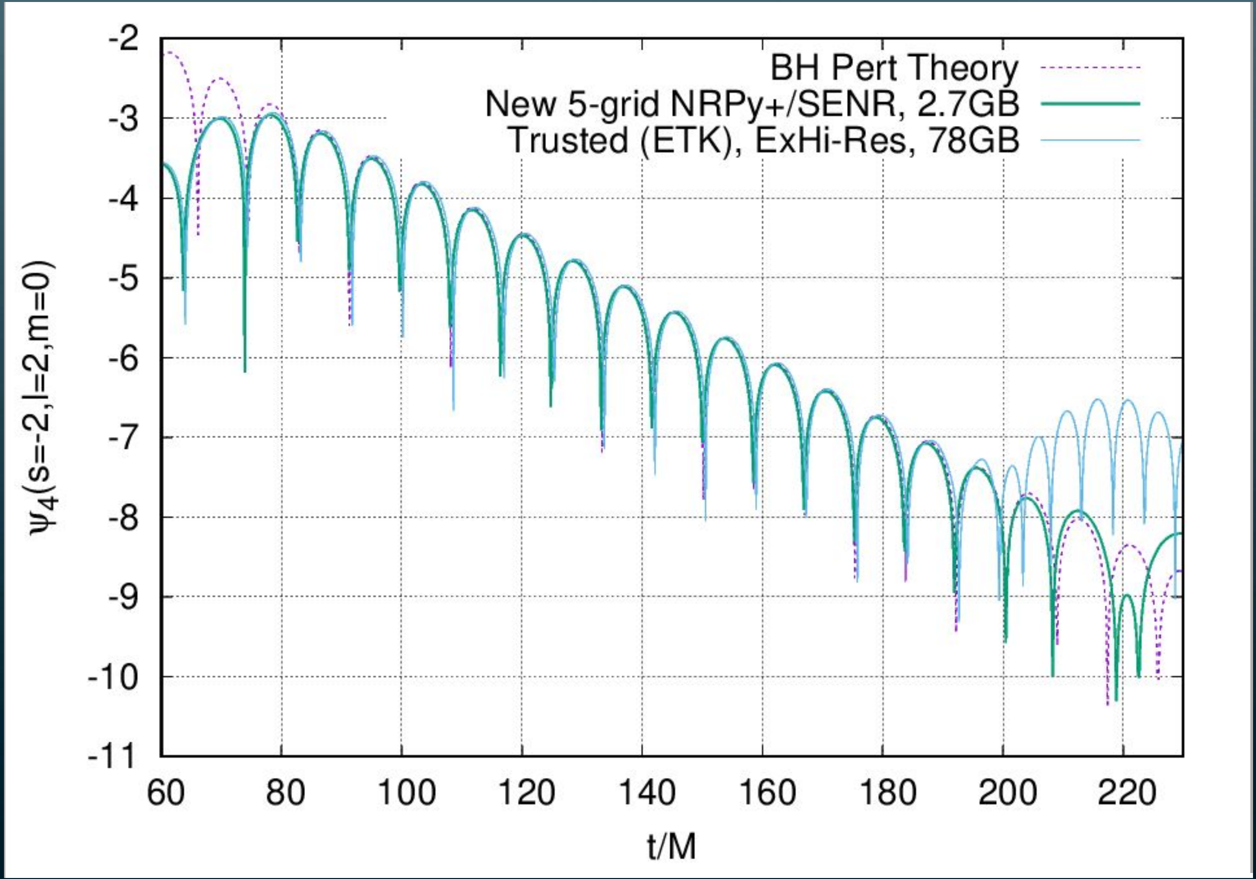
5-grid Bispheres Head-on Results



5-grid Bispheres Head-on Results



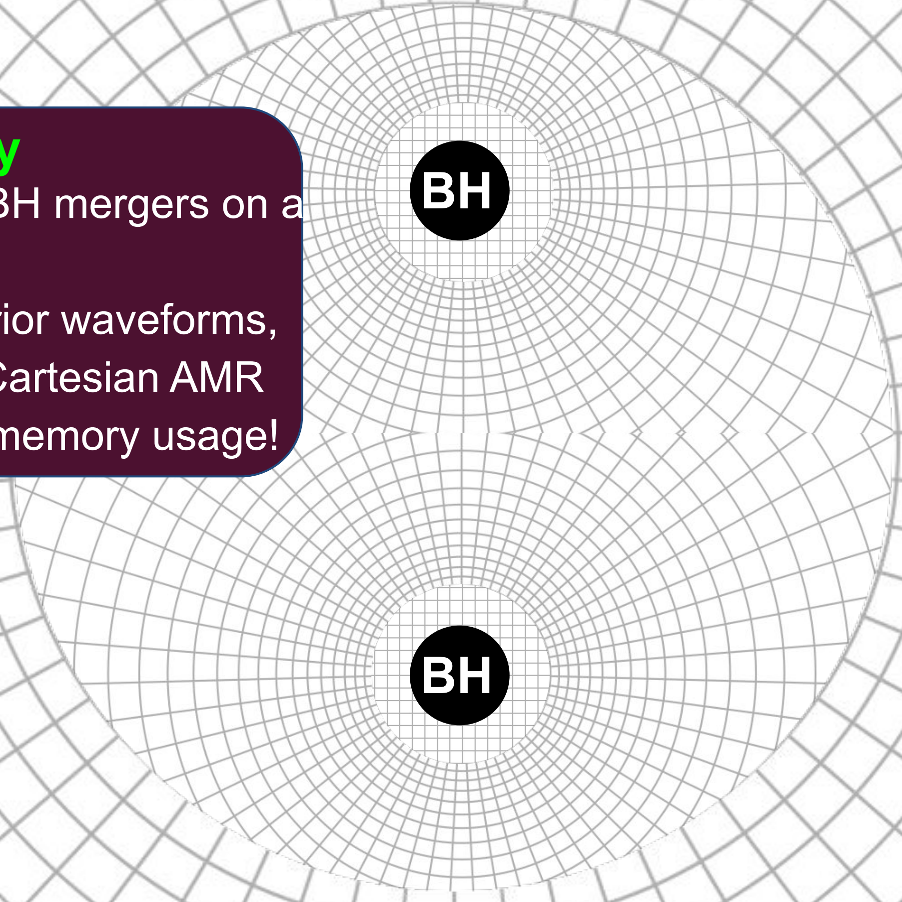
5-grid Bispheres
Head-on Results



Conclusions

Summary

1. (close-separation) BBH mergers on a cellphone!
2. Comparable or superior waveforms, 5-grid Bispheres vs Cartesian AMR
   a. ~20x less memory usage!



Schematic

Conclusions

Summary

1. (close-separation) BBH mergers on a cellphone!
2. Comparable or superior waveforms, 5-grid Bispheres vs Cartesian AMR
   a. ~20x less memory usage!

We're so close now!

-={ Last Steps }=-

1. Current focus: regridding algorithm
   a. Hard part (months) was intergrid interpolations!
2. Small modifications:
   a. GW extraction
   b. TwoPunctures ID
3. Finished, not reported here
   a. PN parameters for quasicircular BBH ID
      i. **NRPyPN**

BH

BH

Schematic

Addressing Issues with Singular Coordinates

Baumgarte, Montero, Cordero-Carrión, Müller (PRD 87, 044026, **2012**)

1. **Tensor components can be singular** ($\rightarrow 0$ or ∞) **at coord singularities**
   - Use cell-centered grids to avoid exact overlap with singularities
   - Singular pieces are multiplicative and known analytically:
     1. Scale out singular pieces & handle spatial derivs analytically
     2. Promote rescaled tensors to evolved quantities

Addressing Issues with Singular Coordinates

Baumgarte, Montero, Cordero-Carrión, Müller (PRD 87, 044026, **2012**)

1. **Tensor components can be singular** (→0 or ∞) **at coord singularities**
   - Use cell-centered grids to avoid exact overlap with singularities
   - Singular pieces are multiplicative and known analytically:
     1. Scale out singular pieces & handle spatial derivs analytically
     2. Promote rescaled tensors to evolved quantities

- *Example*: Smooth spacetime quantity Λ^i
  - **Cartesian**: all components regular; no coord singularities

$$\begin{aligned}\bar{\Lambda}^x &= [\text{smooth}] \\ \bar{\Lambda}^y &= [\text{smooth}] \\ \bar{\Lambda}^z &= [\text{smooth}]\end{aligned}$$

Addressing Issues with Singular Coordinates

Baumgarte, Montero, Cordero-Carrión, Müller (PRD 87, 044026, **2012**)

1. **Tensor components can be singular** (→0 or ∞) **at coord singularities**
   - Use cell-centered grids to avoid exact overlap with singularities
   - Singular pieces are multiplicative and known analytically:
     1. Scale out singular pieces & handle spatial derivs analytically
     2. Promote rescaled tensors to evolved quantities

$$\begin{aligned}\bar{\Lambda}^x &= [\text{smooth}]\\ \bar{\Lambda}^y &= [\text{smooth}]\\ \bar{\Lambda}^z &= [\text{smooth}]\end{aligned}$$

- *Example*: Smooth spacetime quantity Λ^i
  - **Cartesian**: all components regular; no coord singularities
  - **Spherical**: e.g., ϕ component diverges at coord singularity
    - **Idea**: where needed, only take numer. derivatives of smooth part, λ^ϕ
    - Perform *exact* differentiation on singular terms like $1/(r \sin \theta)$

$$\begin{aligned}\bar{\Lambda}^\phi &= \frac{1}{r\sin\theta} \times [\text{smooth part}]\\ &= \frac{1}{r\sin\theta} \times \lambda^\phi\end{aligned}$$

Addressing Issues with Singular Coordinates

Baumgarte, Montero, Cordero-Carrión, Müller (PRD 87, 044026, **2012**)

1. **Tensor components can be singular** (→0 or ∞) **at coord singularities**
   - Use cell-centered grids to avoid exact overlap with singularities
   - Singular pieces are multiplicative and known analytically:
     1. Scale out singular pieces & handle spatial derivs analytically
     2. Promote rescaled tensors to evolved quantities

2. **Divergent multiplicative terms in RHSs of equations**
   - E.g., 1D scalar wave equation:

$$\partial_t^2 u = \partial_r^2 u + \boxed{\frac{2}{r}}\partial_r u$$

   - $\boxed{2/r}$ term "stiffens" the equation
   - Even with cell-centered grids, RK2 timestepping is unstable
     1. Can use PIRK2 (original formulation), but
     2. *Ordinary* RK4 works just fine in 3+1 NR (discovered later)

Addressing Issues with Singular Coordinates

Baumgarte, Montero, Cordero-Carrión, Müller (PRD 87, 044026, **2012**)

1. **Tensor components can be singular** (→0 or ∞) **at coord singularities**
    - Use cell-centered grids to avoid exact overlap with singularities
    - Singular pieces are multiplicative and known analytically:
        1. Scale out singular pieces & handle spatial derivs analytically
        2. Promote rescaled tensors to evolved quantities

2. **Divergent multiplicative terms in RHSs of equations**
    - E.g., 1D scalar wave equation:

$$\partial_t^2 u = \partial_r^2 u + \boxed{\frac{2}{r}} \partial_r u$$

    - $\boxed{2/r}$ term "stiffens" the equation
    - Even with cell-centered grids, RK2 timestepping is unstable
        1. Can use PIRK2 (original formulation), but
        2. *Ordinary* RK4 works just fine in 3+1 NR (discovered later)

***Net result*: Stability & convergence properties *on par* with Cartesian grids**

SENR/NRPy+: Code Validation

http://blackholesathome.net

- Black hole simulation
  - Wormhole initial data
  - Cylindrical coordinates
  - Fourth-order finite differencing

- Excellent convergence
  - at t = 5M, in region unaffected by outer boundary (at r=10M)