Scalar self-force for generic bound orbits on a Kerr background

Zachary Nasipak[1], Thomas Osborn[2], & Charles R. Evans[1]

21st Capra Meeting on Radiation Reaction in General Relativity
25 June 2018

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

[1]University of North Carolina at Chapel Hill; [2]Oxford College at Emory University

Motivation

- **Focus:** Self-force calculations on Kerr
- **Difficulties:**
  - Loss of symmetry compared to Schwarzschild
  - Current computational methods inefficient (algebraic-convergence, precision loss, cancellation errors)
- **Goal:** Use a developmental model to improve self-force calculations
- **Approach:** Build scalar self-force for generic bound orbits on Kerr
  - See Peter Diener's talk on Tuesday for other approaches/uses of scalar self-force code

Motivation

- **Focus:** Self-force calculations on Kerr
- **Difficulties:**
  - Loss of symmetry compared to Schwarzschild
  - Current computational methods inefficient (algebraic-convergence, precision loss, cancellation errors)
- **Goal:** Use a developmental model to improve self-force calculations
- **Approach:** Build scalar self-force for generic bound orbits on Kerr
  - See Peter Diener's talk on Tuesday for other approaches/uses of scalar self-force code

Motivation

- **Focus:** Self-force calculations on Kerr
- **Difficulties:**
  - Loss of symmetry compared to Schwarzschild
  - Current computational methods inefficient (algebraic-convergence, precision loss, cancellation errors)
- **Goal:** Use a developmental model to improve self-force calculations
- **Approach:** Build scalar self-force for generic bound orbits on Kerr
  - See Peter Diener's talk on Tuesday for other approaches/uses of scalar self-force code

Motivation

- **Focus:** Self-force calculations on Kerr
- **Difficulties:**
  - Loss of symmetry compared to Schwarzschild
  - Current computational methods inefficient (algebraic-convergence, precision loss, cancellation errors)
- **Goal:** Use a developmental model to improve self-force calculations
- **Approach:** Build scalar self-force for generic bound orbits on Kerr
  - See Peter Diener's talk on Tuesday for other approaches/uses of scalar self-force code

Motivation

- **Focus:** Self-force calculations on Kerr
- **Difficulties:**
  - Loss of symmetry compared to Schwarzschild
  - Current computational methods inefficient (algebraic-convergence, precision loss, cancellation errors)
- **Goal:** Use a developmental model to improve self-force calculations
- **Approach:** Build scalar self-force for generic bound orbits on Kerr
  - See Peter Diener's talk on Tuesday for other approaches/uses of scalar self-force code

Motivation

- **Focus:** Self-force calculations on Kerr
- **Difficulties:**
  - Loss of symmetry compared to Schwarzschild
  - Current computational methods inefficient (algebraic-convergence, precision loss, cancellation errors)
- **Goal:** Use a developmental model to improve self-force calculations
- **Approach:** Build scalar self-force for generic bound orbits on Kerr
  - See Peter Diener's talk on Tuesday for other approaches/uses of scalar self-force code

Scalar self-force (SSF) literature

- **SSF on Schwarzschild**
  - Circular geodesics
    - Burko (2000) **PRL 84**
    - Diaz-Rivera et al. (2004) **PRD 70**
    - Vega & Detweiler (2008) **PRD 77**
    - Vega et al. (2009) **PRD 80**
    - Dolan & Barack (2011) **PRD 83**
  - Eccentric geodesics
    - Haas (2007) **PRD 75**
    - Canizares et al. (2010) **PRD 82**
    - Vega et al. (2013) **PRD 88**
- **SSF on Kerr**
  - Circular, equatorial geodesics
    - Warburton & Barack (2010) **PRD 81**
  - Eccentric, equatorial geodesics
    - Warburton & Barack (2011) **PRD 83**
    - Thorburg & Wardell (2017) **PRD 95**
  - Circular, inclined geodesics
    - Warburton (2015) **PRD 91**
  - Generic geodesic
    - Nasipak et al. (in progress)

Scalar self-force (SSF) literature

- **SSF on Schwarzschild**
  - Circular geodesics
    - Burko (2000) **PRL 84**
    - Diaz-Rivera et al. (2004) **PRD 70**
    - Vega & Detweiler (2008) **PRD 77**
    - Vega et al. (2009) **PRD 80**
    - Dolan & Barack (2011) **PRD 83**
  - Eccentric geodesics
    - Haas (2007) **PRD 75**
    - Canizares et al. (2010) **PRD 82**
    - Vega et al. (2013) **PRD 88**
- **SSF on Kerr**
  - Circular, equatorial geodesics
    - Warburton & Barack (2010) **PRD 81**
  - Eccentric, equatorial geodesics
    - Warburton & Barack (2011) **PRD 83**
    - Thorburg & Wardell (2017) **PRD 95**
  - Circular, inclined geodesics
    - Warburton (2015) **PRD 91**
  - Generic geodesic
    - Nasipak et al. (in progress)

Scalar self-force (SSF) problem

Gravitational self-force (GSF)

- Equations of motion

$$\mu\, u^\beta \nabla_\beta u^\alpha = F^\alpha_{\mathrm{GSF}} \sim \mathcal{O}(\mu^2/M^2)$$

- GSF equations

$$F^\alpha_{\mathrm{GSF}} = \mu\, P^{\alpha\beta\gamma\delta}\nabla_\beta h^{\mathrm{R}}_{\gamma\delta}$$

- Field equations

$$\left.\begin{aligned} {}_2\mathcal{O}\psi_0 &= 4\pi\Sigma\hat{T}_0 \\ {}_{-2}\mathcal{O}\rho^{-4}\psi_4 &= 4\pi\Sigma\hat{T}_4 \end{aligned}\right\} \Rightarrow h^{\mathrm{ret}}_{\alpha\beta}$$

Scalar self-force (GSF)

- Equations of motion

$$u^\beta \nabla_\beta (\mu u^\alpha) = F^\alpha \sim \mathcal{O}(q^2/M^2)$$

- SSF equations

$$F_\alpha = q\, \nabla_\alpha \Phi^{\mathrm{R}}$$

- Field equations

$$\Box_g \Phi = \nabla_\alpha \nabla^\alpha \Phi = -4\pi T$$

Scalar self-force (SSF) problem

Gravitational self-force (GSF)

- Equations of motion

$$\mu\, u^\beta \nabla_\beta u^\alpha = F^\alpha_{\text{GSF}} \sim \mathcal{O}(\mu^2/M^2)$$

- GSF equations

$$F^\alpha_{\text{GSF}} = \mu\, P^{\alpha\beta\gamma\delta}\nabla_\beta h^{\text{R}}_{\gamma\delta}$$

- Field equations

$$\left.\begin{aligned} {}_2\mathcal{O}\psi_0 &= 4\pi\Sigma\hat{T}_0 \\ {}_{-2}\mathcal{O}\rho^{-4}\psi_4 &= 4\pi\Sigma\hat{T}_4 \end{aligned}\right\} \Rightarrow h^{\text{ret}}_{\alpha\beta}$$

Scalar self-force (GSF)

- Equations of motion

$$u^\beta \nabla_\beta(\mu u^\alpha) = F^\alpha \sim \mathcal{O}(q^2/M^2)$$

- SSF equations

$$F_\alpha = q\, \nabla_\alpha \Phi^{\text{R}}$$

- Field equations

$$\Box_g \Phi = \nabla_\alpha \nabla^\alpha \Phi = -4\pi T$$

Scalar self-force (SSF) problem

Gravitational self-force (GSF)

- Equations of motion

$$\mu\, u^\beta \nabla_\beta u^\alpha = F^\alpha_{\mathrm{GSF}} \sim \mathcal{O}(\mu^2/M^2)$$

- GSF equations

$$F^\alpha_{\mathrm{GSF}} = \mu\, P^{\alpha\beta\gamma\delta} \nabla_\beta h^{\mathrm{R}}_{\gamma\delta}$$

- Field equations

$$\left.\begin{aligned} {}_2\mathcal{O}\psi_0 &= 4\pi\Sigma\hat{T}_0 \\ {}_{-2}\mathcal{O}\rho^{-4}\psi_4 &= 4\pi\Sigma\hat{T}_4 \end{aligned}\right\} \Rightarrow h^{\mathrm{ret}}_{\alpha\beta}$$

Scalar self-force (GSF)

- Equations of motion

$$u^\beta \nabla_\beta (\mu u^\alpha) = F^\alpha \sim \mathcal{O}(q^2/M^2)$$

- SSF equations

$$F_\alpha = q\, \nabla_\alpha \Phi^{\mathrm{R}}$$

- Field equations

$$\Box_g \Phi = \nabla_\alpha \nabla^\alpha \Phi = -4\pi T$$

Scalar self-force (SSF) problem

Gravitational self-force (GSF)

- Equations of motion

$$\mu\, u^\beta \nabla_\beta u^\alpha = F^\alpha_{\text{GSF}} \sim \mathcal{O}(\mu^2/M^2)$$

- GSF equations

$$F^\alpha_{\text{GSF}} = \mu\, P^{\alpha\beta\gamma\delta} \nabla_\beta h^{\text{R}}_{\gamma\delta}$$

- Field equations

$$\left.\begin{aligned} {}_2\mathcal{O}\psi_0 &= 4\pi\Sigma\hat{T}_0 \\ {}_{-2}\mathcal{O}\rho^{-4}\psi_4 &= 4\pi\Sigma\hat{T}_4 \end{aligned}\right\} \Rightarrow h^{\text{ret}}_{\alpha\beta}$$

Scalar self-force (GSF)

- Equations of motion

$$u^\beta \nabla_\beta (\mu u^\alpha) = F^\alpha \sim \mathcal{O}(q^2/M^2)$$

- SSF equations

$$F_\alpha = q\, \nabla_\alpha \Phi^{\text{R}}$$

- Field equations

$$\Box_g \Phi = \nabla_\alpha \nabla^\alpha \Phi = -4\pi T$$

Scalar self-force (SSF) problem

Gravitational self-force (GSF)

- Equations of motion

$$\mu\, u^\beta \nabla_\beta u^\alpha = F^\alpha_{\mathrm{GSF}} \sim \mathcal{O}(\mu^2/M^2)$$

- GSF equations

$$F^\alpha_{\mathrm{GSF}} = \mu\, P^{\alpha\beta\gamma\delta} \nabla_\beta h^{\mathrm{R}}_{\gamma\delta}$$

- Field equations

$$\left.\begin{aligned} {}_2\mathcal{O}\psi_0 &= 4\pi\Sigma\hat{T}_0 \\ {}_{-2}\mathcal{O}\rho^{-4}\psi_4 &= 4\pi\Sigma\hat{T}_4 \end{aligned}\right\} \Rightarrow h^{\mathrm{ret}}_{\alpha\beta}$$

Scalar self-force (GSF)

- Equations of motion

$$u^\beta \nabla_\beta (\mu u^\alpha) = F^\alpha \sim \mathcal{O}(q^2/M^2)$$

- SSF equations

$$F_\alpha = q\, \nabla_\alpha \Phi^{\mathrm{R}}$$

- Field equations

$$\Box_g \Phi = \nabla_\alpha \nabla^\alpha \Phi = -4\pi T$$

Scalar self-force (SSF) problem

Gravitational self-force (GSF)

- Equations of motion

$$\mu\, u^\beta \nabla_\beta u^\alpha = F^\alpha_{\rm GSF} \sim \mathcal{O}(\mu^2/M^2)$$

- GSF equations

$$F^\alpha_{\rm GSF} = \mu\, P^{\alpha\beta\gamma\delta}\nabla_\beta h^{\rm R}_{\gamma\delta}$$

- Field equations

$$\left.\begin{aligned} {}_2\mathcal{O}\psi_0 &= 4\pi\Sigma\hat{T}_0 \\ {}_{-2}\mathcal{O}\rho^{-4}\psi_4 &= 4\pi\Sigma\hat{T}_4 \end{aligned}\right\} \Rightarrow h^{\rm ret}_{\alpha\beta}$$

Scalar self-force (GSF)

- Equations of motion

$$u^\beta \nabla_\beta (\mu u^\alpha) = F^\alpha \sim \mathcal{O}(q^2/M^2)$$

- SSF equations

$$F_\alpha = q\, \nabla_\alpha \Phi^{\rm R}$$

- Field equations

$$\Box_g \Phi = \nabla_\alpha \nabla^\alpha \Phi = -4\pi T$$

Scalar self-force (SSF) problem

Gravitational self-force (GSF)

- Equations of motion

$$\mu\, u^\beta \nabla_\beta u^\alpha = F^\alpha_{\text{GSF}} \sim \mathcal{O}(\mu^2/M^2)$$

- GSF equations

$$F^\alpha_{\text{GSF}} = \mu\, P^{\alpha\beta\gamma\delta} \nabla_\beta h^{\text{R}}_{\gamma\delta}$$

- Field equations

$$\left.\begin{aligned} {}_2\mathcal{O}\psi_0 &= 4\pi\Sigma\hat{T}_0 \\ {}_{-2}\mathcal{O}\rho^{-4}\psi_4 &= 4\pi\Sigma\hat{T}_4 \end{aligned}\right\} \Rightarrow h^{\text{ret}}_{\alpha\beta}$$

Scalar self-force (GSF)

- Equations of motion

$$u^\beta \nabla_\beta (\mu u^\alpha) = F^\alpha \sim \mathcal{O}(q^2/M^2)$$

- SSF equations

$$F_\alpha = q\, \nabla_\alpha \Phi^{\text{R}}$$

- Field equations

$$\Box_g \Phi = \nabla_\alpha \nabla^\alpha \Phi = -4\pi T$$

My SSF code

- Written in Mathematica using FD approach
- Performed w/ arbitrary numerical precision
- Computational time of generic orbit
  - ~ 10,000 CPU hours*
- Some modules inherited
  - Credit: Charles R. Evans, Thomas Osburn, Erik Forseth, & Seth Hopper
  - Black Hole Perturbation Toolkit (bhptoolkit.org)

My SSF code

- Written in MATHEMATICA using FD approach
- Performed w/ arbitrary numerical precision
- Computational time of generic orbit
  - ~ 10,000 CPU hours*
- Some modules inherited
  - Credit: Charles R. Evans, Thomas Osburn, Erik Forseth, & Seth Hopper
  - Black Hole Perturbation Toolkit (bhptoolkit.org)

My SSF code

- Written in MATHEMATICA using FD approach
- Performed w/ arbitrary numerical precision
- Computational time of generic orbit
  - ~ 10,000 CPU hours*
- Some modules inherited
  - Credit: Charles R. Evans, Thomas Osburn, Erik Forseth, & Seth Hopper
  - Black Hole Perturbation Toolkit (bhptoolkit.org)

My SSF code

- Written in MATHEMATICA using FD approach
- Performed w/ arbitrary numerical precision
- Computational time of generic orbit
  - ~ 10,000 CPU hours*

***Depends on initial parameters, specified numerical precision, etc.**

- Some modules inherited
  - Credit: Charles R. Evans, Thomas Osburn, Erik Forseth, & Seth Hopper
  - Black Hole Perturbation Toolkit (bhptoolkit.org)

My SSF code

- Written in MATHEMATICA using FD approach
- Performed w/ arbitrary numerical precision
- Computational time of generic orbit
  - ~ 10,000 CPU hours*
- Some modules inherited
  - Credit: Charles R. Evans, Thomas Osburn, Erik Forseth, & Seth Hopper
  - Black Hole Perturbation Toolkit (bhptoolkit.org)

***Depends on initial parameters, specified numerical precision, etc.**

Building a generic SSF code

- Construct scalar field

$$\Box_g \Phi^{\text{ret}} = -4\pi\rho \qquad \rho = q\int \delta^{(4)}(x^\mu, z^\mu(\tau))d\tau$$

- Separable in FD $(\omega_{mkn} = m\Omega_\varphi + k\Omega_\theta + n\Omega_r)$

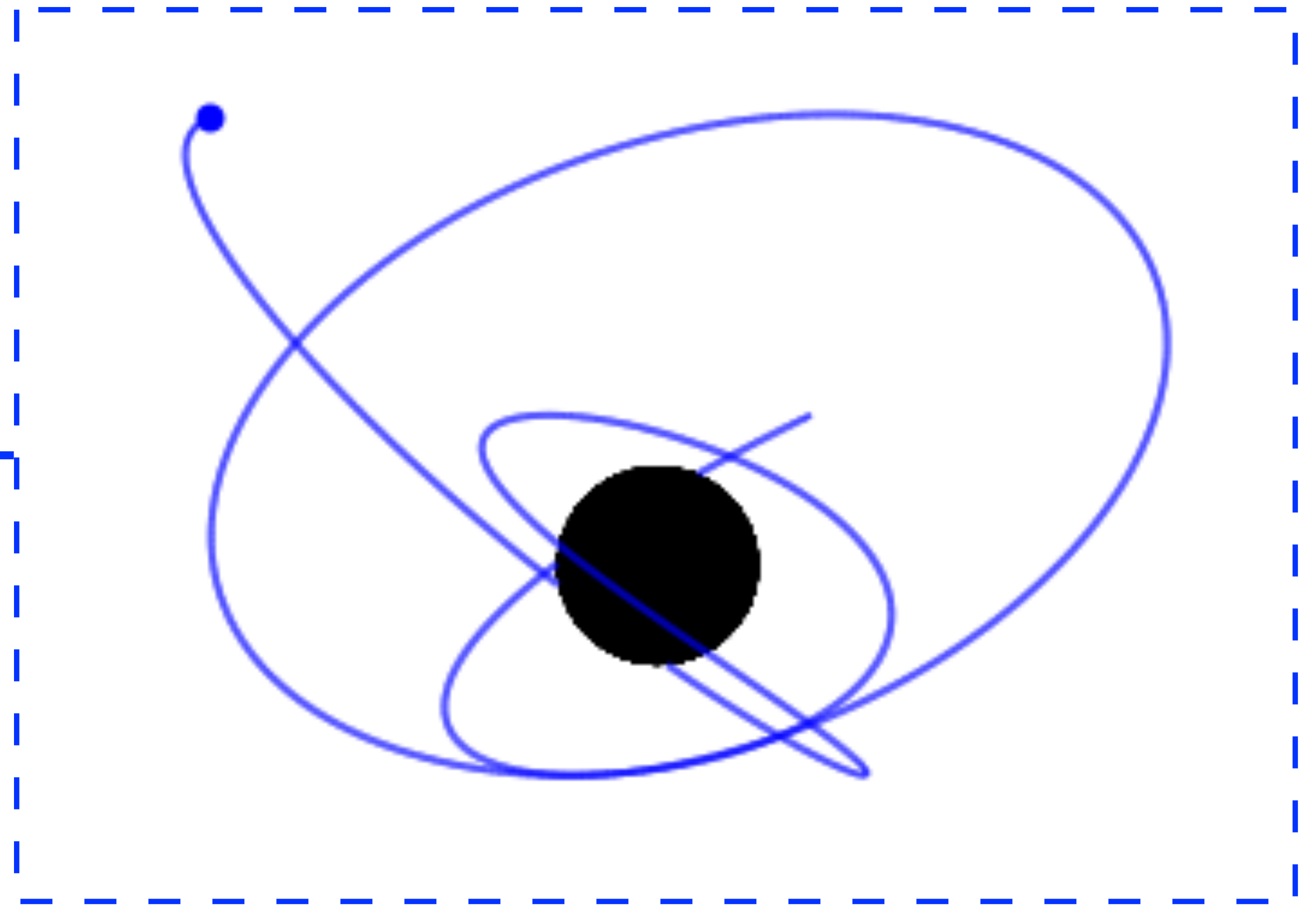
$$\Phi^{\text{ret}} = \frac{1}{\sqrt{r^2+a^2}}\sum_{\hat{l}mkn} X_{\hat{l}mkn}(r)S_{\hat{l}mkn}(\theta)e^{im\varphi}e^{i\omega_{mkn}t}$$

- Decouples into ODEs
  - Solve for $S_{\hat{l}mkn} \rightarrow Y_{lm}$ expansion
  - Solve for $X_{\hat{l}mkn} \rightarrow$ variation of parameters + MST

Building a generic SSF code

- Construct scalar field

$$\Box_g \Phi^{\text{ret}} = -4\pi\rho \qquad \rho = q\int \delta^{(4)}(x^\mu, z^\mu(\tau))d\tau$$



- Separable in FD $(\omega_{mkn} = m\Omega_\varphi + k\Omega_\theta + n\Omega_r)$

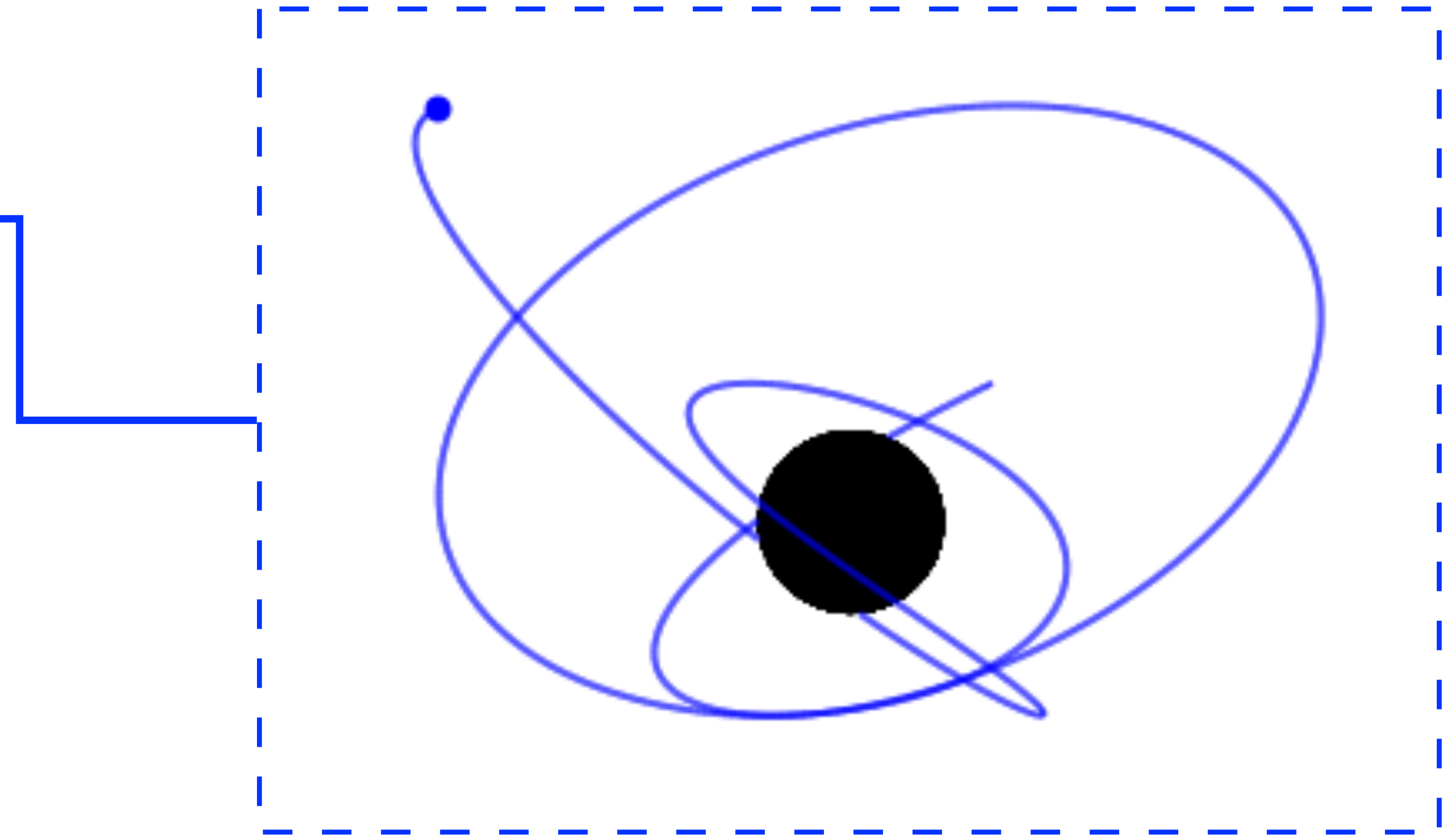
$$\Phi^{\text{ret}} = \frac{1}{\sqrt{r^2+a^2}} \sum_{\hat{l}mkn} X_{\hat{l}mkn}(r) S_{\hat{l}mkn}(\theta) e^{im\varphi} e^{i\omega_{mkn}t}$$

- Decouples into ODEs
  - Solve for $S_{\hat{l}mkn} \rightarrow Y_{lm}$ expansion
  - Solve for $X_{\hat{l}mkn} \rightarrow$ variation of parameters + MST

Building a generic SSF code

- Construct scalar field

$$\Box_g \Phi^{\rm ret} = -4\pi\rho \qquad \rho = q\int \delta^{(4)}(x^\mu, z^\mu(\tau))d\tau$$



- Separable in FD $(\omega_{mkn} = m\Omega_\varphi + k\Omega_\theta + n\Omega_r)$

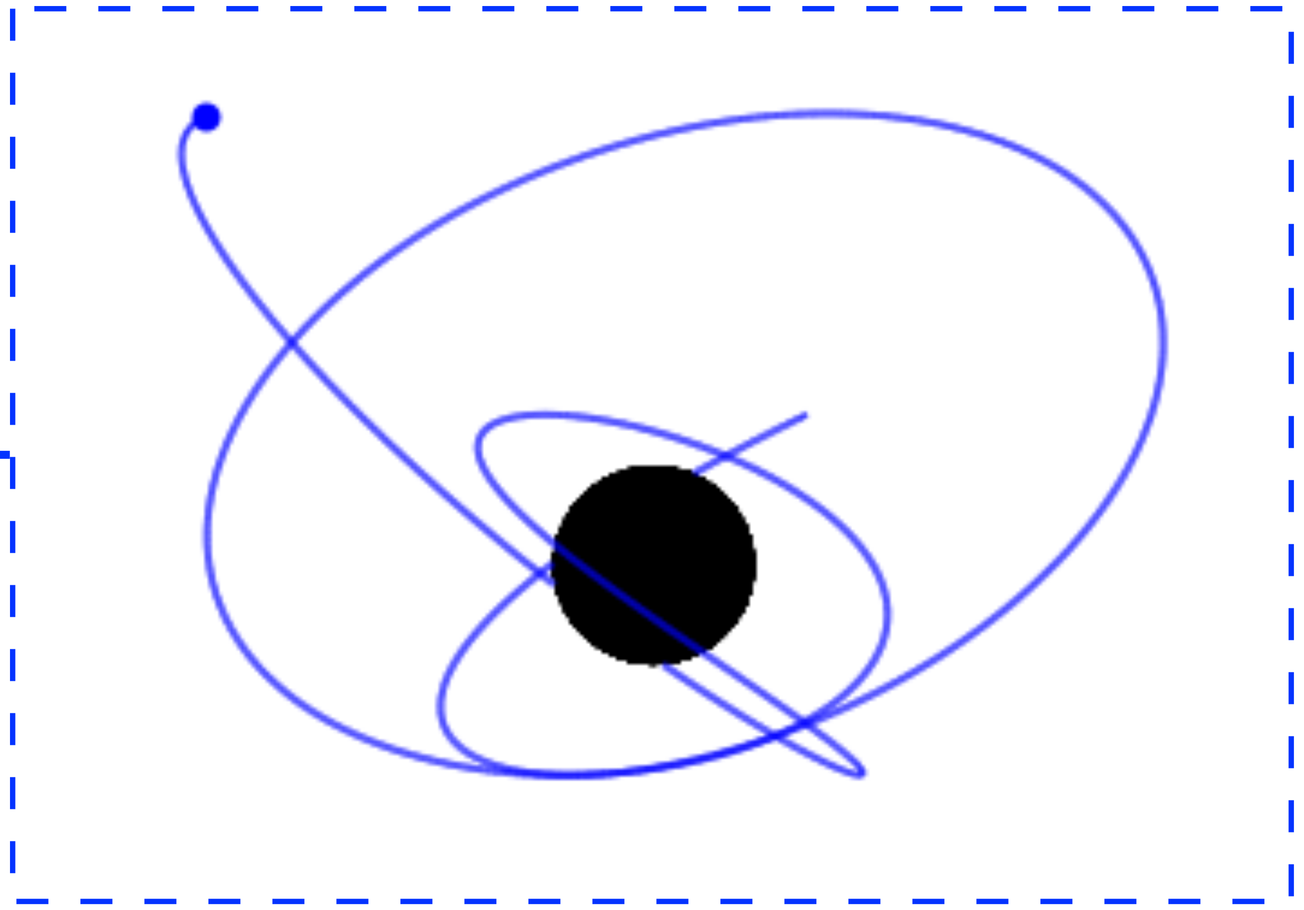
$$\Phi^{\rm ret} = \frac{1}{\sqrt{r^2+a^2}}\sum_{\hat{l}mkn} X_{\hat{l}mkn}(r) S_{\hat{l}mkn}(\theta) e^{im\varphi} e^{i\omega_{mkn}t}$$

- Decouples into ODEs
  - Solve for $S_{\hat{l}mkn} \rightarrow Y_{lm}$ expansion
  - Solve for $X_{\hat{l}mkn} \rightarrow$ variation of parameters + MST

Building a generic SSF code

- Construct scalar field

$$\Box_g \Phi^{\text{ret}} = -4\pi\rho \qquad \rho = q\int \delta^{(4)}(x^\mu, z^\mu(\tau))d\tau$$



- Separable in FD $(\omega_{mkn} = m\Omega_\varphi + k\Omega_\theta + n\Omega_r)$

$$\Phi^{\text{ret}} = \frac{1}{\sqrt{r^2+a^2}} \sum_{\hat{l}mkn} X_{\hat{l}mkn}(r) S_{\hat{l}mkn}(\theta) e^{im\varphi} e^{i\omega_{mkn}t}$$

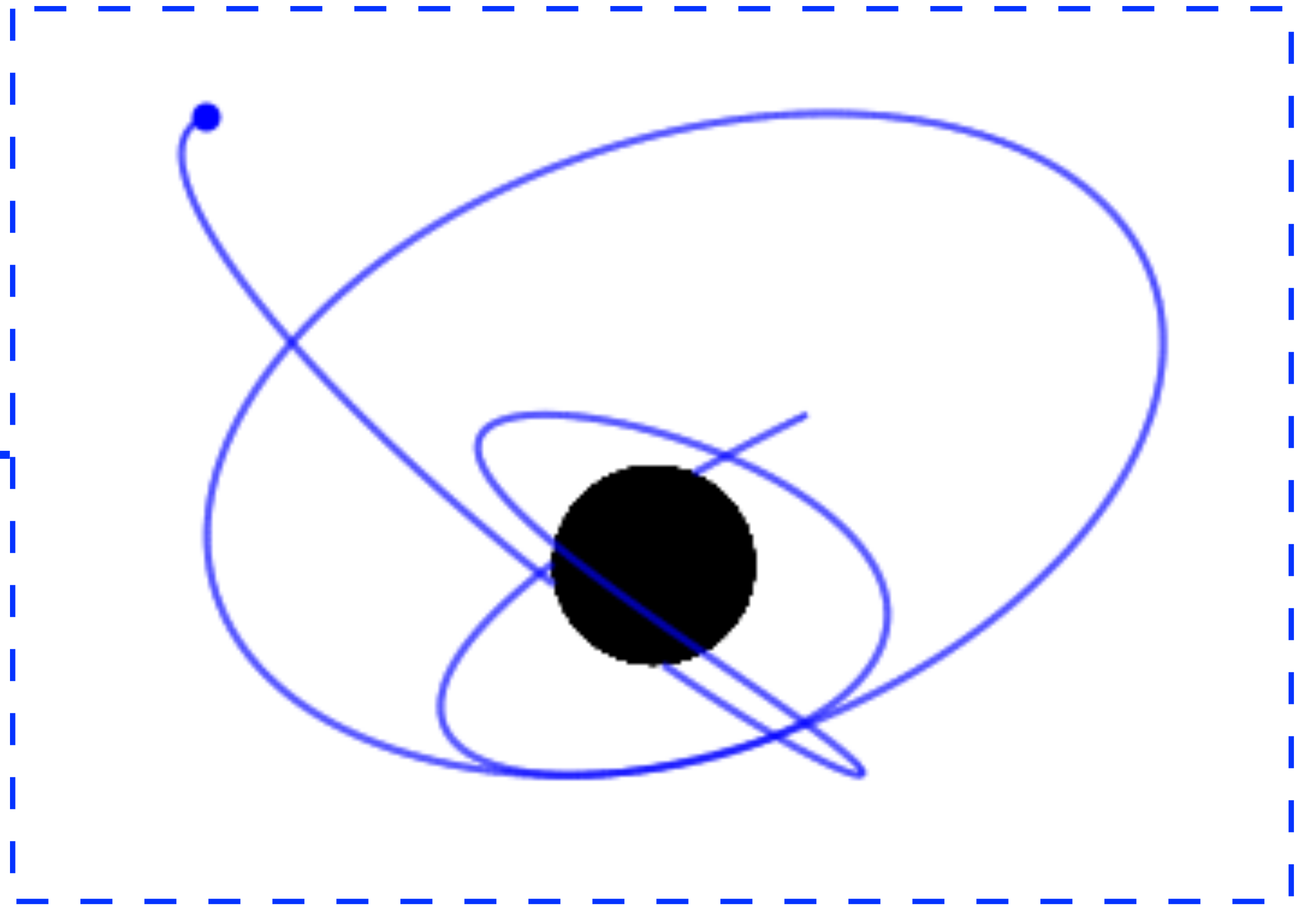
Spheroidal harmonics

- Decouples into ODEs
  - Solve for $S_{\hat{l}mkn} \rightarrow Y_{lm}$ expansion
  - Solve for $X_{\hat{l}mkn} \rightarrow$ variation of parameters + MST

Building a generic SSF code

- Construct scalar field

$$\Box_g \Phi^{\text{ret}} = -4\pi\rho \qquad \rho = q\int \delta^{(4)}(x^\mu, z^\mu(\tau))d\tau$$



- Separable in FD $(\omega_{mkn} = m\Omega_\varphi + k\Omega_\theta + n\Omega_r)$

$$\Phi^{\text{ret}} = \frac{1}{\sqrt{r^2+a^2}}\sum_{\hat{l}mkn} X_{\hat{l}mkn}(r) S_{\hat{l}mkn}(\theta) e^{im\varphi} e^{i\omega_{mkn}t}$$

$$X(r) = \sqrt{r^2+a^2}\, R(r)$$

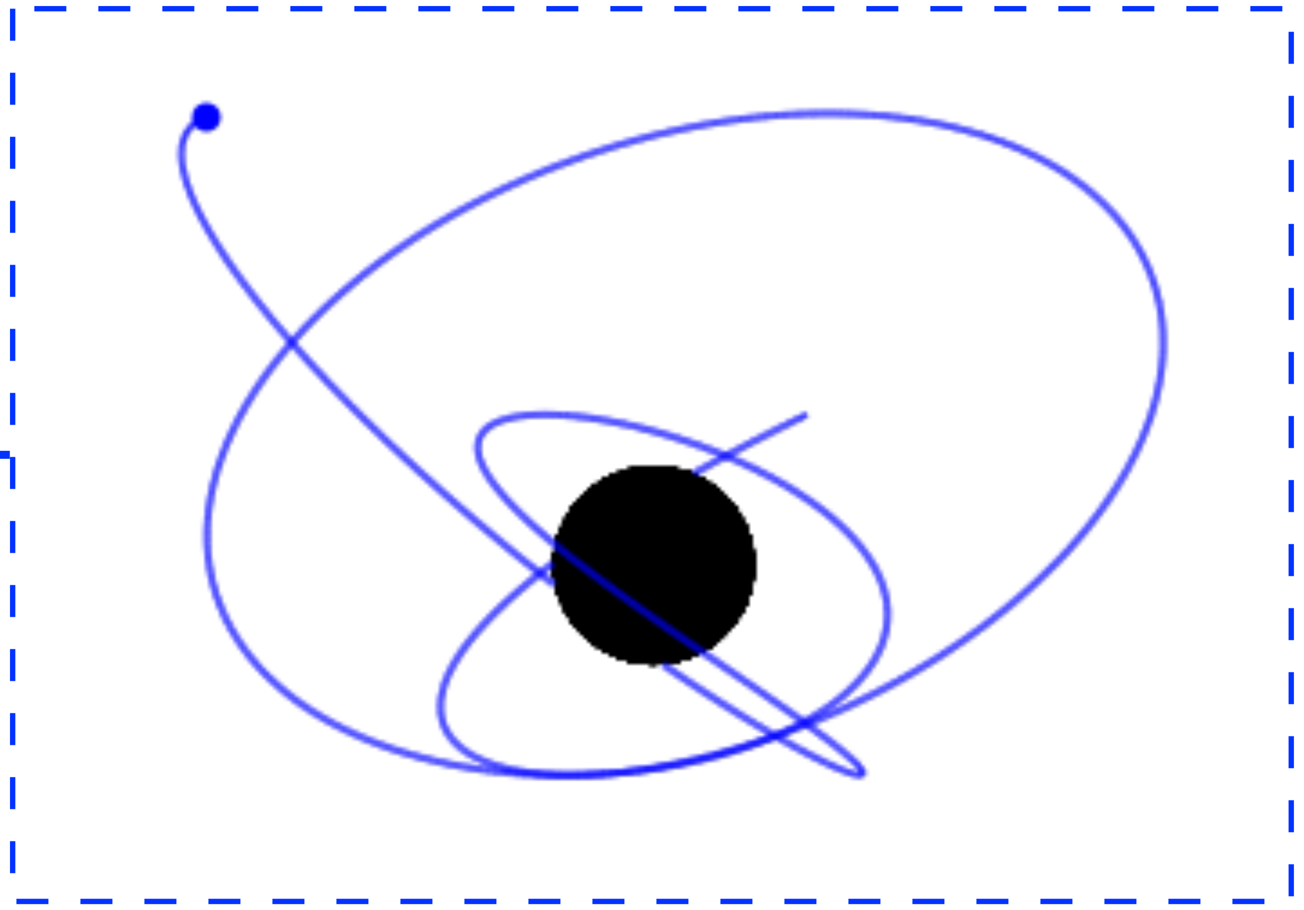
Spheroidal harmonics

- Decouples into ODEs
  - Solve for $S_{\hat{l}mkn} \rightarrow Y_{lm}$ expansion
  - Solve for $X_{\hat{l}mkn} \rightarrow$ variation of parameters + MST

Building a generic SSF code

- Construct scalar field

$$\Box_g \Phi^{\text{ret}} = -4\pi\rho \qquad \rho = q\int \delta^{(4)}(x^\mu, z^\mu(\tau))d\tau$$



- Separable in FD $(\omega_{mkn} = m\Omega_\varphi + k\Omega_\theta + n\Omega_r)$

$$\Phi^{\text{ret}} = \frac{1}{\sqrt{r^2+a^2}} \sum_{\hat{l}mkn} X_{\hat{l}mkn}(r) S_{\hat{l}mkn}(\theta) e^{im\varphi} e^{i\omega_{mkn}t}$$

$$X(r) = \sqrt{r^2+a^2}\, R(r)$$

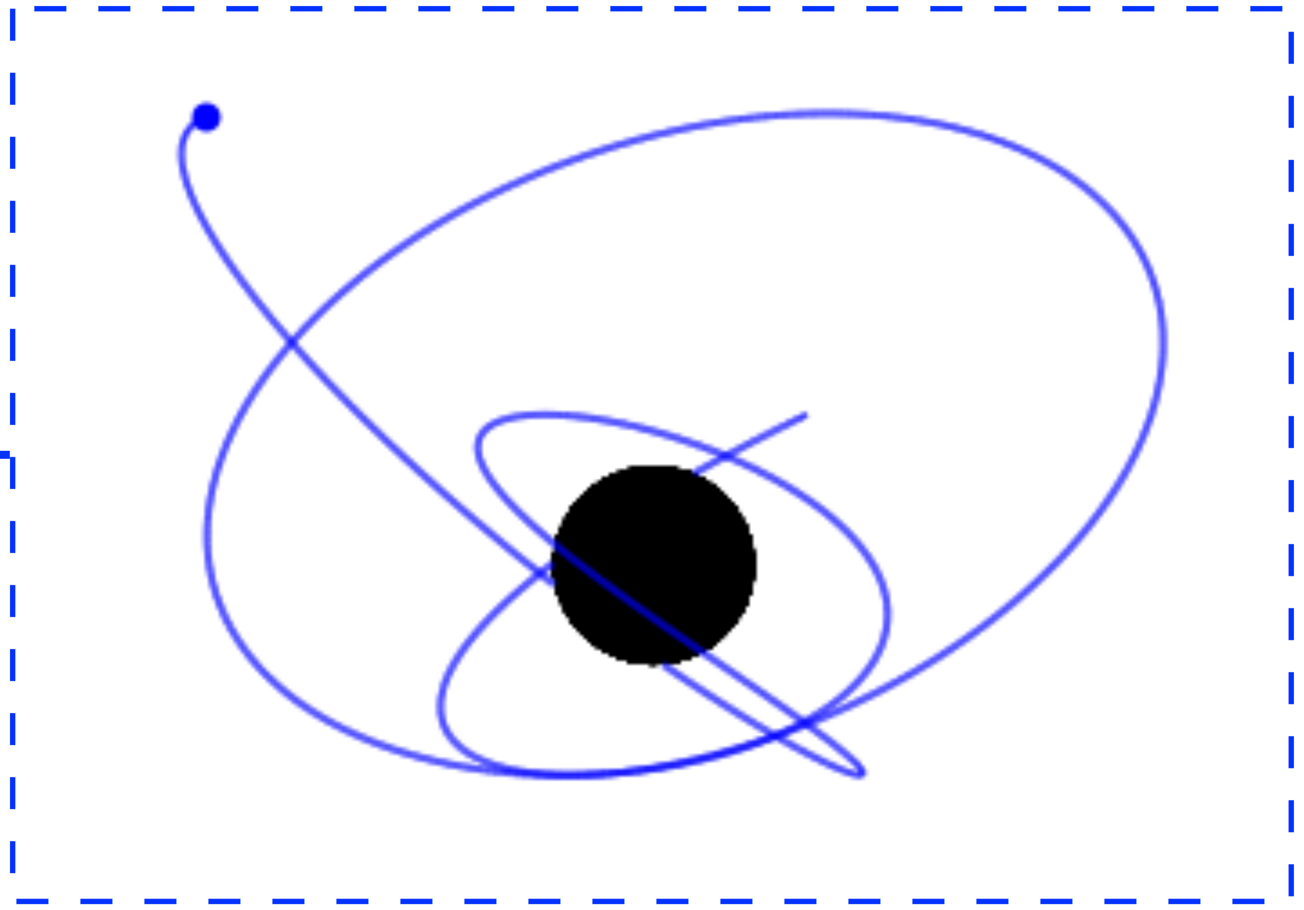
Spheroidal harmonics

- Decouples into ODEs
  - Solve for $S_{\hat{l}mkn} \rightarrow Y_{lm}$ expansion
  - Solve for $X_{\hat{l}mkn} \rightarrow$ variation of parameters + MST

Building a generic SSF code

- Construct scalar field

$$\Box_g \Phi^{\text{ret}} = -4\pi\rho \qquad \rho = q\int \delta^{(4)}(x^\mu, z^\mu(\tau))d\tau$$



- Separable in FD $(\omega_{mkn} = m\Omega_\varphi + k\Omega_\theta + n\Omega_r)$

$$\Phi^{\text{ret}} = \frac{1}{\sqrt{r^2+a^2}} \sum_{\hat{l}mkn} X_{\hat{l}mkn}(r) S_{\hat{l}mkn}(\theta) e^{im\varphi} e^{i\omega_{mkn}t}$$

$$X(r) = \sqrt{r^2+a^2}\, R(r)$$

Spheroidal harmonics

- Decouples into ODEs
  - Solve for $S_{\hat{l}mkn} \to Y_{lm}$ expansion $\implies S_{\hat{l}mkn}(\theta)e^{im\varphi} = \sum_{l=|m|}^{\infty} b^l_{\hat{l}m}(a^2\omega^2_{mkn}) Y_{lm}(\theta,\varphi)$
  - Solve for $X_{\hat{l}mkn} \to$ variation of parameters + MST

Building a generic SSF code

- Solving the radial ODEs

$$\frac{dr_*}{dr} = \frac{r^2 + a^2}{r^2 - 2Mr + a^2}$$

- Variation of parameters

Building a generic SSF code

- Solving the radial ODEs

$$\frac{dr_*}{dr} = \frac{r^2 + a^2}{r^2 - 2Mr + a^2}$$

Tortoise coordinate

- Variation of parameters

Building a generic SSF code

- Solving the radial ODEs

$$\frac{dr_*}{dr} = \frac{r^2 + a^2}{r^2 - 2Mr + a^2}$$ **Tortoise coordinate** $$\rightarrow \left[\frac{d^2}{dr_*^2} - U_{\hat{l}mkn}(r)\right] X_{\hat{l}mkn} = \tilde{\rho}_{\hat{l}mkn}$$

- Variation of parameters

Building a generic SSF code

- Solving the radial ODEs

Generalized Sasaki-Nakamura Eqn

$$\frac{dr_*}{dr} = \frac{r^2 + a^2}{r^2 - 2Mr + a^2}$$ **Tortoise coordinate**

$$\rightarrow \left[\frac{d^2}{dr_*^2} - U_{\hat{l}mkn}(r)\right] X_{\hat{l}mkn} = \tilde{\rho}_{\hat{l}mkn}$$

- Variation of parameters

Building a generic SSF code

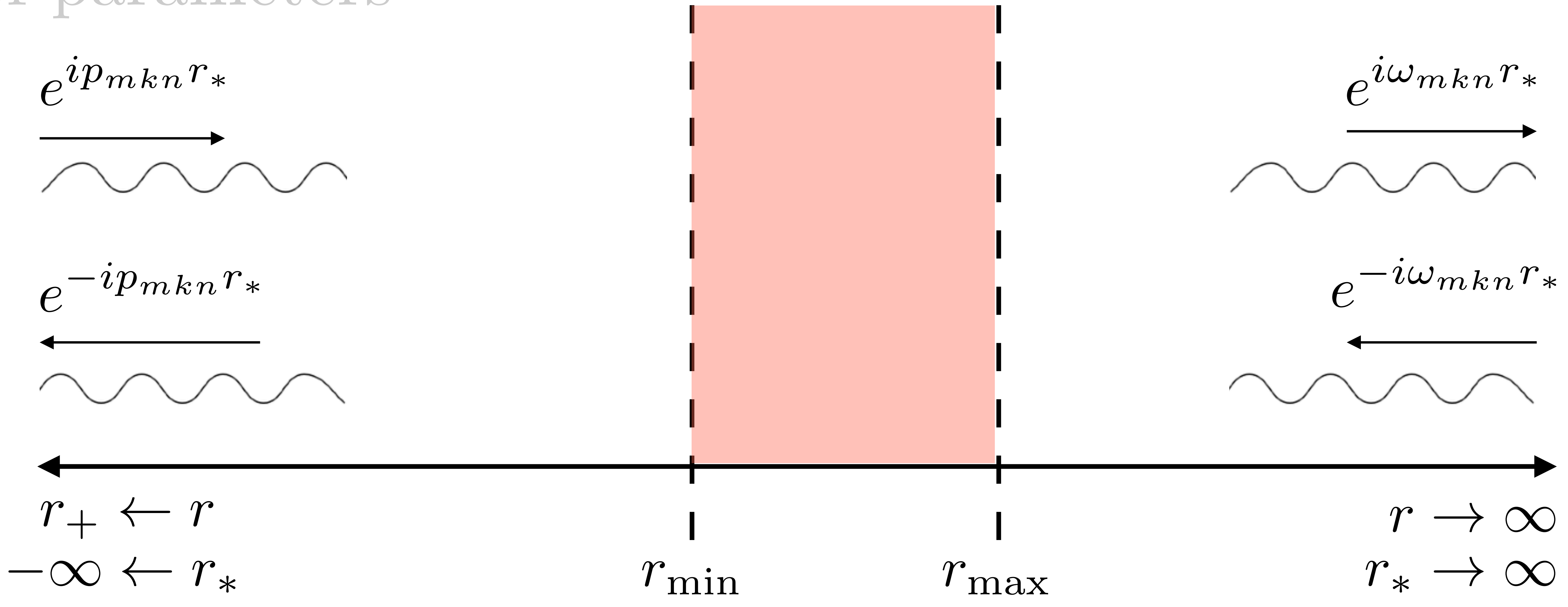
- Solving the radial ODEs

$$\frac{dr_*}{dr} = \frac{r^2 + a^2}{r^2 - 2Mr + a^2}$$ **Tortoise coordinate**

$$\rightarrow \left[\frac{d^2}{dr_*^2} - U_{\hat{l}mkn}(r)\right] X_{\hat{l}mkn} = \tilde{\rho}_{\hat{l}mkn}$$

Generalized Sasaki-Nakamura Eqn

- Variation of parameters



Building a generic SSF code

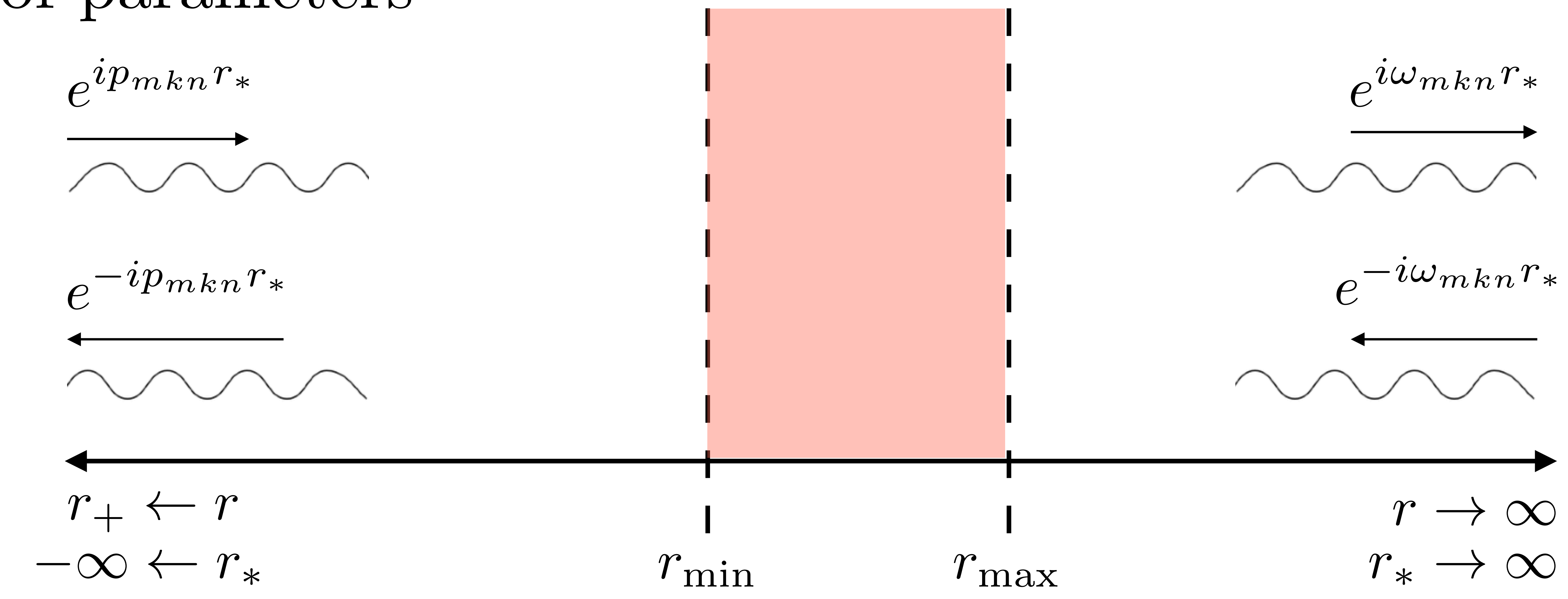
- Solving the radial ODEs

Generalized Sasaki-Nakamura Eqn

$$\frac{dr_*}{dr} = \frac{r^2 + a^2}{r^2 - 2Mr + a^2}$$ **Tortoise coordinate**

$$\rightarrow \left[\frac{d^2}{dr_*^2} - U_{\hat{l}mkn}(r)\right] X_{\hat{l}mkn} = \tilde{\rho}_{\hat{l}mkn}$$

- Variation of parameters



Building a generic SSF code

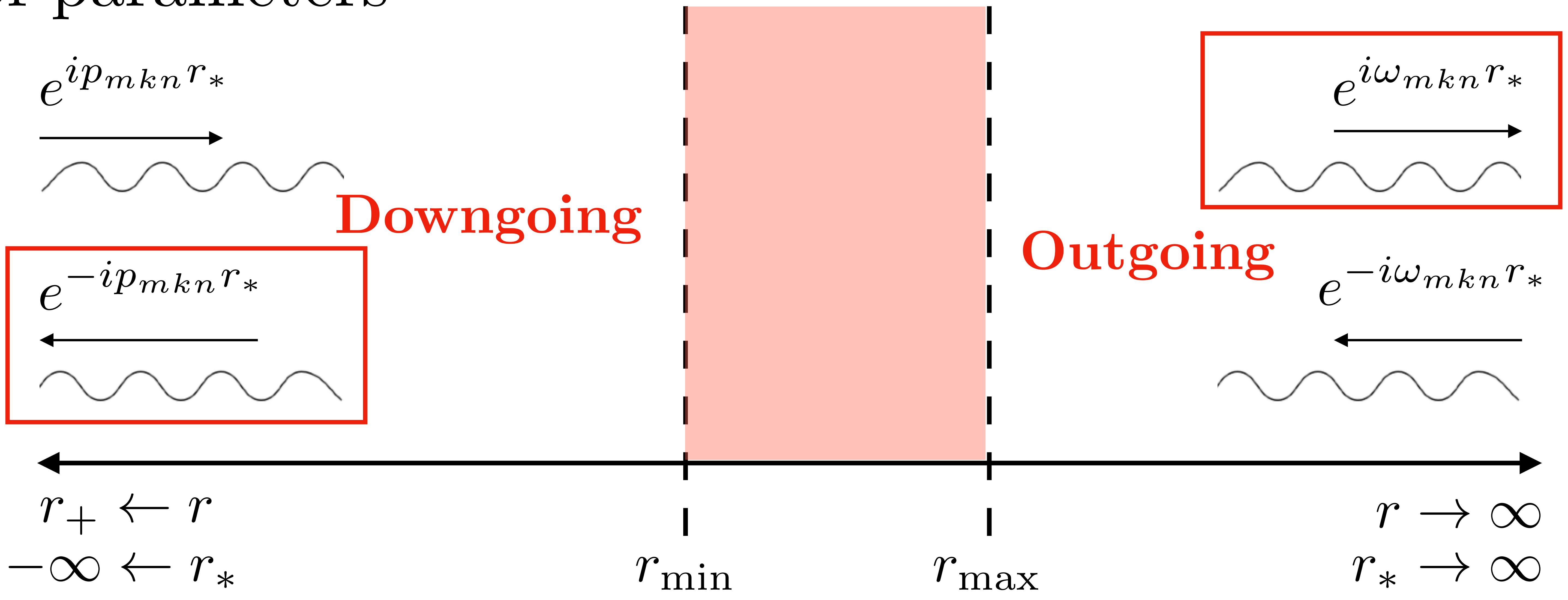
- Solving the radial ODEs

Generalized Sasaki-Nakamura Eqn

$$\frac{dr_*}{dr} = \frac{r^2 + a^2}{r^2 - 2Mr + a^2}$$ **Tortoise coordinate**

$$\rightarrow \left[\frac{d^2}{dr_*^2} - U_{\hat{l}mkn}(r)\right] X_{\hat{l}mkn} = \tilde{\rho}_{\hat{l}mkn}$$

- Variation of parameters



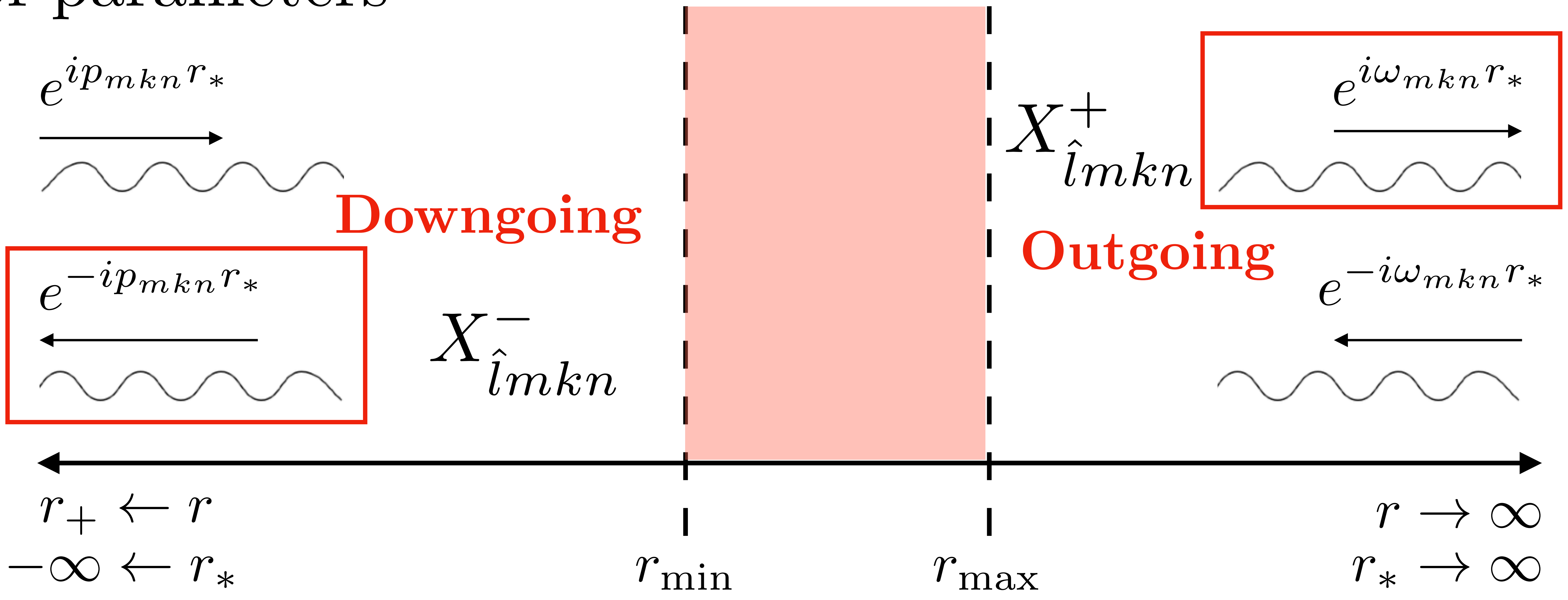
Building a generic SSF code

- Solving the radial ODEs

Generalized Sasaki-Nakamura Eqn

$$\frac{dr_*}{dr} = \frac{r^2 + a^2}{r^2 - 2Mr + a^2}$$ **Tortoise coordinate** $$\rightarrow \left[\frac{d^2}{dr_*^2} - U_{\hat{l}mkn}(r)\right] X_{\hat{l}mkn} = \tilde{\rho}_{\hat{l}mkn}$$

- Variation of parameters



Building a generic SSF code

- Finding homogeneous solutions
  - **Mano-Suzuki-Takasugi (MST)** function expansion formalism
    - Series solution w/ introduction of free parameter ν

$$R^{-}_{\hat{l}mkn}(r) \rightarrow R^{-}_{\hat{l}m}(\omega, r) = \sum_{n=-\infty}^{\infty} f^{\nu}(\omega, x) a_n^{\nu} F(c^{\nu}_{n,1}, c^{\nu}_{n,2}; c^{\nu}_{n,3}; x)$$

$$R^{+}_{\hat{l}mkn}(r) \rightarrow R^{+}_{\hat{l}m}(\omega, r) = \sum_{n=-\infty}^{\infty} g^{\nu}(\omega, z) b_n^{\nu} (2z)^n \Psi(c^{\nu}_{n,4}, c^{\nu}_{n,5}; -2iz)$$

$$c^{\nu}_{n,i} = c^{\nu}_{n,i}(\omega)$$

Building a generic SSF code

- Finding homogeneous solutions
  - **Mano-Suzuki-Takasugi (MST)** function expansion formalism
    - Series solution w/ introduction of free parameter ν

Sasaki & Tagoshi (2003) **LRR 6**

$$R^{-}_{\hat{l}mkn}(r) \to R^{-}_{\hat{l}m}(\omega, r) = \sum_{n=-\infty}^{\infty} f^{\nu}(\omega, x) a^{\nu}_{n} F(c^{\nu}_{n,1}, c^{\nu}_{n,2}; c^{\nu}_{n,3}; x)$$

$$R^{+}_{\hat{l}mkn}(r) \to R^{+}_{\hat{l}m}(\omega, r) = \sum_{n=-\infty}^{\infty} g^{\nu}(\omega, z) b^{\nu}_{n} (2z)^{n} \Psi(c^{\nu}_{n,4}, c^{\nu}_{n,5}; -2iz)$$

$$c^{\nu}_{n,i} = c^{\nu}_{n,i}(\omega)$$

Building a generic SSF code

- Finding homogeneous solutions
  - **Mano-Suzuki-Takasugi (MST)** function expansion formalism
    - Series solution w/ introduction of free parameter ν

Sasaki & Tagoshi (2003) **LRR 6**

Hypergeometric function

$$R^{-}_{\hat{l}mkn}(r) \to R^{-}_{\hat{l}m}(\omega, r) = \sum_{n=-\infty}^{\infty} f^{\nu}(\omega, x) a^{\nu}_{n} F(c^{\nu}_{n,1}, c^{\nu}_{n,2}; c^{\nu}_{n,3}; x)$$

Irregular confluent hypergeometric function

$$R^{+}_{\hat{l}mkn}(r) \to R^{+}_{\hat{l}m}(\omega, r) = \sum_{n=-\infty}^{\infty} g^{\nu}(\omega, z) b^{\nu}_{n} (2z)^{n} \Psi(c^{\nu}_{n,4}, c^{\nu}_{n,5}; -2iz)$$

$$c^{\nu}_{n,i} = c^{\nu}_{n,i}(\omega)$$

Building a generic SSF code

- Finding homogeneous solutions
  - **Mano-Suzuki-Takasugi (MST)** function expansion formalism
    - Series solution w/ introduction of free parameter ν

Sasaki & Tagoshi (2003) **LRR 6**

Hypergeometric function

$$R^-_{\hat{l}mkn}(r) \to R^-_{\hat{l}m}(\omega, r) = \sum_{n=-\infty}^{\infty} f^{\nu}(\omega, x) a_n^{\nu} F(c^{\nu}_{n,1}, c^{\nu}_{n,2}; c^{\nu}_{n,3}; x)$$

Irregular confluent hypergeometric function

$$R^+_{\hat{l}mkn}(r) \to R^+_{\hat{l}m}(\omega, r) = \sum_{n=-\infty}^{\infty} g^{\nu}(\omega, z) b_n^{\nu} (2z)^n \Psi(c^{\nu}_{n,4}, c^{\nu}_{n,5}; -2iz)$$

$$c^{\nu}_{n,i} = c^{\nu}_{n,i}(\omega)$$

ν Renormalized angular momentum

Building a generic SSF code

- Finding homogeneous solutions
  - **Mano-Suzuki-Takasugi (MST)** function expansion formalism
    - Series solution w/ introduction of free parameter ν

Sasaki & Tagoshi (2003) **LRR 6**

Hypergeometric function

$$R^{-}_{\hat{l}mkn}(r) \to R^{-}_{\hat{l}m}(\omega, r) = \sum_{n=-\infty}^{\infty} f^{\nu}(\omega, x) a_n^{\nu} F(c^{\nu}_{n,1}, c^{\nu}_{n,2}; c^{\nu}_{n,3}; x)$$

Irregular confluent hypergeometric function

$$R^{+}_{\hat{l}mkn}(r) \to R^{+}_{\hat{l}m}(\omega, r) = \sum_{n=-\infty}^{\infty} g^{\nu}(\omega, z) b_n^{\nu} (2z)^n \Psi(c^{\nu}_{n,4}, c^{\nu}_{n,5}; -2iz)$$

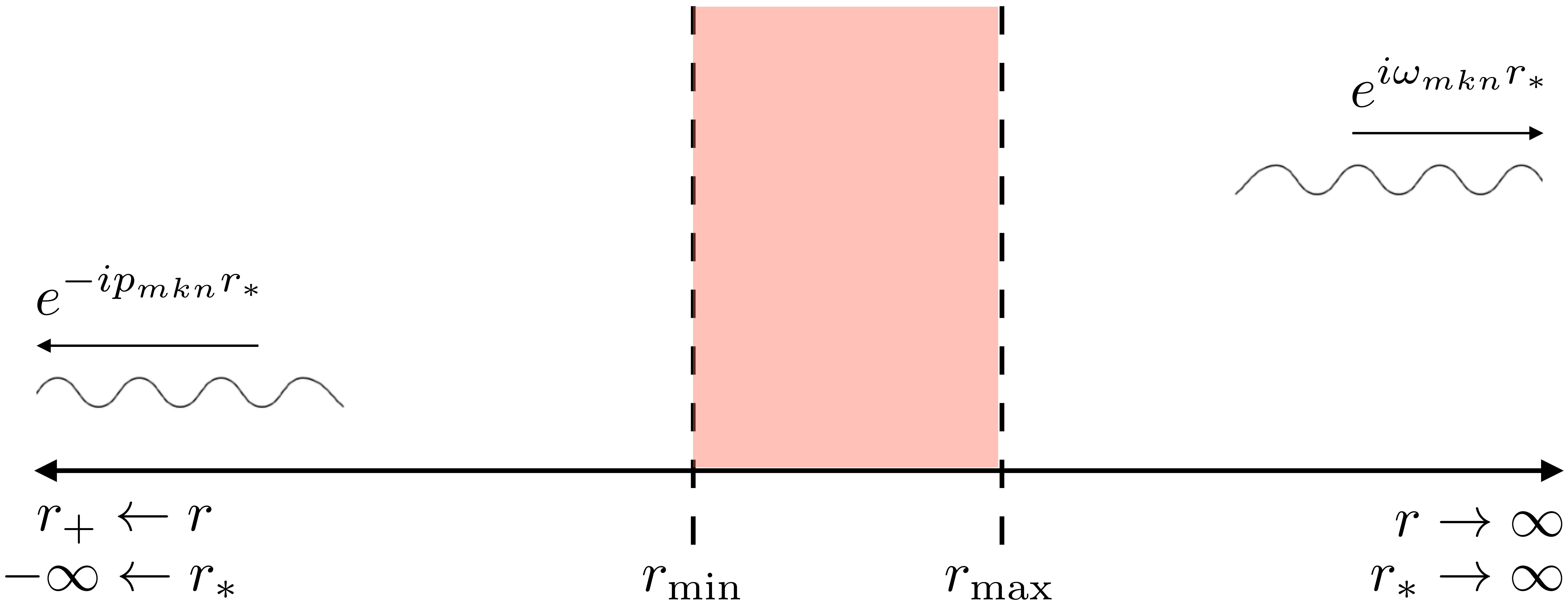
$$c^{\nu}_{n,i} = c^{\nu}_{n,i}(\omega)$$

ν Renormalized angular momentum

Black hole monodromy

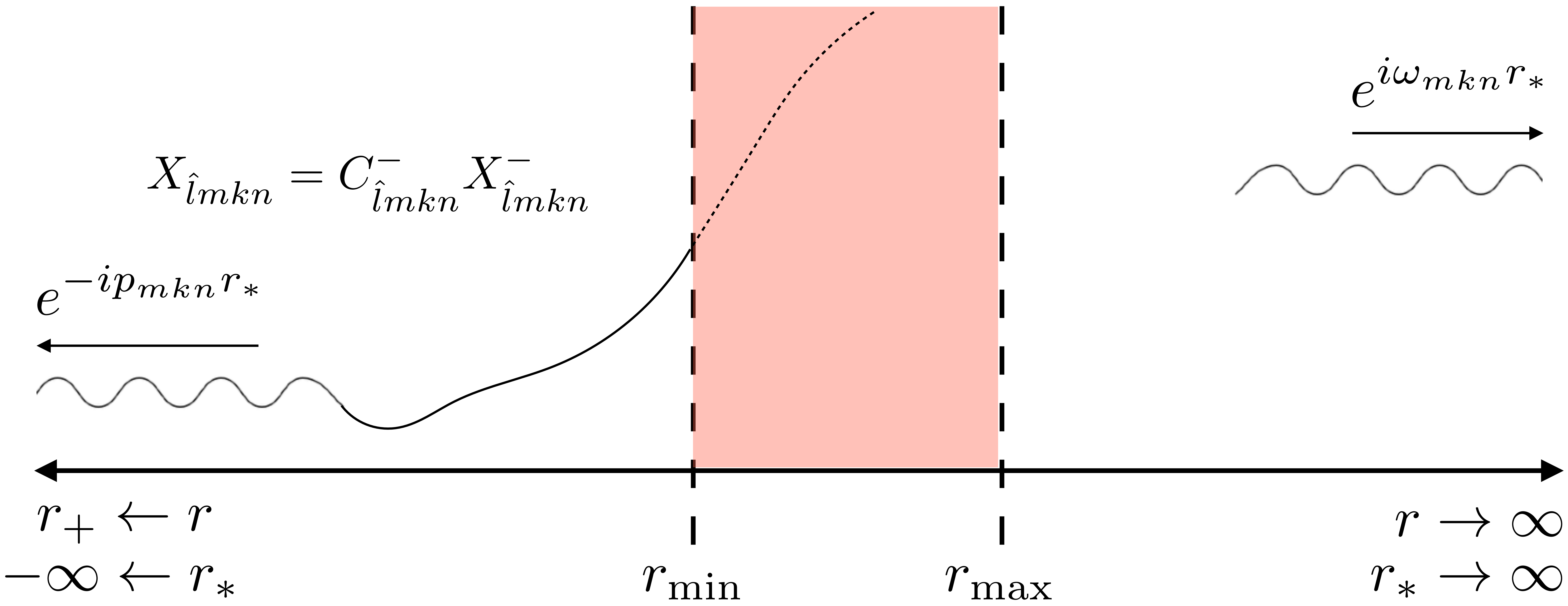
Building a generic SSF code

- Source integration & TD reconstruction



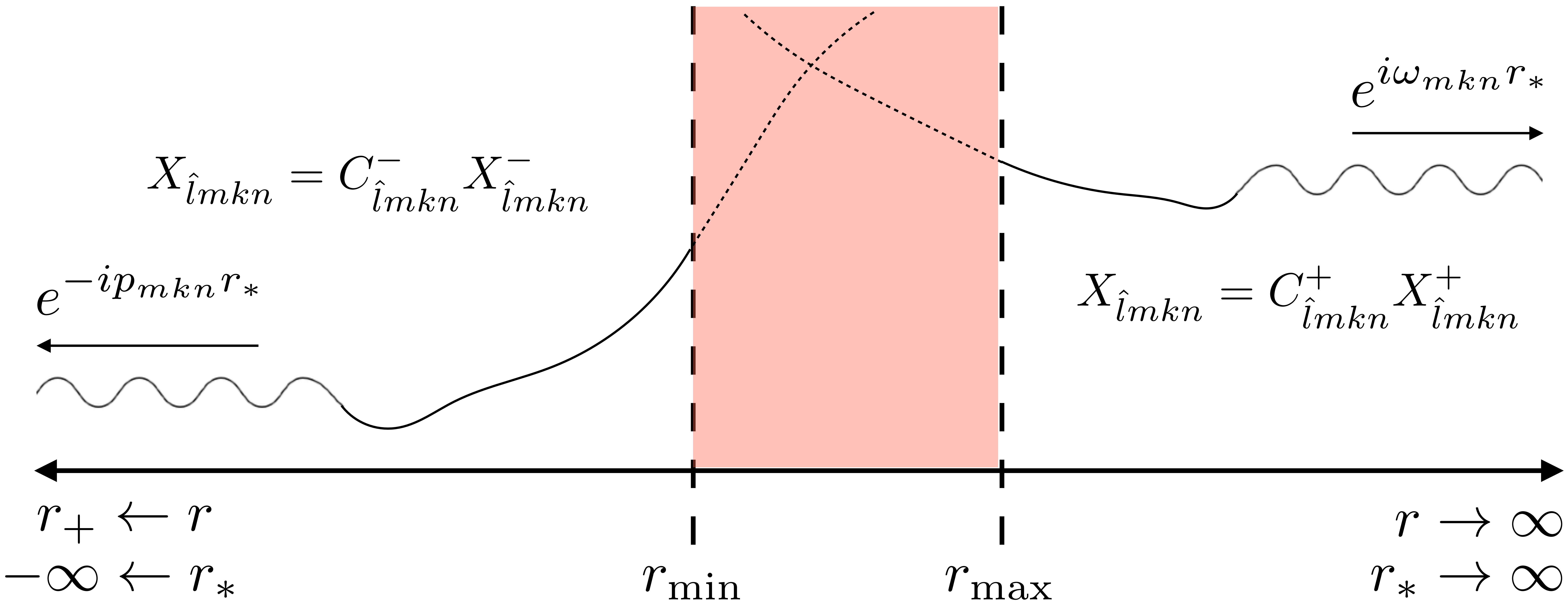
Building a generic SSF code

- Source integration & TD reconstruction



Building a generic SSF code

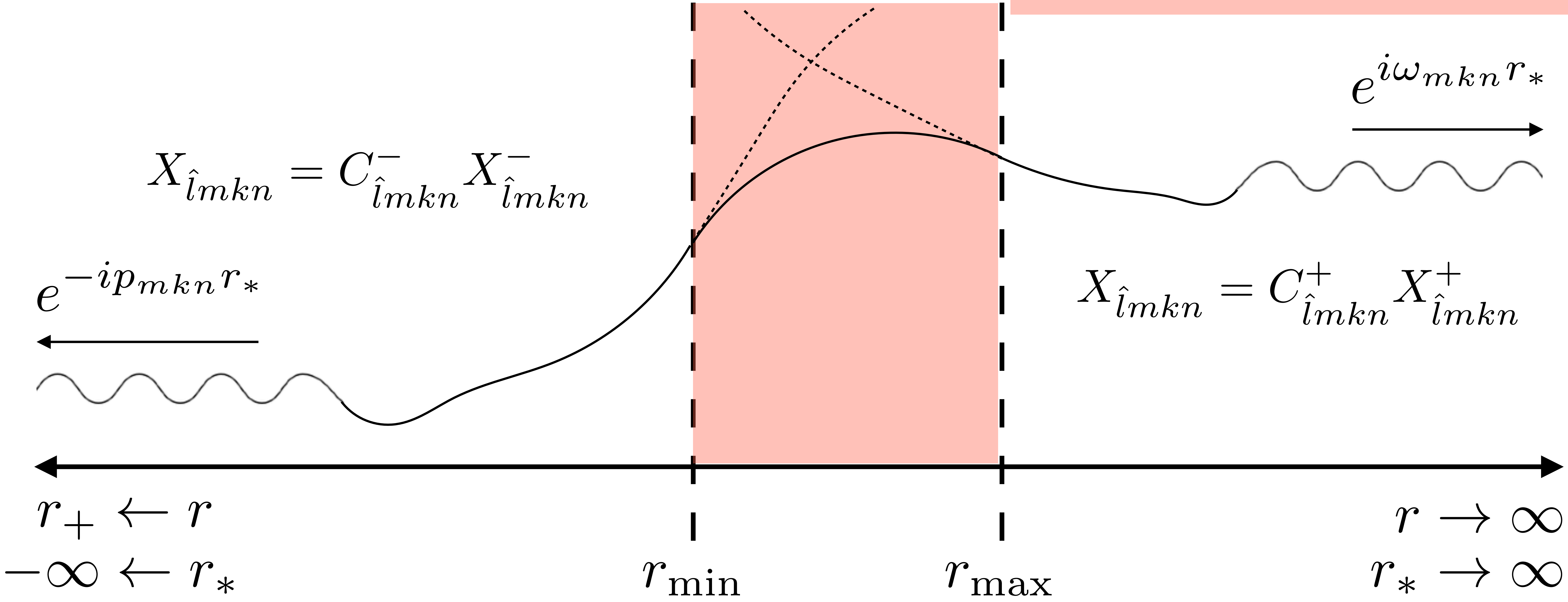
- Source integration & TD reconstruction



Building a generic SSF code

- Source integration & TD reconstruction

$$X_{\hat{l}mkn} = c^-_{\hat{l}mkn}(r)X^-_{\hat{l}mkn}(r) + c^+_{\hat{l}mkn}(r)X^+_{\hat{l}mkn}(r)$$



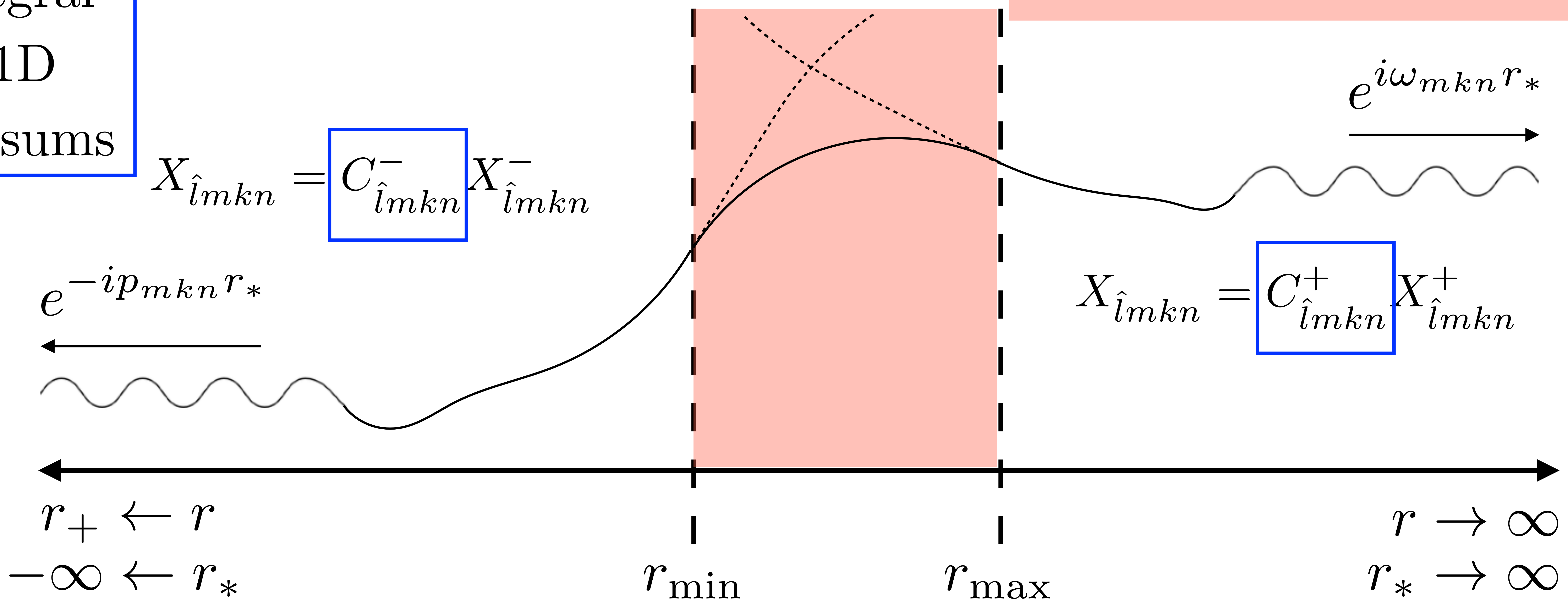
THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Building a generic SSF code

- Source integration & TD reconstruction

2D integral
$\to$ 4 1D
Fourier sums

$$X_{\hat{l}mkn} = c^-_{\hat{l}mkn}(r) X^-_{\hat{l}mkn}(r) + c^+_{\hat{l}mkn}(r) X^+_{\hat{l}mkn}(r)$$

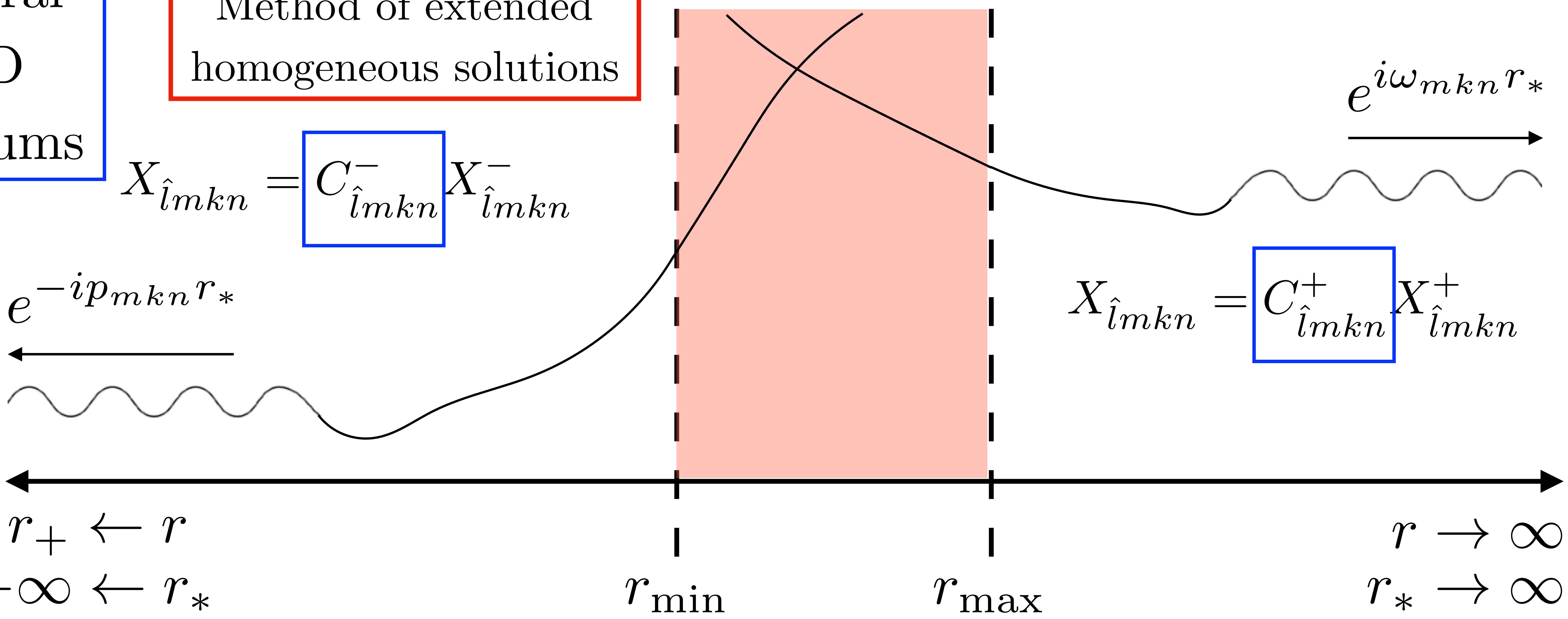


Building a generic SSF code

- Source integration & TD reconstruction

2D integral
→ 4 1D
Fourier sums

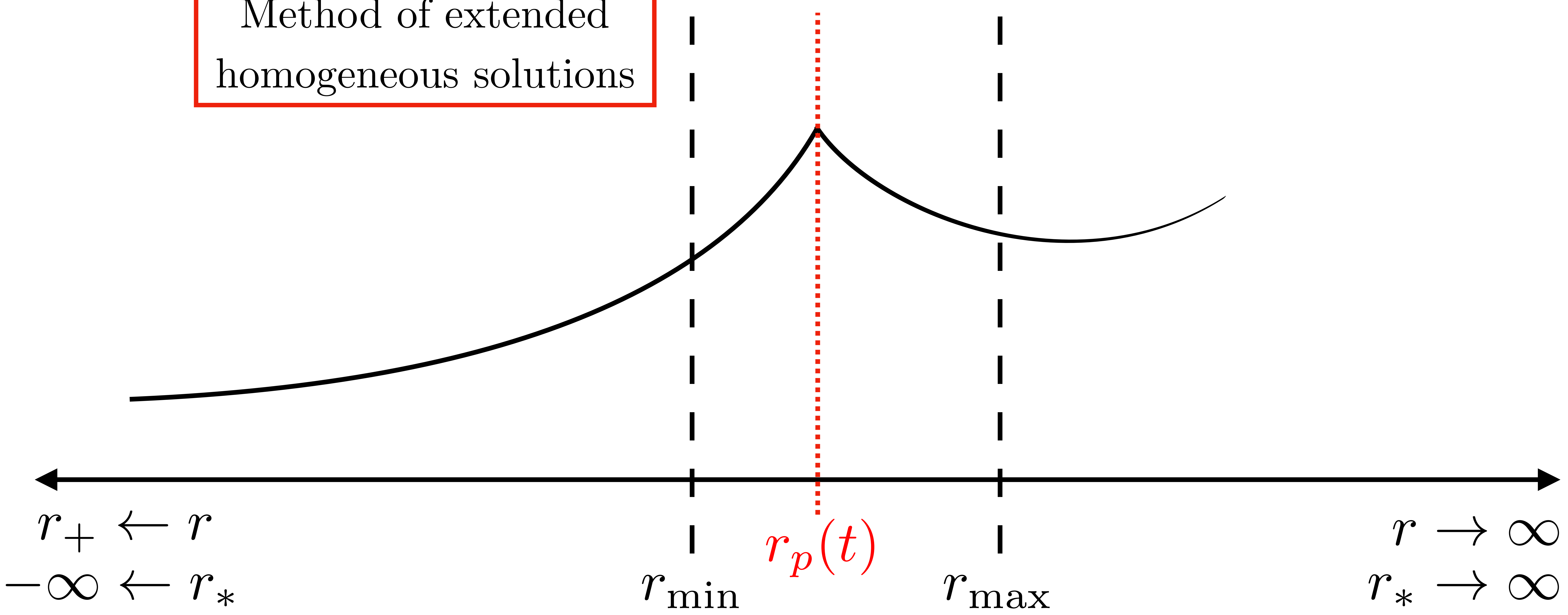
Method of extended
homogeneous solutions



Building a generic SSF code

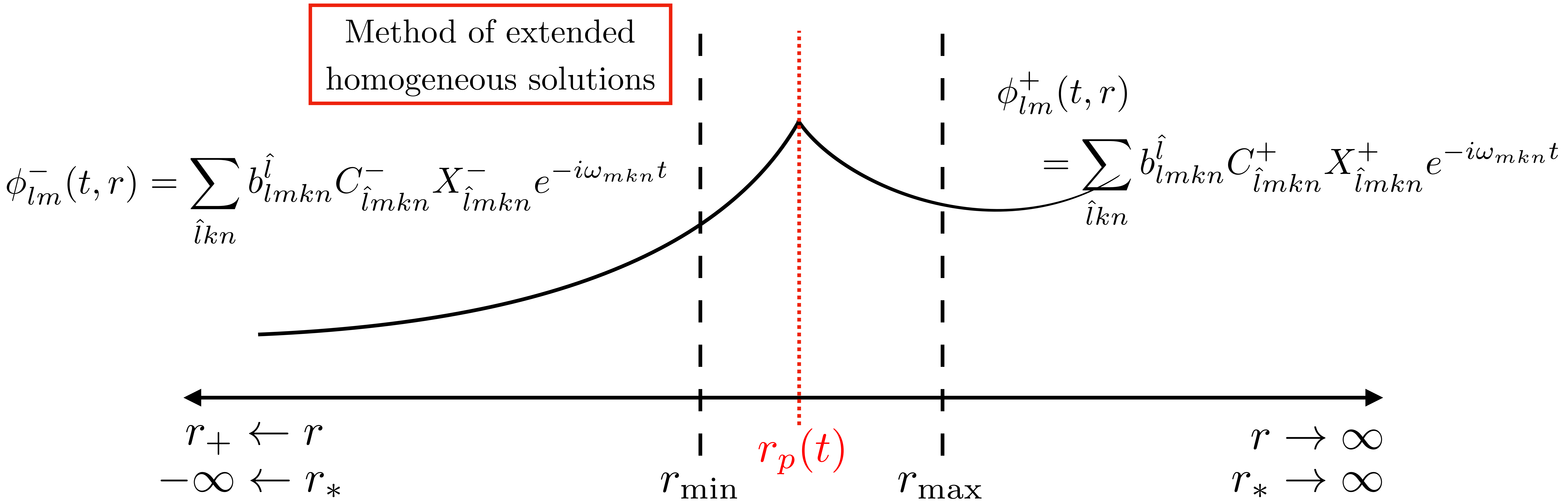
- Source integration & TD reconstruction

Method of extended homogeneous solutions



Building a generic SSF code

- Source integration & TD reconstruction



Building a generic SSF code

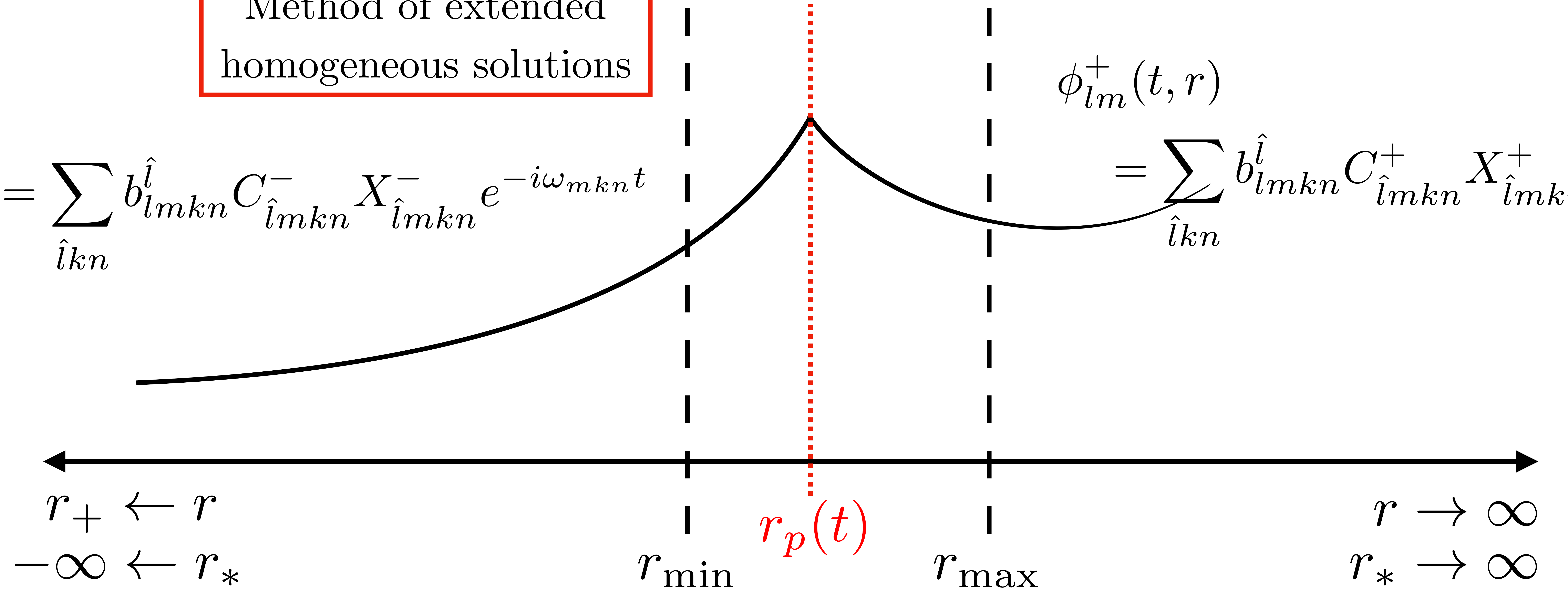
- Source integration & TD reconstruction

$$\Phi^{\text{ret}} = \sum_{lm} \phi^{\pm}_{lm}(t,r) Y_{lm}(\theta,\varphi)$$

Method of extended homogeneous solutions

$$\phi^{-}_{lm}(t,r) = \sum_{\hat{l}kn} b^{\hat{l}}_{lmkn} C^{-}_{\hat{l}mkn} X^{-}_{\hat{l}mkn} e^{-i\omega_{mkn}t}$$

$$\phi^{+}_{lm}(t,r) = \sum_{\hat{l}kn} b^{\hat{l}}_{lmkn} C^{+}_{\hat{l}mkn} X^{+}_{\hat{l}mkn} e^{-i\omega_{mkn}t}$$



Building a generic SSF code

- Mode-sum regularization

$$F_\mu = \sum_{l=0}^{\infty} (F_\mu^{\mathrm{ret},l} - F_\mu^{\mathrm{S},l})$$

$$F_\mu^{\mathrm{S},l} = A_\mu (l + 1/2) + B_\mu + \sum_{n=2}^{\infty} D_\mu^{2n} \left[\prod_{k=1}^{n-1} (2l - 2k + 1)(2l + 2k + 1) \right]^{-1}$$

$$\sum_{l=0}^{\infty} \left[\prod_{k=1}^{n-1} (2l - 2k + 1)(2l + 2k + 1) \right]^{-1} = 0$$

Building a generic SSF code

- Mode-sum regularization

$$F_\mu = \sum_{l=0}^{\infty} (F_\mu^{\mathrm{ret},l} - F_\mu^{\mathrm{S},l})$$

$$F_\mu^{\mathrm{S},l} = A_\mu (l+1/2) + B_\mu + \sum_{n=2}^{\infty} D_\mu^{2n} \left[\prod_{k=1}^{n-1} (2l-2k+1)(2l+2k+1) \right]^{-1}$$

$$\sum_{l=0}^{\infty} \left[\prod_{k=1}^{n-1} (2l-2k+1)(2l+2k+1) \right]^{-1} = 0$$

Building a generic SSF code

- Mode-sum regularization

$$F_\mu = \sum_{l=0}^{\infty} (F_\mu^{\mathrm{ret},l} - F_\mu^{\mathrm{S},l})$$

$$F_\mu^{\mathrm{S},l} = A_\mu (l + 1/2) + B_\mu + \sum_{n=2}^{\infty} D_\mu^{2n} \left[\prod_{k=1}^{n-1} (2l - 2k + 1)(2l + 2k + 1) \right]^{-1}$$

$$\sum_{l=0}^{\infty} \left[\prod_{k=1}^{n-1} (2l - 2k + 1)(2l + 2k + 1) \right]^{-1} = 0$$

Building a generic SSF code

- Mode-sum regularization

$$F_\mu = \sum_{l=0}^{\infty} (F_\mu^{\mathrm{ret},l} - F_\mu^{\mathrm{S},l})$$

$$F_\mu^{\mathrm{S},l} = A_\mu (l + 1/2) + B_\mu + \sum_{n=2}^{\infty} D_\mu^{2n} \left[\prod_{k=1}^{n-1} (2l - 2k + 1)(2l + 2k + 1) \right]^{-1}$$

$$\sum_{l=0}^{\infty} \left[\prod_{k=1}^{n-1} (2l - 2k + 1)(2l + 2k + 1) \right]^{-1} = 0$$

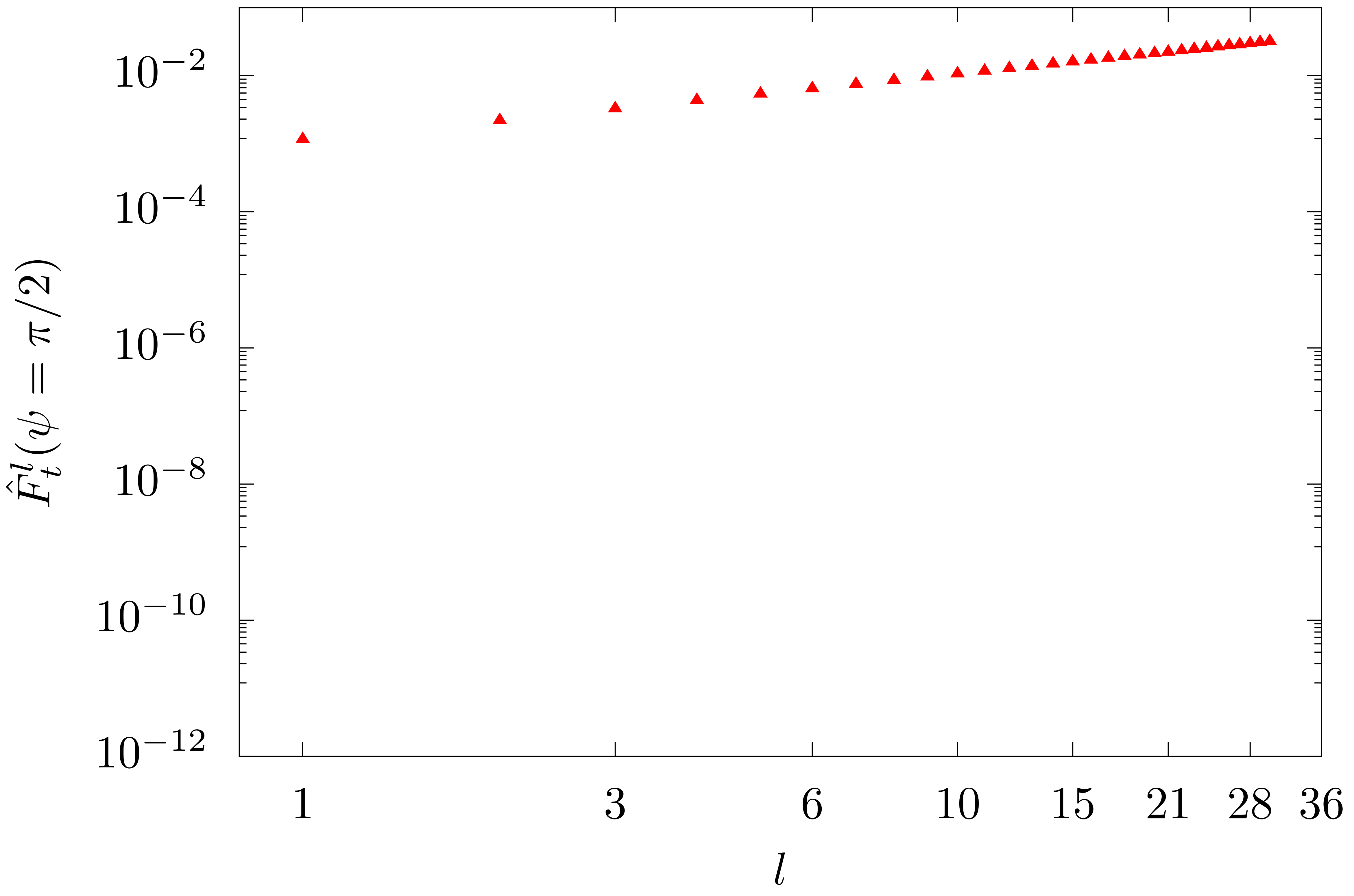
Building a generic SSF code

- Mode-sum regularization

$$F_\mu = \sum_{l=0}^{\infty} (F_\mu^{\mathrm{ret},l} - F_\mu^{\mathrm{S},l})$$

$$F_\mu^{\mathrm{S},l} = A_\mu (l+1/2) + B_\mu + \sum_{n=2}^{\infty} D_\mu^{2n} \left[\prod_{k=1}^{n-1} (2l-2k+1)(2l+2k+1) \right]^{-1}$$

$$\boxed{\sum_{l=0}^{\infty} \left[\prod_{k=1}^{n-1} (2l-2k+1)(2l+2k+1) \right]^{-1} = 0}$$



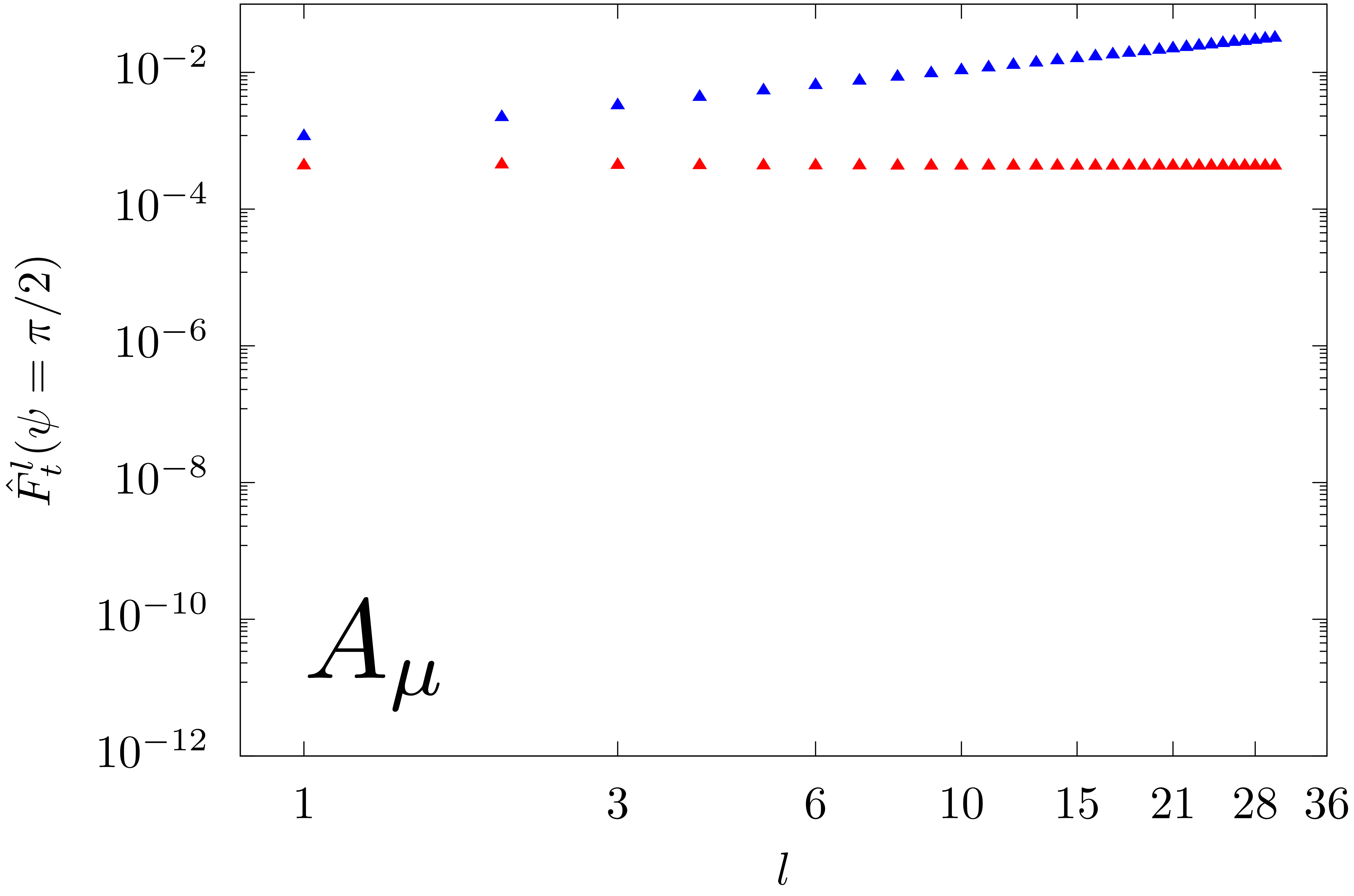
Building a generic SSF code

- Mode-sum regularization

$$F_\mu = \sum_{l=0}^{\infty} (F_\mu^{\text{ret},l} - F_\mu^{\text{S},l})$$

$$F_\mu^{\text{S},l} = A_\mu (l + 1/2) + B_\mu + \sum_{n=2}^{\infty} D_\mu^{2n} \left[\prod_{k=1}^{n-1} (2l - 2k + 1)(2l + 2k + 1) \right]^{-1}$$

$$\sum_{l=0}^{\infty} \left[\prod_{k=1}^{n-1} (2l - 2k + 1)(2l + 2k + 1) \right]^{-1} = 0$$



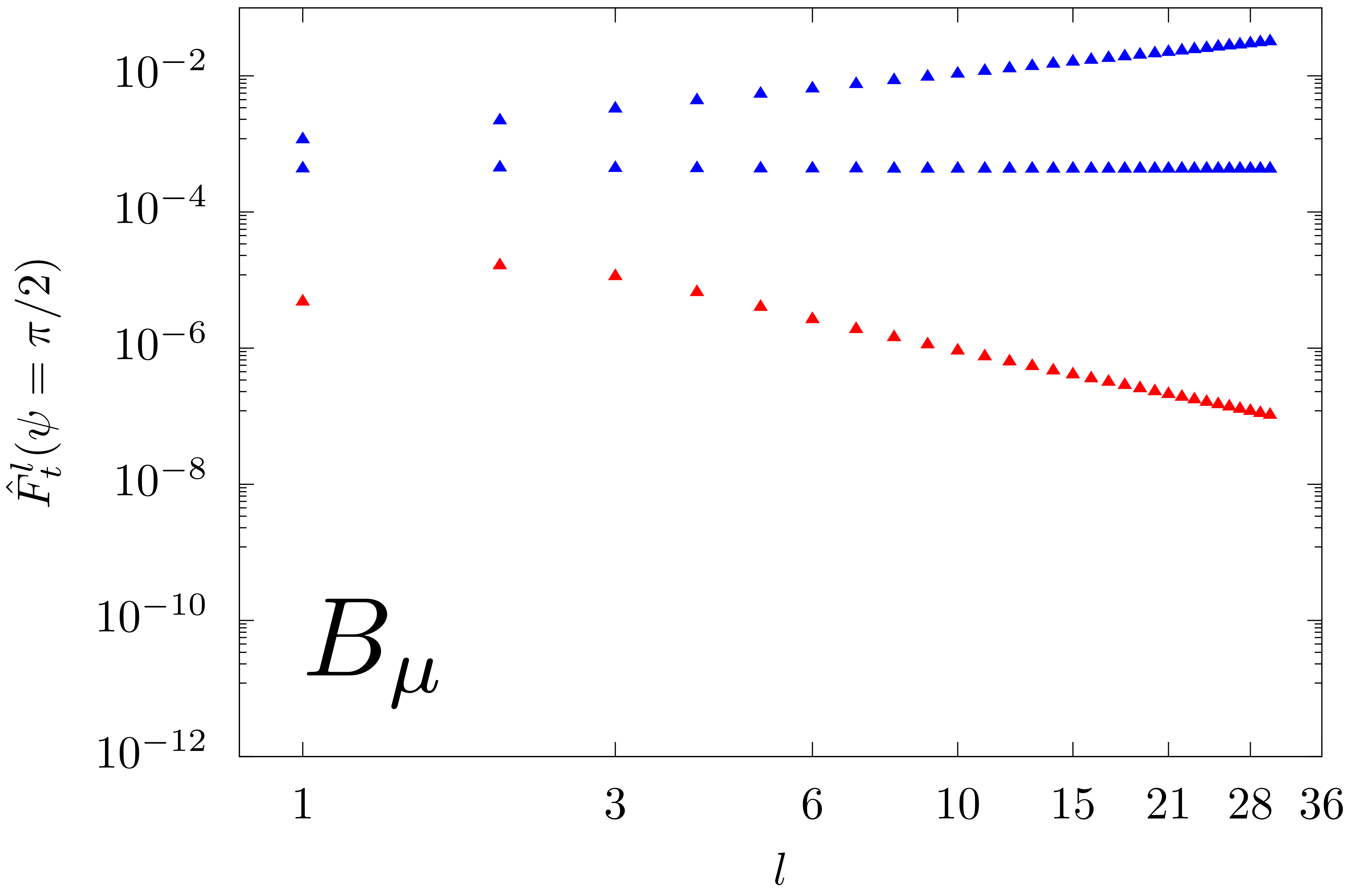
Building a generic SSF code

- Mode-sum regularization

$$F_\mu = \sum_{l=0}^{\infty} (F_\mu^{\mathrm{ret},l} - F_\mu^{\mathrm{S},l})$$

$$F_\mu^{\mathrm{S},l} = A_\mu (l+1/2) + B_\mu + \sum_{n=2}^{\infty} D_\mu^{2n} \left[\prod_{k=1}^{n-1} (2l-2k+1)(2l+2k+1) \right]^{-1}$$

$$\sum_{l=0}^{\infty} \left[\prod_{k=1}^{n-1} (2l-2k+1)(2l+2k+1) \right]^{-1} = 0$$



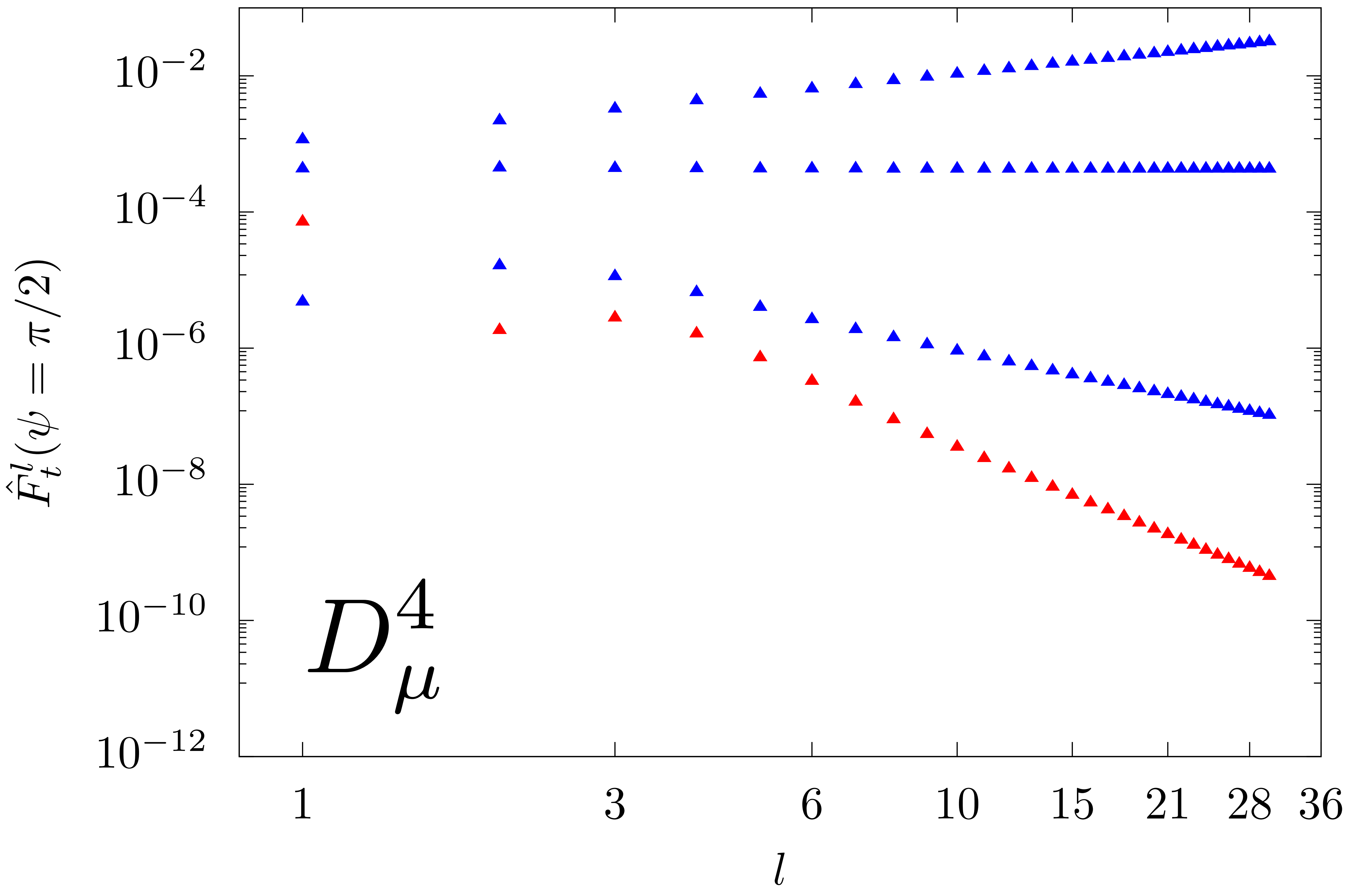
Building a generic SSF code

- Mode-sum regularization

$$F_\mu = \sum_{l=0}^{\infty} (F_\mu^{\mathrm{ret},l} - F_\mu^{\mathrm{S},l})$$

$$F_\mu^{\mathrm{S},l} = A_\mu(l+1/2) + B_\mu + \sum_{n=2}^{\infty} D_\mu^{2n} \left[\prod_{k=1}^{n-1}(2l-2k+1)(2l+2k+1)\right]^{-1}$$

$$\sum_{l=0}^{\infty} \left[\prod_{k=1}^{n-1}(2l-2k+1)(2l+2k+1)\right]^{-1} = 0$$



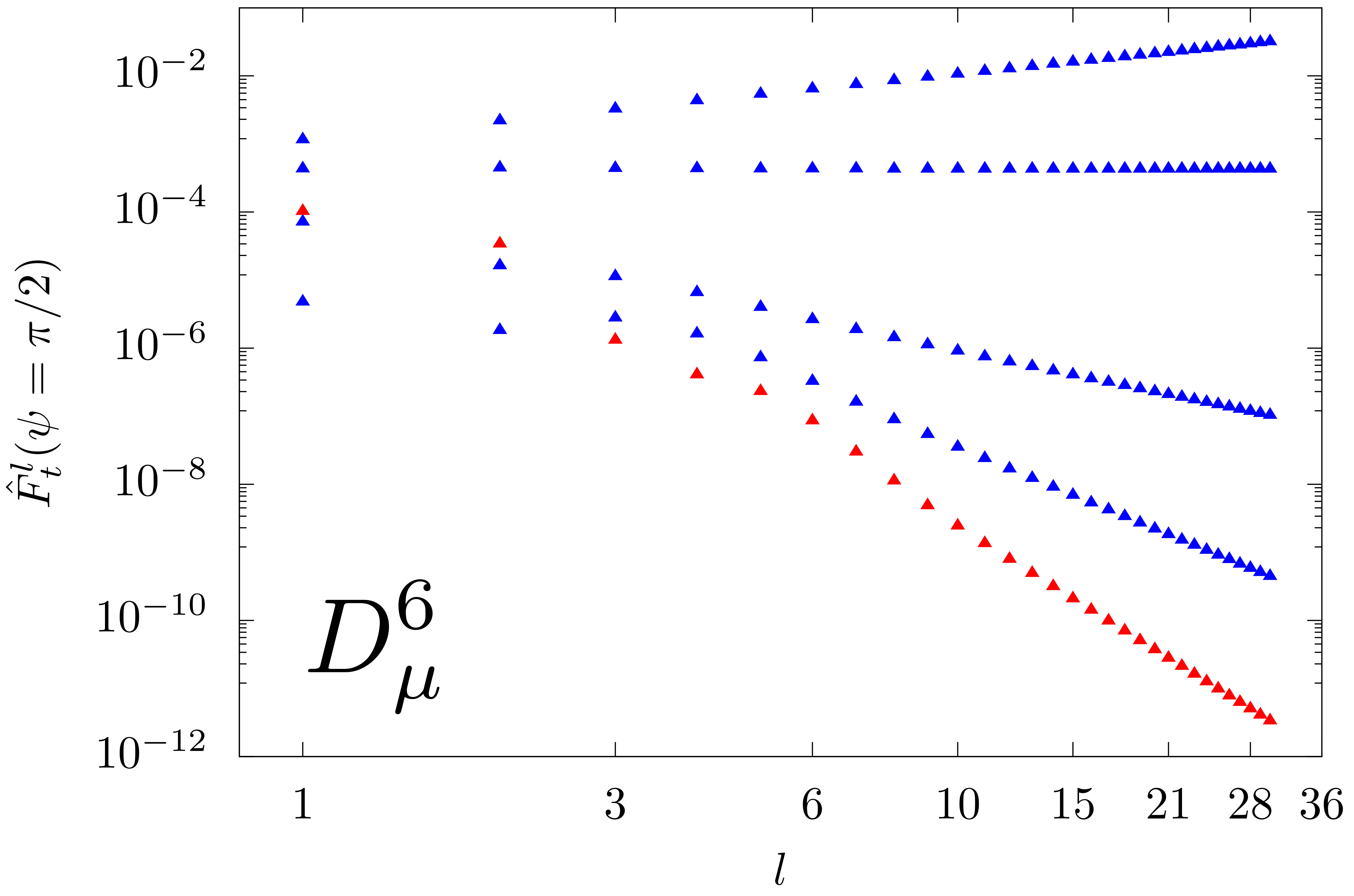
Building a generic SSF code

- Mode-sum regularization

$$F_\mu = \sum_{l=0}^{\infty} (F_\mu^{\mathrm{ret},l} - F_\mu^{\mathrm{S},l})$$

$$F_\mu^{\mathrm{S},l} = A_\mu (l + 1/2) + B_\mu + \sum_{n=2}^{\infty} D_\mu^{2n} \left[\prod_{k=1}^{n-1} (2l - 2k + 1)(2l + 2k + 1) \right]^{-1}$$

$$\sum_{l=0}^{\infty} \left[\prod_{k=1}^{n-1} (2l - 2k + 1)(2l + 2k + 1) \right]^{-1} = 0$$



Building a generic SSF code

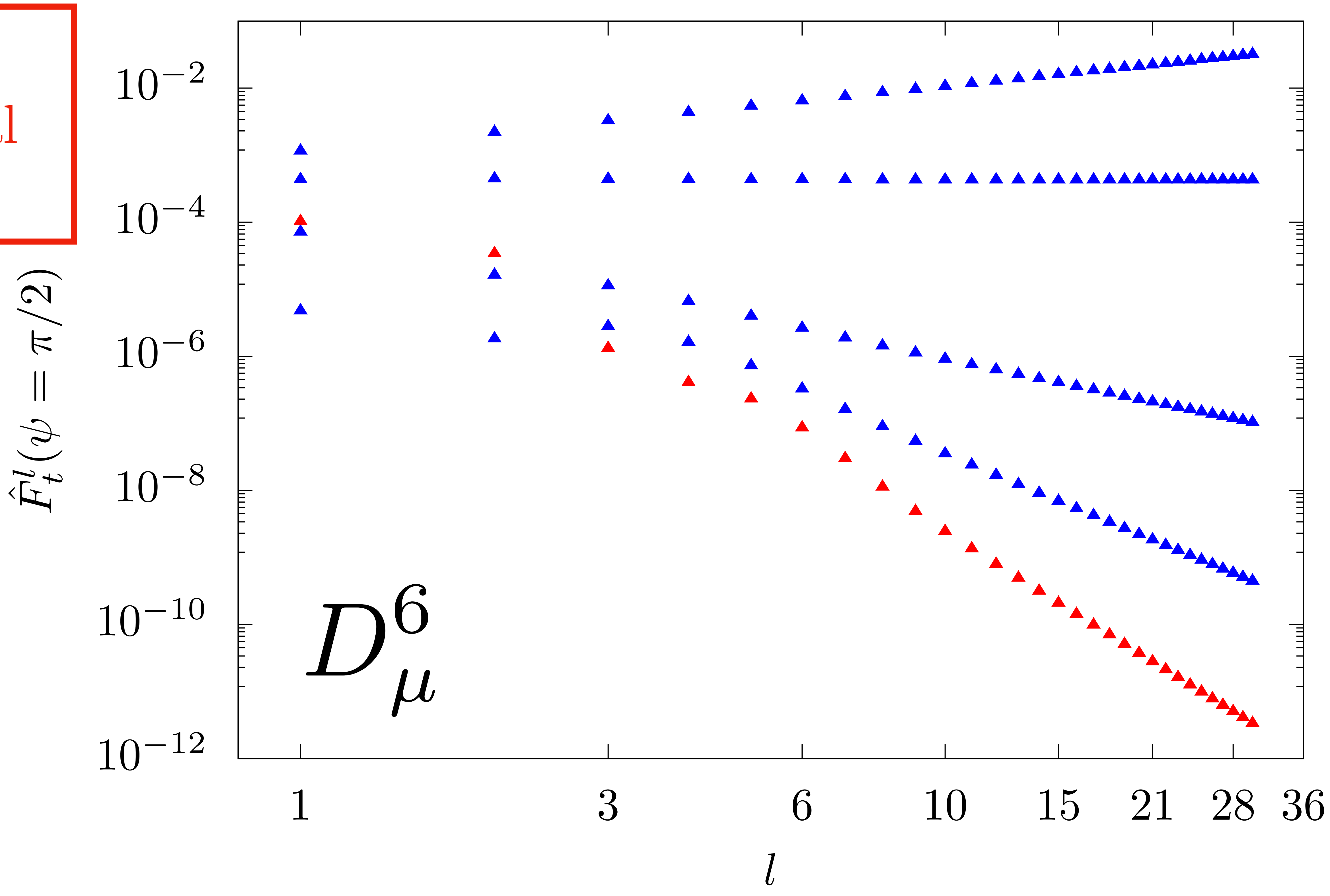
- Mode-sum regularization

$$F_\mu = \sum_{l=0}^{\infty} (F_\mu^{\mathrm{ret},l} - F_\mu^{\mathrm{S},l})$$

Higher-order RPs known for equatorial orbits

$$F_\mu^{\mathrm{S},l} = A_\mu (l+1/2) + B_\mu + \sum_{n=2}^{\infty} D_\mu^{2n} \left[\prod_{k=1}^{n-1} (2l-2k+1)(2l+2k+1) \right]^{-1}$$

$$\sum_{l=0}^{\infty} \left[\prod_{k=1}^{n-1} (2l-2k+1)(2l+2k+1) \right]^{-1} = 0$$



Building a generic SSF code

- Mode-sum regularization

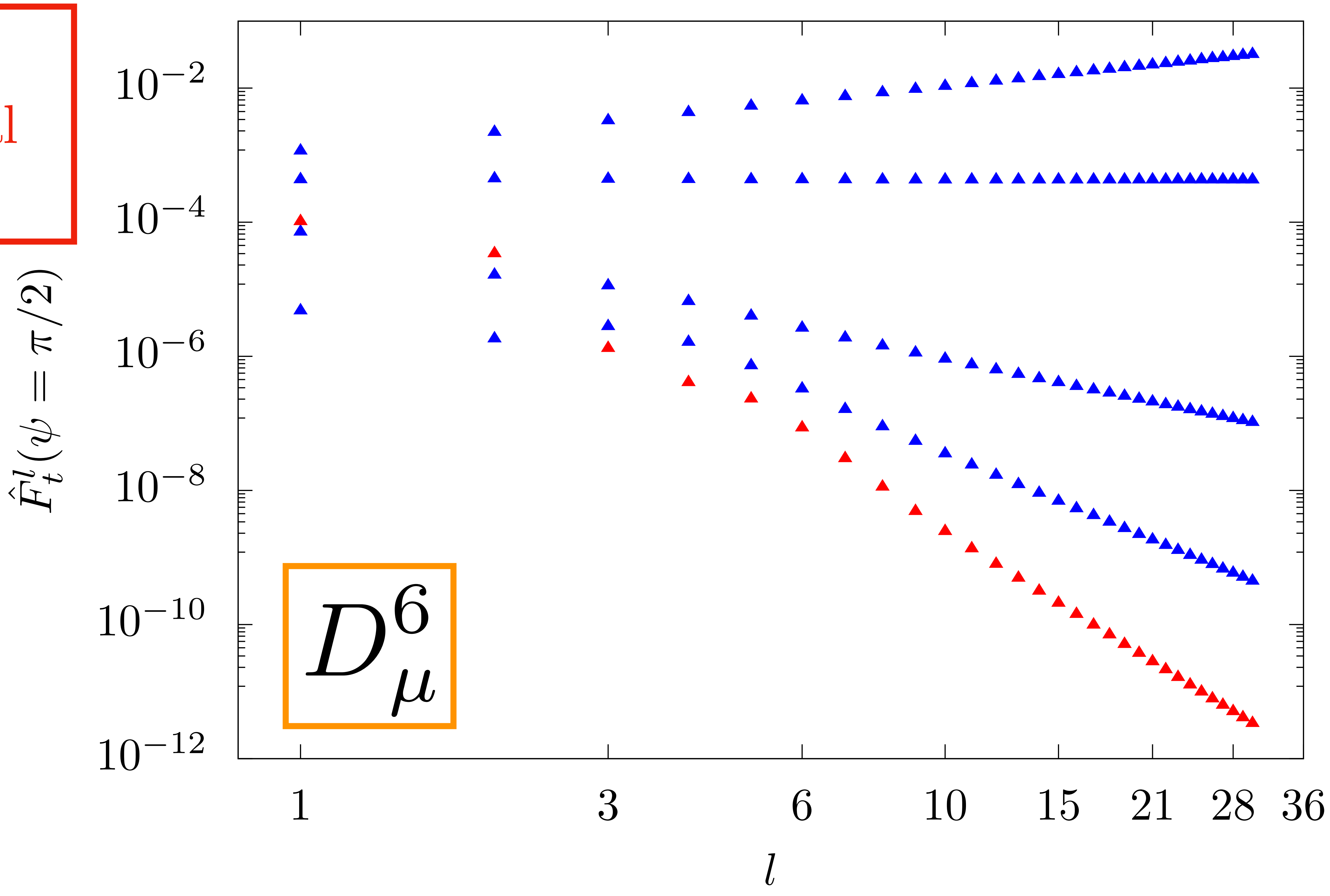
$$F_\mu = \sum_{l=0}^{\infty} (F_\mu^{\mathrm{ret},l} - F_\mu^{\mathrm{S},l})$$

Higher-order RPs known for equatorial orbits

$$F_\mu^{\mathrm{S},l} = A_\mu (l+1/2) + B_\mu + \sum_{n=2}^{\infty} D_\mu^{2n} \left[\prod_{k=1}^{n-1} (2l-2k+1)(2l+2k+1) \right]^{-1}$$

$$\sum_{l=0}^{\infty} \left[\prod_{k=1}^{n-1} (2l-2k+1)(2l+2k+1) \right]^{-1} = 0$$

Higher-order parameters for generic orbits?
See Anna Heffernan's talk Tuesday



Schwarzschild test

- **Equatorial v. Inclined**

$p = 10, e = 0.5, \iota = 0, a = 0$

$p = 10, e = 0.5, \iota = \pi/5, a = 0$

Schwarzschild test

- **Equatorial v. Inclined**

$p = 10, e = 0.5, \iota = 0, a = 0$

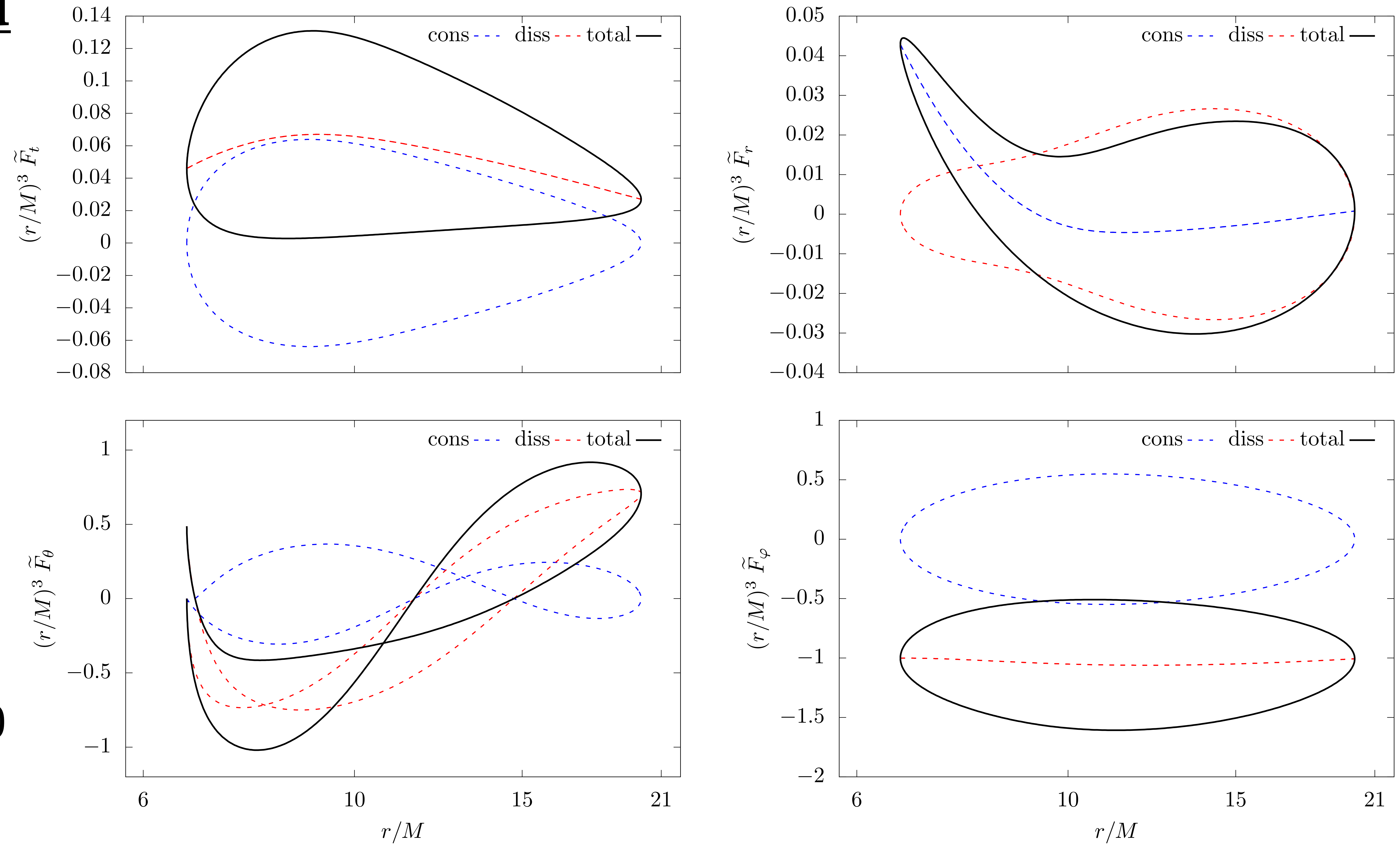
$p = 10, e = 0.5, \iota = \pi/5, a = 0$

Schwarzschild test

- **<u>Equatorial v. Inclined</u>**

  $p = 10, e = 0.5, \iota = 0, a = 0$

$p = 10, e = 0.5, \iota = \pi/5, a = 0$

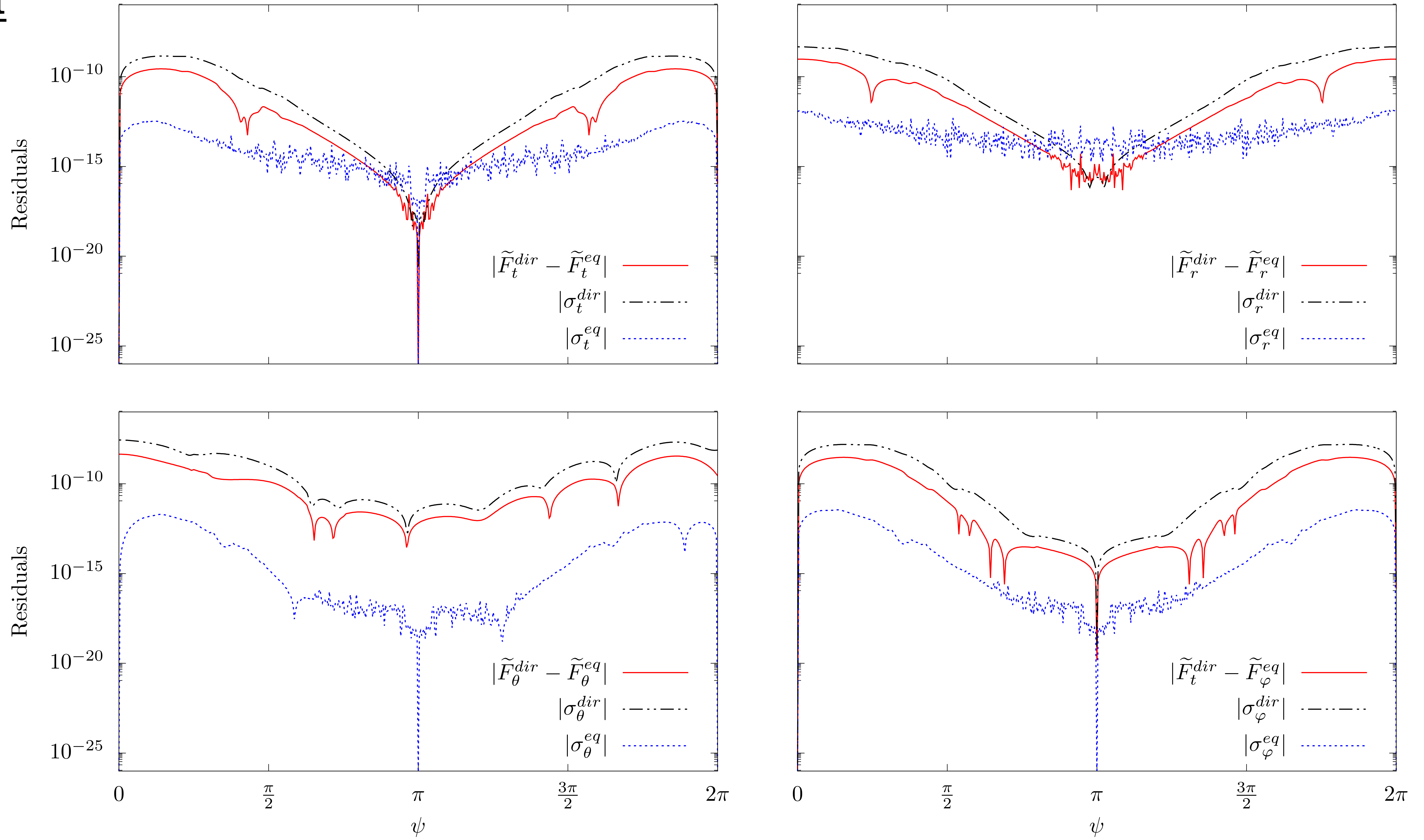


Schwarzschild test

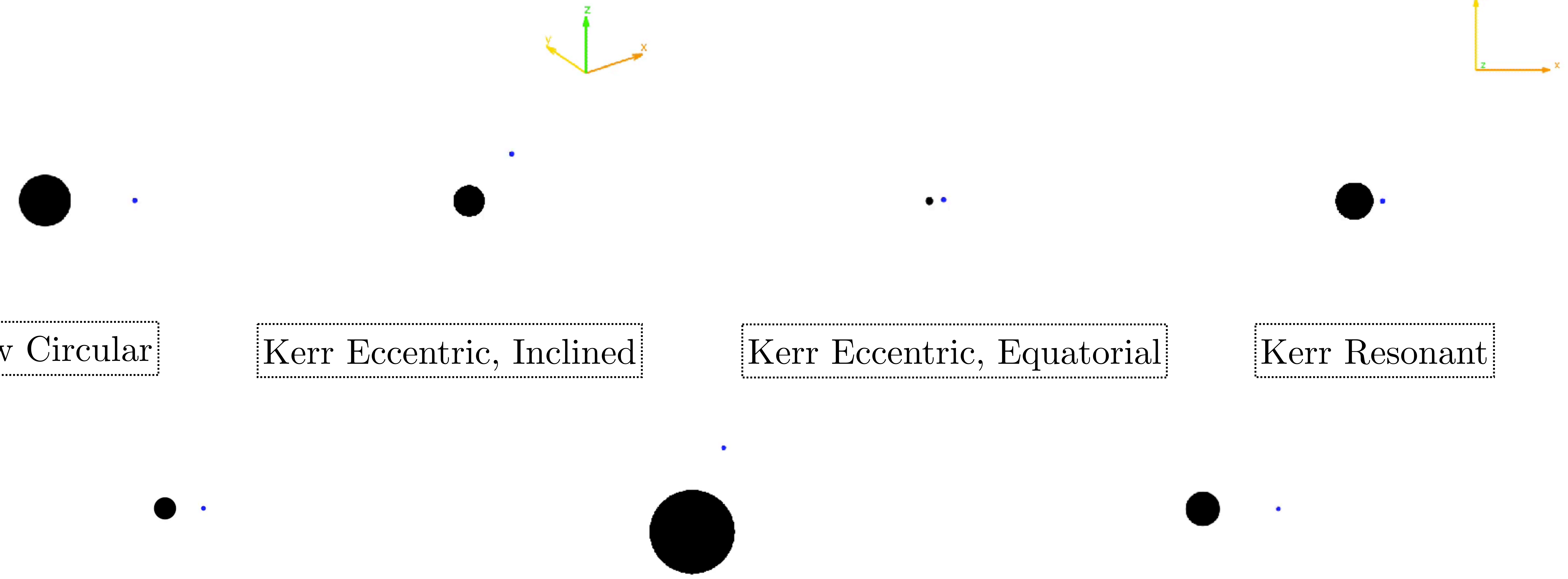
- **Equatorial v. Inclined**

$p = 10, e = 0.5, \iota = 0, a = 0$

$p = 10, e = 0.5, \iota = \pi/5, a = 0$



Orbits analyzed by my SSF code



Schw Circular

Kerr Eccentric, Inclined

Kerr Eccentric, Equatorial

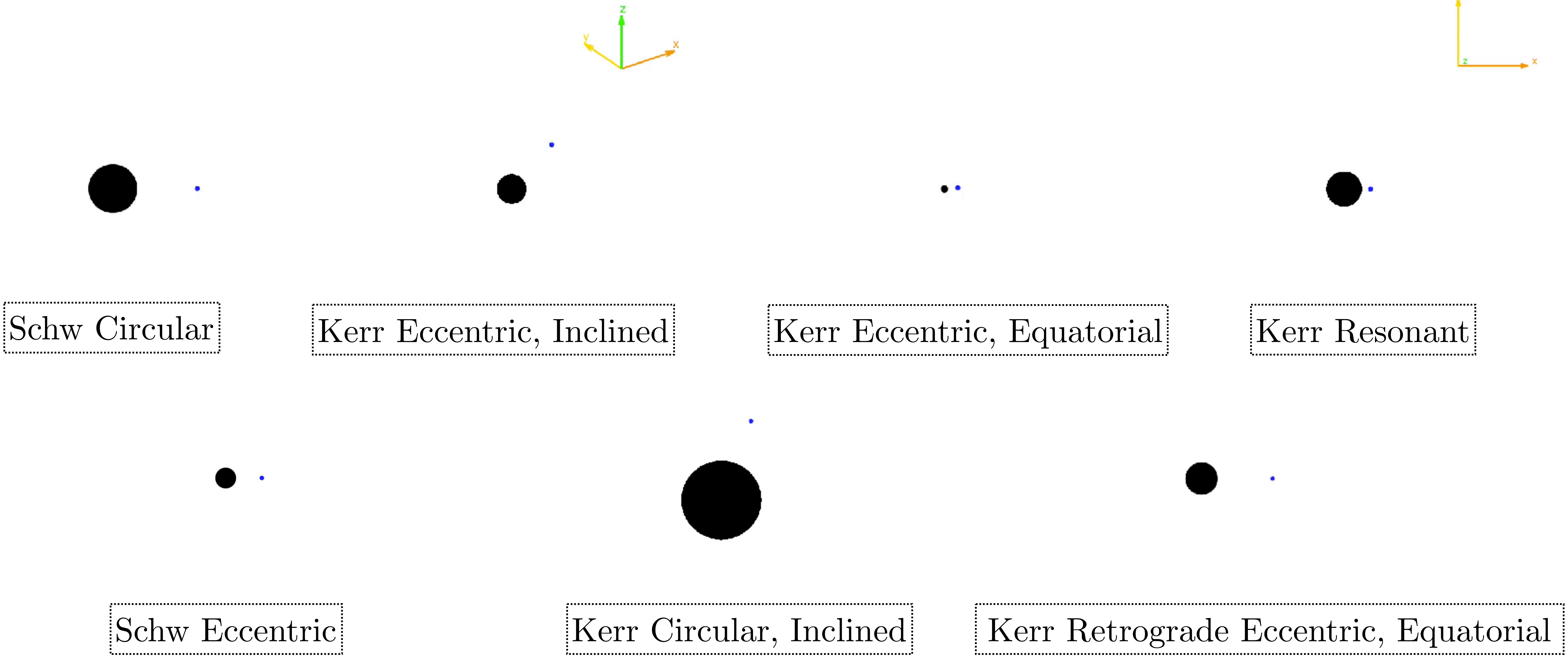
Kerr Resonant

Schw Eccentric

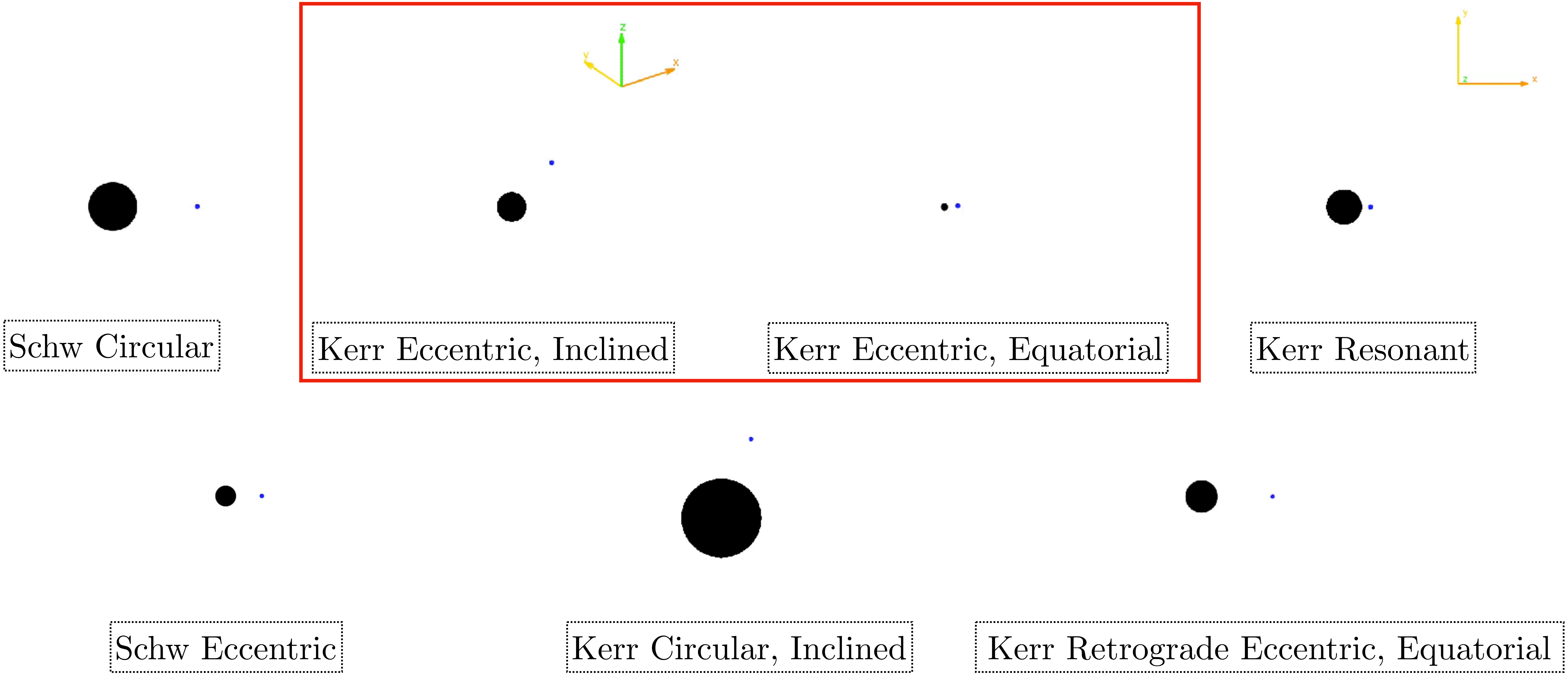
Kerr Circular, Inclined

Kerr Retrograde Eccentric, Equatorial

Orbits analyzed by my SSF code



Orbits analyzed by my SSF code



Highly eccentric orbit

$p = 8, e = 0.8, \iota = 0, a = 0.99M$

Highly eccentric orbit

$p = 8, e = 0.8, \iota = 0, a = 0.99M$

Highly eccentric orbit

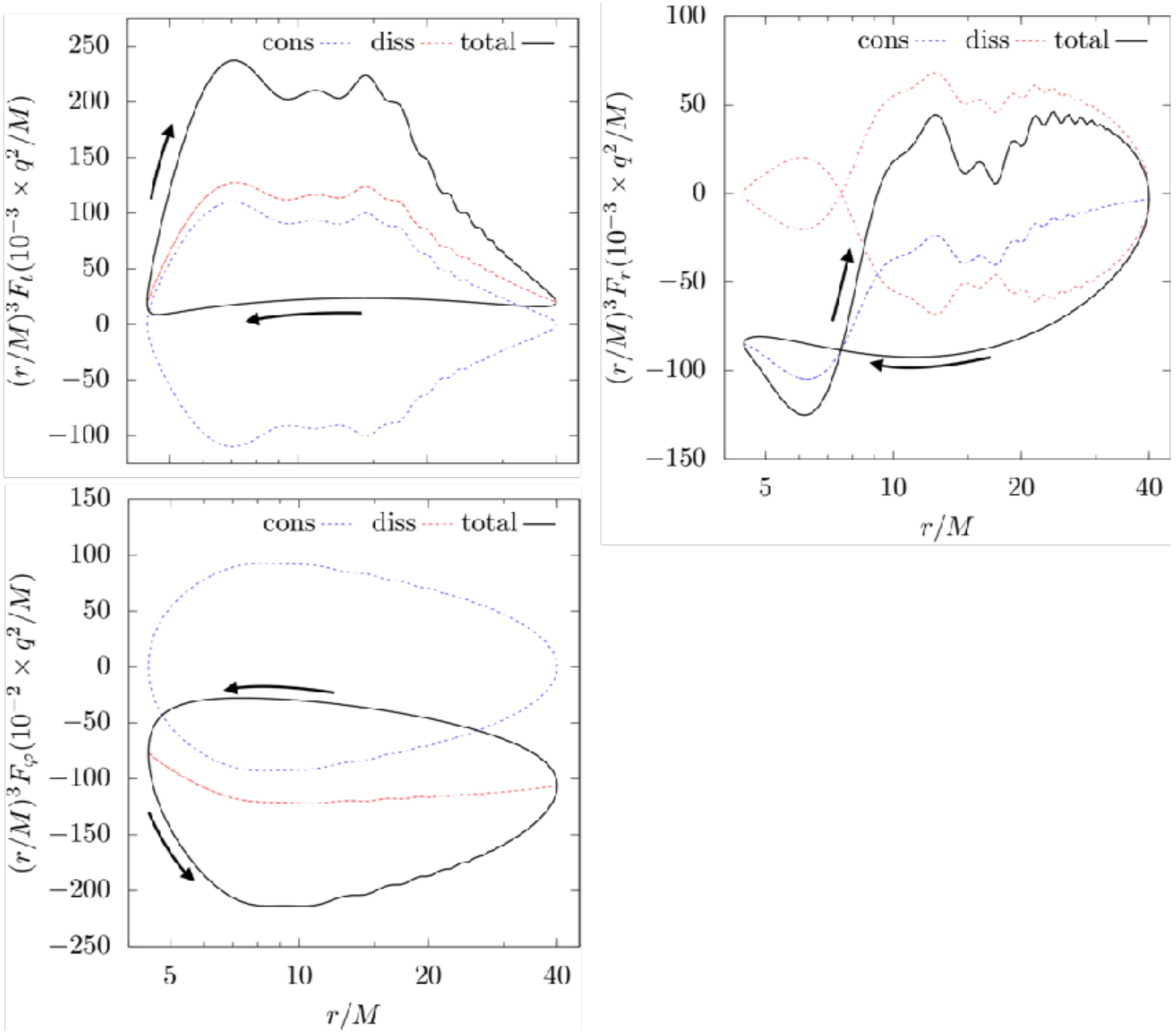
$p = 8, e = 0.8, \iota = 0, a = 0.99M$

[1610.09319]

- Inspired by Thornburg & Wardell (2017): Time-domain, equatorial Kerr SSF code

Highly eccentric orbit

$p = 8, e = 0.8, \iota = 0, a = 0.99M$



[1610.09319]

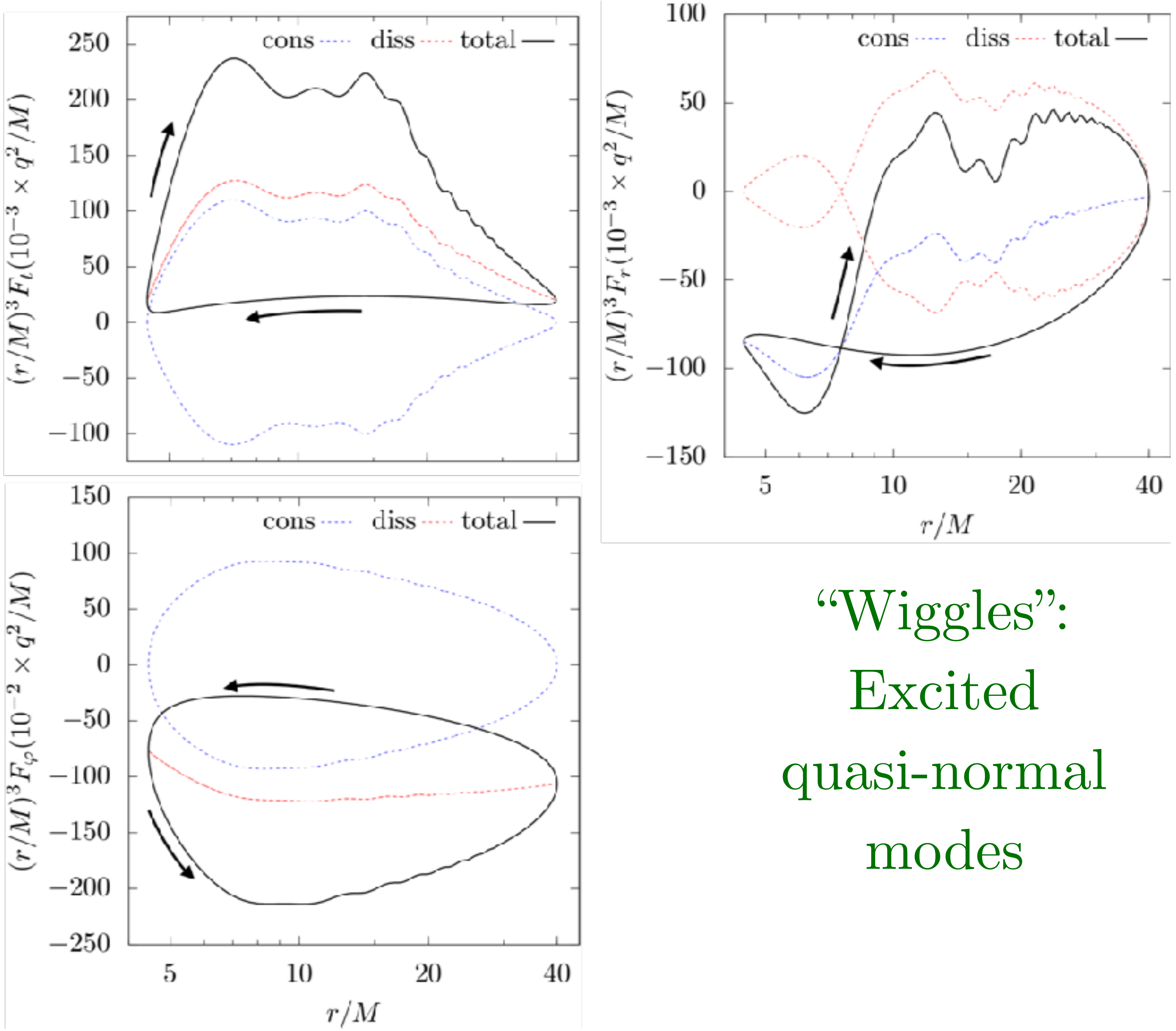
- Inspired by Thornburg & Wardell (2017): Time-domain, equatorial Kerr SSF code

Highly eccentric orbit

$p = 8, e = 0.8, \iota = 0, a = 0.99M$

[1610.09319]

- Inspired by Thornburg & Wardell (2017): Time-domain, equatorial Kerr SSF code

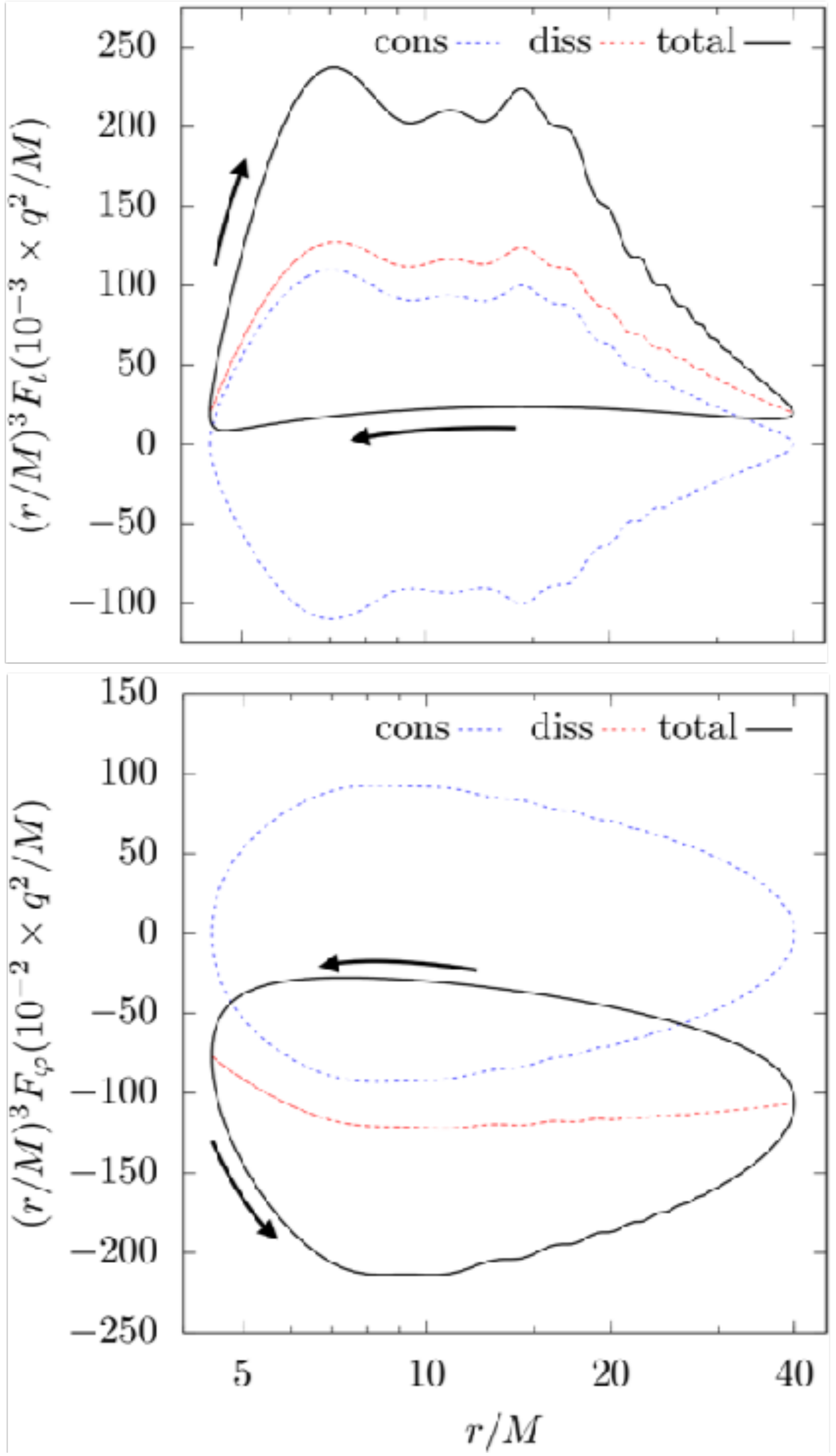


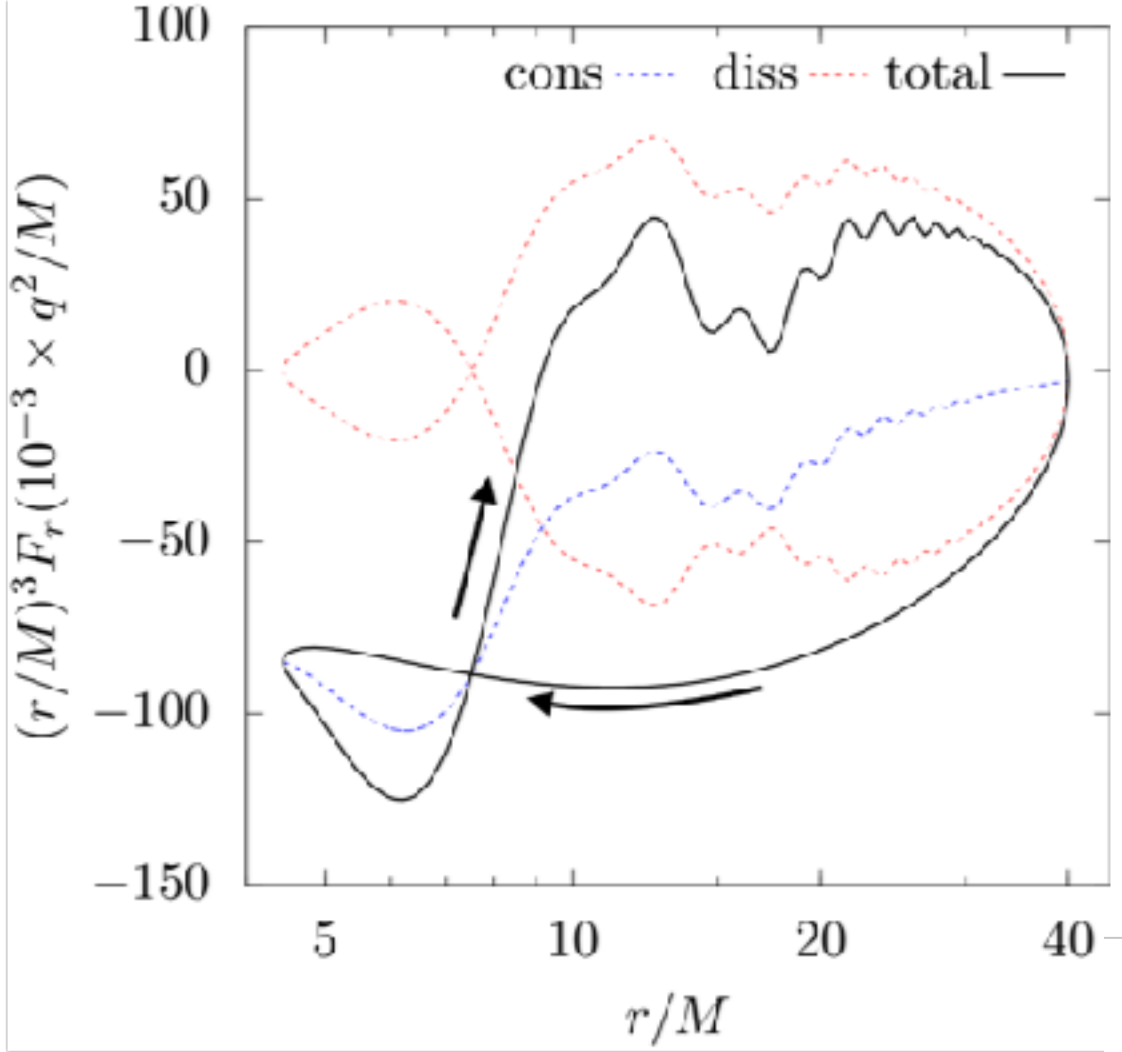
"Wiggles": Excited quasi-normal modes

Highly eccentric orbit

$p = 8, e = 0.8, \iota = 0, a = 0.99M$

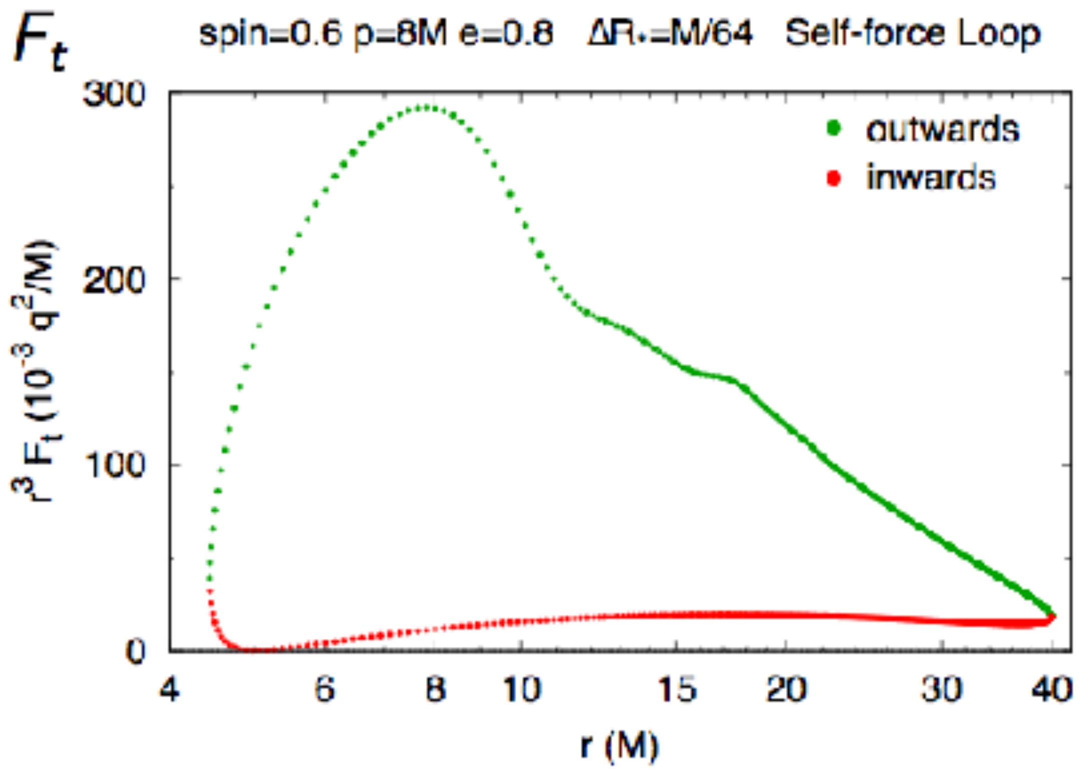
[1610.09319]



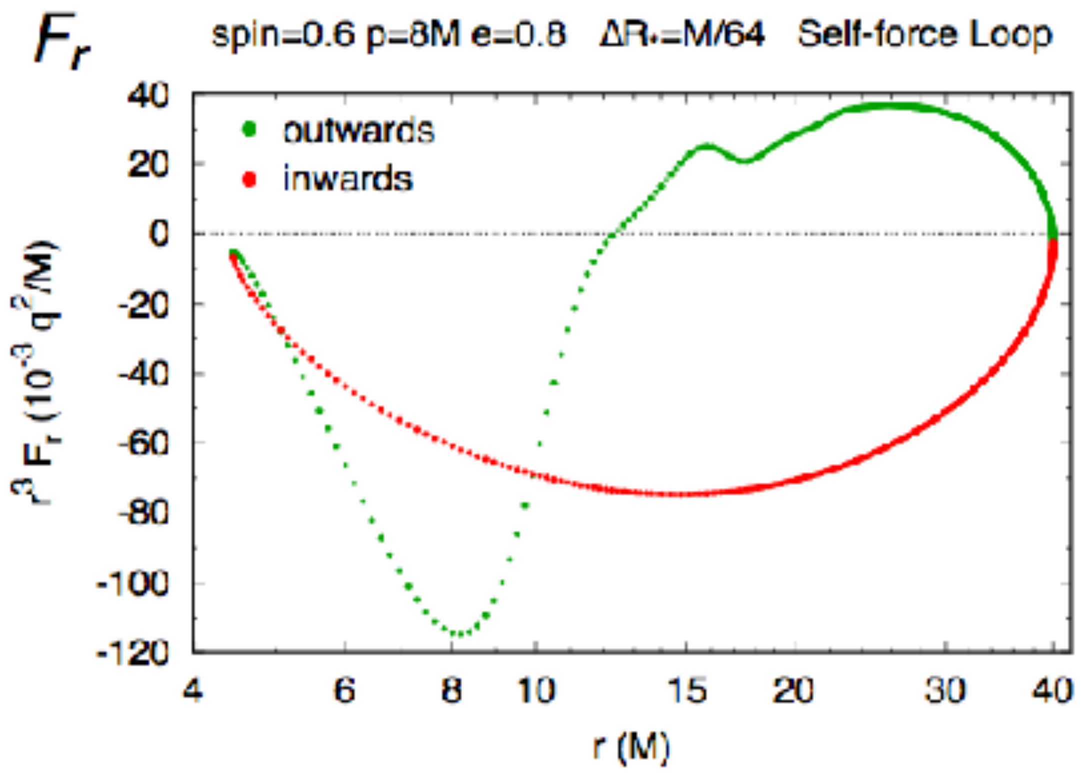


- Inspired by Thornburg & Wardell (2017): Time-domain, equatorial Kerr SSF code

$$p = 8, e = 0.8, \iota = 0, a = 0.6M$$



“Wiggles”: Excited quasi-normal modes



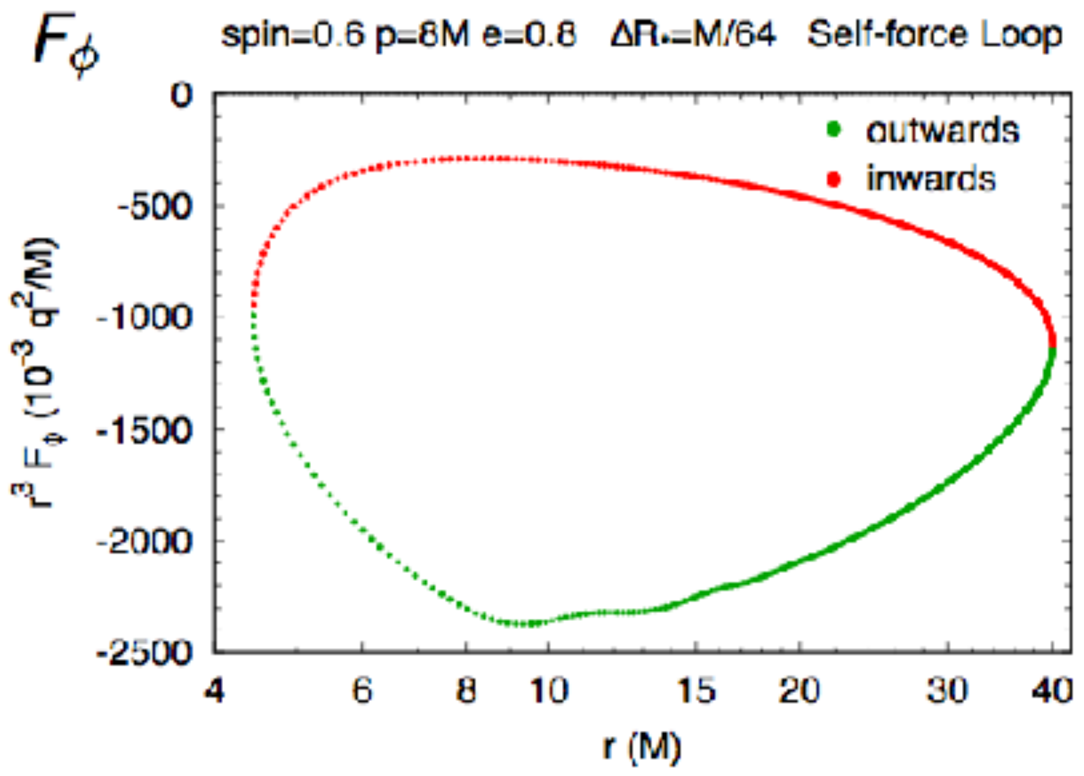
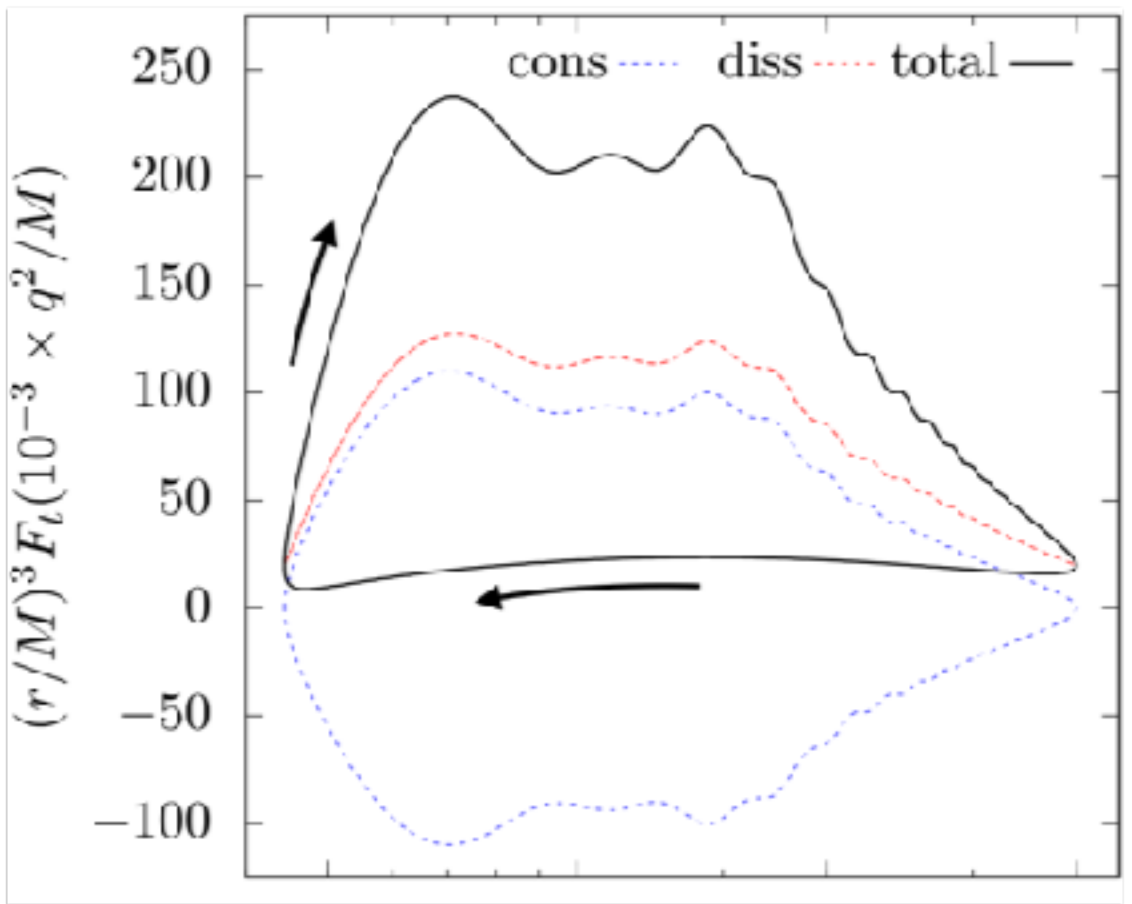


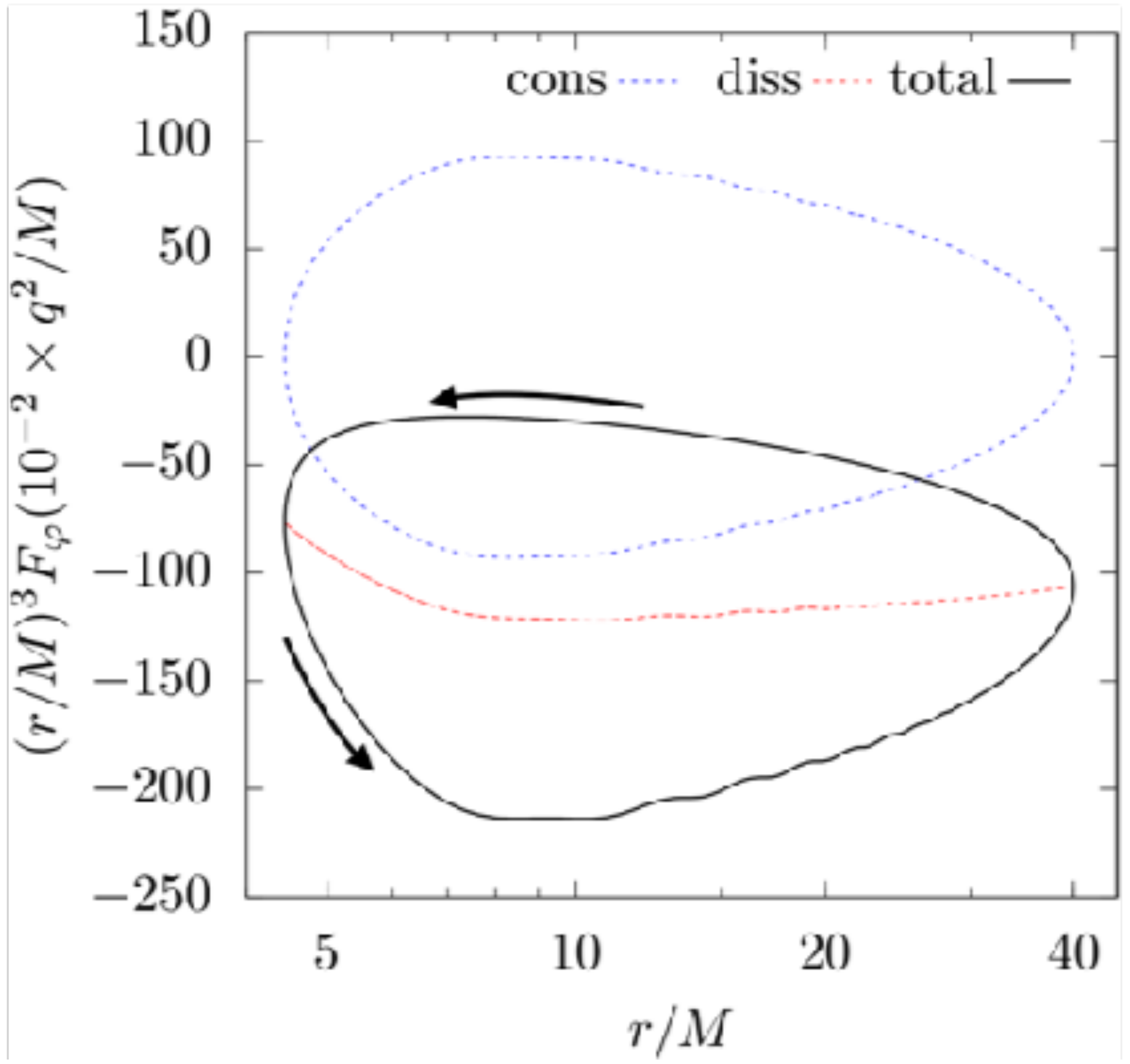
Image Credit: Thornburg 20th Capra

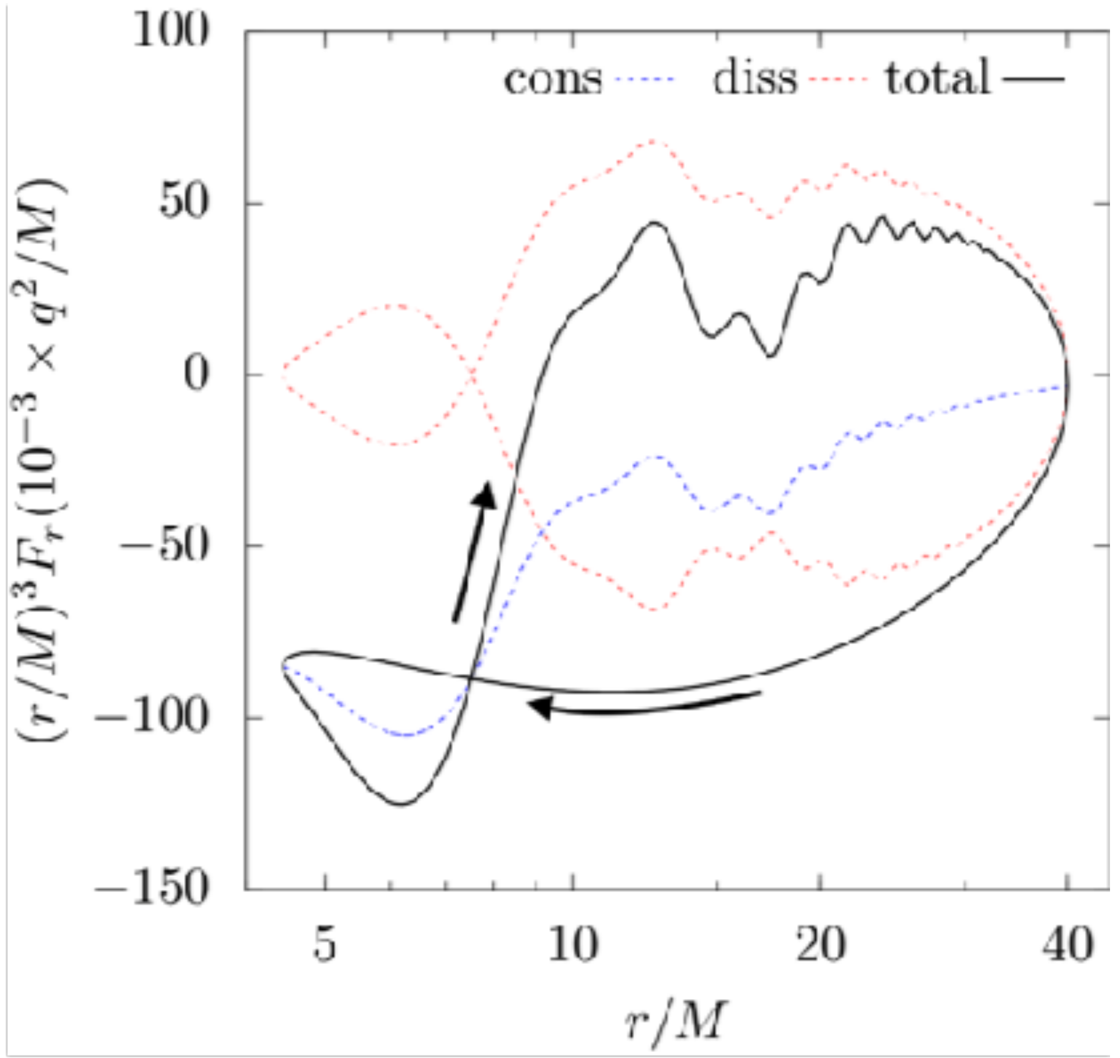
Highly eccentric orbit

$p = 8, e = 0.8, \iota = 0, a = 0.99M$

[1610.09319]



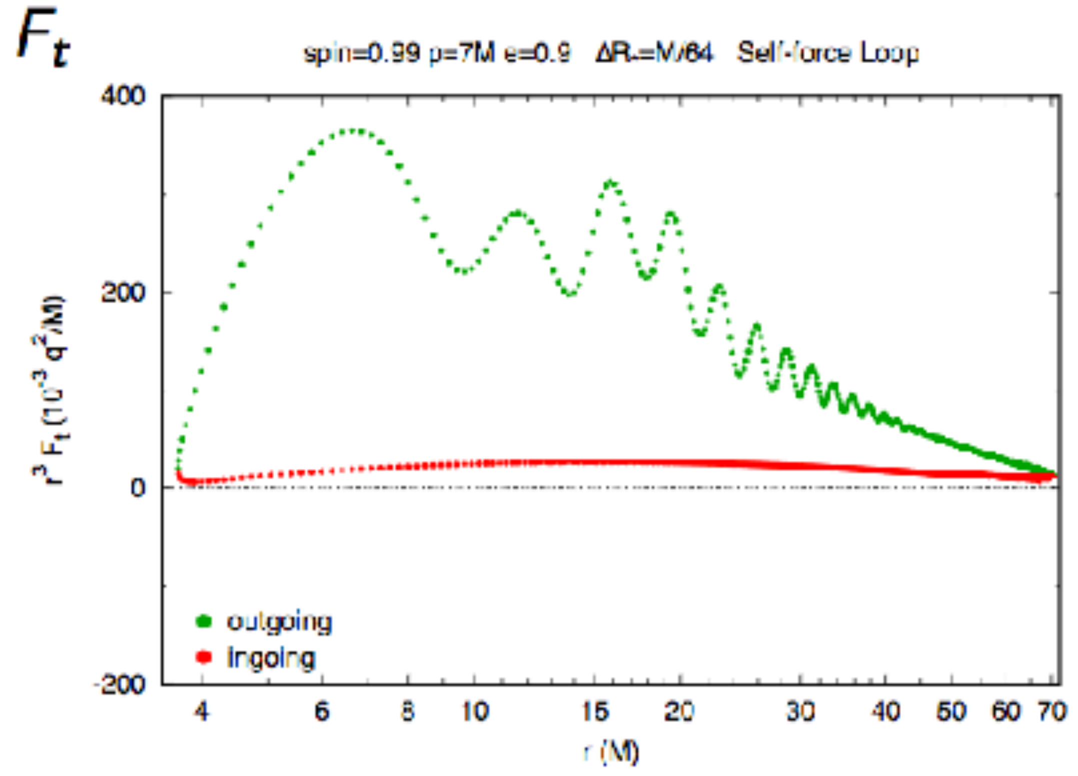


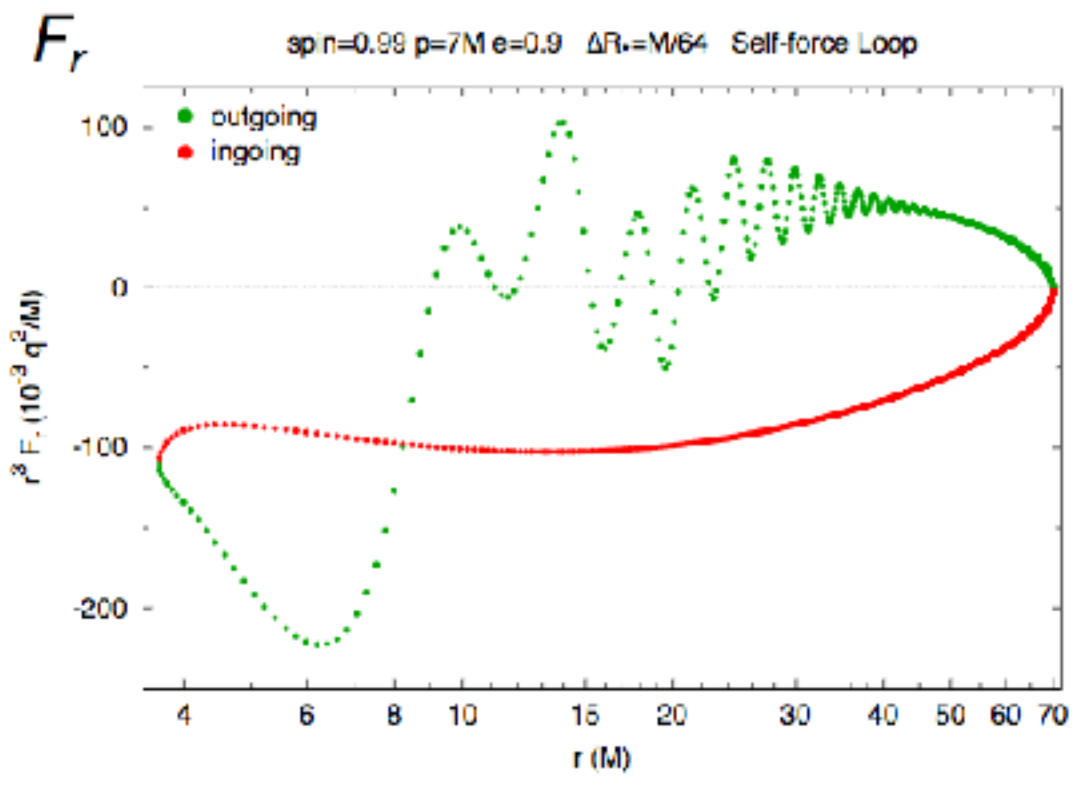


"Wiggles": Excited quasi-normal modes

- Inspired by Thornburg & Wardell (2017): Time-domain, equatorial Kerr SSF code

$p = 7, e = 0.9, \iota = 0, a = 0.99M$





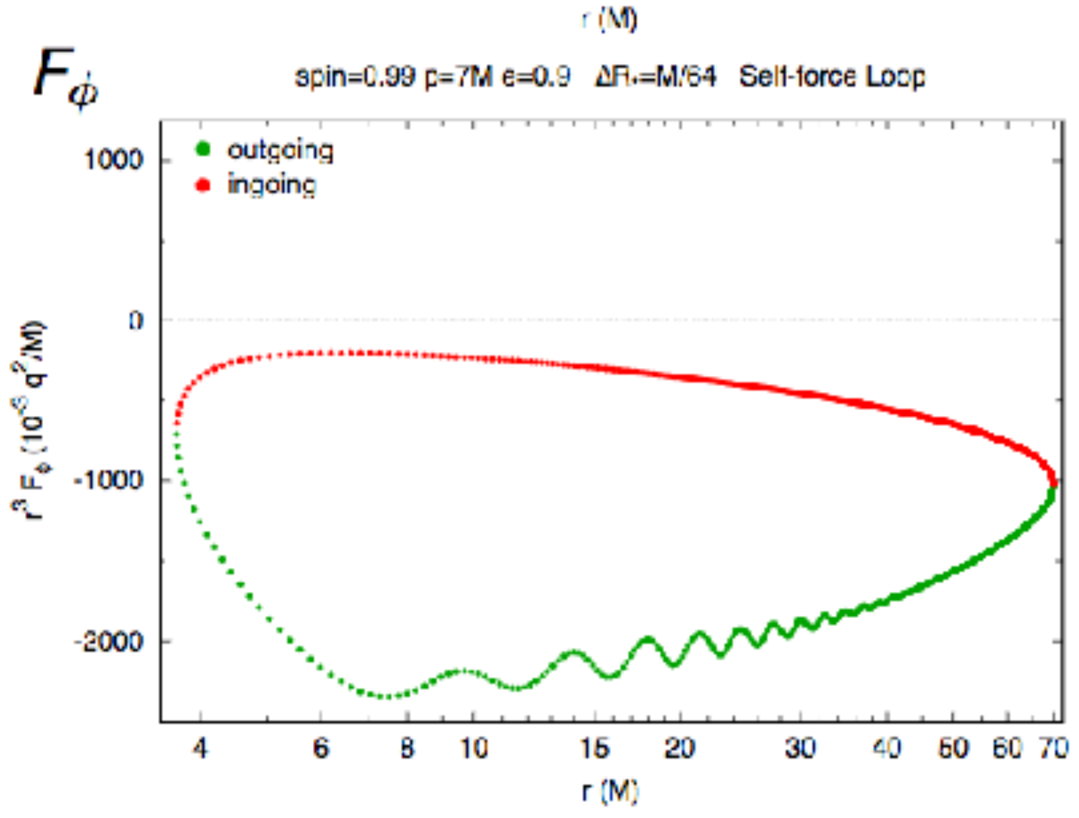
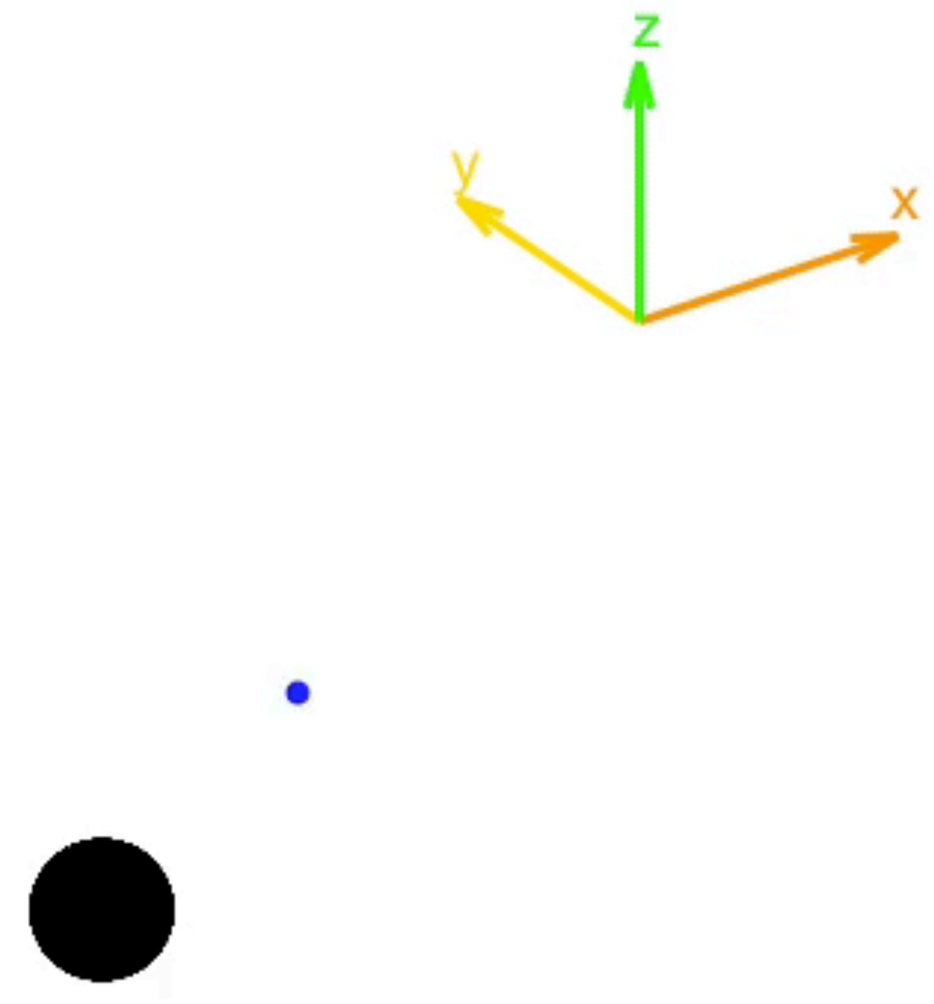


Image Credit: Thornburg 20th Capra

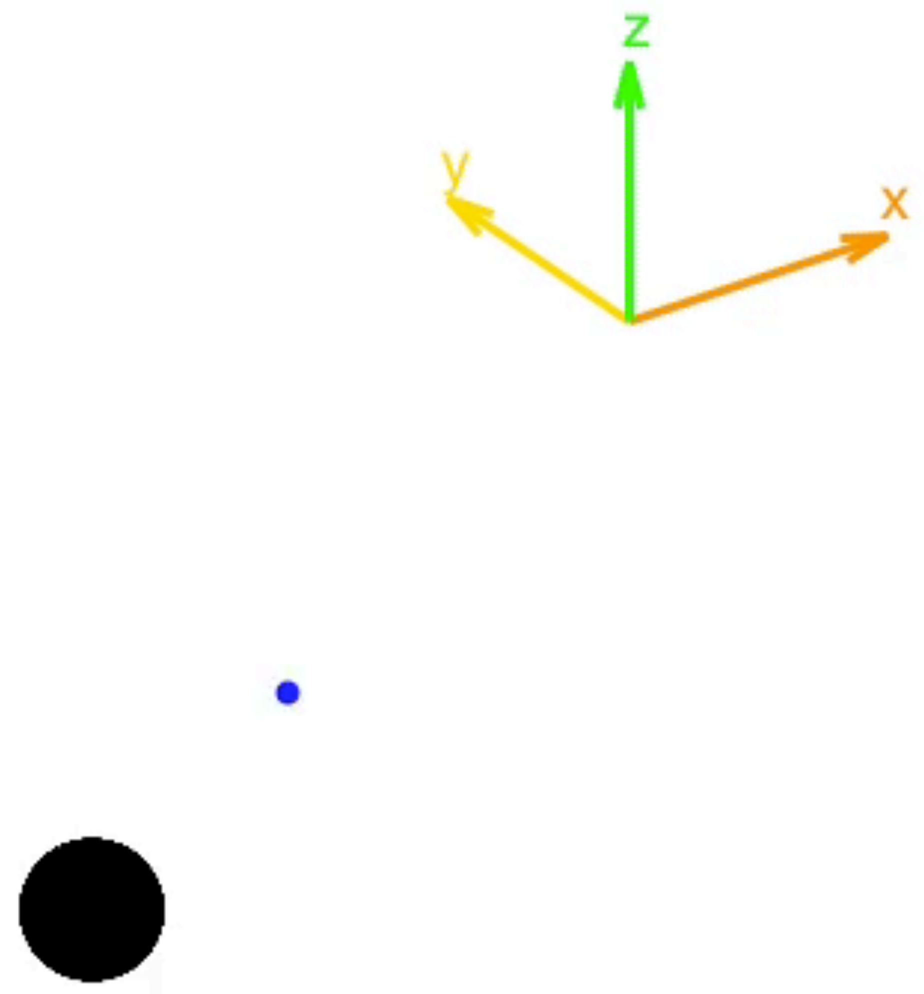
Generic orbit

$p = 10, e = 0.1, \iota = \pi/5, a = 0.5M$



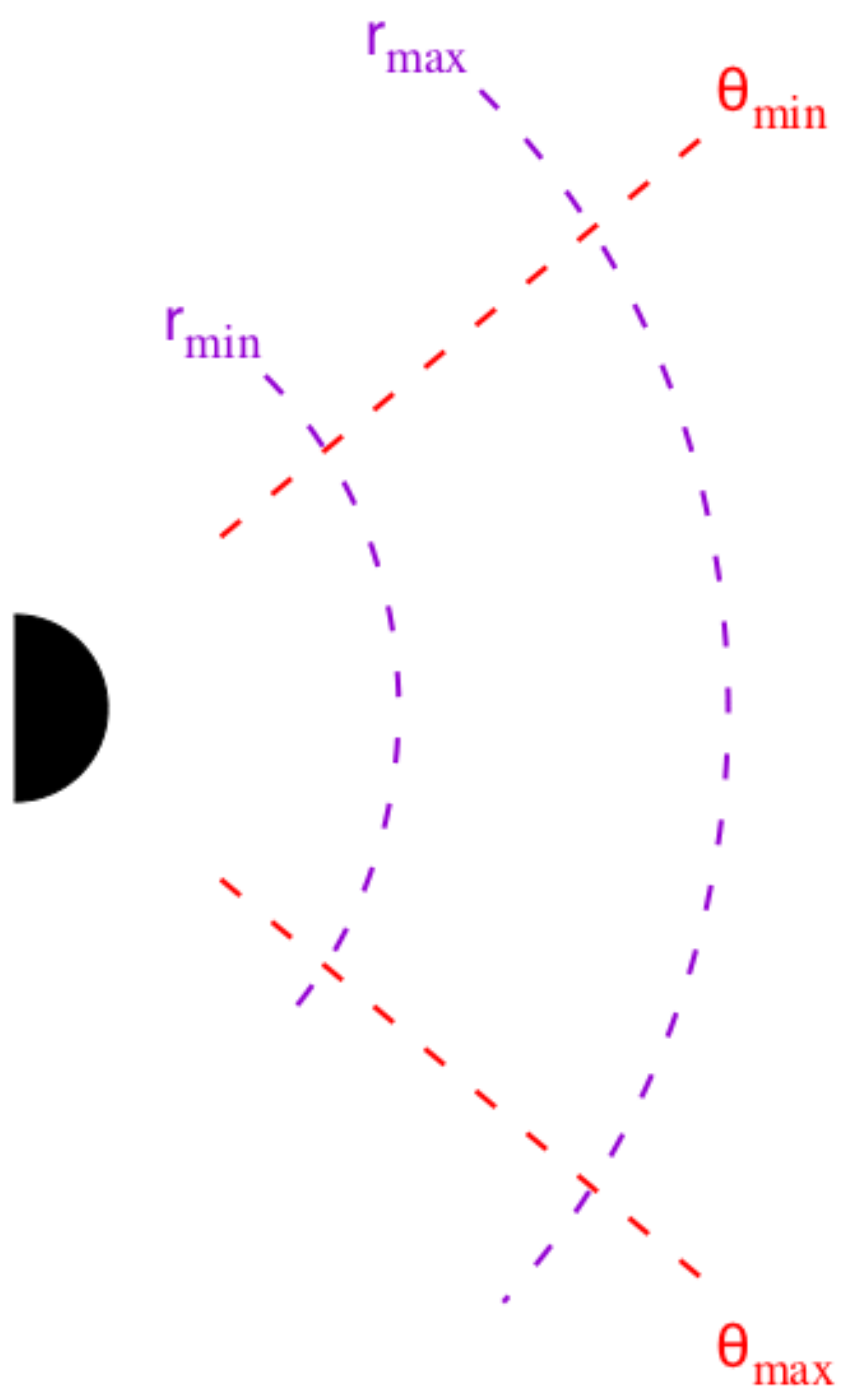
Generic orbit

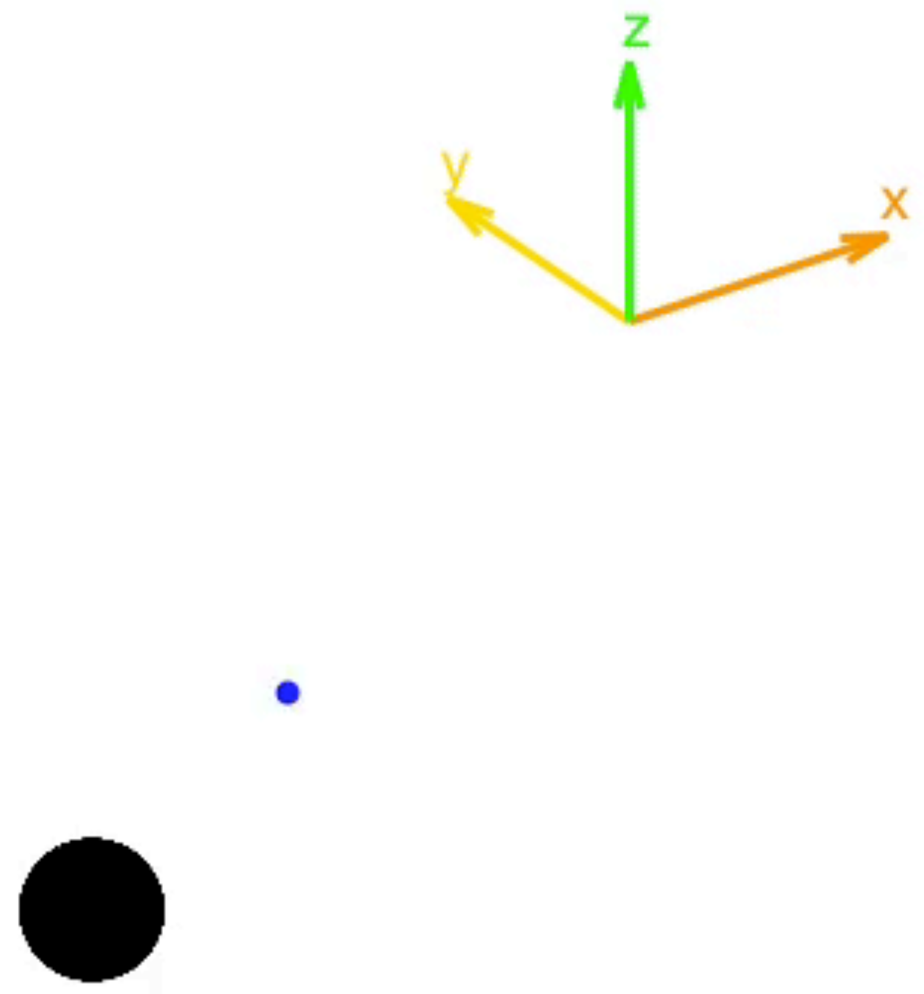
$p = 10, e = 0.1, \iota = \pi/5, a = 0.5M$



Generic orbit

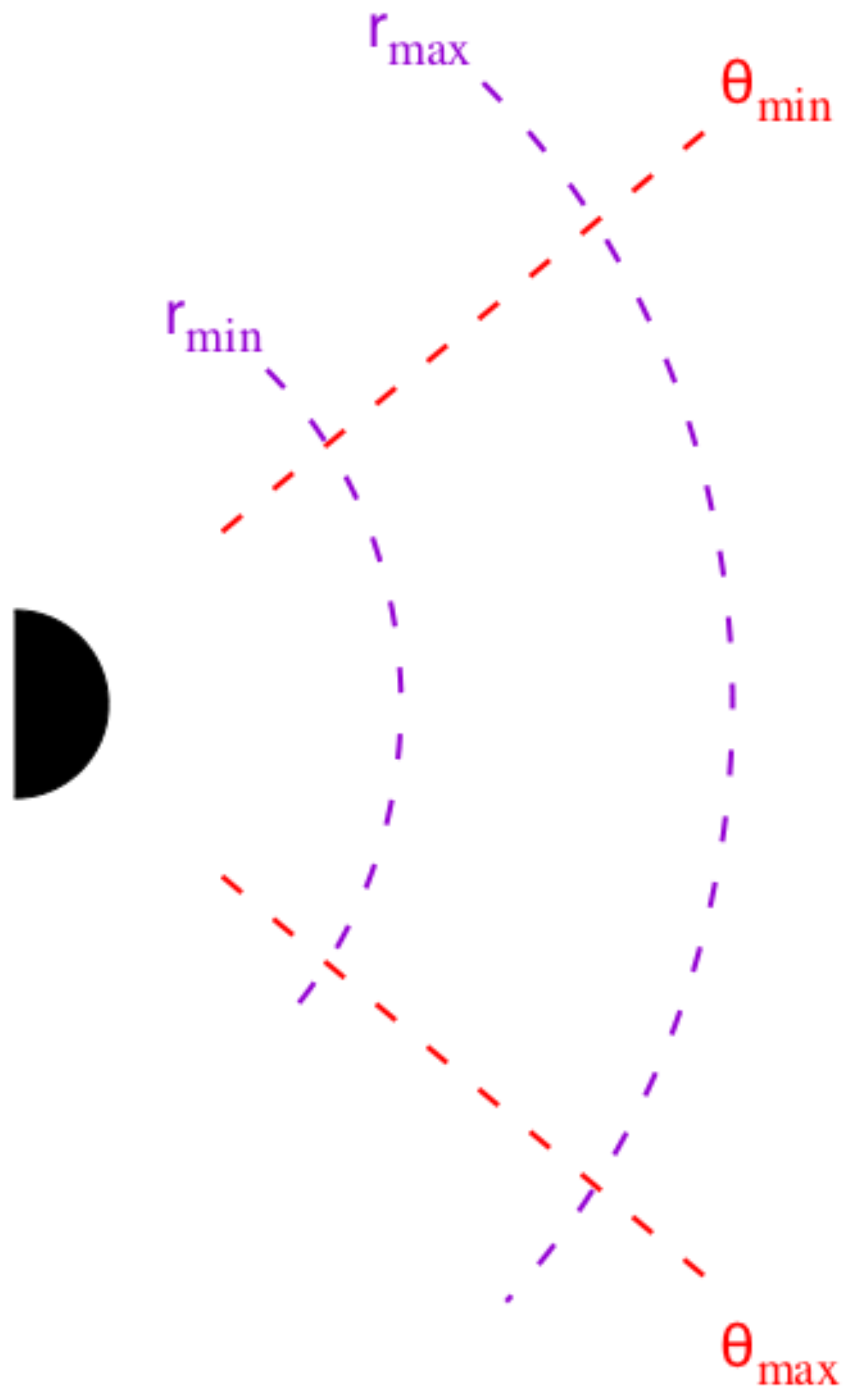
$p = 10, e = 0.1, \iota = \pi/5, a = 0.5M$

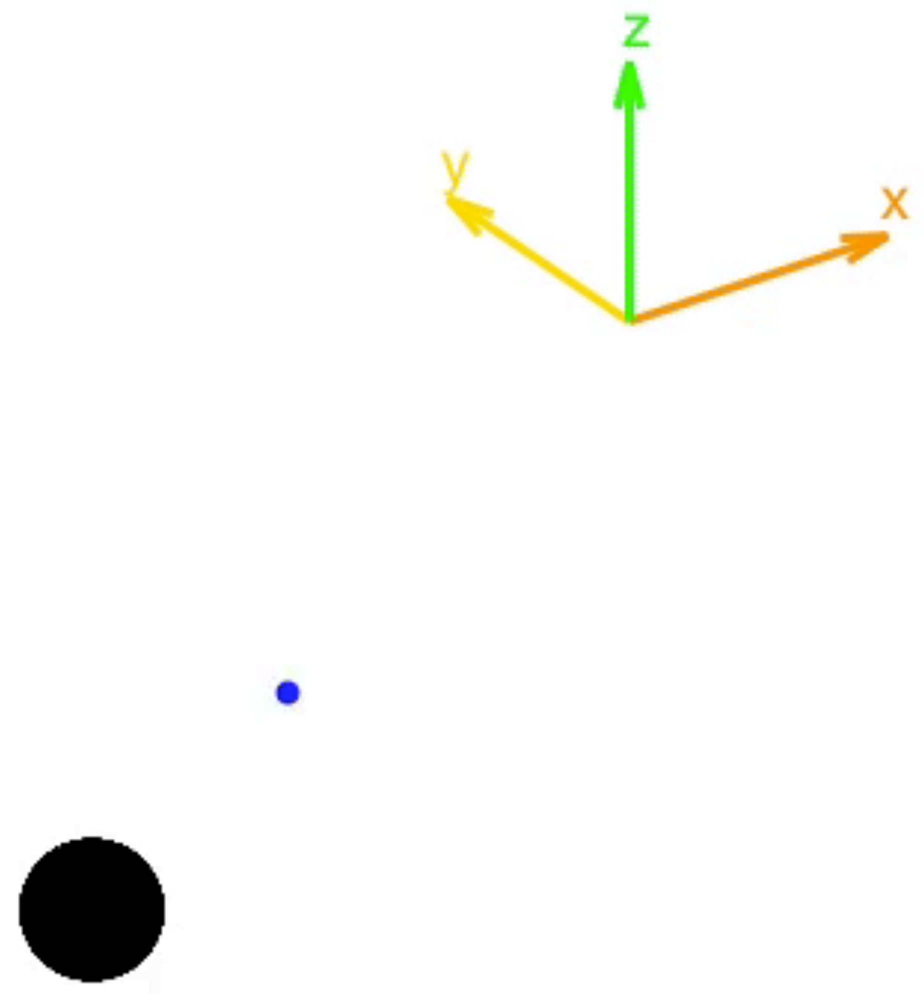




Generic orbit

$p = 10, e = 0.1, \iota = \pi/5, a = 0.5M$

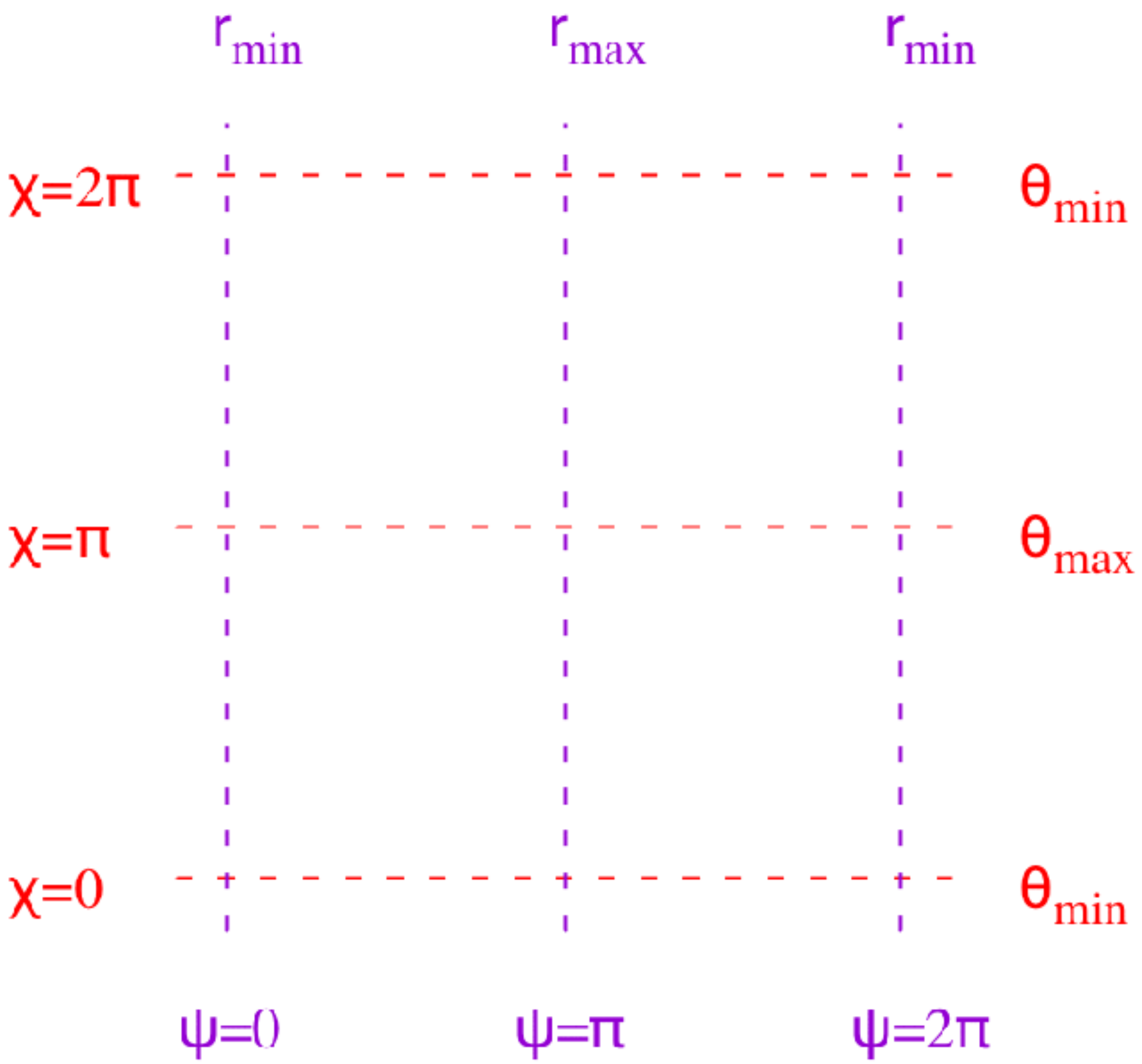


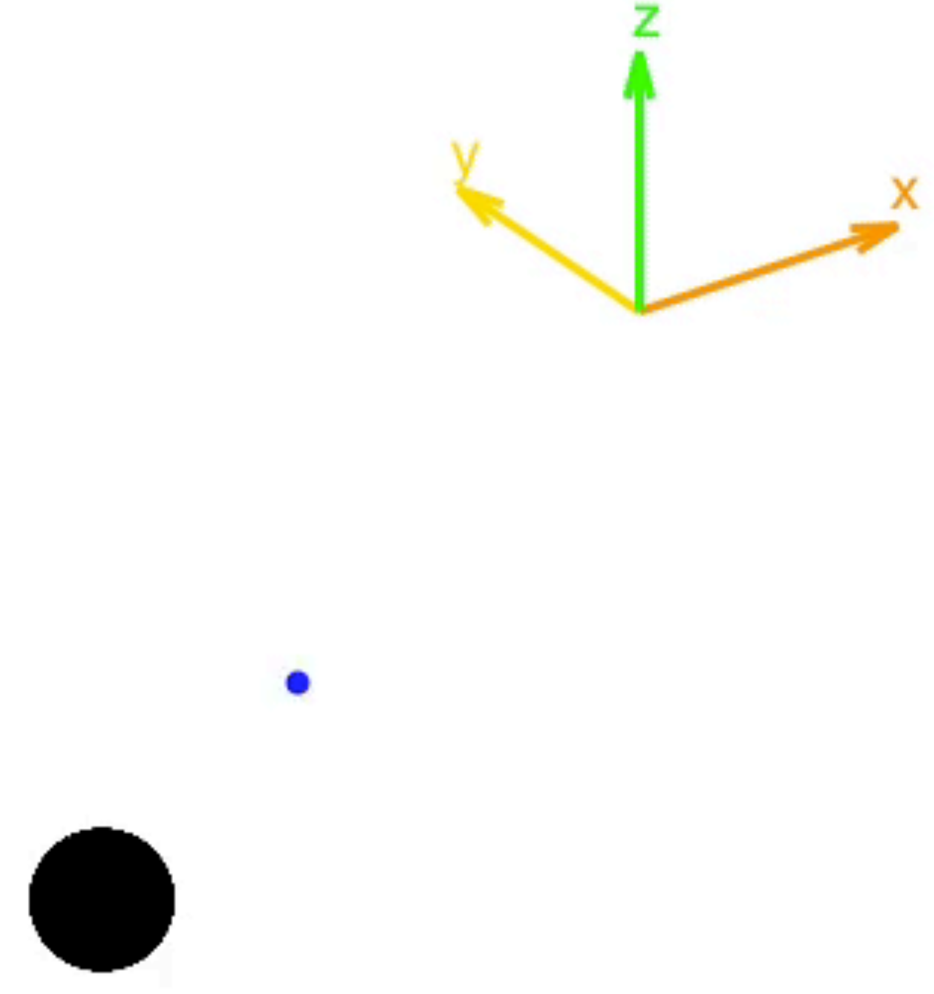


$$r_p(\psi) = \frac{pM}{1 + e\cos\psi} \qquad \cos\theta_p(\chi) = \cos\theta_{\min}\cos\chi$$

Generic orbit

$p = 10, e = 0.1, \iota = \pi/5, a = 0.5M$

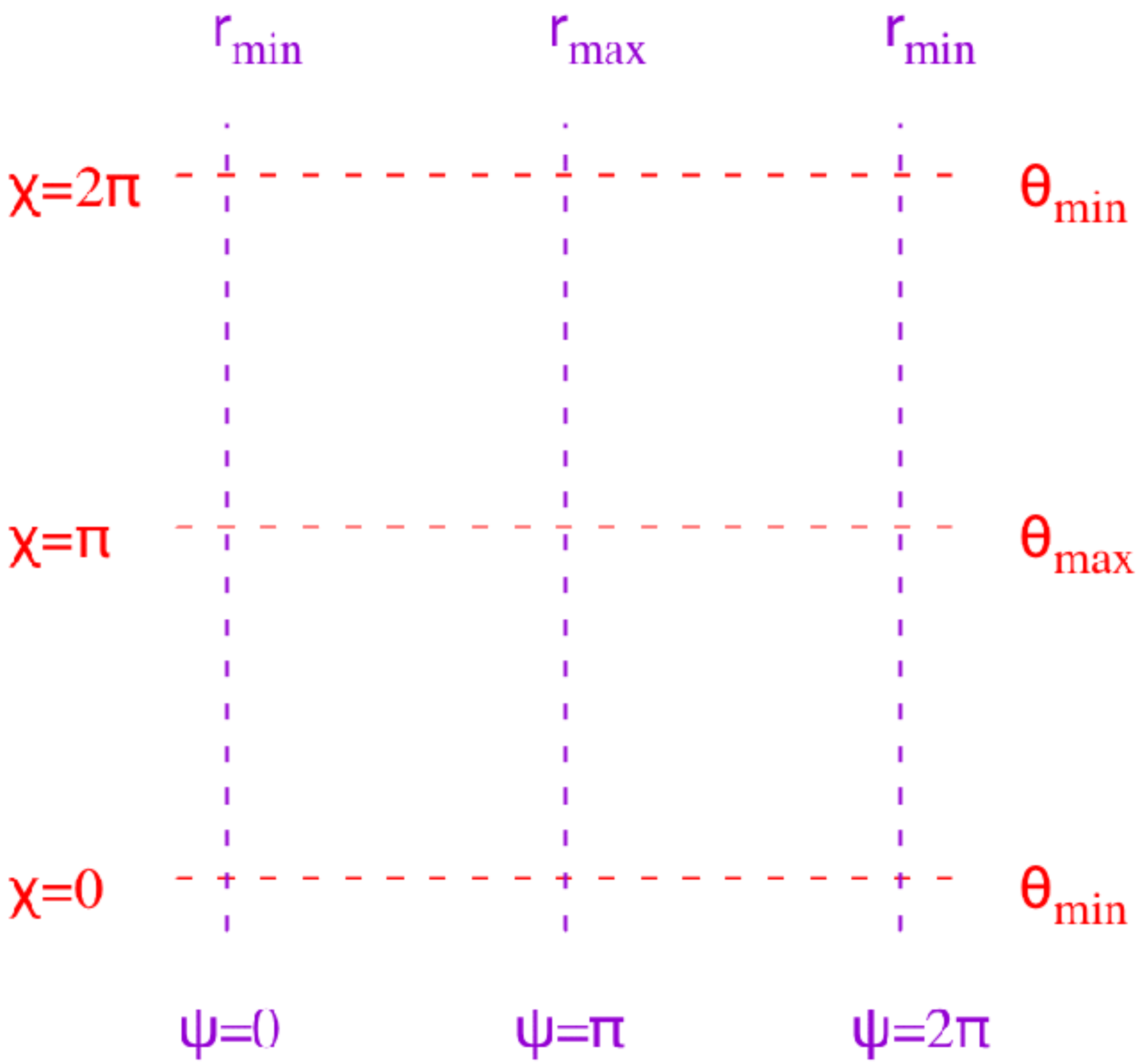




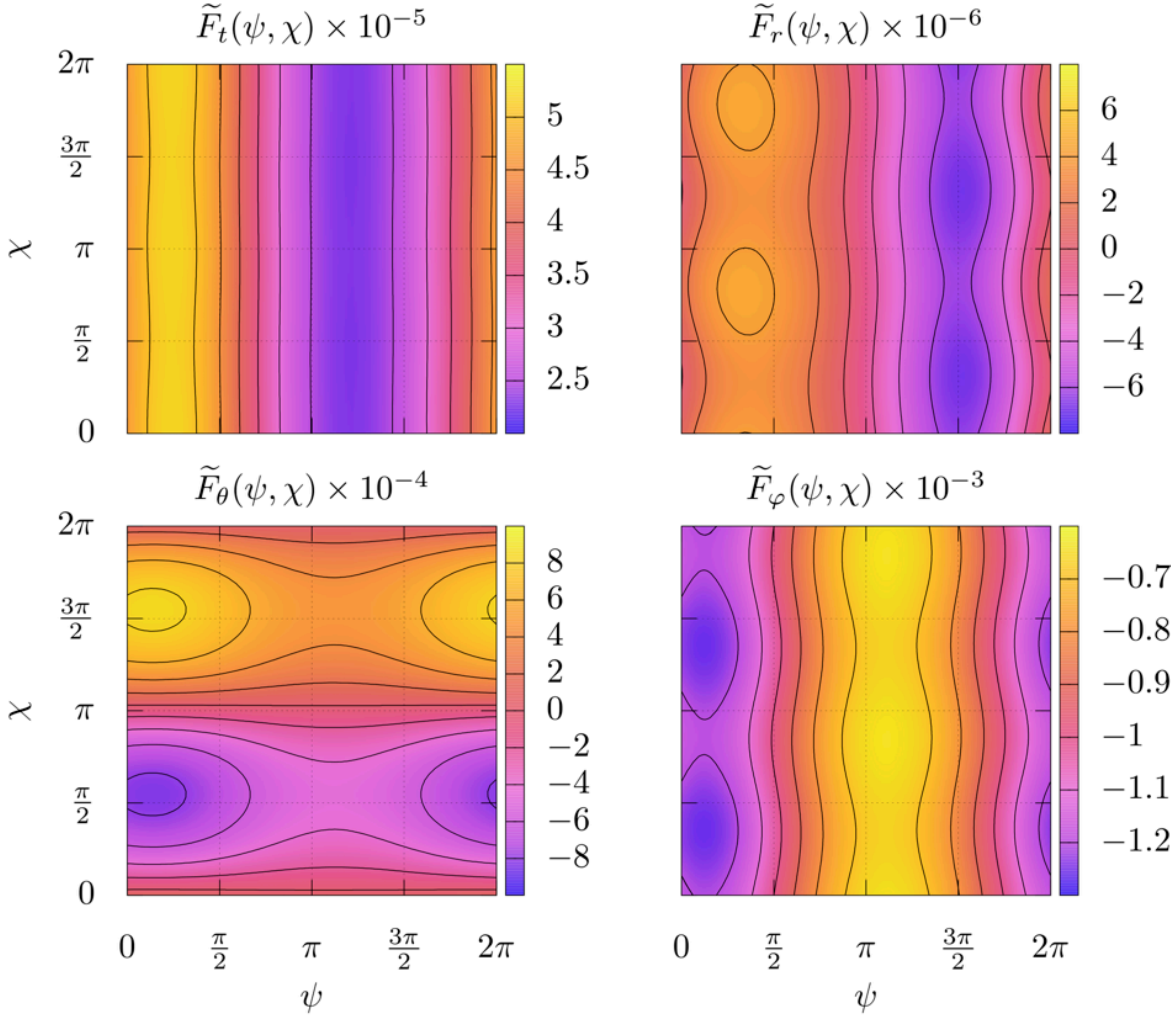
$$r_p(\psi) = \frac{pM}{1 + e\cos\psi} \qquad \cos\theta_p(\chi) = \cos\theta_{\min}\cos\chi$$

Generic orbit

$p = 10, e = 0.1, \iota = \pi/5, a = 0.5M$



$$r_p(\psi) = \frac{pM}{1 + e\cos\psi} \qquad \cos\theta_p(\chi) = \cos\theta_{\min}\cos\chi$$



Generic orbit comparison

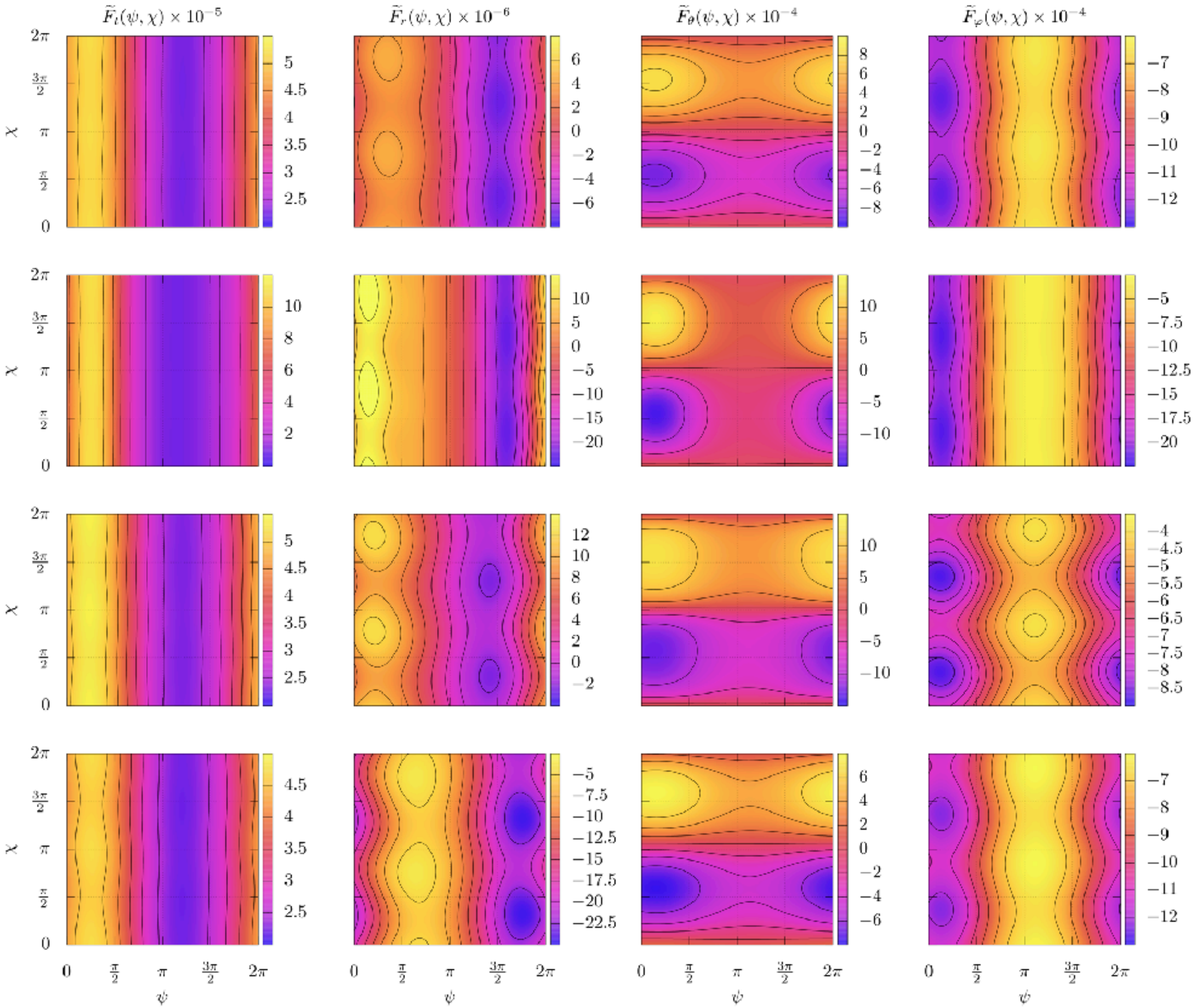
F_t | F_r | F_θ | F_φ

$p = 10, e = 0.1, \iota = \pi/5, a = 0.5M$

$p = 10, e = 0.3, \iota = \pi/5, a = 0.5M$

$p = 10, e = 0.1, \iota = \pi/3, a = 0.5M$

$p = 10, e = 0.1, \iota = \pi/5, a = 0.9M$



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

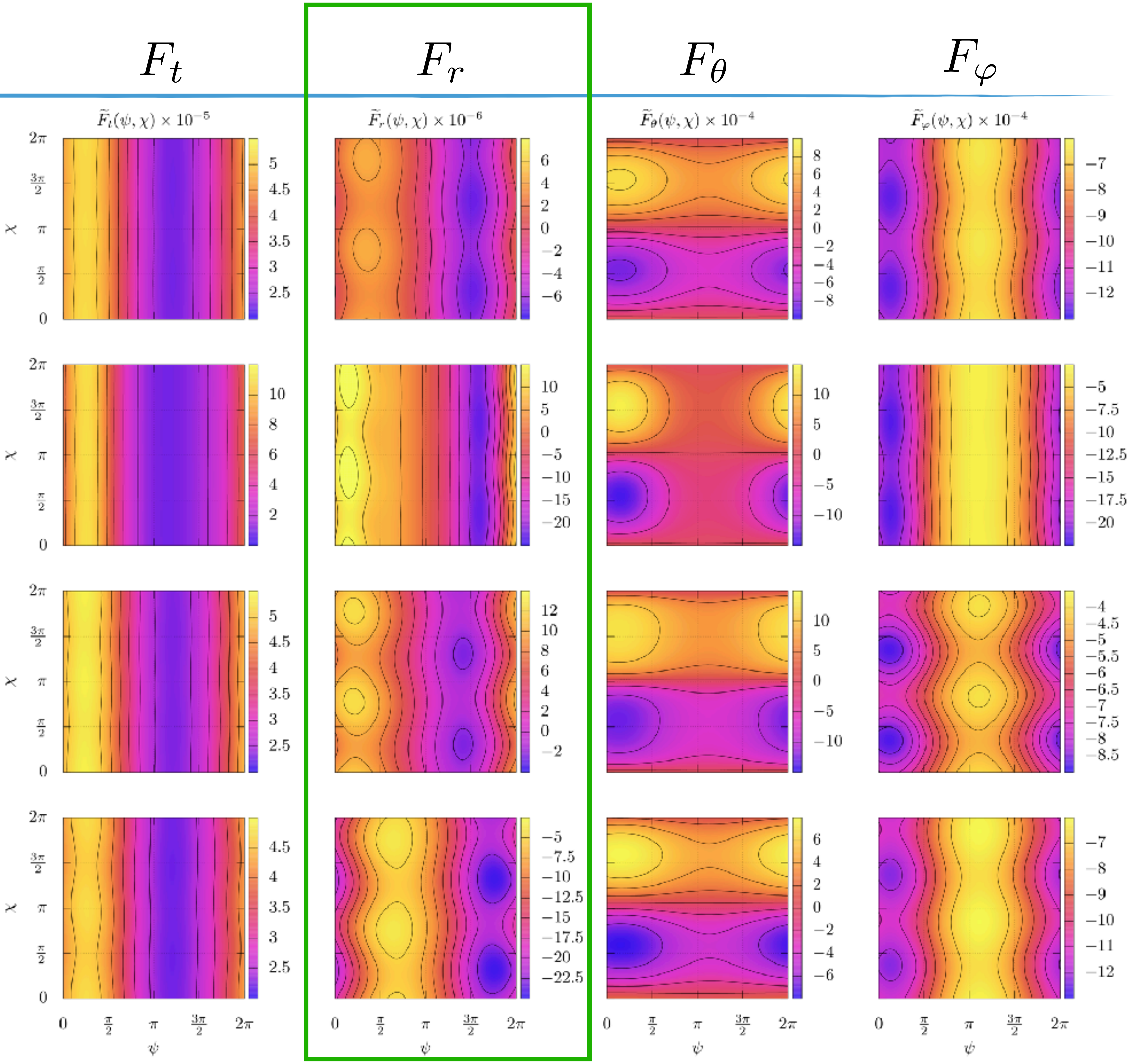
Generic orbit comparison

$p = 10, e = 0.1, \iota = \pi/5, a = 0.5M$

$p = 10, e = 0.3, \iota = \pi/5, a = 0.5M$

$p = 10, e = 0.1, \iota = \pi/3, a = 0.5M$

$p = 10, e = 0.1, \iota = \pi/5, a = 0.9M$



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Generic orbit comparison

- **Radial-component F_r**

$p = 10, e = 0.1, \iota = \pi/5, a = 0.5M$

Baseline $\qquad e = 0.3 \qquad \iota = \pi/3 \qquad a = 0.9M$

Validation tests - fluxes

- **Flux balance**

$$\left\langle \dot{E}^{\mathcal{H}} \right\rangle + \left\langle \dot{E}^{\infty} \right\rangle = \frac{1}{T}\int_0^T \frac{F_t}{u^t}\, dt$$

Radiated energy = Local work

Validation tests - fluxes

- **Flux balance**

$$\left\langle \dot{E}^{\mathcal{H}} \right\rangle + \left\langle \dot{E}^{\infty} \right\rangle = \frac{1}{T}\int_0^T \frac{F_t}{u^t}\, dt$$

Radiated energy = Local work

Radiated energy	Fractional Error
2.917529922×10^{-5}	5×10^{-14}
2.9610263×10^{-5}	9×10^{-14}
2.994475370×10^{-5}	0×10^{-11}
2.745901231×10^{-5}	7×10^{-12}

Validation tests - fluxes

- **Flux balance**

$$\left\langle \dot{E}^{\mathcal{H}} \right\rangle + \left\langle \dot{E}^{\infty} \right\rangle = \frac{1}{T}\int_0^T \frac{F_t}{u^t}\, dt$$

Radiated energy = Local work

$$\left\langle \dot{L}_z^{\mathcal{H}} \right\rangle + \left\langle \dot{L}_z^{\infty} \right\rangle = -\frac{1}{T}\int_0^T \frac{F_\varphi}{u^t}\, dt$$

Radiated angular momentum = Local torque

Radiated energy	Fractional Error
2.917529922×10^{-5}	5×10^{-14}
2.9610263×10^{-5}	9×10^{-14}
2.994475370×10^{-5}	0×10^{-11}
2.745901231×10^{-5}	7×10^{-12}

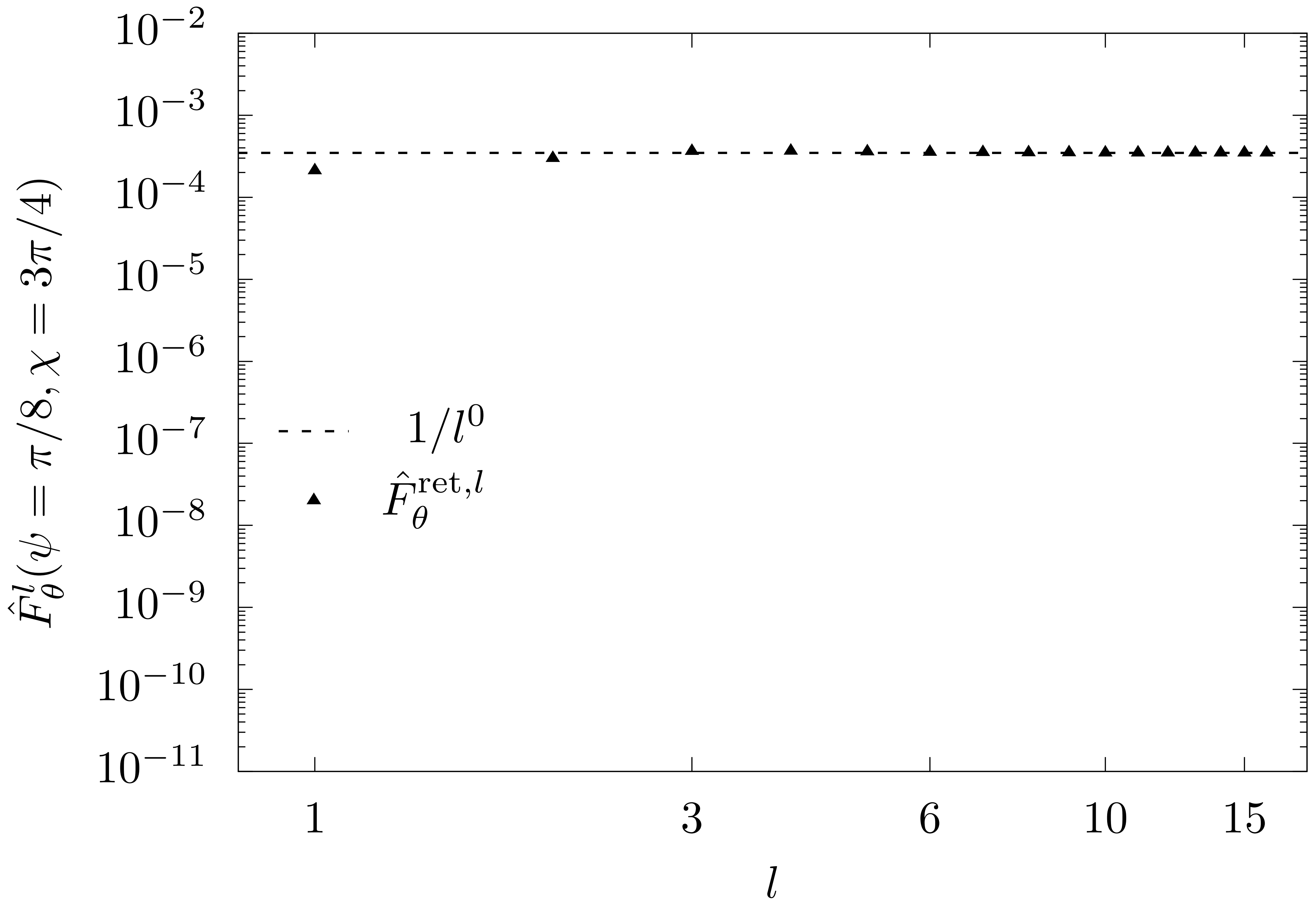
Angular momentum	Fraction Error
7.56756034×10^{-4}	6×10^{-15}
4.93896206×10^{-4}	4×10^{-14}
6.9840212×10^{-4}	0×10^{-12}
7.281232718×10^{-4}	0×10^{-11}

Validation tests

- **Regularization**

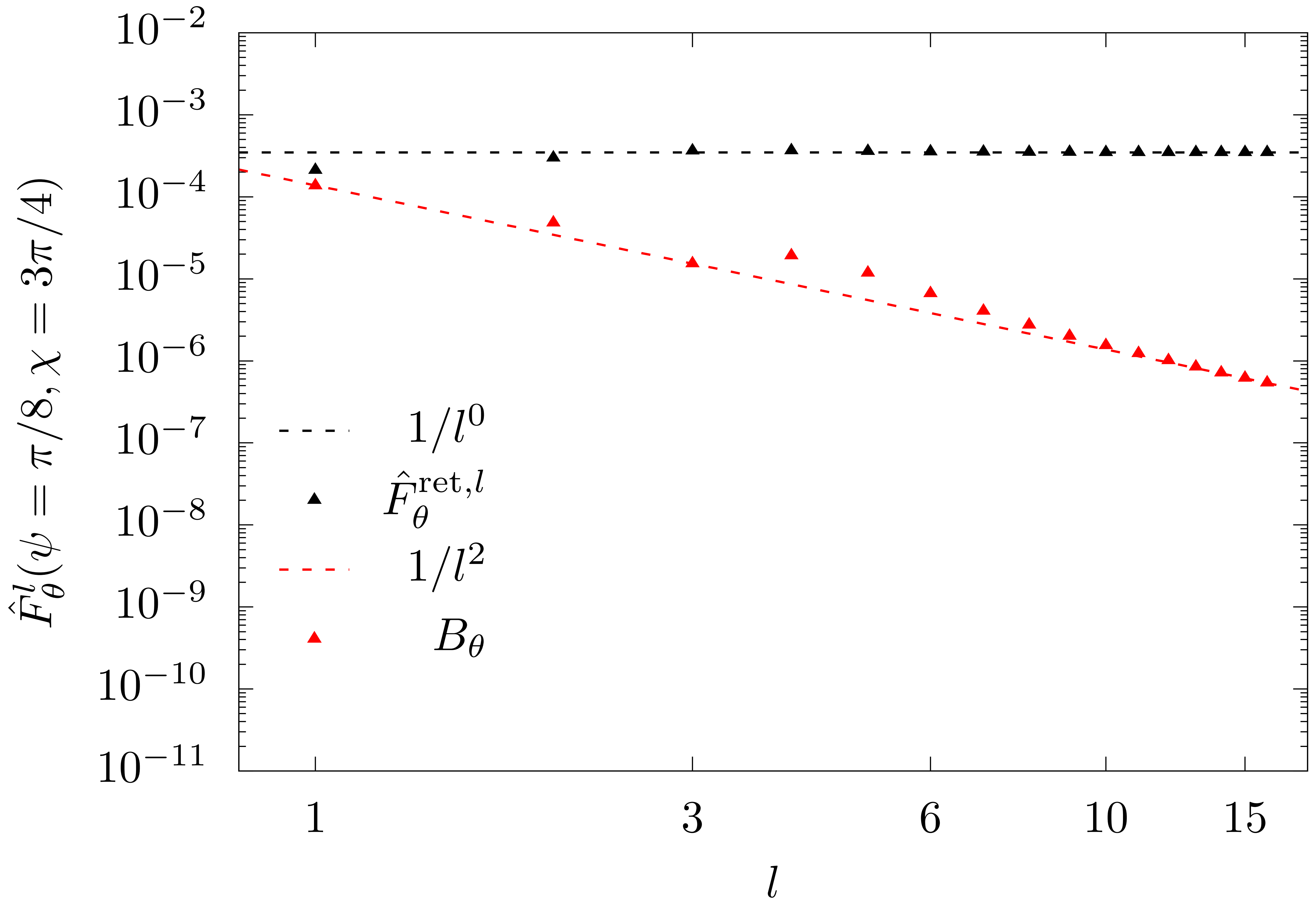
Validation tests

- **Regularization**



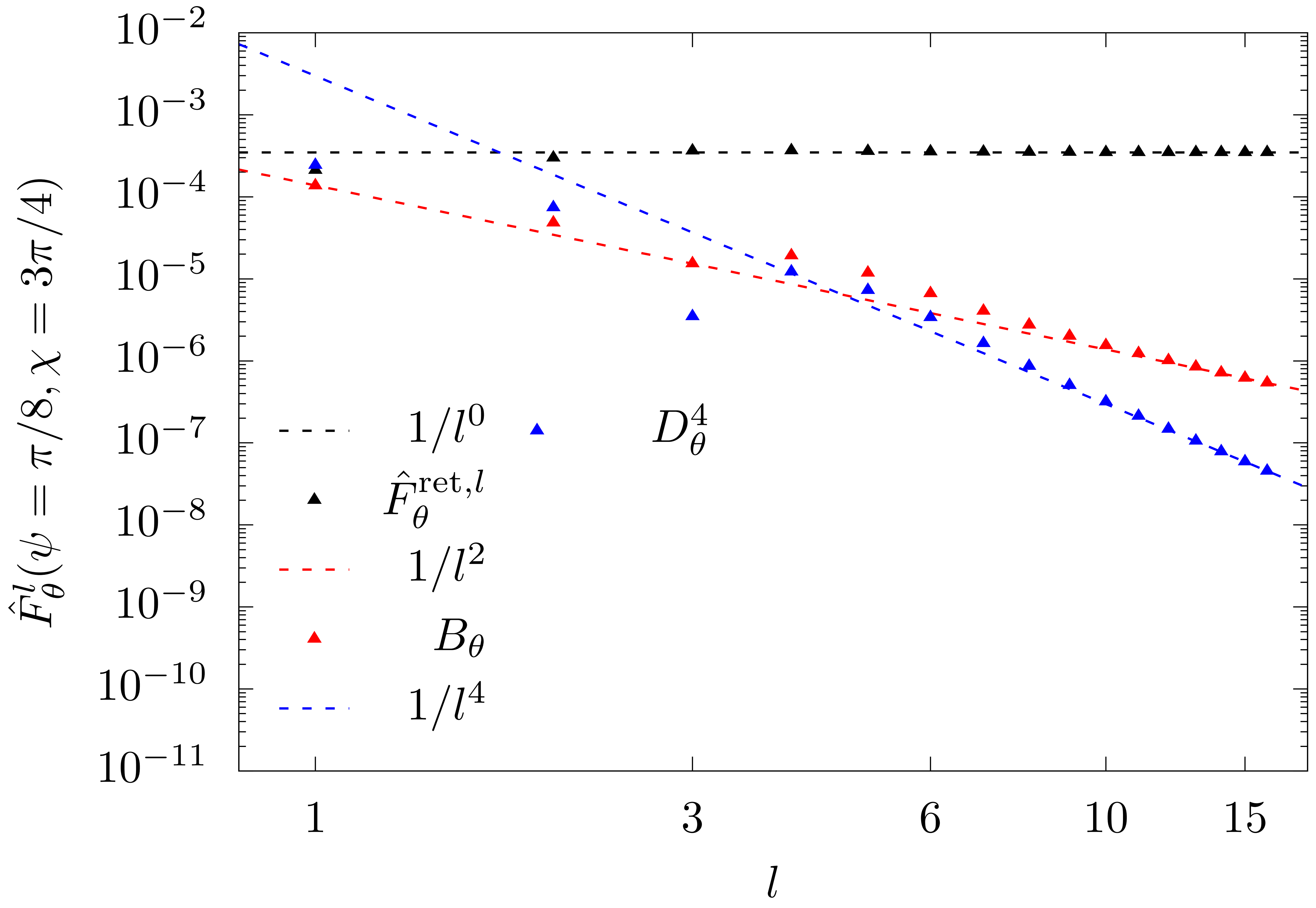
Validation tests

- **Regularization**



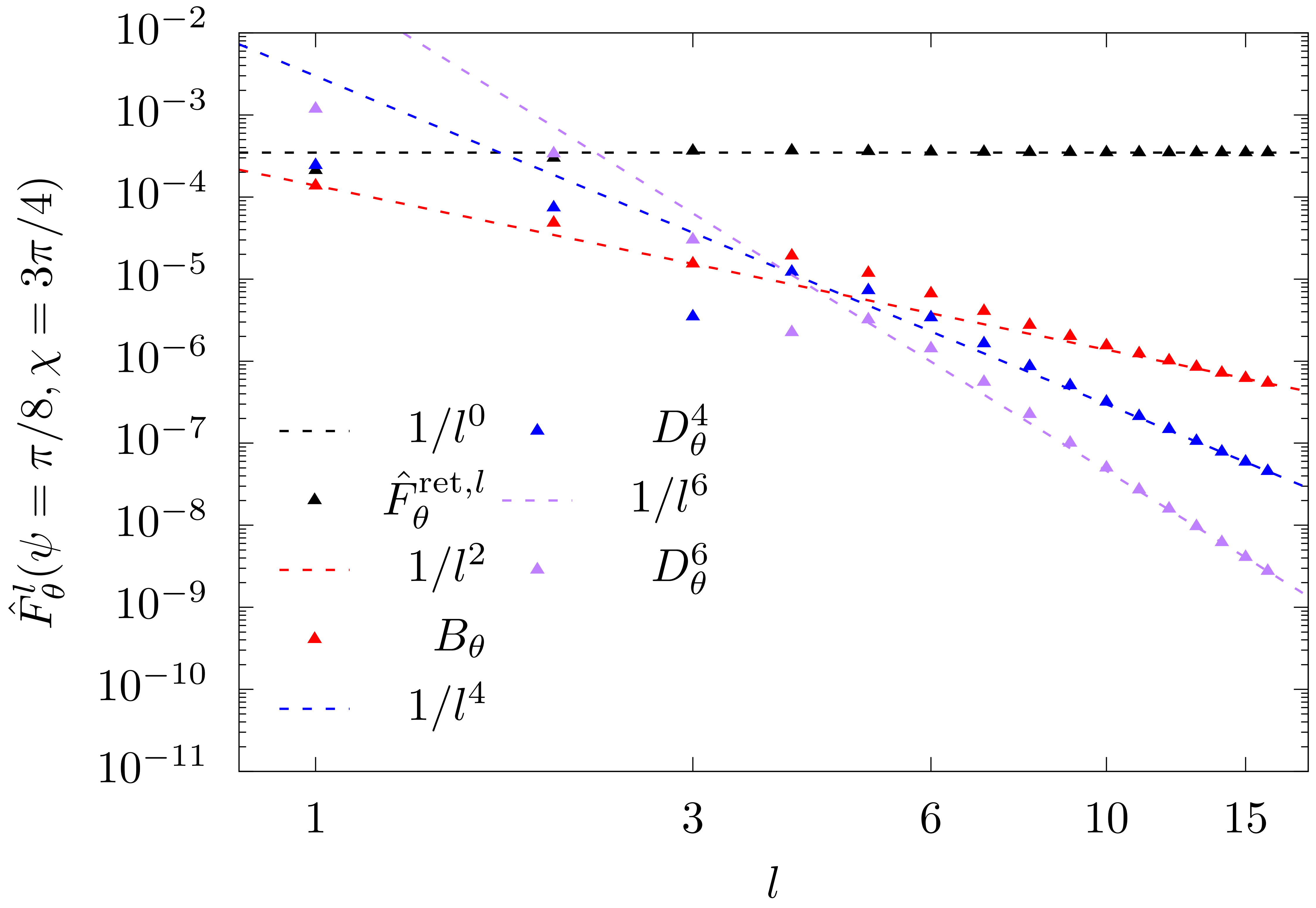
Validation tests

- **Regularization**



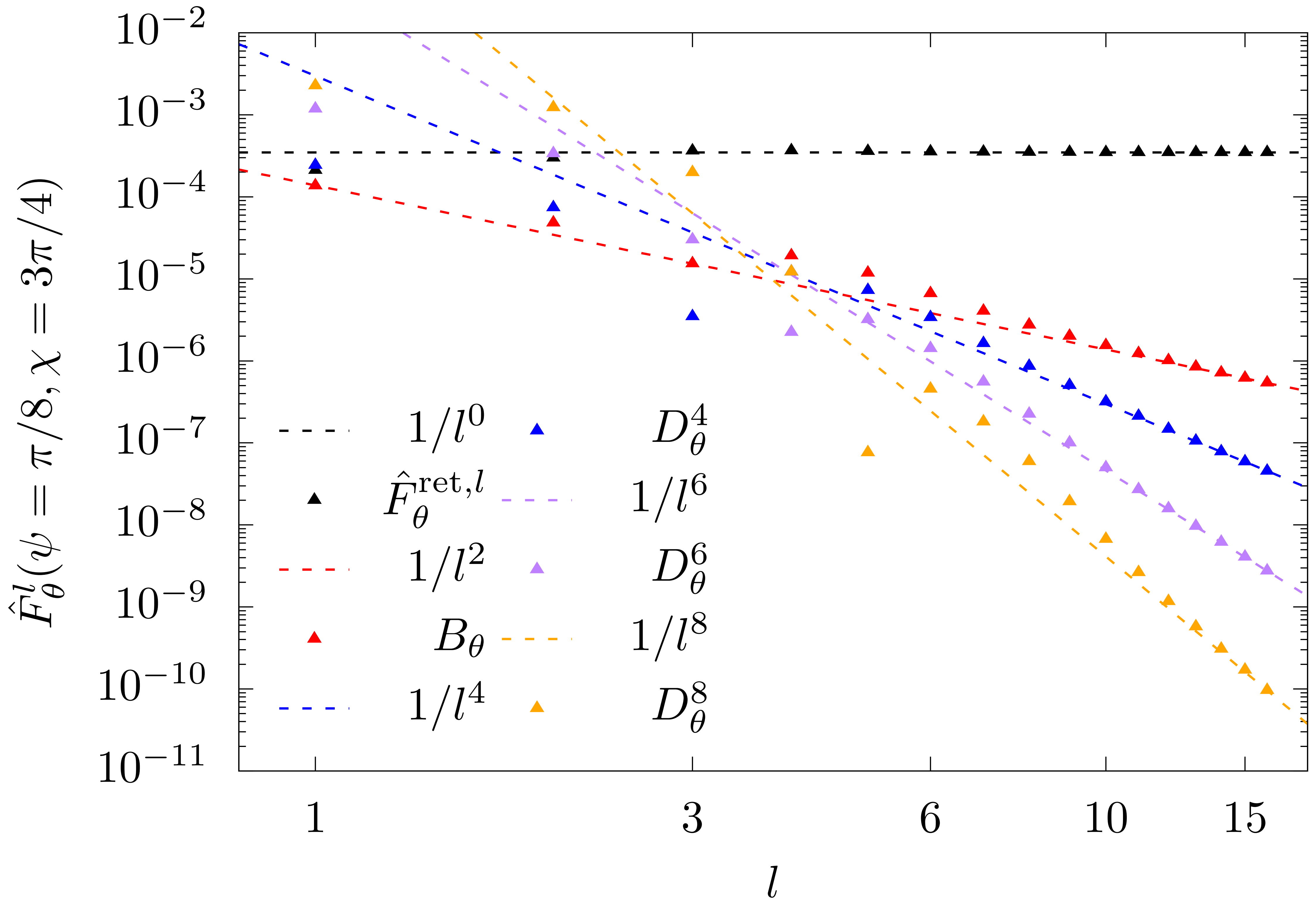
Validation tests

- **Regularization**



Validation tests

- **Regularization**



Conclusions

- Built functioning scalar self-force (SSF) code in MATHEMATICA
- Passed latest round of validation tests
  - Successful comparison w/ other results in literature
- Calculations still computationally expensive, but easier to implement than full GSF calculation
- Possibilities moving forward:
  - Perform calculations of & around resonant orbits
  - Explore alternative analytical & numerical approaches to SSF calculations

Conclusions

- Built functioning scalar self-force (SSF) code in Mathematica
- Passed latest round of validation tests
  - Successful comparison w/ other results in literature
- Calculations still computationally expensive, but easier to implement than full GSF calculation
- Possibilities moving forward:
  - Perform calculations of & around resonant orbits
  - Explore alternative analytical & numerical approaches to SSF calculations

Conclusions

- Built functioning scalar self-force (SSF) code in MATHEMATICA
- Passed latest round of validation tests
  - Successful comparison w/ other results in literature
- Calculations still computationally expensive, but easier to implement than full GSF calculation
- Possibilities moving forward:
  - Perform calculations of & around resonant orbits
  - Explore alternative analytical & numerical approaches to SSF calculations

Conclusions

- Built functioning scalar self-force (SSF) code in MATHEMATICA
- Passed latest round of validation tests
  - Successful comparison w/ other results in literature
- Calculations still computationally expensive, but easier to implement than full GSF calculation
- Possibilities moving forward:
  - Perform calculations of & around resonant orbits
  - Explore alternative analytical & numerical approaches to SSF calculations

Conclusions

- Built functioning scalar self-force (SSF) code in MATHEMATICA
- Passed latest round of validation tests
  - Successful comparison w/ other results in literature
- Calculations still computationally expensive, but easier to implement than full GSF calculation
- Possibilities moving forward:
  - Perform calculations of & around resonant orbits
  - Explore alternative analytical & numerical approaches to SSF calculations

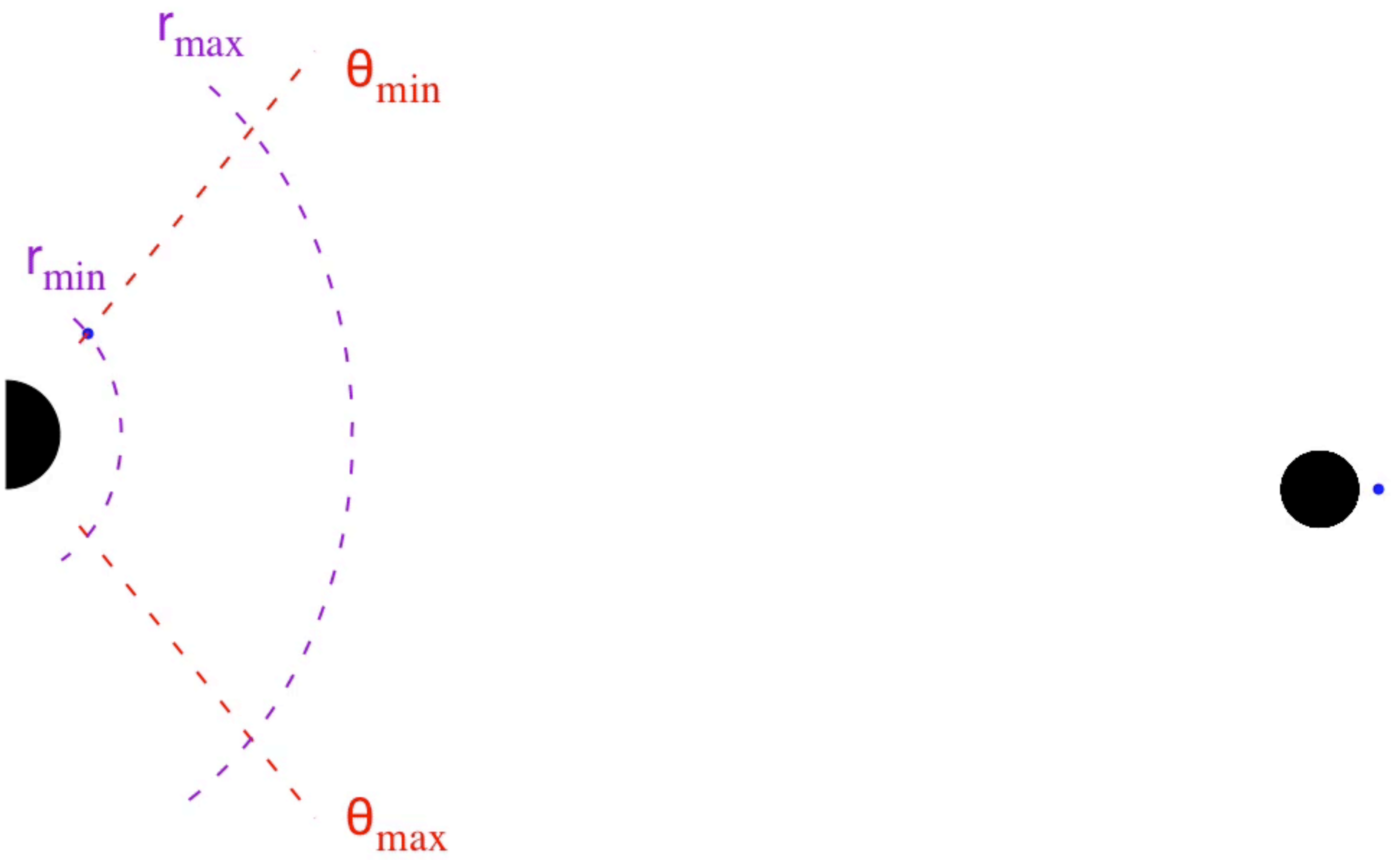
Acknowledgements

- Dr. Charles R. Evans
- Dr. Thomas Osburn
- Black Hole Perturbation Toolkit (bhptoolkit.org)
- National Science Foundation (PHY-1506182)
- North Carolina Space Grant Consortium

Questions?

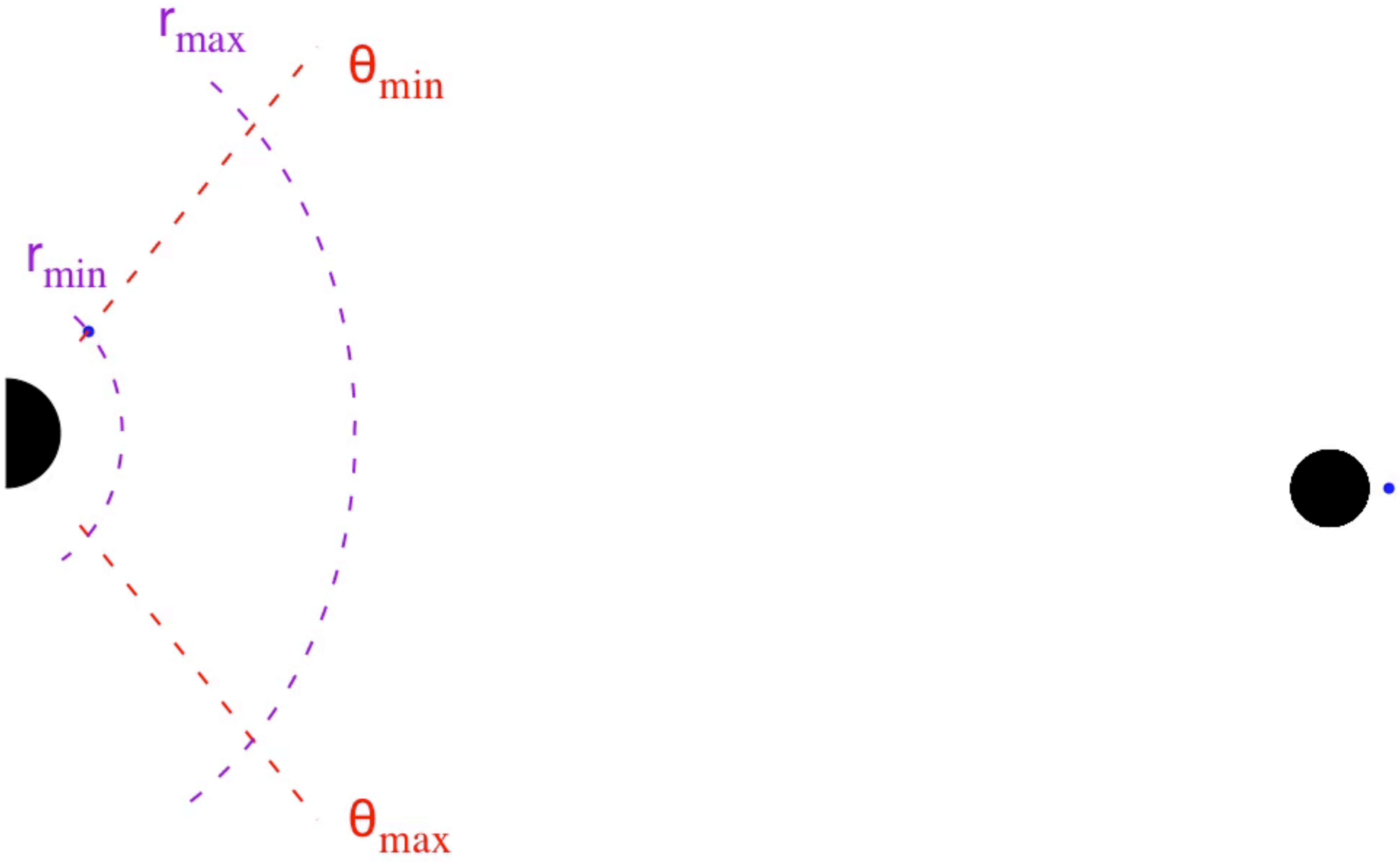
2:1 resonant orbit $(\Omega_\theta/\Omega_r = 2)$

$p \approx 4.6, e = 0.5, \iota \approx 0.79, a = 0.9M$



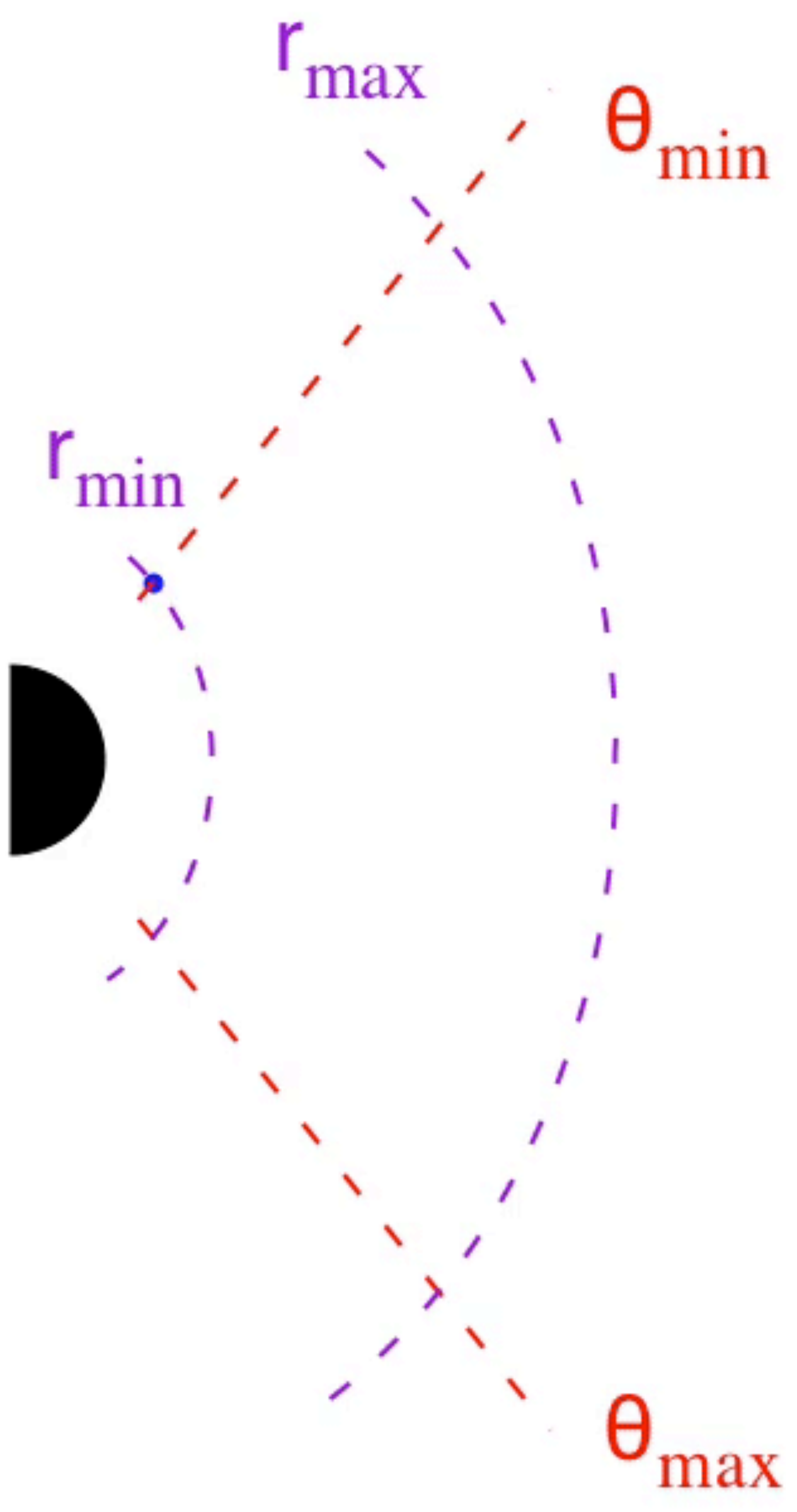
2:1 resonant orbit $(\Omega_\theta/\Omega_r = 2)$

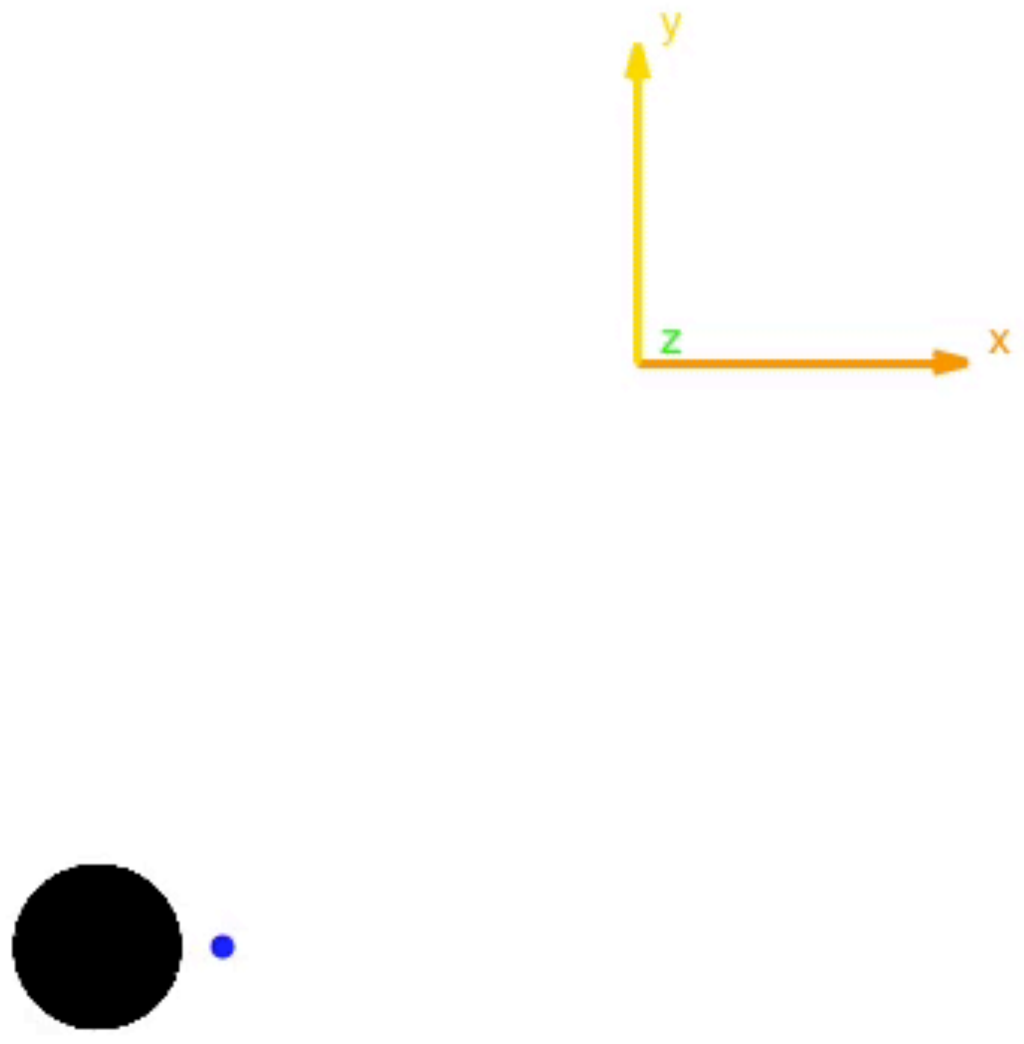
$p \approx 4.6, e = 0.5, \iota \approx 0.79, a = 0.9M$



2:1 resonant orbit $(\Omega_\theta/\Omega_r = 2)$

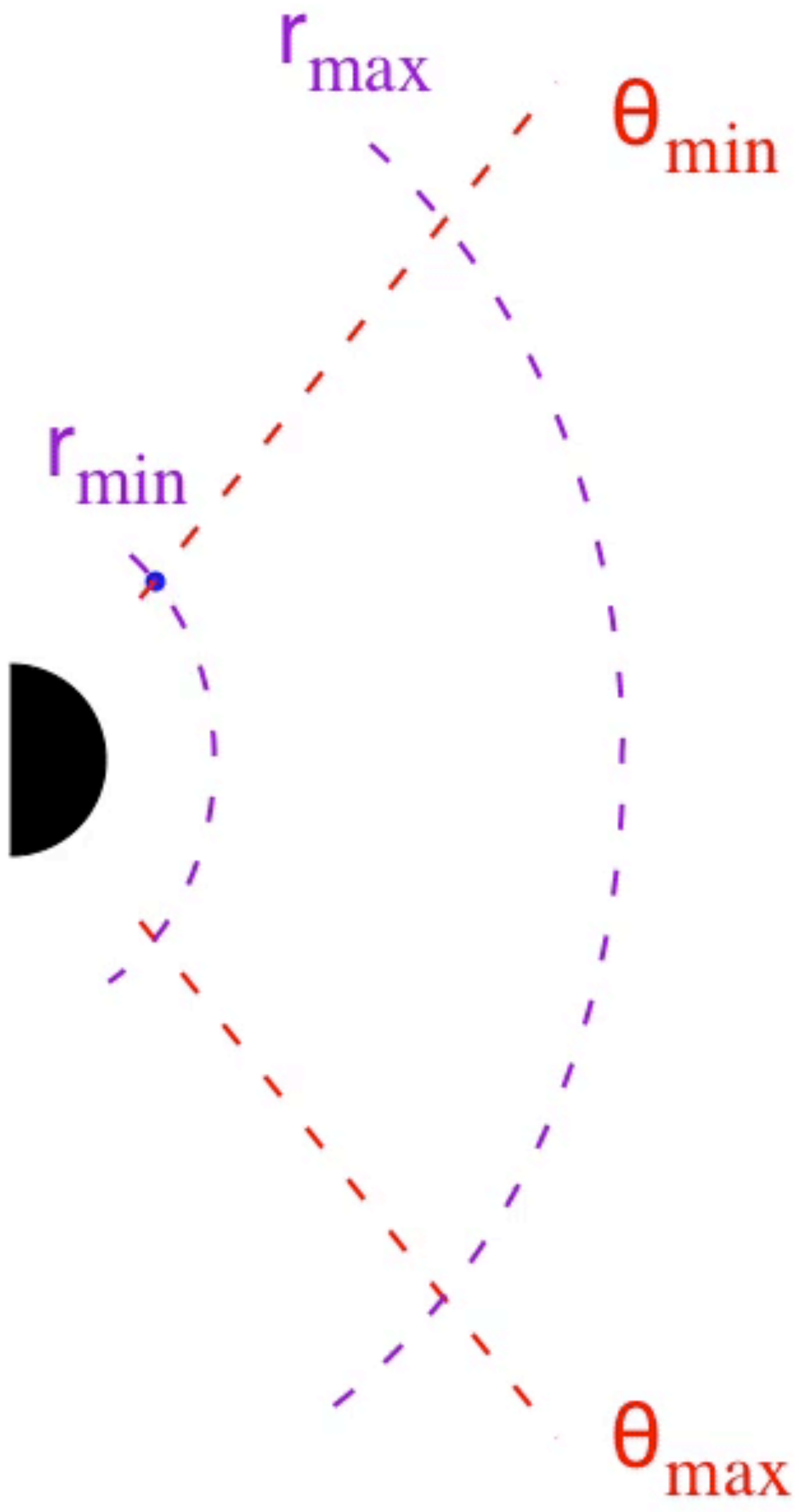
$p \approx 4.6, e = 0.5, \iota \approx 0.79, a = 0.9M$

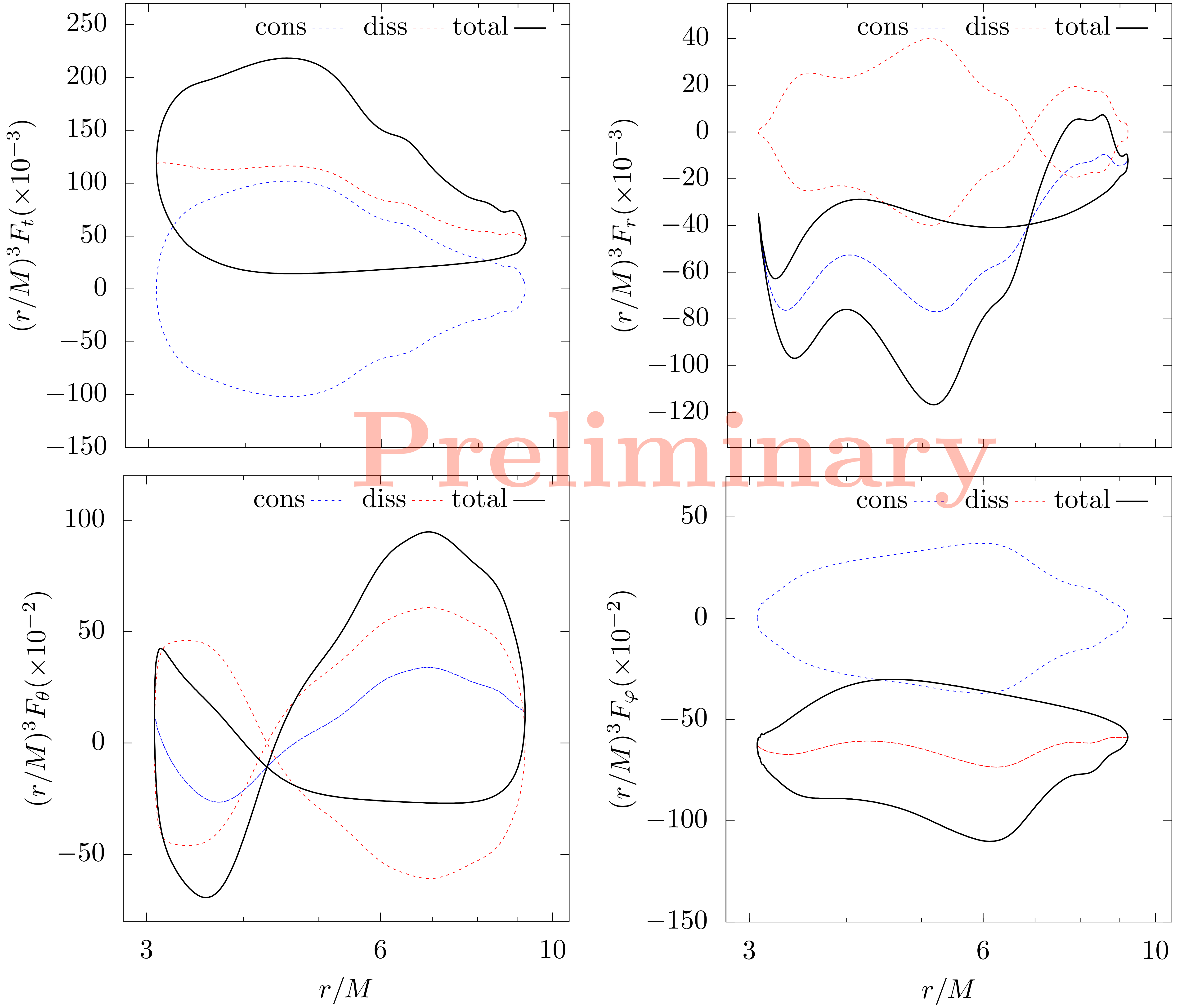




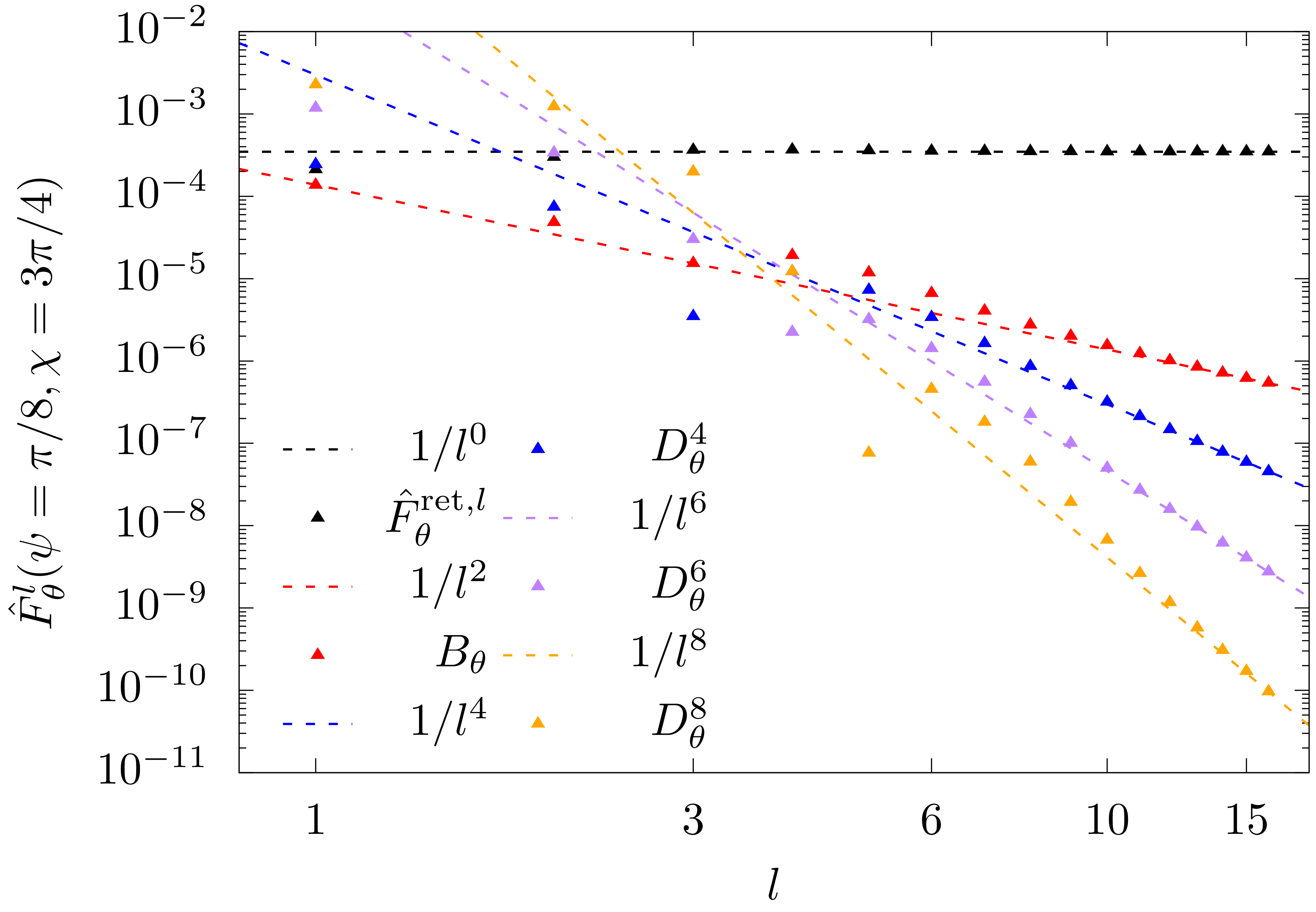
2:1 resonant orbit $(\Omega_\theta/\Omega_r = 2)$

$p \approx 4.6, e = 0.5, \iota \approx 0.79, a = 0.9M$

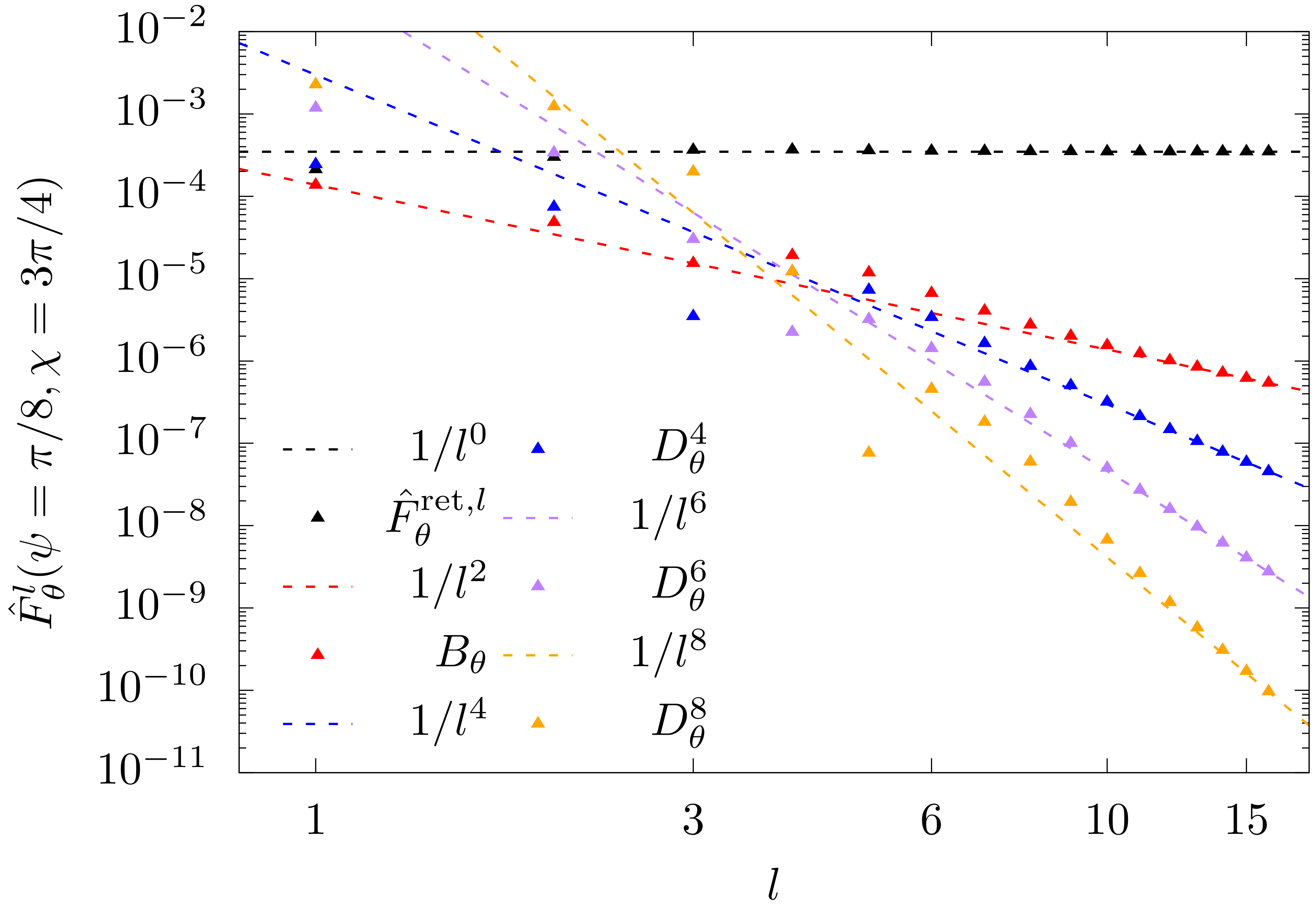


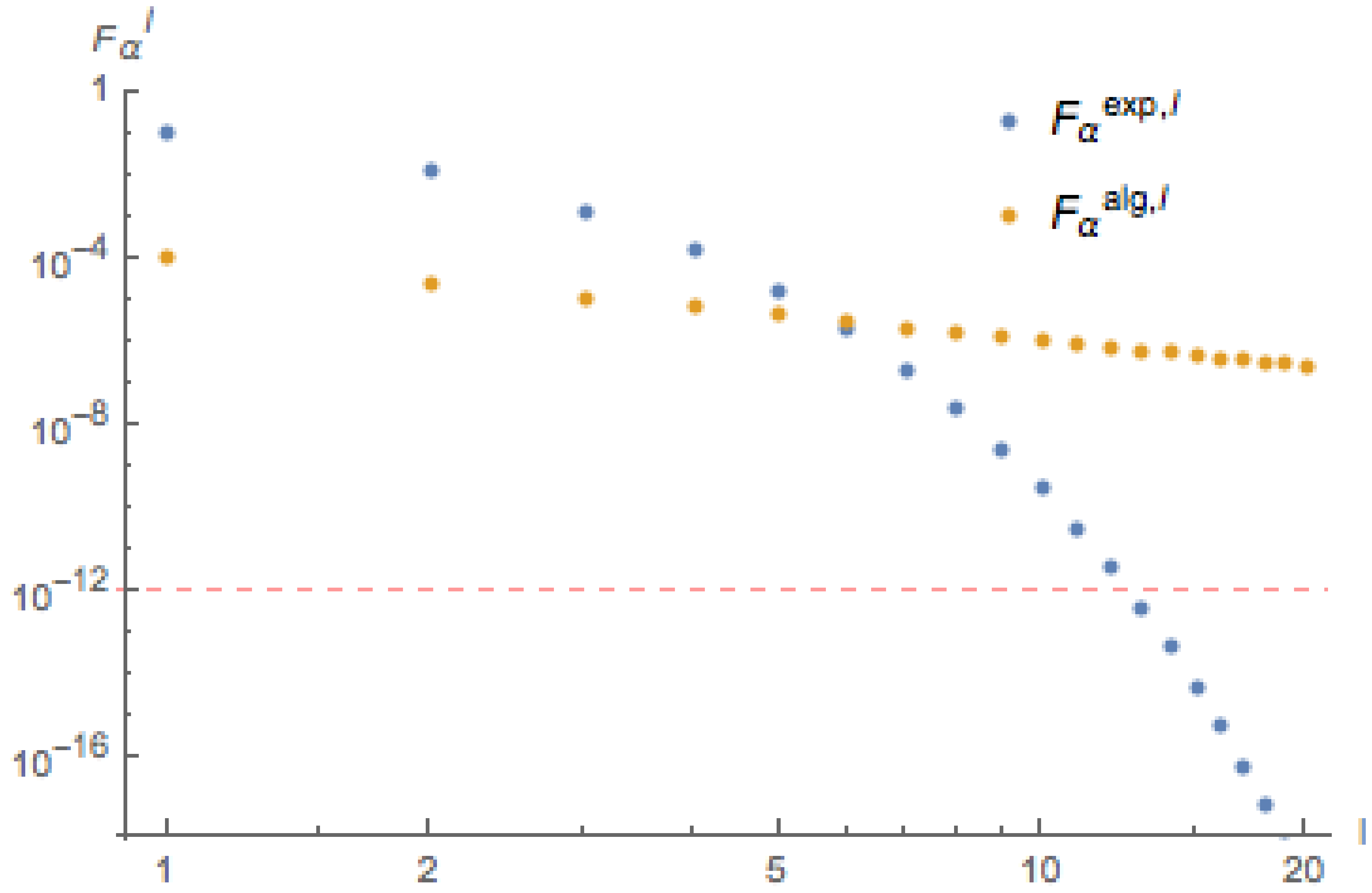


Regularization scheme

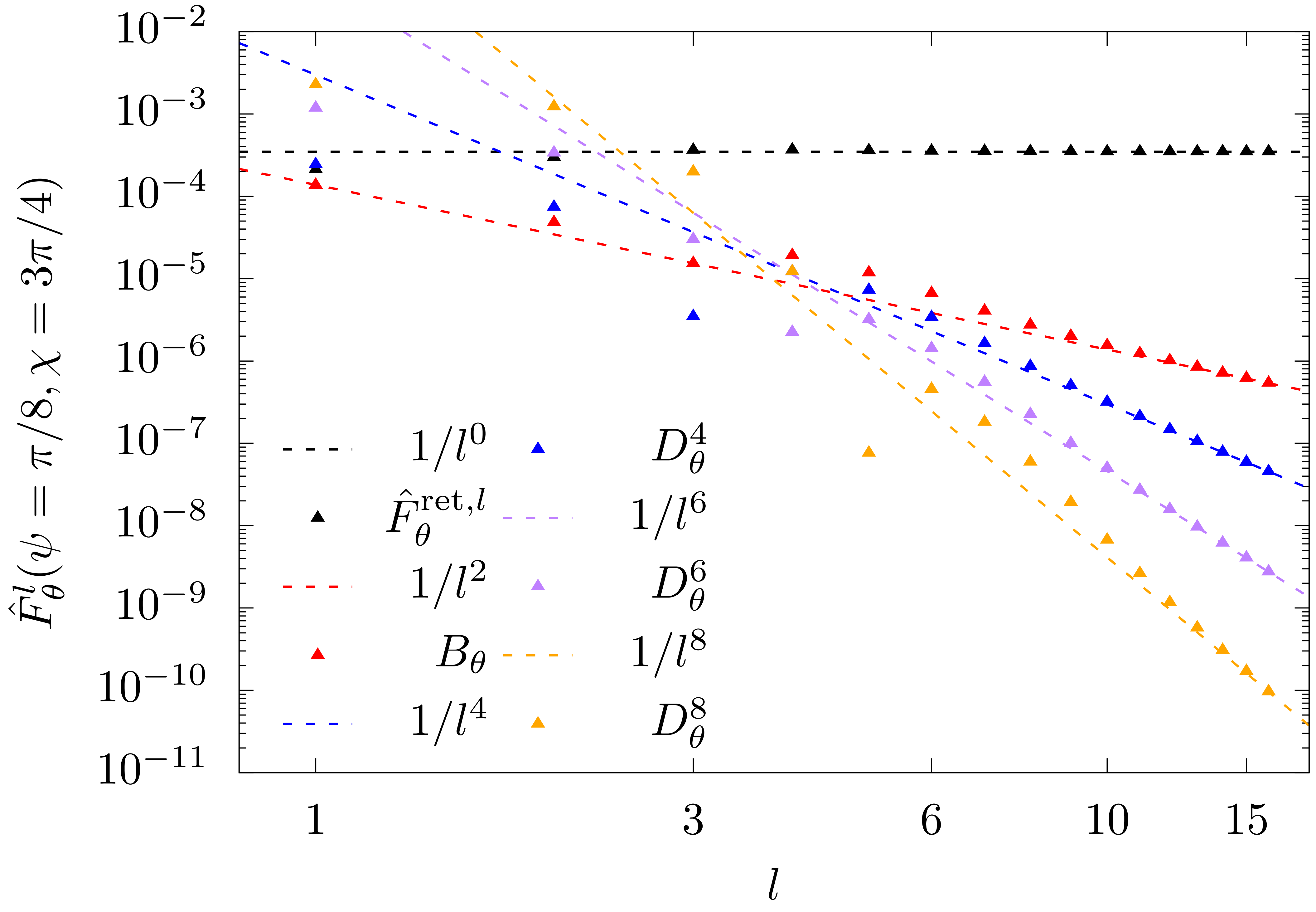


Regularization scheme





Regularization scheme



Regularization scheme