A finite element framework for stationary solutions in curved spacetime

Romain Gervalle

XVIII Black Holes Workshop

December 18, 2025

R. Gervalle, *Spinning mini-Boson Stars* (2025), GitHub: romaingrvl/spinning_mini_BS

universidade de aveiro

CIDMA

Gr@v

fct Fundação para a Ciência e a Tecnologia

Numerical methods in curved spacetime

Einstein's field equations are **highly nonlinear.**

$\rightarrow$ **Numerical methods** are essential for constructing solutions and complementing analytical approaches.

$\rightarrow$ Here: a **Finite Element** code for stationary problems using `FreeFem`.

FREEFEM

F. Hecht, *J. Numer. Math. 20* (2012) 251, website: https://freefem.org/tryit

Finite elements in curved spacetime

We use *quasi-isotropic coords.* (t, r, θ, φ) + *compactification* $r = cx/(1-x)$,

$$ds^2 = -e^{2F_0} dt^2 + e^{2F_1} \left(r'^2 dx^2 + r^2 d\theta^2\right) + e^{2F_2} r^2 \sin^2\theta \left(d\varphi - \frac{w}{r^2} dt\right)^2 .$$

Unknowns: $\mathcal{F} = \{F_0, F_1, F_2, w, \dots\}$.

Finite elements in curved spacetime

We use *quasi-isotropic coords.* (t, r, θ, φ) + *compactification* $r = cx/(1-x)$,

$$ds^2 = -e^{2F_0} dt^2 + e^{2F_1}\left(r'^2 dx^2 + r^2 d\theta^2\right) + e^{2F_2} r^2 \sin^2\theta \left(d\varphi - \frac{w}{r^2} dt\right)^2 .$$

<u>Unknowns:</u> $\mathcal{F} = \{F_0, F_1, F_2, w, \dots\}$.

`FreeFem` requires a **weak formulation** of the equations:

$$\Delta_g \mathcal{F} + \dots = 0 \quad \rightarrow \quad \boxed{\int_\Omega \left(g^{\mu\nu}\partial_\mu \mathcal{F} \partial_\nu v_{\mathcal{F}} + \dots\right)\sqrt{-g}\, dx\, d\theta = 0.}$$

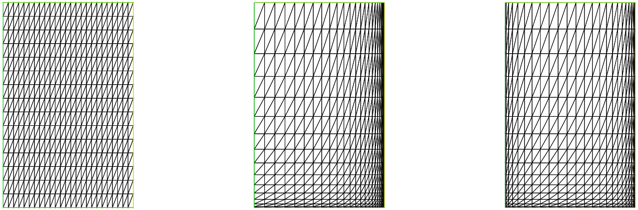
Discretization

→ Introduce a **finite dimensional** space V_h (piecewise polynomial functions).

→ Expanding $\mathcal{F}$ and $v_{\mathcal{F}}$ in a basis $\mathfrak{B}$ of V_h, the problem reduces to a **nonlinear algebraic system** for $(\mathcal{F}_1, \mathcal{F}_2, \dots)$.

Computational domain: $\Omega = [0,1] \times [0, \pi/2]$.

`FreeFem` provides a built-in **mesh generator.**

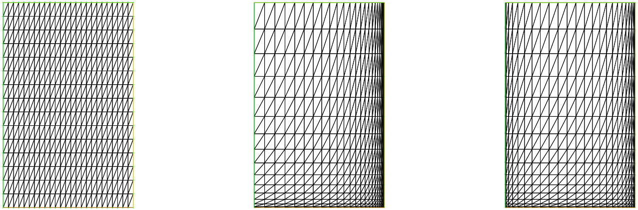


- **Parameters:** N_x, N_θ (edges along along x, θ).
- Supports **non-uniform** meshes.

Mesh generation

Computational domain: $\Omega = [0,1] \times [0,\pi/2]$.

`FreeFem` provides a built-in **mesh generator.**



- **Parameters:** N_x, N_θ (edges along along x, θ).
- Supports **non-uniform** meshes.

We work with $V_h = \mathbb{P}_h^2$: continuous piecewise **quadratic** functions.

$\rightarrow$ On a single cell, $\mathcal{F}(x,\theta) = c_1 + c_2 x + c_3 \theta + c_4 x^2 + c_5 x\,\theta + c_6 \theta^2$

The boson star factory (with C. A. R. Herdeiro, E. Radu, E. S. Costa Filho, *et al.*)

Work in progress: comparison of different numerical approaches for constructing **axisymmtric** Boson Stars (BSs).

Model: **Complex scalar field** Φ minimally coupled to **Einstein's gravity.**

$$\boxed{\mathcal{L} = \frac{R}{16\pi G} - \frac{1}{2} g^{\alpha\beta} \left(\partial_\alpha \Phi^* \partial_\beta \Phi + \partial_\beta \Phi^* \partial_\alpha \Phi \right) - \mu^2 \Phi^* \Phi, \quad \Phi = \phi(r,\theta) e^{im\varphi - i\omega t}.}$$

The boson star factory (with C. A. R. Herdeiro, E. Radu, E. S. Costa Filho, *et al.*)

Work in progress: comparison of different numerical approaches for constructing **axisymmtric** Boson Stars (BSs).

Model: **Complex scalar field** Φ minimally coupled to **Einstein's gravity.**

$$\boxed{\mathcal{L} = \frac{R}{16\pi G} - \frac{1}{2} g^{\alpha\beta}\left(\partial_\alpha \Phi^* \partial_\beta \Phi + \partial_\beta \Phi^* \partial_\alpha \Phi\right) - \mu^2 \Phi^* \Phi, \quad \Phi = \phi(r,\theta) e^{im\varphi - i\omega t}.}$$

Input parameters: ω, m, $\mathcal{P} = \pm 1$ (ϕ parity).

Output parameters:

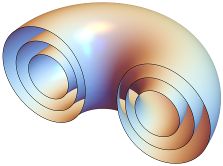
M_∞ : mass evaluated at **infinity,**

$M_{\rm bulk}$: mass from **bulk integral,**

J_∞ : angular momentum at **infinity,**

$J_{\rm bulk}$: angular momentum from **bulk.**

Single **rotating** BSs:



parity-even ϕ, $m = 1$

Monitoring the errors

Errors estimates

The code evaluates errors using the **relative differences,**

$$\left|\frac{\Delta M}{M}\right| = \left|1 - \frac{M_{\text{bulk}}}{M_\infty}\right|, \qquad \left|\frac{\Delta J}{J}\right| = \left|1 - \frac{J_{\text{bulk}}}{J_\infty}\right|,$$

+ a **virial** identity, $v_{\mathrm{m}} = \int_\Omega T^{\mu\nu}\nabla_\mu \zeta_\nu \sqrt{-g}\, dx\, d\theta$, see [**R. G.**, Volkov, 2025].

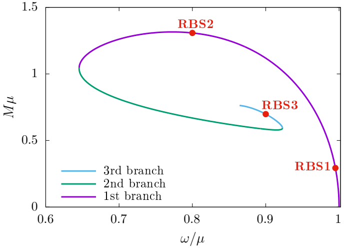
Monitoring the errors

Errors estimates

The code evaluates errors using the **relative differences,**

$$\left|\frac{\Delta M}{M}\right| = \left|1 - \frac{M_{\text{bulk}}}{M_\infty}\right|, \qquad \left|\frac{\Delta J}{J}\right| = \left|1 - \frac{J_{\text{bulk}}}{J_\infty}\right|,$$

+ a **virial** identity, $v_{\rm m} = \int_\Omega T^{\mu\nu}\nabla_\mu \zeta_\nu \sqrt{-g}\, dx\, d\theta$, see [**R. G.**, Volkov, 2025].



$\boxed{\text{Results for } N_x = 137,\ N_\theta = 54}$

Runtime for 1 solution: $\sim$**50sec.** (4 laptop CPUs).

- Errors on 1st branch: $10^{-11} - 10^{-8}$,
- Errors on 2nd branch: $10^{-9} - 10^{-6}$,
- Errors on 3rd branch: $10^{-8} - 10^{-4}$.

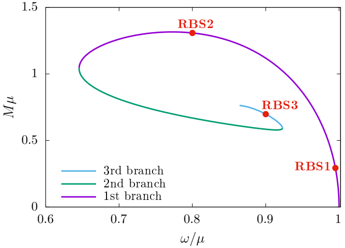
Monitoring the errors

Errors estimates

The code evaluates errors using the **relative differences,**

$$\left|\frac{\Delta M}{M}\right| = \left|1 - \frac{M_{\text{bulk}}}{M_\infty}\right|, \qquad \left|\frac{\Delta J}{J}\right| = \left|1 - \frac{J_{\text{bulk}}}{J_\infty}\right|,$$

+ a **virial** identity, $v_{\rm m} = \int_\Omega T^{\mu\nu}\nabla_\mu \zeta_\nu \sqrt{-g}\, dx\, d\theta$, see [**R. G.**, Volkov, 2025].



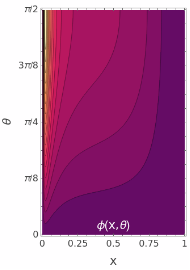
Pushing the solver to the limits

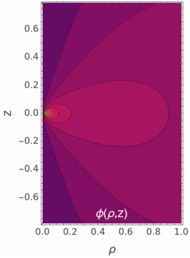
Solution	$\lvert\Delta M/M\rvert$	$\lvert\Delta J/J\rvert$	$\lvert v_{\rm m}\rvert$
RBS1	10^{-14}	10^{-11}	10^{-14}
RBS2	10^{-11}	10^{-14}	10^{-11}
RBS3	10^{-10}	10^{-10}	10^{-13}

($N_x \times N_\theta \sim 1800 \times 100$, runtime $\sim$**20min.**)

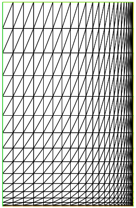
Improving quality with different meshes

RBS3 ϕ-profile:

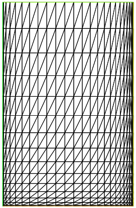




$\rightarrow$ **Very sharp** profile close to the origin ($x = 0$).



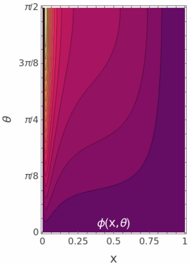
Mesh I

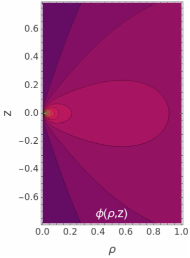


Mesh II

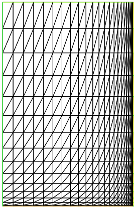
Improving quality with different meshes

RBS3 ϕ-profile:

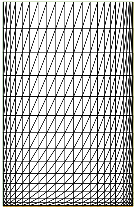




$\rightarrow$ **Very sharp** profile close to the origin ($x = 0$).



Mesh I



Mesh II

$\rightarrow$ Refining the mesh near $x = 0$ improves accuracy **a lot!**
(Keeping the overall resolution $N_x \times N_\theta$ fixed)

Mesh	$\lvert\Delta M/M\rvert$	$\lvert\Delta J/J\rvert$	$\lvert v_{\mathrm{m}}\rvert$
I	10^{-3}	10^{-3}	10^{-6}
II	10^{-7}	10^{-7}	10^{-8}

Conclusion

My code is able to construct **stationary curved-spacetime solutions** using the **Finite Element Method** (`FreeFem`).

Parallelized for **fast, high-accuracy** computations.

→ GitHub: `romaingrvl/spinning_mini_BS`

Further applications:

- Kerr black holes (validation against analytical expressions).
- Chains of rotating BSs (with self-interactions).

  R.G., *Phys.Rev.D* 105 (2022) 124052 – arXiv:2206.03982
- Black holes with electroweak hair (**10 PDEs** to solve!)

  R.G., M. S. Volkov, *Phys.Rev.Lett.* 133 (2024) 171402 – arXiv:2406.14357
  R.G., M. S. Volkov, *Phys.Rev.D* 112 (2025) 064018 – arXiv:2504.09304

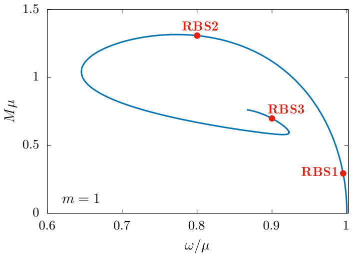
Future directions: Modified gravity, extension to 3D problems, time evolution?

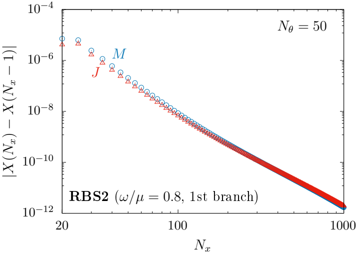
I am open to collaborations!

Thank you for your attention.

This work is supported by CIDMA under the Portuguese Foundation for Science and Technology (FCT, `https://ror.org/00snfqn58`) Multi-Annual Financing Program for RD Units, grants UID/4106/2025, UID/PRR/4106/2025, 2022.04560.PTDC and 2024.05617.CERN.

Convergence





$\rightarrow$ Increasing the resolution yields **good convergence** of output quantities X, with $\log|X(N_x) - X(N_x - 1)|$ vs. $\log N_x$ approaching a slope $\sim$**4.**

Pushing the solver to the limits: ($N_x \times N_\theta \sim 1800 \times 100$, runtime $\sim$**20min.**)

Solution	$\lvert\Delta M/M\rvert$	$\lvert\Delta J/J\rvert$	$\lvert v_{\mathrm{m}}\rvert$
RBS1	6.7×10^{-14}	9.2×10^{-11}	4.8×10^{-14}
RBS2	5.0×10^{-11}	6.9×10^{-14}	3.7×10^{-11}
RBS3	2.6×10^{-10}	2.4×10^{-10}	4.2×10^{-13}