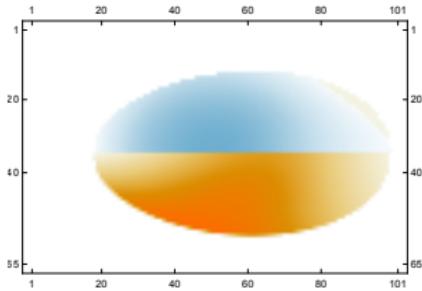


# Keplerian toroids in puncture framework - numerical convergence

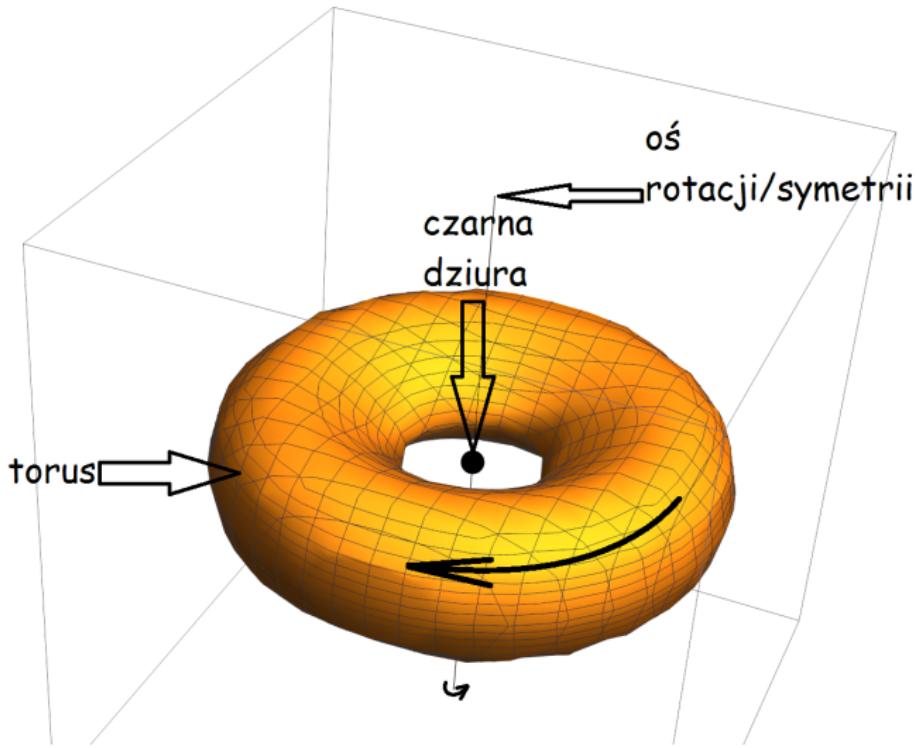
Andrzej Odrzywołek

Zakład Teorii Względności i Astrofizyki, Instytut Fizyki UJ

17 stycznia 2018



# Sketch of physical object



# Spacetime 4-metric

$$g_{\mu\nu} = \begin{pmatrix} r^2 \sin^2(\theta) \beta^2 \psi^4 - \alpha^2 & 0 & 0 & r^2 \sin^2(\theta) \beta \psi^4 \\ 0 & e^{2q} \psi^4 & 0 & 0 \\ 0 & 0 & e^{2q} r^2 \psi^4 & 0 \\ r^2 \sin^2(\theta) \beta \psi^4 & 0 & 0 & r^2 \sin^2(\theta) \psi^4 \end{pmatrix}$$

$$\alpha \equiv \alpha(r, \theta), \beta \equiv \beta(r, \theta), q \equiv q(r, \theta), \psi \equiv \psi(r, \theta)$$

$$ds^2 = -\alpha^2 dt^2 + \psi^4 \left[ e^{2q} (dr^2 + r^2 d\theta^2) + r^2 \sin^2 \theta (\beta dt + d\varphi)^2 \right] \quad (1)$$

# Velocity field

$$U^\mu = \{u^t[r, \theta], 0, 0, u^\varphi[r, \theta]\}$$

$$U_\mu = \{u_t[r, \theta], 0, 0, u_\varphi[r, \theta]\}$$

$$U^\mu = \left\{ -\frac{u_t - \beta u_\varphi}{\alpha^2}, 0, 0, \frac{u_t - \beta u_\varphi}{\alpha^2} \beta + \frac{u_\varphi}{\psi^4 r^2 \sin^2 \theta} \right\}$$

$$U_\mu = \{-\alpha^2 u^t + \psi^4 r^2 \sin^2 \theta (u^\varphi + u^t \beta) \beta, 0, 0, \psi^4 r^2 \sin^2 \theta (u^\varphi + u^t \beta)\}$$

$$\Omega \equiv \frac{u^\varphi}{u^t}, \quad j \equiv = u^t u_\varphi$$

$$T_{\mu\nu} = \rho h U_\mu U_\nu + P g_{\mu\nu}, \quad P = K \rho^\Gamma$$

$$\frac{1}{j(\Omega)} = -\kappa \Omega + w^{-4/3} \Omega^{1/3}$$

$$\left( \partial_{rr} + \frac{1r^2 - 1r_s^2}{r(r^2 - r_s^2)} \right) \partial_r + \frac{1}{r^2} \partial_{\theta\theta} + \frac{0 \cot \theta}{r^2} \partial_\theta \Big) q = S_q, \quad (7a)$$

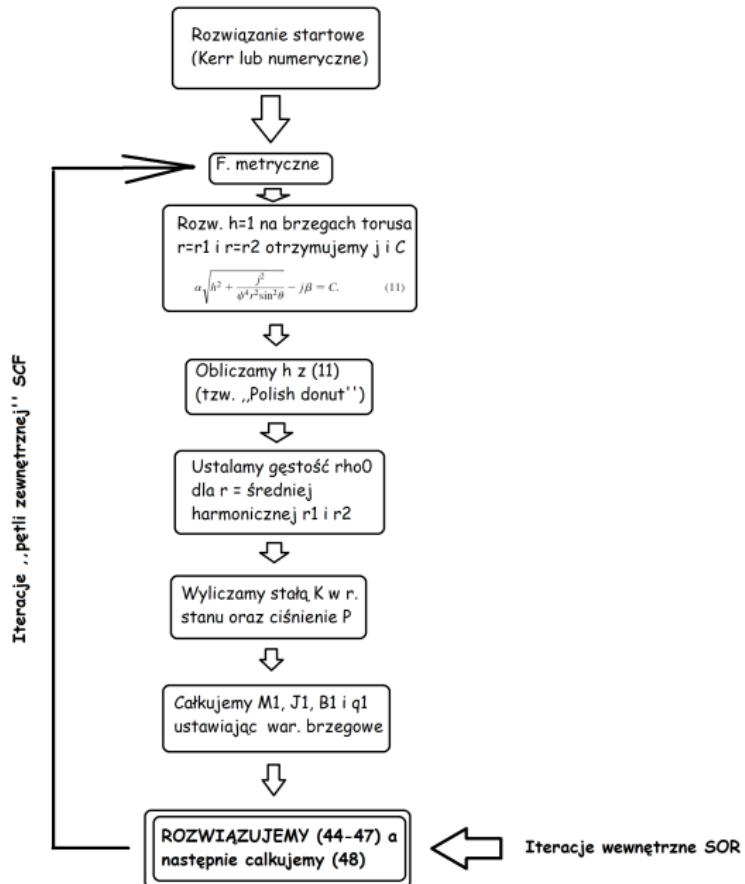
$$\left( \partial_{rr} + \frac{2r^2 + 0r_s^2}{r(r^2 - r_s^2)} \right) \partial_r + \frac{1}{r^2} \partial_{\theta\theta} + \frac{1 \cot \theta}{r^2} \partial_\theta \Big) \phi = S_\phi \quad (7b)$$

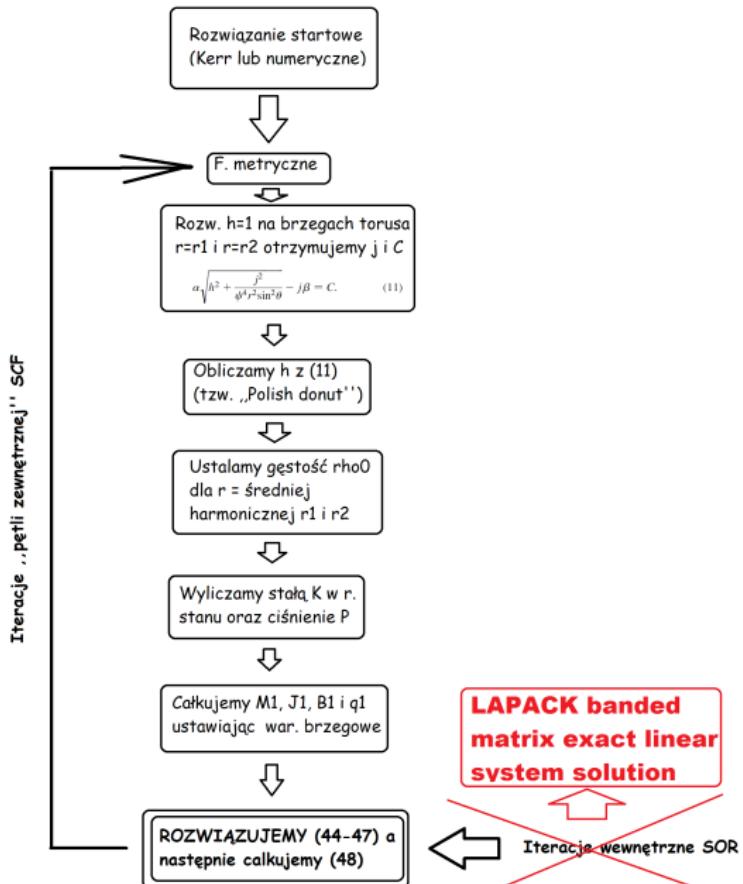
$$\left( \partial_{rr} + \frac{3r^2 + 1r_s^2}{r(r^2 - r_s^2)} \right) \partial_r + \frac{1}{r^2} \partial_{\theta\theta} + \frac{2 \cot \theta}{r^2} \partial_\theta \Big) B = S_B, \quad (7c)$$

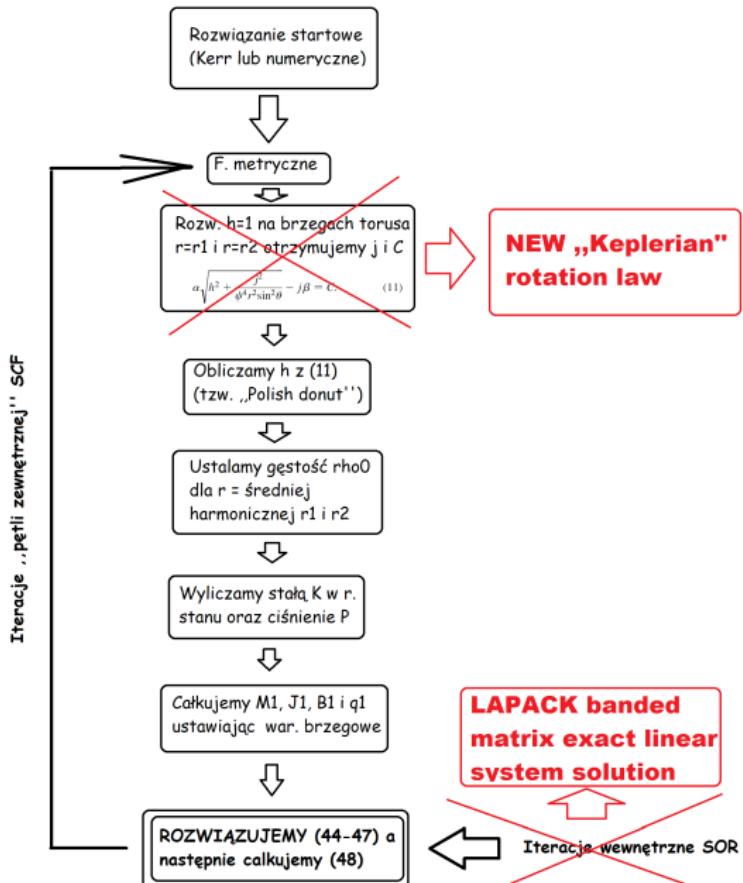
$$\left( \partial_{rr} + \frac{4r^2 + 2r_s^2}{r(r^2 - r_s^2)} - \frac{8r_s}{r^2 - r_s^2} \right) \partial_r + \frac{1}{r^2} \partial_{\theta\theta} + \frac{3 \cot \theta}{r^2} \partial_\theta \Big) \beta_T = S_{\beta_T}, \quad (7d)$$

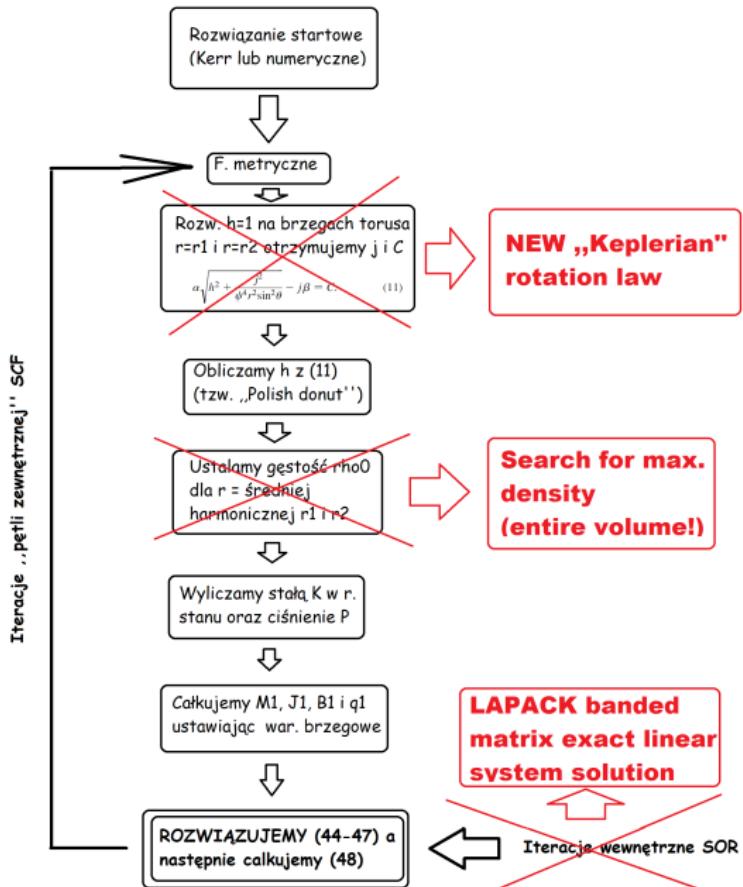
Algorithm solves four above equations, additional eq. for  $\beta_K$  alternatively with algebraic „Bernoulli” equation.

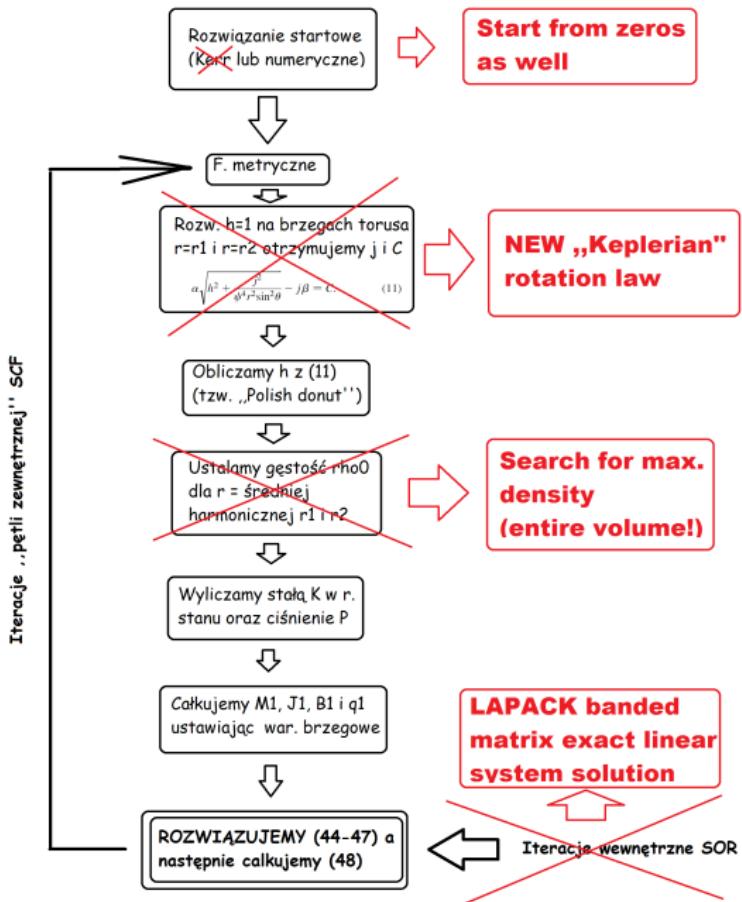
This is known as **Self Consistent Field** method (SCF) [hSCF].



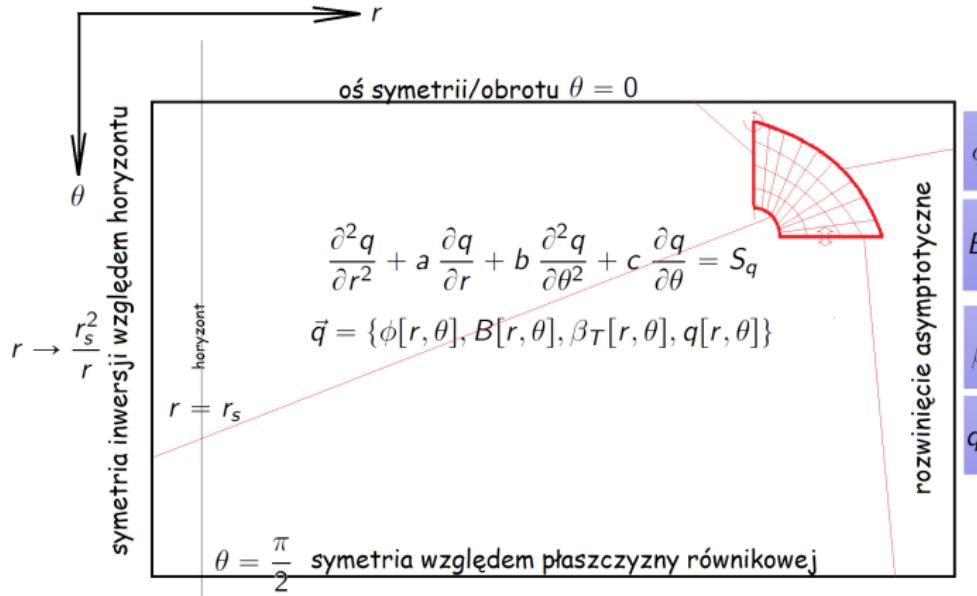








# Warunki brzegowe



$$\phi \rightarrow \frac{M_1}{2r}$$

$$B \rightarrow 1 - \frac{B_1}{r^2}$$

$$\beta_T \rightarrow -\frac{2J_1}{r^3}$$

$$q \rightarrow \frac{q_1 \sin^2 \theta}{r^2}$$

rozwiniecie asymptotyczne

# General remarks

- mixed surface capturing/tracking scheme
- horizon is **tracked** (i.e. grid aligned) via puncture scheme
- toroid surface is **captured**
- we know exact horizon location, but only approximate disk location (up to nearest grid point)

## Physical params ( $G = c = 1$ )

- ①  $m, a$  — „Kerr” mass & spin ( $m = 1.0, a = 0.6$  i.e.  $r_s = 0.4$ )
- ②  $r_1, r_2$  — inner and outer torus radii ( $r_1 = 8, r_2 = 20$ )
- ③  $\rho_0$  — maximum density ( $\rho_0 = 4 \times 10^{-4}$ )

## Numerical parameters (Shibata defaults)

- ① nr, nt — radial & angular grid size ( nr=800, nt=100 )
- ② rOUT — outer grid radius ( rOUT = 5700 rs )
- ③ niter — main operator nesting depth (inaplicable)

## Tested factors related to accuracy of the final result

- ① main loop iterations niter
- ② radial grid resolution nr
- ③ angular grid resolution nt
- ④ size of the grid (outer radius extension)
- ⑤ alignment of radial grid with torus inner and outer radii

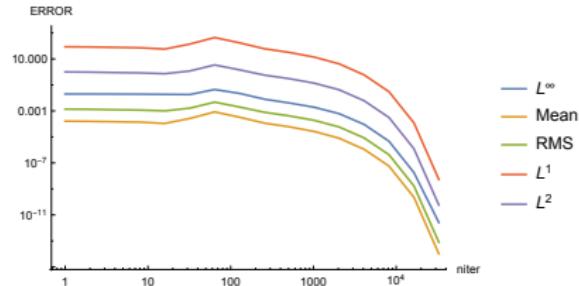
# Remark on norms used to test convergence

## Definition of absolute error

$$\text{errorABS} = ||f - f_{\text{ref}}||$$

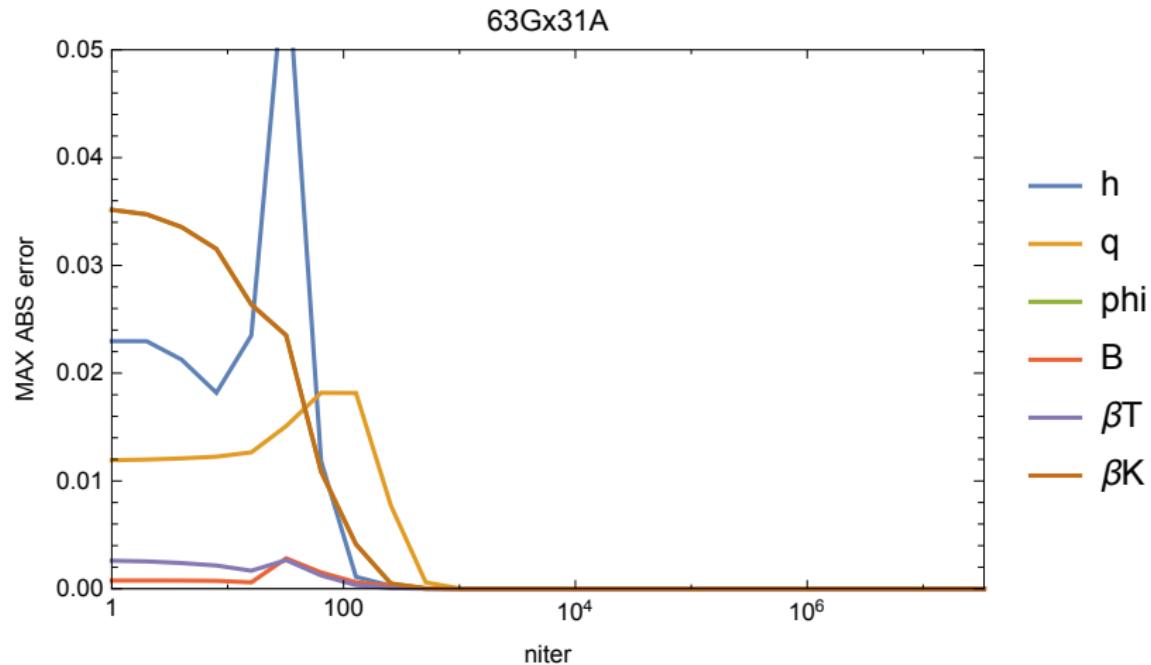
Matrix  $f$  is any of  $\phi, B, q, \beta_T, \beta_K, h$ . Reference model  $f_{\text{ref}}$  is either latest of sequence or „manufactured” best possible result (hardware/memory/CPU limited, **2047x255**).

Various „norms” could be used to quantify convergence:

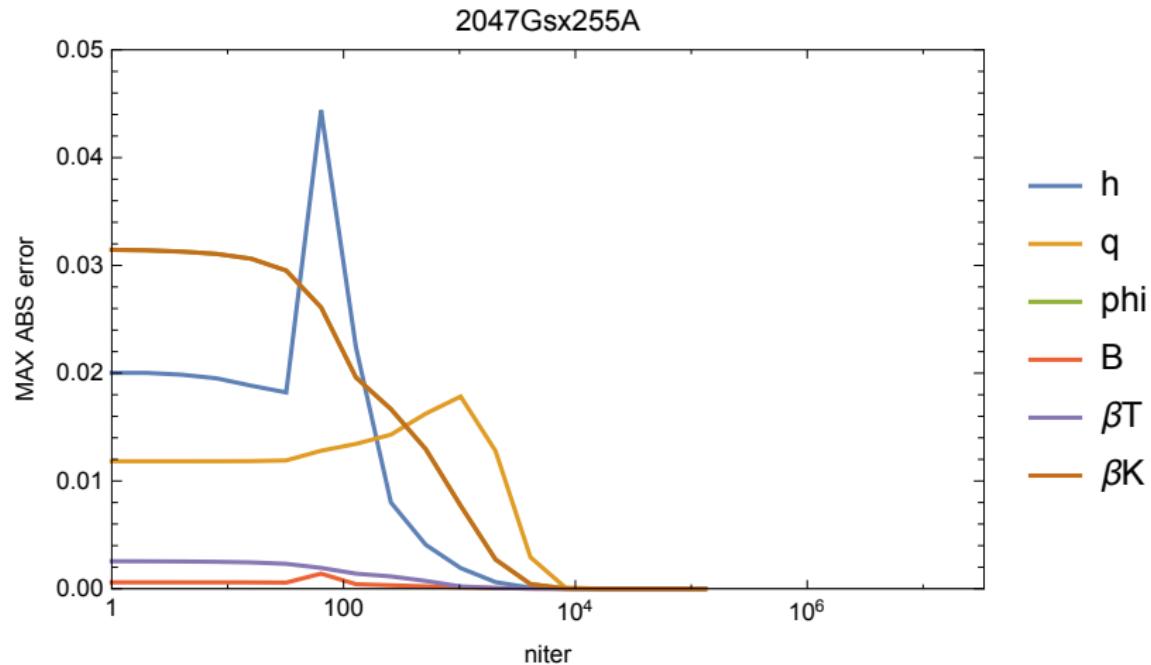


Later  $|| \cdot ||_\infty$  norm (maximum) will be used.

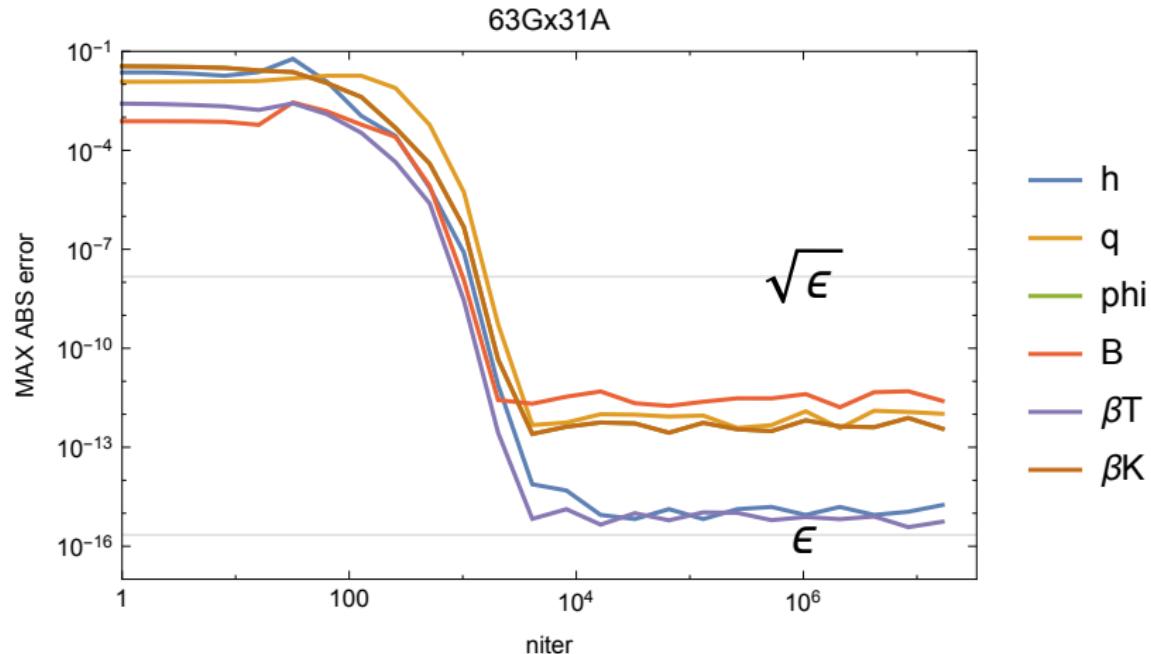
# Test 1: solution behaviour with increasing niter



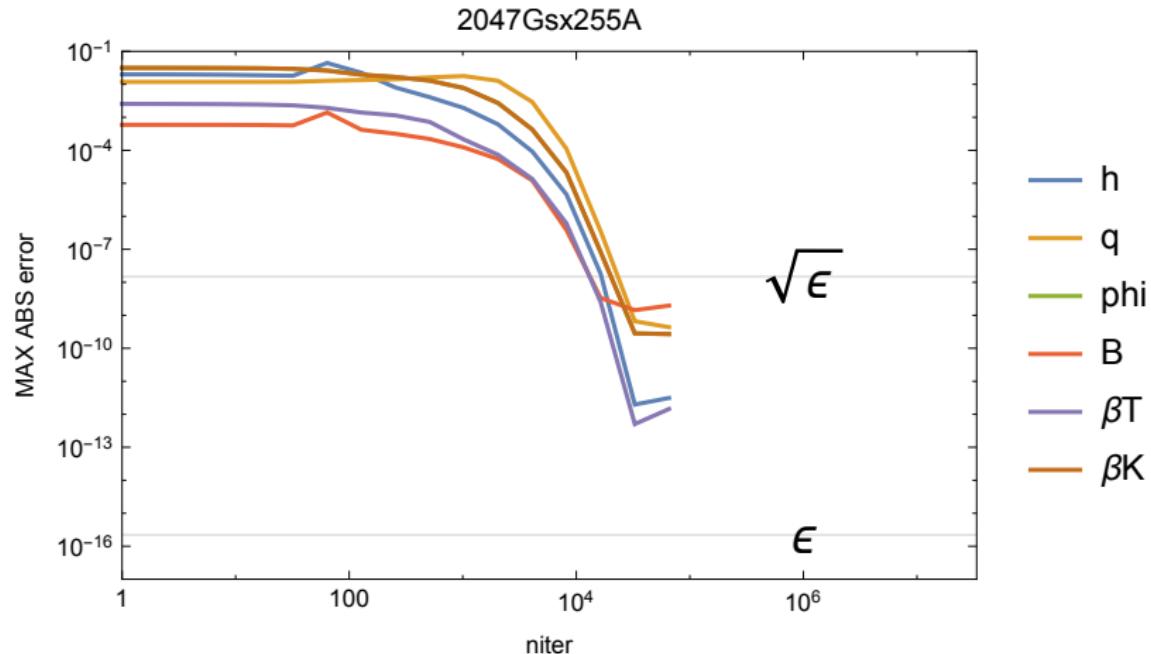
# Test 1: solution behaviour with increasing niter



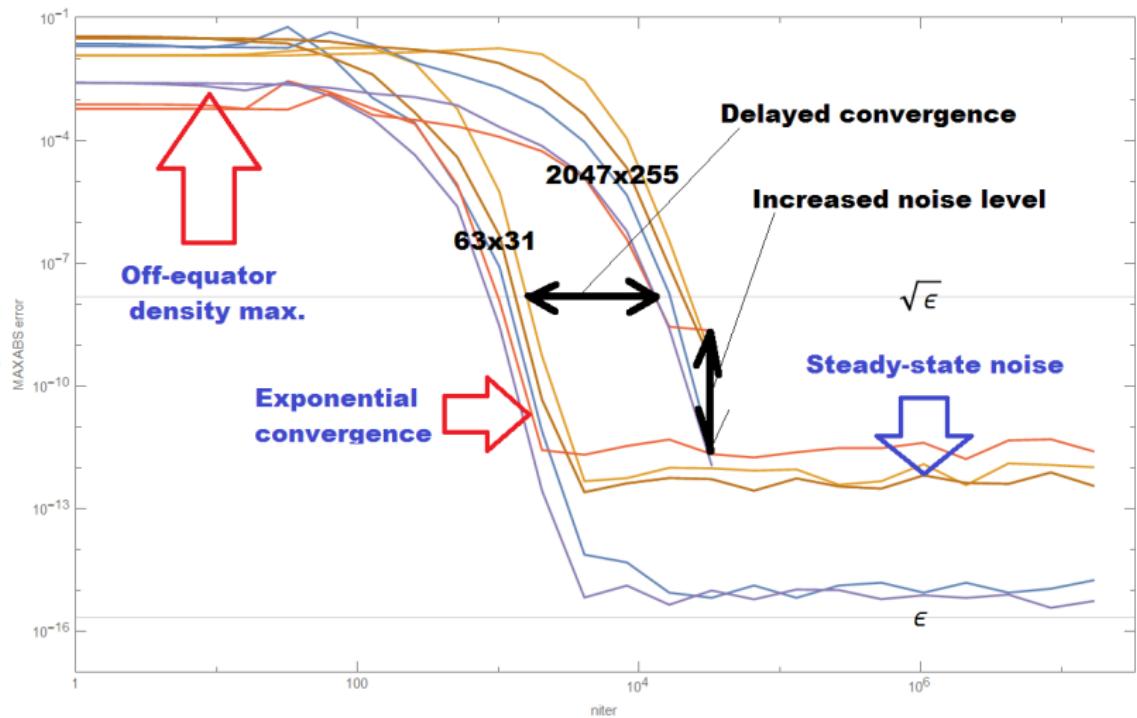
# Test 1: solution behaviour with increasing niter



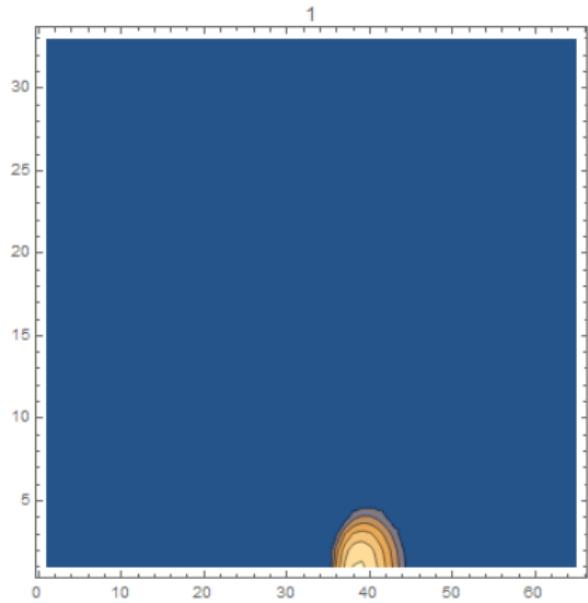
# Test 1: solution behaviour with increasing niter



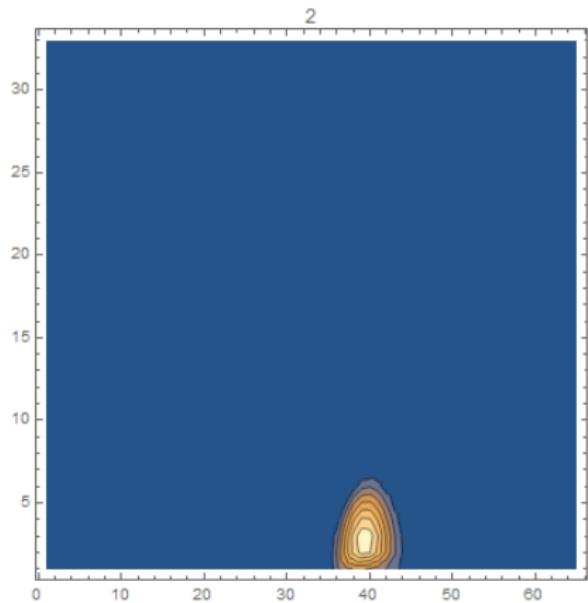
# Test 1: conclusions



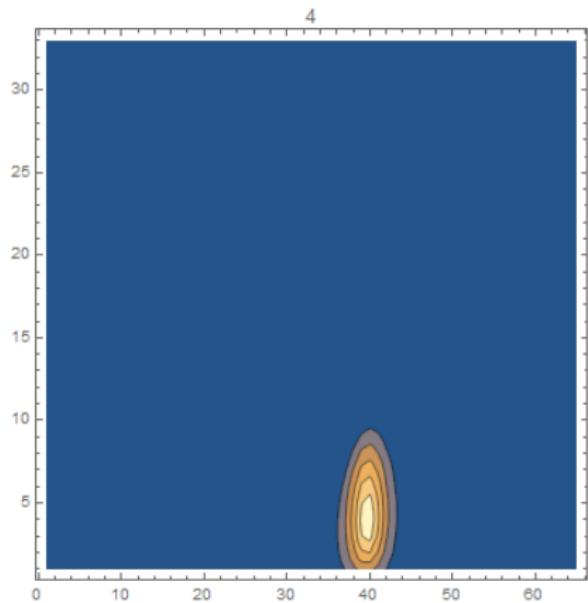
# Test 1: visualization



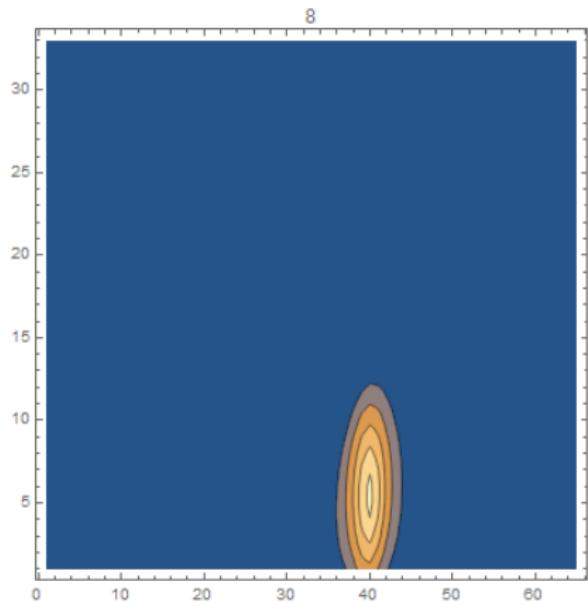
# Test 1: visualization



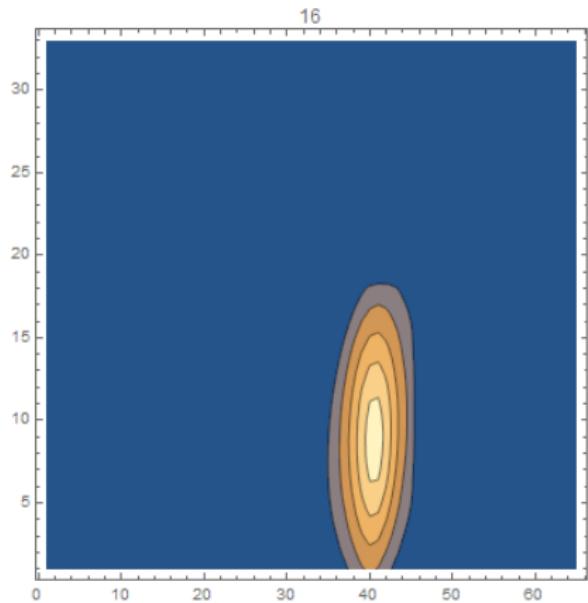
# Test 1: visualization



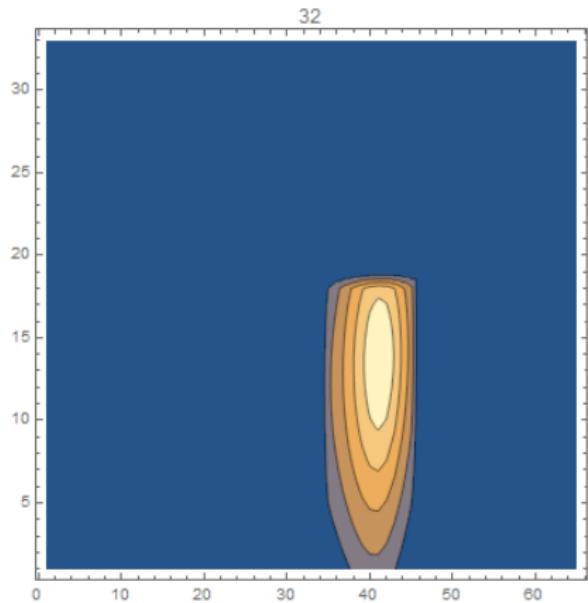
# Test 1: visualization



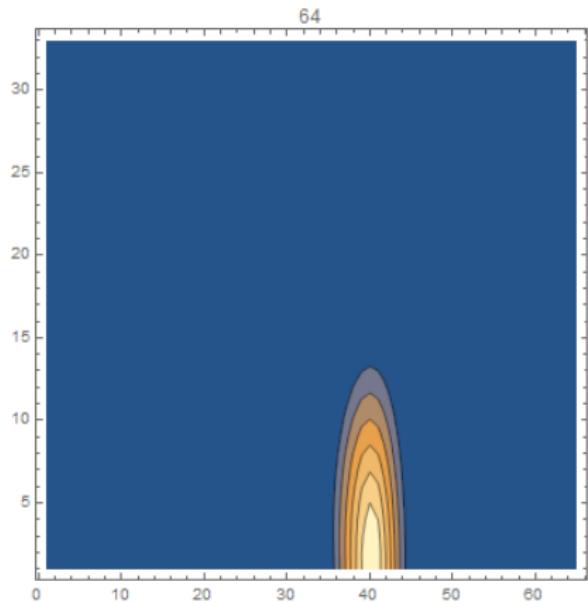
# Test 1: visualization



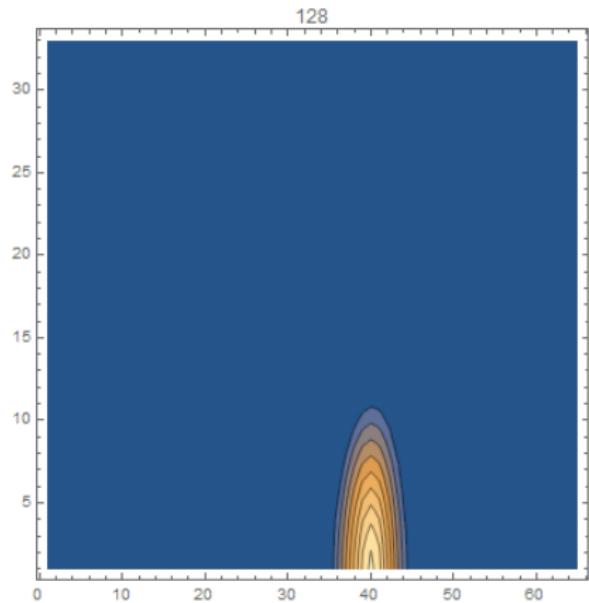
# Test 1: visualization



# Test 1: visualization

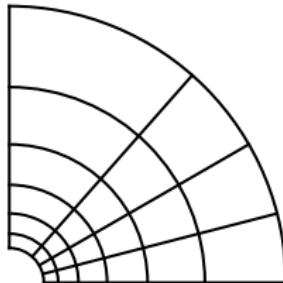


# Test 1: visualization



# Grid resolution test methodology

- ① base grid:  $2^k - 1 \times 2^m - 1$ , e.g.  
 $7 \times 3$
- ② doubling radial grid
- ③ doubling angular grid
- ④ extending radial grid

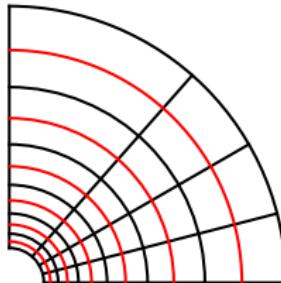


To facilitate procedure odd-odd ( geometric for  $r$  and arithmetic for  $\cos\theta$  ) grids were used:

$$r_i = r_s q^i, \quad i = 0 \dots 2^k \quad \theta_j = \arccos((2^m - j)/2^m), \quad j = 0 \dots 2^m$$

# Grid resolution test methodology

- ➊ base grid:  $2^k - 1 \times 2^m - 1$ , e.g.  
 $7 \times 3$
- ➋ doubling radial grid
- ➌ doubling angular grid
- ➍ extending radial grid

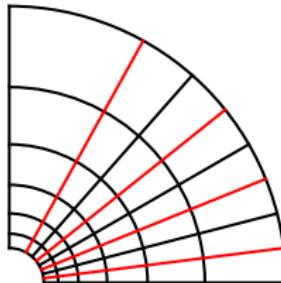


To facilitate procedure odd-odd ( geometric for  $r$  and arithmetic for  $\cos\theta$  ) grids were used:

$$r_i = r_s q^i, \quad i = 0 \dots 2^k \quad \theta_j = \arccos((2^m - j)/2^m), \quad j = 0 \dots 2^m$$

# Grid resolution test methodology

- ① base grid:  $2^k - 1 \times 2^m - 1$ , e.g.  
 $7 \times 3$
- ② doubling radial grid
- ③ doubling angular grid
- ④ extending radial grid

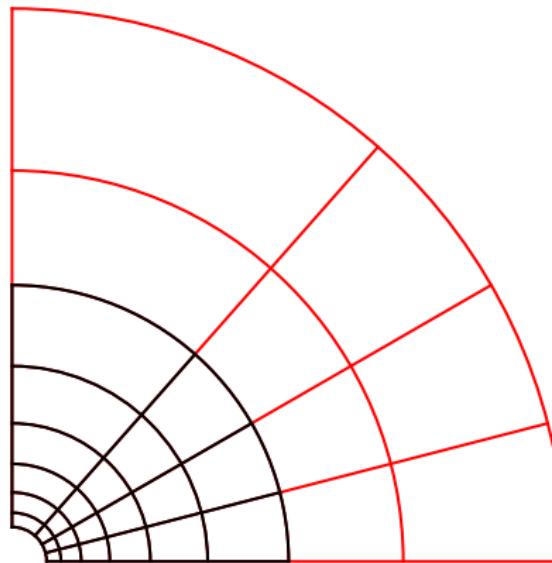


To facilitate procedure odd-odd ( geometric for  $r$  and arithmetic for  $\cos\theta$  ) grids were used:

$$r_i = r_s q^i, \quad i = 0 \dots 2^k \quad \theta_j = \arccos((2^m - j)/2^m), \quad j = 0 \dots 2^m$$

# Grid resolution test methodology

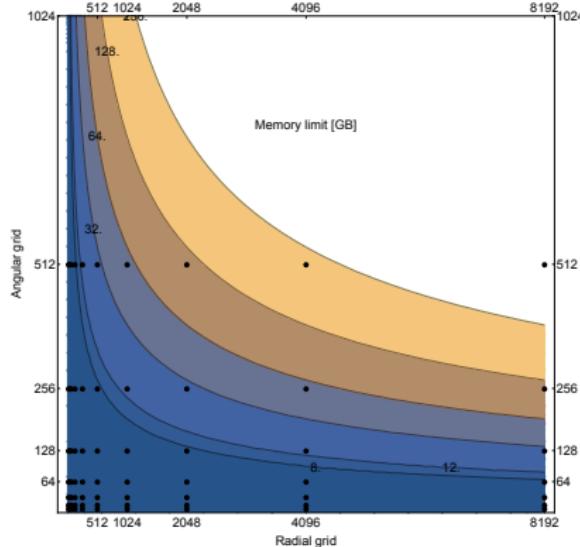
- ➊ base grid:  $2^k - 1 \times 2^m - 1$ , e.g.  
 $7 \times 3$
- ➋ doubling radial grid
- ➌ doubling angular grid
- ➍ extending radial grid



To facilitate procedure odd-odd ( geometric for  $r$  and arithmetic for  $\cos\theta$  ) grids were used:

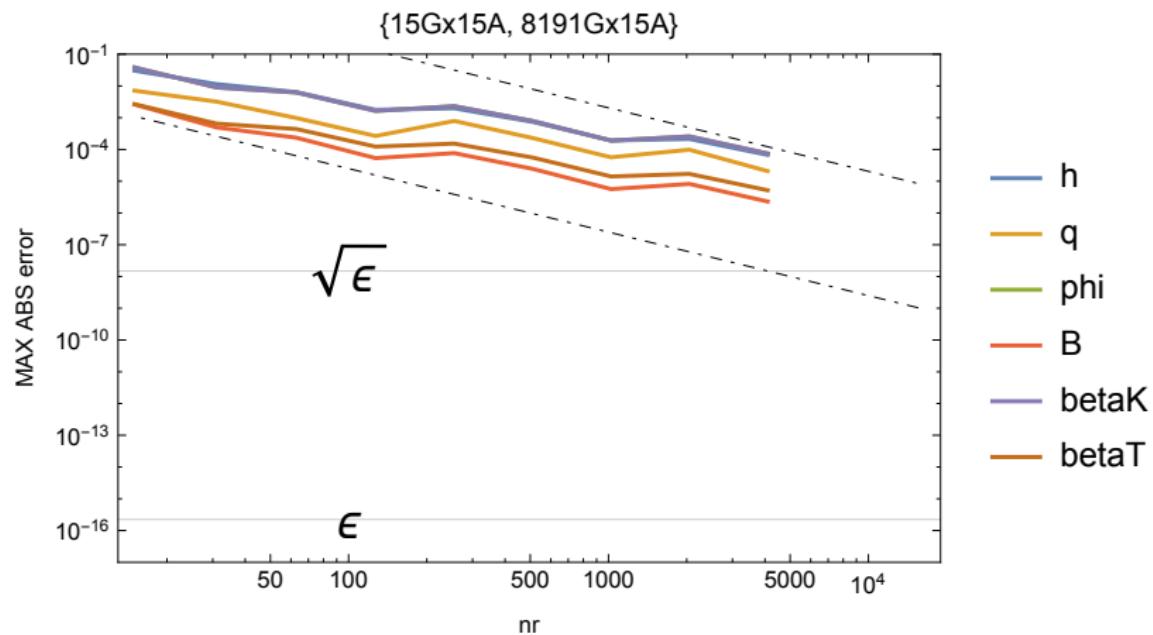
$$r_i = r_s q^i, \quad i = 0 \dots 2^k \quad \theta_j = \arccos(2^m - j)/2^m, \quad j = 0 \dots 2^m$$

# Memory limit

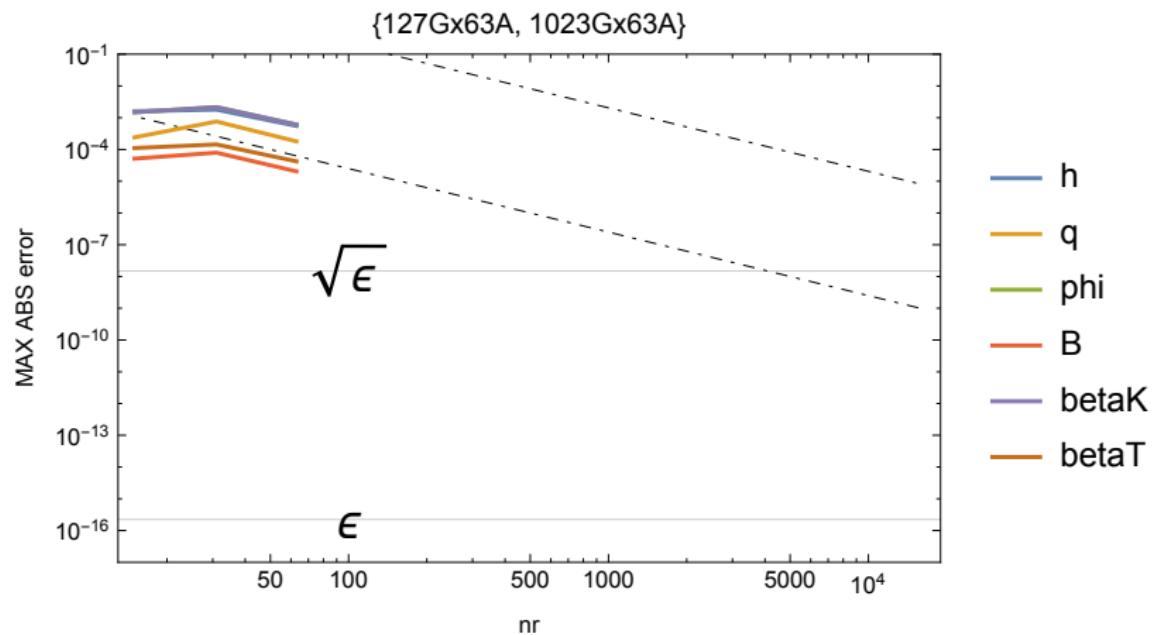


Memory required is linear with radial grid size, but quadratic with angular grid size.

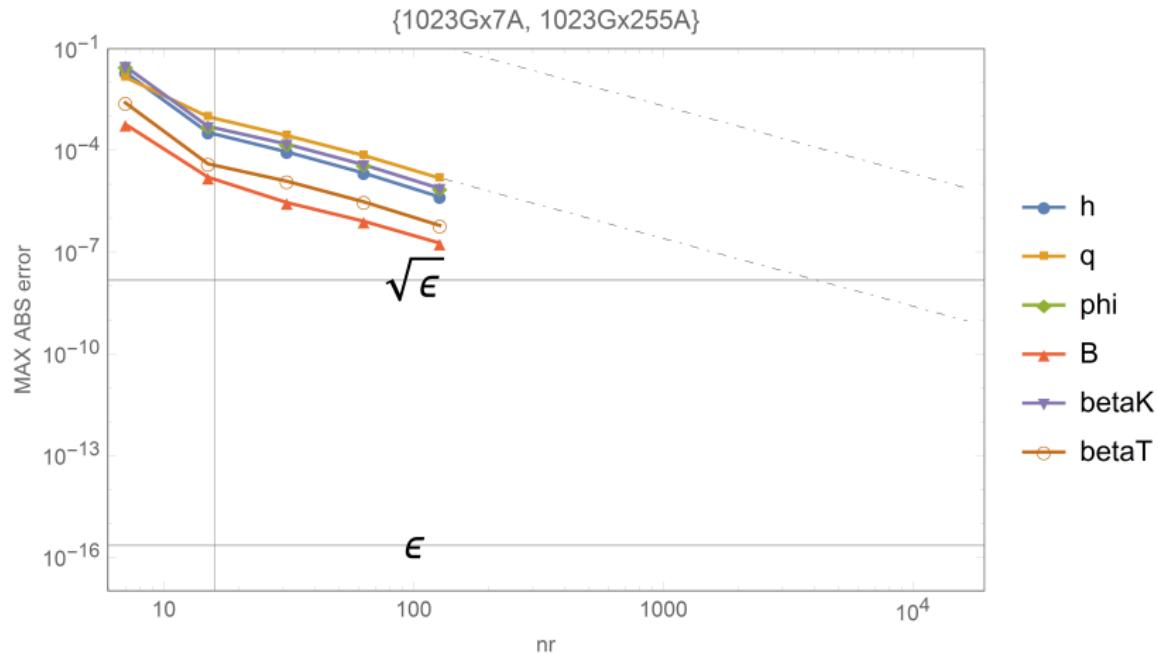
# Test 2: doubling radial grid



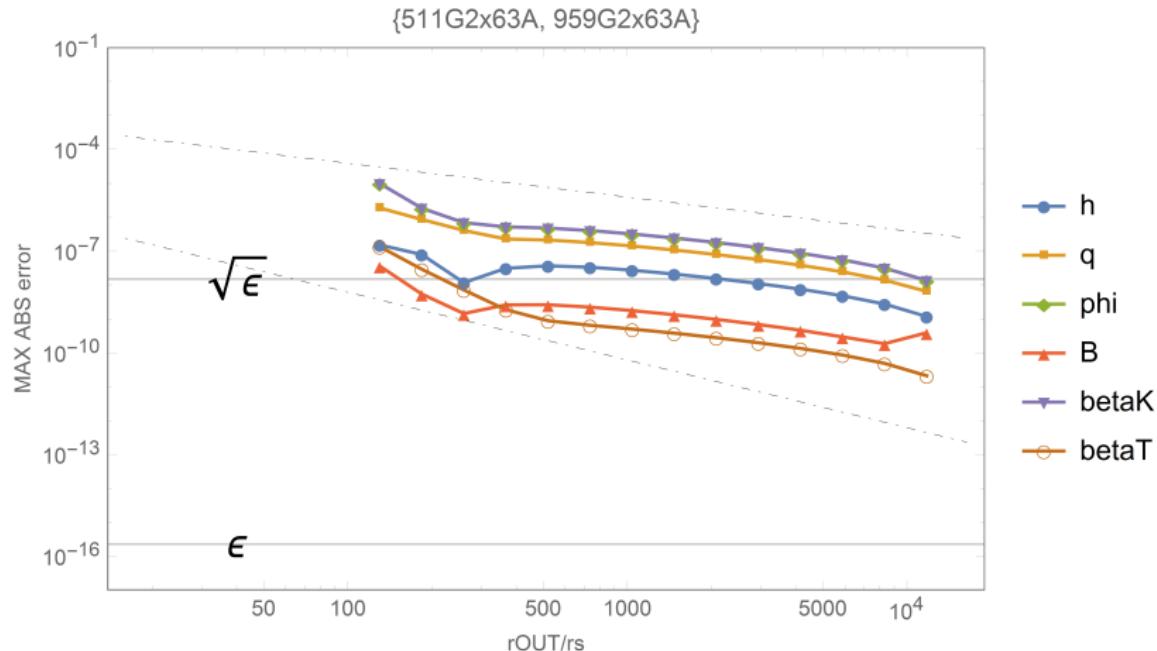
## Test 2: doubling radial grid



# Test 3: doubling angular grid



# Test 4: extending radial grid



## Typical absolute errors ( $L^\infty$ )

- ① radial grid nr:  $10^{-3} \dots 10^{-5}$
- ② angular grid nt:  $10^{-5} \dots 10^{-7}$
- ③ radial grid extension:  $10^{-8} \dots 10^{-10}$
- ④ main loop iterations:  $10^{-11} \dots 10^{-16}$

Main concern is radial resolution, least important - iteration count.

But what is the actual error of the solution?

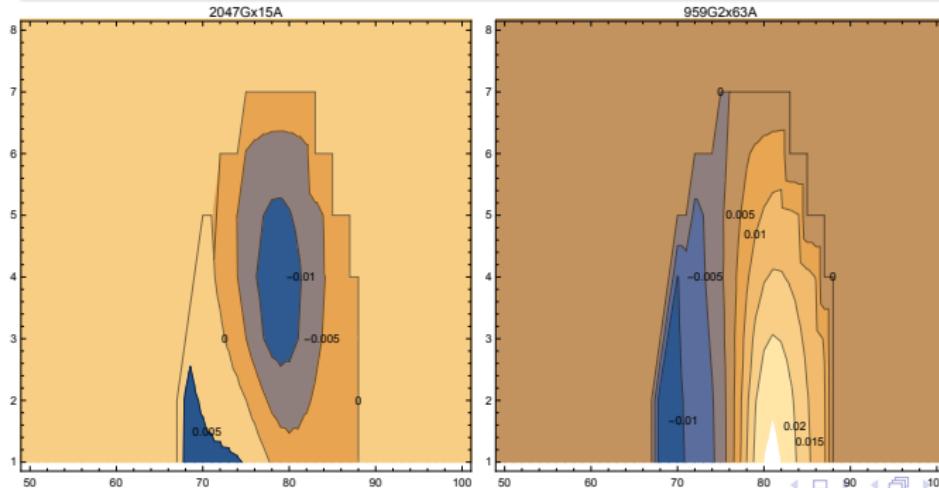
# Ordering models according to given norm

## Example

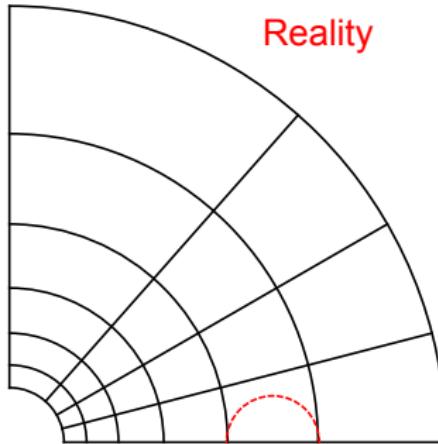
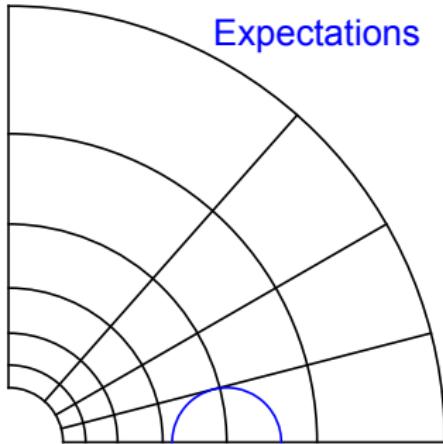
For which one enthalpy  $h$  is more accurate compared to „etalon” [2047x255,  $10^5 r_s$ , niter=2<sup>17</sup>]:

- 959x62 with rOUT = 30000 rs
- 2047x16 with rOUT = 300 rs

?



# Aliasing error



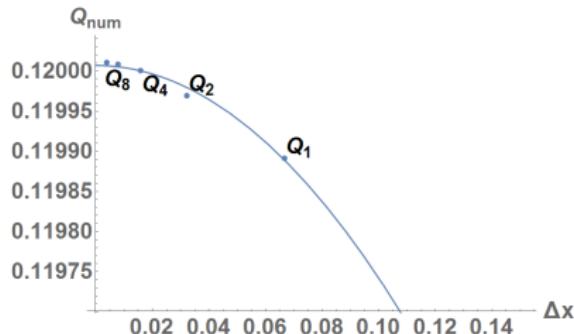
## Questions

- How convergence and model error differ between grid-aligned and random models?
- What is optimal: increase grid size or simply align grid with  $r_1$  and  $r_2$ ?

# Convergence order test

Using  $n$ -th order polynomial discrete derivatives with respect to  $x$  we expect:

$$Q_{\text{num}} \simeq Q_{\text{true}} + \lambda \Delta x^n$$



Denoting:

$$Q_1 = Q_{\text{true}} + \lambda \Delta x^n, \quad Q_2 = Q_{\text{true}} + \lambda (2 \Delta x)^n, \quad Q_4 = Q_{\text{true}} + \lambda (4 \Delta x)^n$$

we have:

$$R_{\text{con}} \equiv \frac{Q_4 - Q_1}{Q_2 - Q_1} = 2^n + 1.$$

In particular, for  $n = 2$   $R_{\text{con}} = 5$ !

# Conclusions

- ① robust code working for variety of physical parameters: crashes or divergence highly unlikely
- ② initial data: zeros or other model
- ③ grid resolution as low as 16x16 and as high as 16536x512 available on typical modern computers with 32 GB-256 GB memory
- ④ reasonable computational time of  $\sim$  hours/model on standard laptop
- ⑤ Kerr parameter up to at least 0.99
- ⑥ any rotation law  $j(\Omega)$  (demonstrated on Keplerian-Kerr formula)
- ⑦ large unexplored potential for improvement, e.g. extrapolation, periodic boundary in radial direction
- ⑧ realistic error estimate hard: supercomputer-class „etalon” solution required (one/code version seems enough)