Elliptic gauges for numerical spacetimes

Markus P. Rumpfkeil
AEI, Potsdam Golm, Germany

Supervised by Denis Pollney, AEI

References

- Miguel Alcubierre *et.al.* :

  Gauge conditions for long-term numerical black hole evolutions without excision

  July 2002, gr-qc/0206072

- Masaru Shibata:

  Fully general relativistic simulation of merging binary clusters - Spatial gauge condition –

  May 1999, gr-qc/9905058

Outline

- The BSSN System
- Slicing conditions
- Spatial Gauge conditions
- Implementation issues
- Results
- Problems / Future work

Our „playground“

- The BSSN variables are

$$
\begin{aligned}
A_{ij} &= K_{ij} - \frac{1}{3}\gamma_{ij}K \\
\phi &= \ln\psi = \frac{1}{12}\ln\gamma, \\
K &= \gamma_{ij}K^{ij}, \\
\tilde{\gamma}_{ij} &= e^{-4\phi}\gamma_{ij}, \\
\tilde{A}_{ij} &= e^{-4\phi}A_{ij}
\end{aligned}
$$

$$
\tilde{\Gamma}^i = \tilde{\gamma}^{jk}\tilde{\Gamma}^i{}_{jk} = -\partial_j\tilde{\gamma}^{ij}
$$

- We then have the constraint equations for the vacuum case

$$R = K_{ij}K^{ij} - K^2 = \tilde{A}_{ij}\tilde{A}^{ij} - \frac{2}{3}K^2$$

$$\partial_j \tilde{A}^{ij} = -\tilde{\Gamma}^i{}_{jk}\tilde{A}^{jk} - 6\tilde{A}^{ij}\partial_j\phi + \frac{2}{3}\tilde{\gamma}^{ij}\partial_j K$$

$$\tilde{\gamma} = 1$$

- and the evolution equations

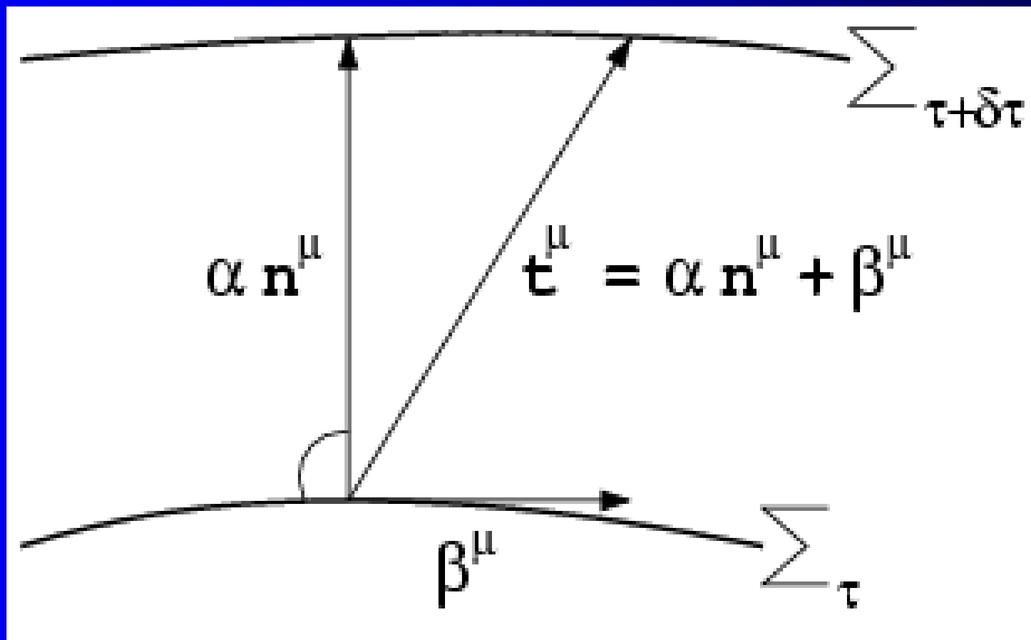
$$
\begin{aligned}
(\partial_t - \mathcal{L}_\beta)\,\phi &= -\frac{1}{6}\alpha K \\
(\partial_t - \mathcal{L}_\beta)\,\tilde{\gamma}_{ij} &= -2\alpha \tilde{A}_{ij} \\
(\partial_t - \mathcal{L}_\beta)\,\tilde{A}_{ij} &= e^{-4\phi}[-D_i D_j \alpha + \alpha R_{ij}]^{TF} \\
&\quad + \alpha(K\tilde{A}_{ij} - 2\tilde{A}_{ik}\tilde{A}^k{}_j)
\end{aligned}
$$

$$
\begin{aligned}
\partial_t K &= -\gamma^{ij} D_j D_i \alpha + \alpha \tilde{A}_{ij}\tilde{A}^{ij} \\
&\quad + \frac{1}{3}\alpha K^2 + \beta^i \partial_i K
\end{aligned}
$$

- In particular for $\tilde{\Gamma}^i$ we have

$$\partial_t \tilde{\Gamma}^i = -\partial_j \partial_t \tilde{\gamma}^{ij} = -2\left(\alpha \partial_j \tilde{A}^{ij} + \tilde{A}^{ij} \partial_j \alpha\right) - \partial_j \left(\mathcal{L}_\beta \tilde{\gamma}^{ij}\right)$$

- One important issue is now to determine the lapse α and shift β^i on each time slice:



Slicing conditions

- „K-freezing" condition $\partial_t K = 0$ leads to

$$D^i D_i \alpha = \alpha K_{ij} K^{ij} + \beta^i \partial_i K$$

- This reduces for *K=0* to the well known maximal slicing condition

$$D^i D_i \alpha = \alpha K_{ij} K^{ij}$$

- Imposing the condition by simply not evolving the trace of the extrinsic curvature in the BSSN formalism

- Max. slicing equation actually looks like

$$
\begin{aligned}
& \partial_x(\sqrt{\gamma}(\gamma^{xx}\partial_x\alpha + \gamma^{xy}\partial_y\alpha + \gamma^{xz}\partial_z\alpha)) \\
+ \quad & \partial_y(\sqrt{\gamma}(\gamma^{xy}\partial_x\alpha + \gamma^{yy}\partial_y\alpha + \gamma^{yz}\partial_z\alpha)) \\
+ \quad & \partial_z(\sqrt{\gamma}(\gamma^{xz}\partial_x\alpha + \gamma^{yz}\partial_y\alpha + \gamma^{zz}\partial_z\alpha)) \\
= \quad & \sqrt{\gamma}K^{ab}K_{ab}\alpha
\end{aligned}
$$

- I use second order acurate, central finite differences with a nineteen-point stencil

- The finite differenced form of one of the terms is e.g. ($f \equiv \sqrt{\gamma}\gamma^{xy}$):

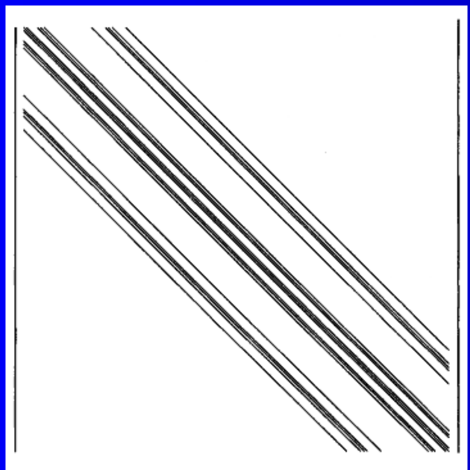
$$
\begin{aligned}
\partial_x(f\partial_y\alpha) \quad = \quad & \frac{1}{4\Delta x\Delta y}[f_{i+1,j,k}(\alpha_{i+1,j+1,k} - \alpha_{i+1,j-1,k}) \\
- \quad & f_{i-1,j,k}(\alpha_{i-1,j+1,k} - \alpha_{i-1,j-1,k})]
\end{aligned}
$$

- This leads to a set of N linear equations in N unknowns (N total number of grid zones):

$$J\alpha = b$$

J -Jacobi Matrix b -contains source terms

- J is a N x N very sparse square Matrix with a diagonal structure:



Each diagonal corresponds to a finite difference coefficient in the nineteen-point stencil

- J would be even symmetric without the application of boundary conditions

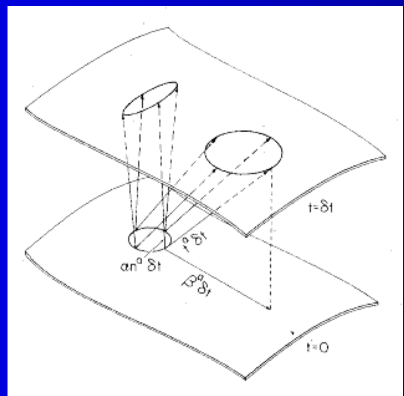
- It is computationally way cheaper to use a hyperbolic slicing e.g. the Bona-Masso family of slicing („K-driver“) conditions:

$$\partial_t \alpha = -\alpha^2 f(\alpha) K \quad \text{with } v_\alpha = \alpha \sqrt{f(\alpha)}$$

- Maximal slicing corresponds (at least formally) to $f \to \infty$
- Popular choice is $f(\alpha) = \frac{2}{\alpha}$ referred to as 1+log slicing since integration yields $\alpha = 1 + \ln \gamma$
- The gauge speed is $v_\alpha = \sqrt{2/\alpha} \cdot v_{null} > v_{null}$

Spatial gauge conditions

- Smarr and York (1978) defined a (traceless) "distortion" tensor as $\Sigma_{ij} = \frac{1}{2}\gamma^{1/3}\partial_t\tilde{\gamma}_{ij}$



- Minimizing the „shear stretching energy“

$$S[\beta] = \frac{1}{2}\int d^3x \; \Sigma_{ij}\Sigma^{ij}\sqrt{\gamma}$$

with respect to β^i yields the (covariant) „**Minimal distortion condition**“ (MD):

$$D_j \Sigma^{ij} \stackrel{!}{=} 0 \quad \Leftrightarrow D_j(e^{-4\phi}\partial_t \tilde{\gamma}^{ij}) = 0$$

$$\Leftrightarrow \tilde{D}_j(e^{6\phi}\partial_t \tilde{\gamma}^{ij}) = 0$$

- Shibata proposed to minimize the „shear stretching energy“ in the conformal three space instead, which yields the „**Shibata condition**“ : $\tilde{D}_j(\partial_t \tilde{\gamma}^{ij}) = 0$
- A merit of this condition is that we don‘t have a coupling term between β^i and ϕ
- In a second step he rewrote the covariant derivative operator as a partial derivative, leading to the „**AMD condition**“:

$$\tilde{D}_i(\partial_t \tilde{\gamma}^{ij}) \approx \tilde{\gamma}^{jk}\partial_j\partial_k\beta^i + \frac{1}{3}\tilde{\gamma}^{ij}\partial_j\partial_k\beta^k$$
$$-2\tilde{A}^{ij}\partial_j\alpha - \frac{4}{3}\alpha\tilde{\gamma}^{ij}\partial_j K$$
$$+12\alpha\tilde{A}^{ij}\partial_j\phi$$
$$\stackrel{!}{=} 0$$

- Miguel Alcubierre *et.al.* obtained the „**Gamma-freezing condition**“ by demanding that the conformal connection functions don‘t evolve:

$$\partial_t \tilde{\Gamma}^i \stackrel{!}{=} 0$$

- Imposing the condition by simply not evolving the $\tilde{\Gamma}^i$
- This condition is related to MD via:

$$\partial_t \tilde{\Gamma}^i = 2e^{4\phi}\left[D_j\Sigma^{ij} - \tilde{\Gamma}^i_{jk}\Sigma^{jk} - 6\Sigma^{ij}\partial_j\phi\right]$$

- Furthermore they introduced a „**Gamma-driver condition**“ (Gamma2) :

$$\partial_t^2 \beta^i = F\,\partial_t \tilde{\Gamma}^i - \eta\,\partial_t \beta^i$$

F and η are positive functions of space and time, usually $F \propto \alpha$

$$\partial_t \beta^i = B^i \qquad \partial_t B^i = F \partial_t \tilde{\Gamma}^i - \eta B^i$$

- One can calculate the gauge speed in regions where the spacetime is almost flat:

$$v_{\mathrm{long}} = 2\sqrt{F/3} \qquad v_{\mathrm{trans}} = \sqrt{F}$$

- To have the longitudinal part of the shift propagate with the speed of light they choose

$$F(\alpha) \equiv \frac{3}{4}\alpha$$

Implementation issues

- One subtlety is the use of $\tilde{\Gamma}^i$: It seems to be numerically the most stable to compute finite differences from the indepent evolved variable for partial derivatives $\partial_j \tilde{\Gamma}^i$, but to substitute its definition in all expressions that just require $\tilde{\Gamma}^i$
- A crucial point for the stability is the use of the constraints in the derivation of the evolution equations
- Boundary conditions are playing an important rule, I used octant mode implemented through reflection symmetry

- and Robin BC at the outer boundary, this is

$$f(r) = f_0 + \frac{k}{r^n} \quad \Rightarrow \quad \frac{\partial f}{\partial r} = -n\frac{(f - f_0)}{r}$$

k a constant
n the decay rate
f_0 the value at infinity

Considering now a given cartesian direction *x* we get:

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial r}\frac{\partial r}{\partial x} = \frac{x}{r}\frac{\partial f}{\partial r} = -n(f - f_0)\frac{x}{r^2}$$

Finite differencing around the grid point *i + 1/2* yields

$$f_{i+1} - f_i = -n\Delta x((f_{i+1} + f_i) - 2f_0)\frac{x_{i+1} + x_i}{(r_{i+1} + r_i)^2}$$

- To save computational time I solve the elliptic equations only on every *5th* time slice. Otherwise I keep the values from the previous one. This doesn‘t seem to have too much influence on the timepoint of the breakdown

- For the initial guess of the solution I use the result of the previous time slice

- I use the Generalized Minimum Residual (GMRES) method implemented in PETSc („The portable, extensible Toolkit for Scientific Computation“), which is designed to solve nonsymmetric linear systems

- The Jacobi Matrix for the elliptic equation is ill conditioned (Condition numbers of 10^{40} are not uncommon if no excision is used)
- A rescaling of the Jacobi Matrix saves about *20% – 40%* of the computational time in my problems
- I rescaled in the following way:

$$Ax = b \Leftrightarrow DAx = Db \quad \text{with} \quad D = [d_{ij}] \quad d_{ij} = \delta_{ij}(a_{ij})^{-1}$$

- Another possibility is

$$D = [\hat{d}_{ij}] \quad \hat{d}_{ij} = \delta_{ij} \; (\sum_{l=1}^{n} \mid a_{il} \mid)^{-1}$$

Results

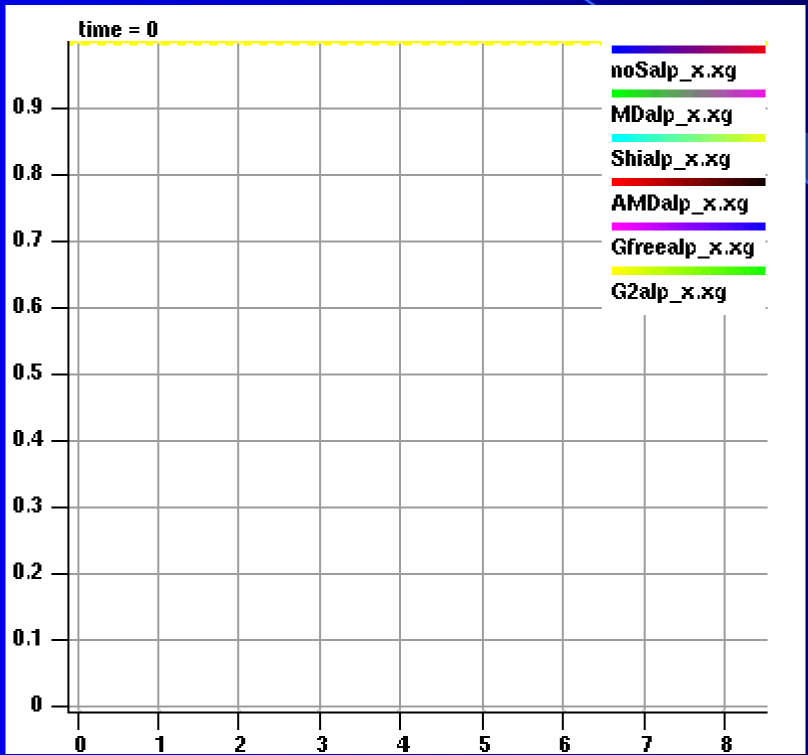
- My test problem is a single black hole sitting in the origin with unit mass, but no spin, momentum or charge and modeled with puncture initial data
- I evolve the sytem using the Thorn BSSN_MoL of the Cactus Code Environment with radiative BC and third order in time ICN
- I use my own Thorns to set up the elliptic equations for the lapse and the shift

- Following table shows main result: The breakdown times of the Code (in M) for several gauge conditions

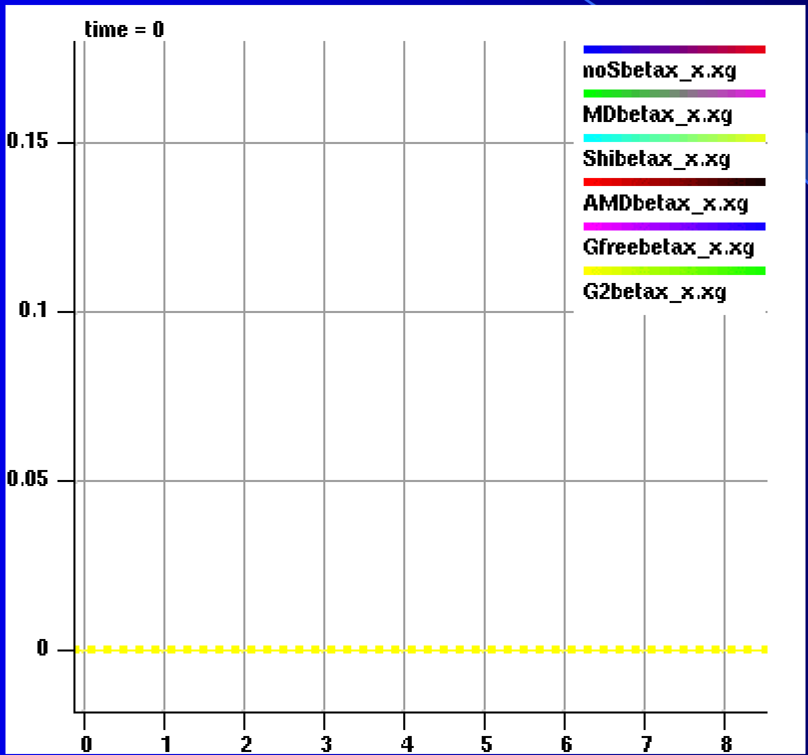
	no shift	MD	Shibata	AMD	$\partial_t \Gamma^i = 0$	Γ_2
1+log	48	3	4	6	9	> 2000
maximal	69	70	40	150	20	34

- I am going to show now some 1D movies to compare the results with each other

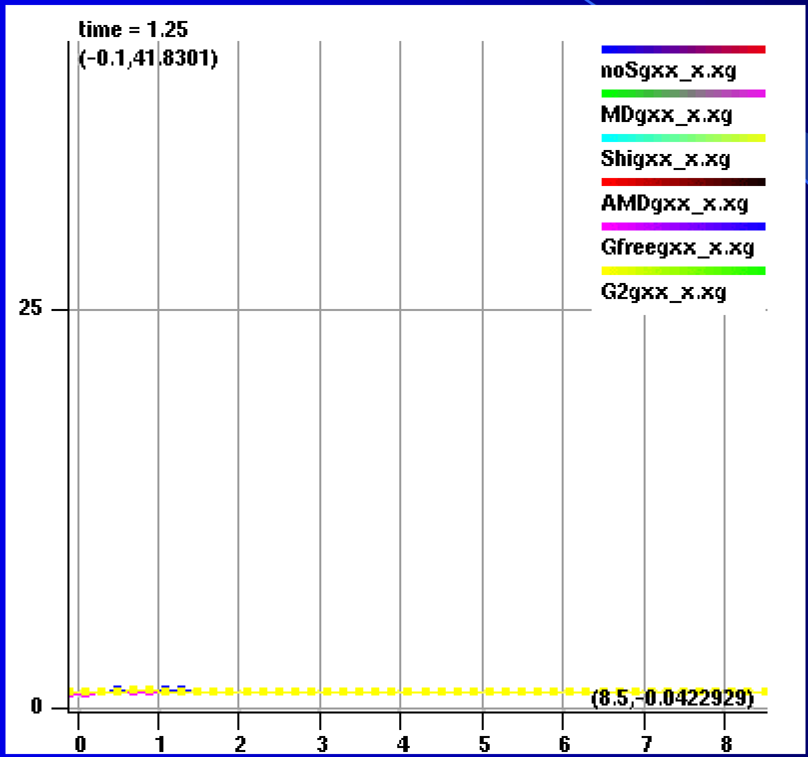
Max. slicings for several shift conditions

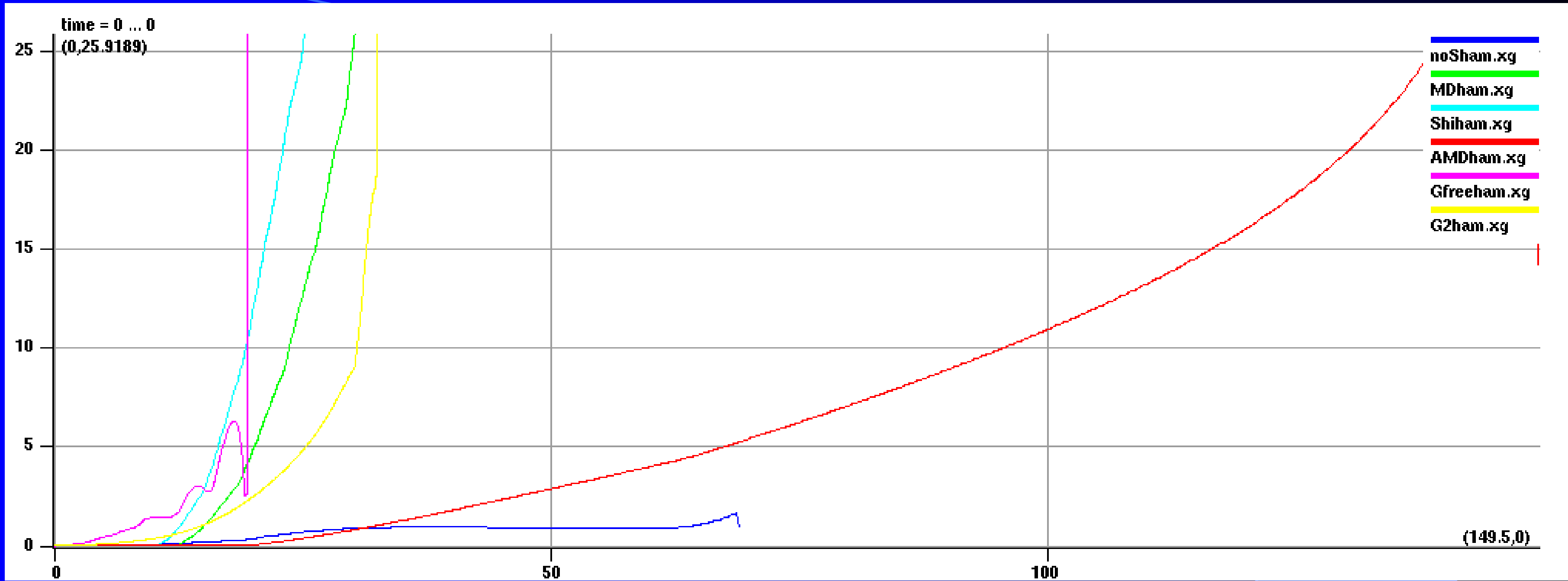


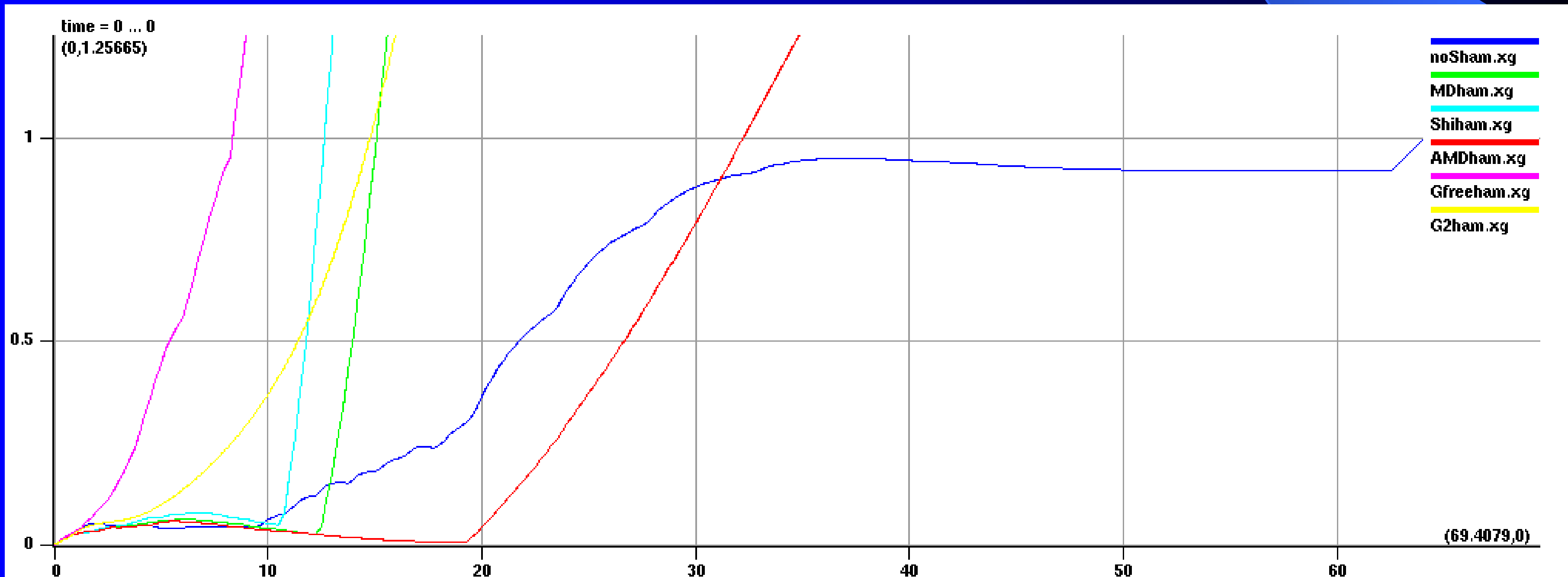
Several shifts for maximal slicing



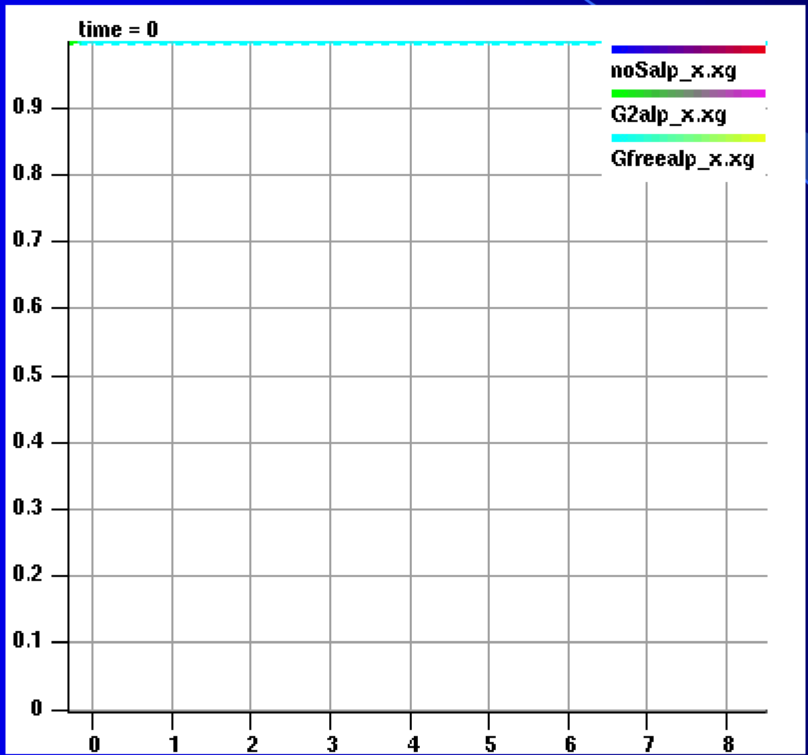
Grid stretching



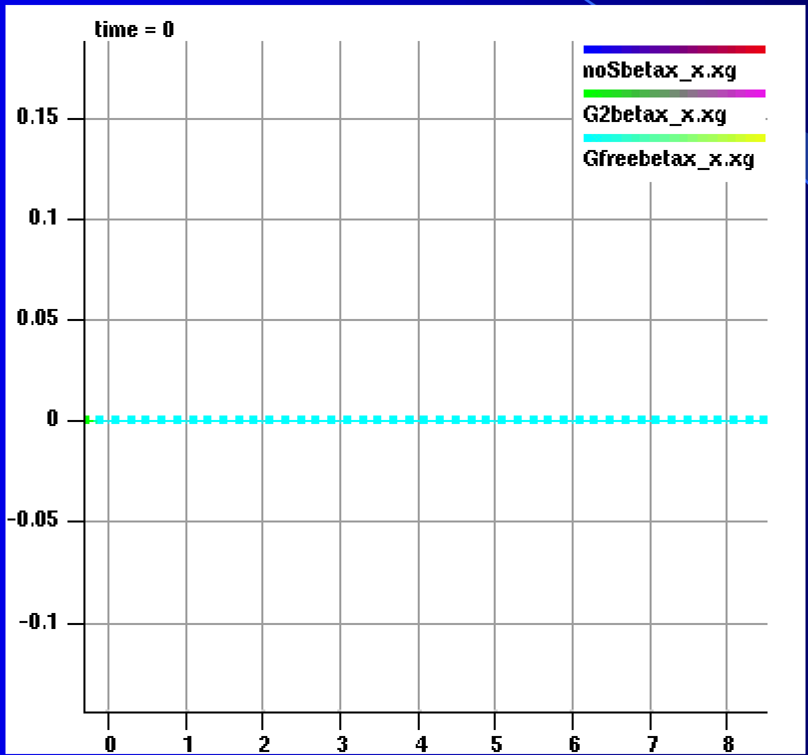

time = 0 ... 0
(0,25.9189)
25
20
15
10
5
0
50
100
(149.5,0)
noSham.xg
MDham.xg
Shiham.xg
AMDham.xg
Gfreeham.xg
G2ham.xg


time = 0 ... 0
(0,1.25665)
1
0.5
0
10
20
30
40
50
60
(69.4079,0)
noSham.xg
MDham.xg
Shiham.xg
AMDham.xg
Gfreeham.xg
G2ham.xg

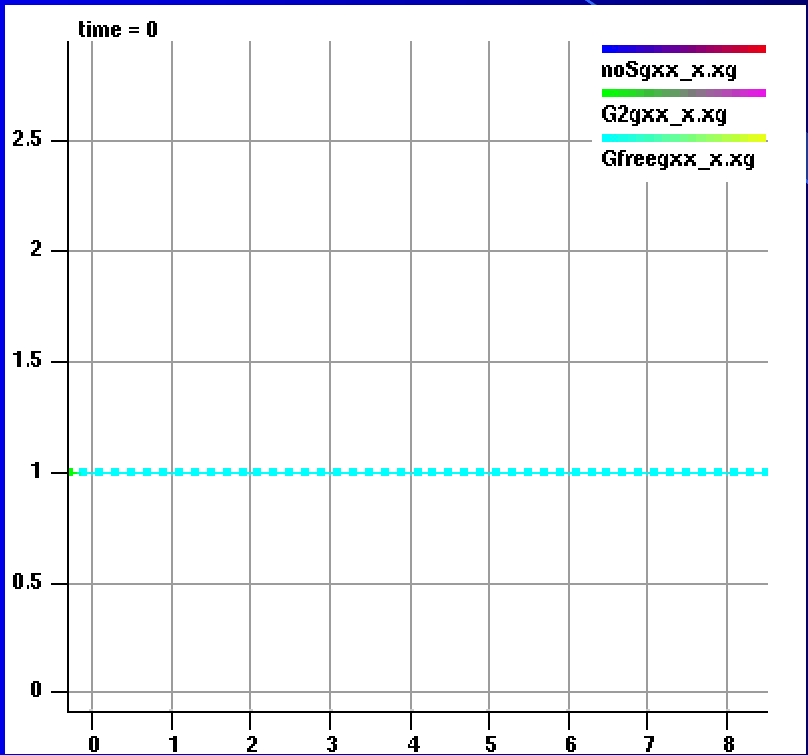
1+log slicings for several shift conditions

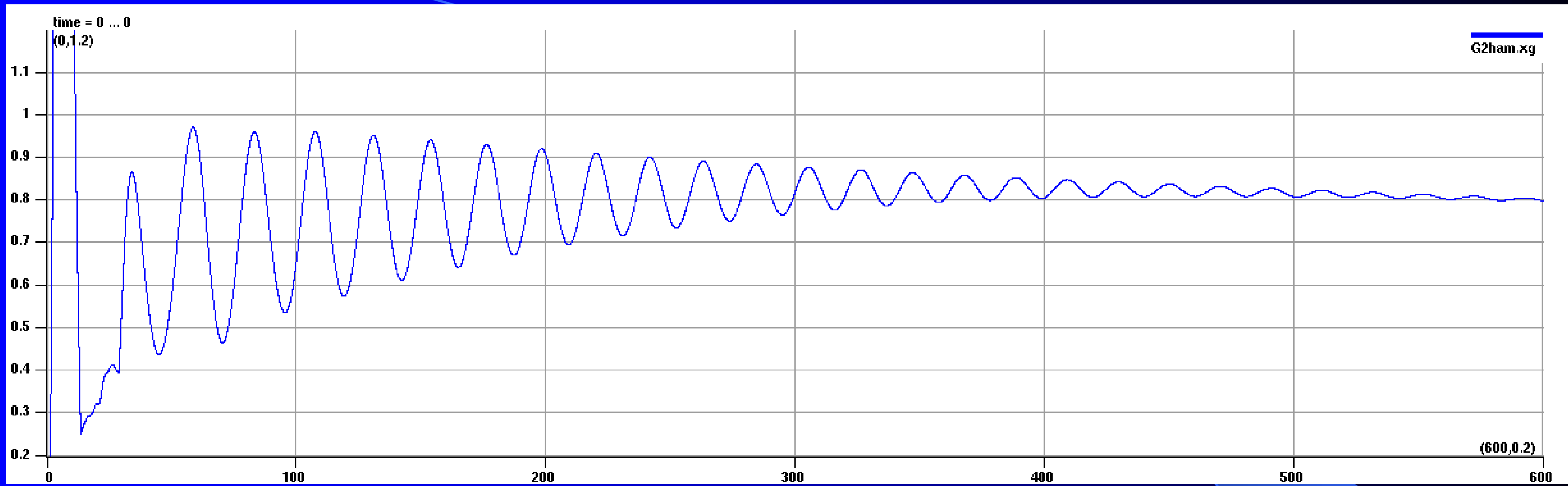


Several shifts for 1+log slicing

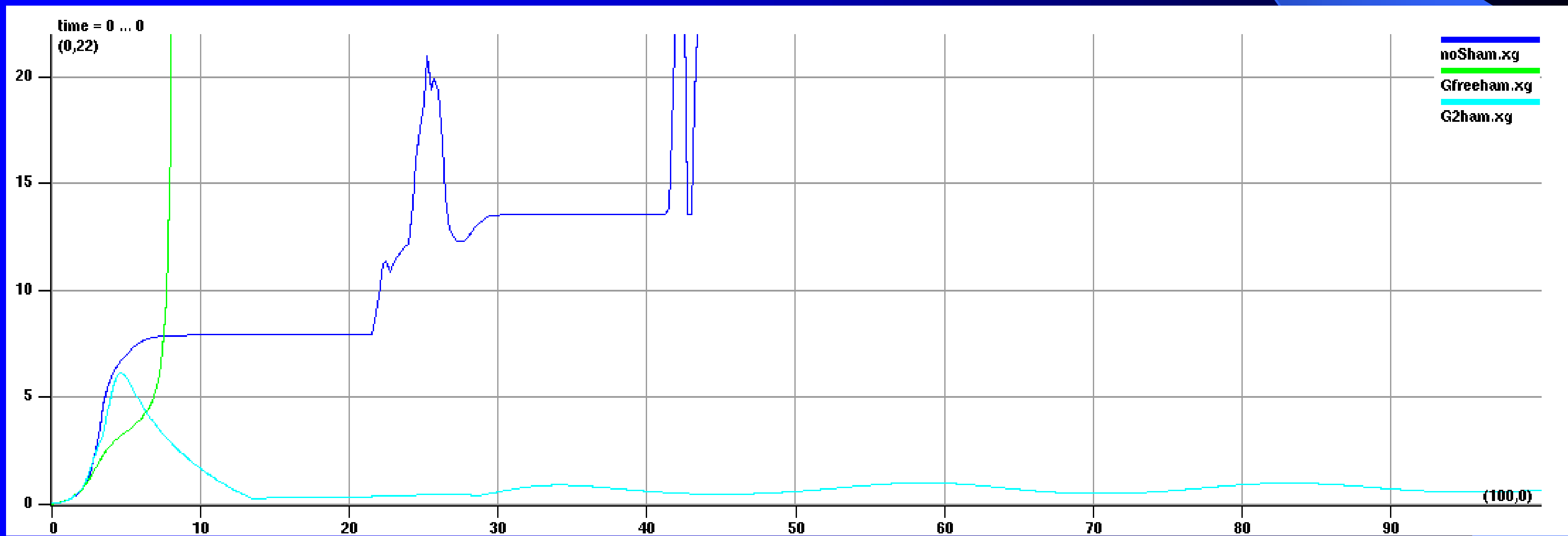


Grid stretching





time = 0 ... 0
(0,1.2)
G2ham.xg
(600,0.2)
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
100
200
300
400
500
600



time = 0 ... 0
(0,22)
noSham.xg
Gfreeham.xg
G2ham.xg
(100,0)
20
15
10
5
0
0
10
20
30
40
50
60
70
80
90

Problems / Future work

- Solving the elliptic equations is computational very expensive, therefore it would be nice to have a multigrid solver $\Rightarrow$ need of a wrapper routine to use the multigrid solver in PETSc with Cactus
- Excision techniques would be desirable, but a problem with the inner Boundary Conditions arises immediately

Thanks for listening