Gmunu: Toward multigrid based Einstein field equations solver for general-relativistic hydrodynamics simulations

2019

CHEONG, Chi-Kit (Patrick)
Tjonnie G. F. LI, L. M. LIN

Department of Physics
The Chinese University of Hong Kong

Oct 14 2019

香港中文大學
The Chinese University of Hong Kong

Outline

Introduction

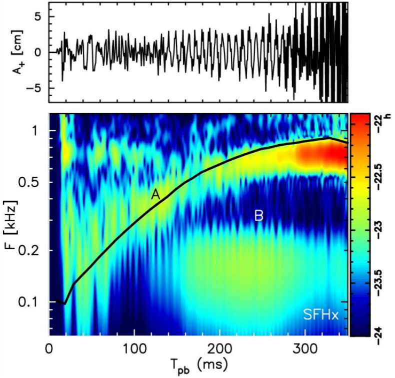
- Implementation details of `Gmunu` (P. C. K. CHEONG et al., in prep.)
- Why numerical simulations
- Introduction of CCSNe and its gravitational waves
- Develop a code for:
  - Core-collapse supernovae
  - Isolated Neutron stars

Supernovae

- Type II, Ib, Ic
- Iron core of a Massive star ($\sim 10 \sim 100 M_{\odot}$)
- $O(10^{53})$ ergs released in gravitational collapse
- $\sim 99\%$ of the energy is radiated in neutrinos
- Neutron star or black hole remanant

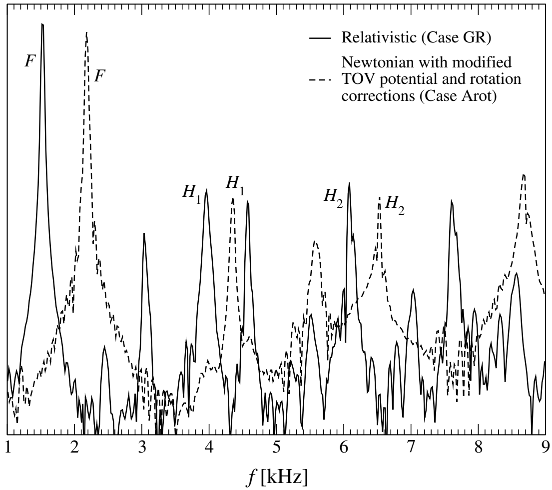
Why we need simulations

We need numerical simulations!



T. Kuroda+ 2016

Introduction



B. Muller, et al. 2008

General relativistic simulations is needed!

Introduction

Different ways to solve Einstein equations

- "free-evolution" formulations
- constrained formulation
  - elliptic sector and hyperbolic sector

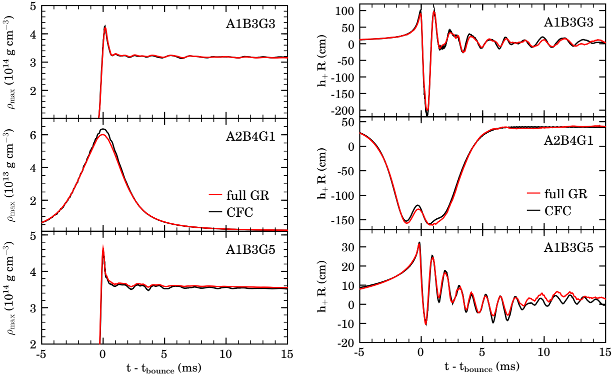
Conformally fatness approximation (CFC) is applied successfully in various astrophysical problems even in spherical-polar coordinate
However...

- high computational cost for elliptic equations
- on the fly simulation?

Conformally flatness approximation

CFC approximation

$$g_{\mu\nu} = \begin{bmatrix} -\alpha^2 + \beta_i\beta^i & \beta_1 & \beta_2 & \beta_3 \\ \beta_1 & \varphi^4 & 0 & 0 \\ \beta_2 & 0 & \varphi^4 r^2 & 0 \\ \beta_3 & 0 & 0 & \varphi^4 r^2 \sin^2\theta \end{bmatrix}$$



Ott, et al. 2006

Hydro

The 3+1 "Valencia" formulation

$$\partial_t(\sqrt{\gamma}\vec{U}) + \partial_i(\sqrt{\gamma}\vec{F^i}) = \vec{S}$$

where

$$\boldsymbol{U} = \begin{bmatrix} D \\ S_j \\ \tau \end{bmatrix} = \begin{bmatrix} \rho W \\ \rho h W^2 v_j \\ \rho h W^2 - p - D \end{bmatrix},$$

$$\boldsymbol{F^i} = \begin{bmatrix} D\left(v^i - \beta^i/\alpha\right) \\ S_j\left(v^i - \beta^i/\alpha\right) + \delta^i_j D \\ \tau\left(v^i - \beta^i/\alpha\right) + P v^i \end{bmatrix},$$

$$\boldsymbol{Q} = \begin{bmatrix} 0 \\ T^{\mu\nu}\left(\frac{\partial g_{\nu j}}{\partial x^\mu} - \Gamma^\lambda_{\mu\nu} g_{\lambda j}\right) \\ \alpha\left(T^{\mu 0}\frac{\partial \ln \alpha}{\partial x^\mu} - T^{\mu\nu}\Gamma^0_{\mu\nu}\right) \end{bmatrix}$$

Plus EOS

xCFC scheme

Cordero-Carrion et al. (2009)

$$\tilde{\Delta} X^i + \frac{1}{3}\tilde{\nabla}^i \left(\tilde{\nabla}_j X^j\right) = 8\pi \tilde{S}^i$$

$$\tilde{A}^{ij} \approx \tilde{\nabla}^i X^j + \tilde{\nabla}^j X^i - \frac{2}{3}\tilde{\nabla}_k X^k f^{ij}$$

$$\tilde{\Delta}\psi = -2\pi \tilde{E}\psi^{-1} - \frac{1}{8} f_{ik} f_{jl} \tilde{A}^{kl} \tilde{A}^{ij} \psi^{-7}$$

$$\tilde{\Delta}(\alpha\psi) = (\alpha\psi)\left[2\pi\left(\tilde{E} + 2\tilde{S}\right)\psi^{-2} + \frac{7}{8} f_{ik} f_{jl} \tilde{A}^{kl} \tilde{A}^{ij} \psi^{-8}\right]$$

$$\tilde{\Delta}\beta^i + \frac{1}{3}\tilde{\nabla}^i \left(\tilde{\nabla}_j \beta^j\right) = 16\pi\alpha\psi^{-6} f^{ij} \tilde{S}_i + 2\tilde{A}^{ij}\tilde{\nabla}_j \left(\alpha\psi^{-6}\right)$$

Boundary condiditon

Robin B. C.

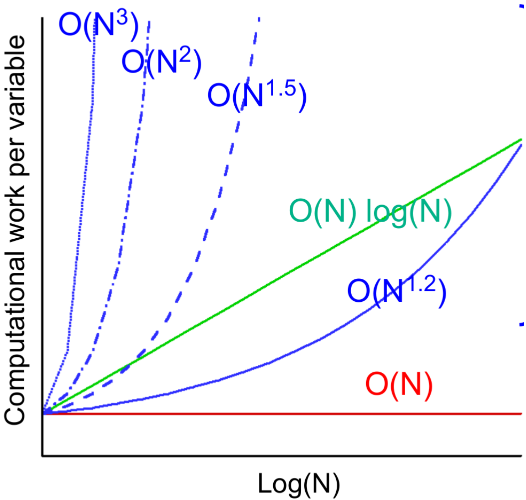
$$\left.\frac{\partial \psi}{\partial r}\right|_{r_{\text{max}}} = \frac{1-\psi}{r}$$
$$\left.\frac{\partial \alpha}{\partial r}\right|_{r_{\text{max}}} = \frac{1-\alpha}{r}$$
$$\left.\beta^i\right|_{r_{\text{max}}} = 0$$

Elliptic solvers

Table: Elliptic solvers

Method	Complexity (accuracy ϵ)
Gauss Elimination	$O(N^2)$
Jacobi iteration	$O(N^2)\log(\epsilon)$
Gauss-Seidel iteration	$O(N^2)\log(\epsilon)$
SOR	$O(N^{3/2})\log(\epsilon)$
Conjugate Gradient	$O(N^{3/2})\log(\epsilon)$
ADI	$O(N\log(N))\log(\epsilon)$
FFT	$O(N\log(N))$
Multigrid (iterative)	$O(N)\log(\epsilon)$
Multigrid (FMG)	$O(N)$

Elliptic solvers



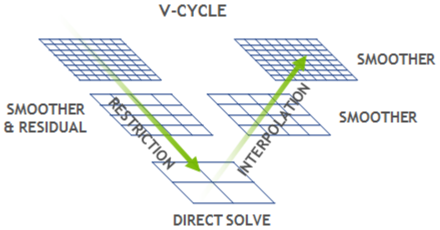
credit: Klaus Stuben

Elliptic solvers

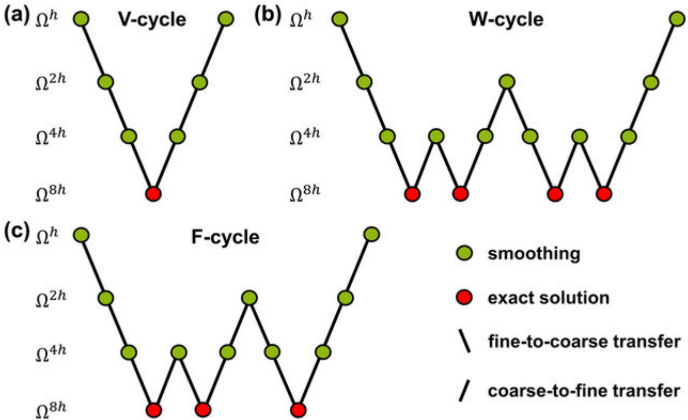
- LORENE using spectral methods for CFC
- require accurate grid communication

Multigrid Method

- A strategy, not a single method
- various possible implementation
- fast and efficient
- suitable for non-linear problems
- no grid communication is needed

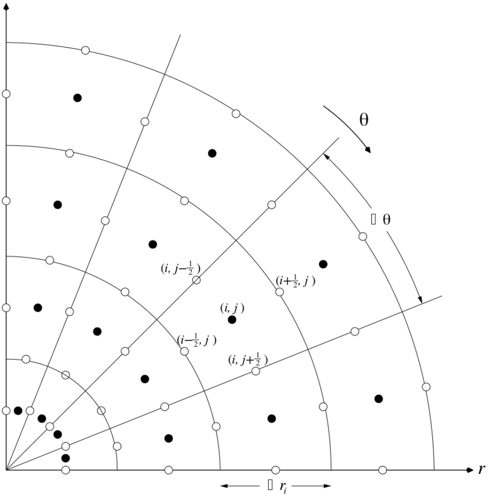


Multigrid Method



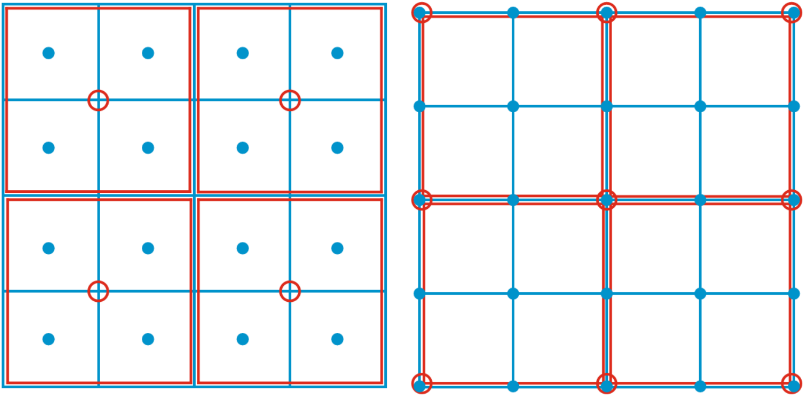
credit: Jaysaval et al. (2016)

Multigrid Method



cell-centered grid
Dimmelmeier et al. (2002)

Multigrid Method

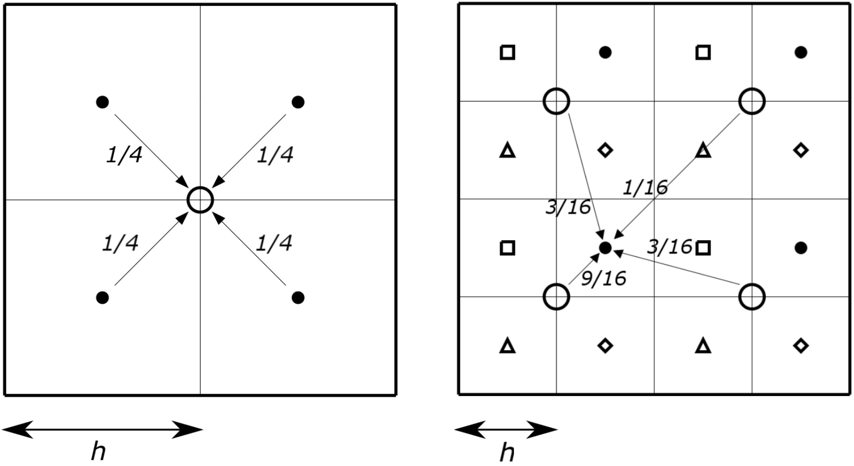


(a) Cell-Centered Grid (b) Vertex-Centered Grid

credit: Hooshyar (2014)

Multigrid Method

- Restriction (Left): piecewise constant
- Prolongation (Right): bi-linear



credit: F. W. Yang et al. (2015)

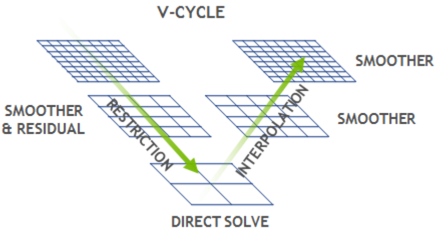
Code features

Gmunu (General-relativistic multigrid numerical Einstein solver)

- 3+1 in Spherical polar coordinates (1,2,3-D)
- Hydro: high-resolution shock-capturing (HRSC) methods
  - Reconstruction method: Piecewise-Constant (PC), TVD, (5-th order)WENO, MP5
  - Riemann solver: HLL, HLLE, Marquina
  - Time update: RK3 methods
  - Conserved variables to Primitive variables: Regula-Falsi method
- Conformally flatness condition(CFC) metric evolution:
  - extended CFC (xCFC) scheme (Cordero-Carrion et al. (2009))
  - Multigrid solver for the elliptic non-linear coupled equations

Key features of our MG solver

- Nonlinear FAS (Full Approximation Storage) algorithm
- Cycles options: V, W and F
- Cell-Centered discretization (no grid communication is needed)
- Smoother: Red-Black Gauss-Seidel relaxation
- Prolongation: bi-linear
- Restriction: piecewise constant



Algorithom

$$L(u) = f$$

V-cycle

if $l = 1$ **then**

  $u_l \leftarrow \texttt{Solve}(u_l)$

else

  $u_l \leftarrow \texttt{Smoothing}(u_l, f_l)$! Pre-smoothing

  $r_l \leftarrow f_l - L_l(u_l)$! residual

  $\texttt{Restriction}(r_l, u_l)$

  $f_{l-1} \leftarrow r_{l-1} + L_{l-1}(u_{l-1})$

  $v \leftarrow u_{l-1}$

  ! Recursive call for the coarse grid correction

  $v \leftarrow \texttt{MG}(v, f_{l-1})$

  $u_l \leftarrow u_l + \texttt{Prolongation}(v - u_{l-1})$

  $u_l \leftarrow \texttt{Smoothing}(u_l, f_l)$! Post-smoothing

end

Implementation note

Define

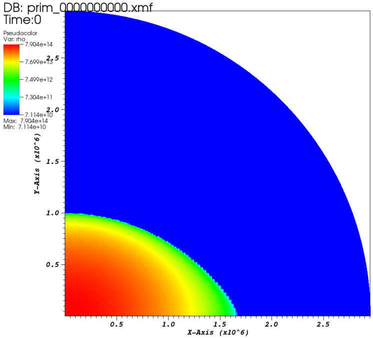
$$\tilde{\boldsymbol{U}} \equiv \psi^6 \boldsymbol{U}$$
$$\tilde{\boldsymbol{F}}^{\boldsymbol{i}} \equiv \psi^6 \boldsymbol{F}^{\boldsymbol{i}}$$
$$\tilde{\boldsymbol{Q}} \equiv \psi^6 \boldsymbol{Q}$$

$$\frac{\partial \tilde{\boldsymbol{U}}}{\partial t} + \frac{1}{r^2}\frac{\partial}{\partial r}\left(\alpha r^2 \tilde{\boldsymbol{F}}^{\boldsymbol{r}}\right) + \frac{1}{\sin\theta}\frac{\partial}{\partial \theta}\left(\alpha \sin\theta \tilde{\boldsymbol{F}}^{\boldsymbol{\theta}}\right) = \alpha \tilde{\boldsymbol{Q}}.$$

Models

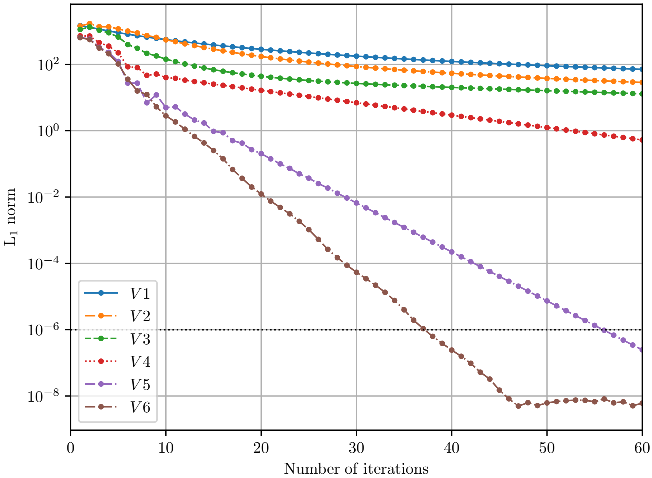
Table: $\Gamma = 2$, $K = 100$,$(c = G = M_{sun} = 1)$.

Model	ρ_c $[10^{-3}]$	Ω $[10^{-2}]$
BU0	1.28	0.000
SU	8.00	0.000
BU2	1.28	1.509
BU8	1.28	2.633



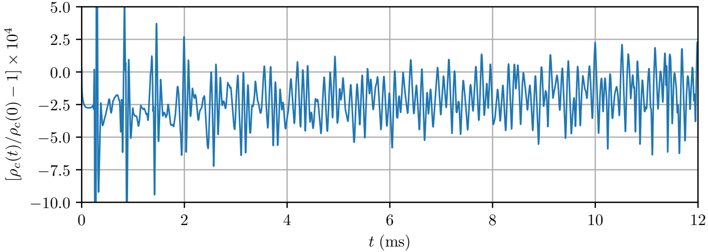
Convergence properties

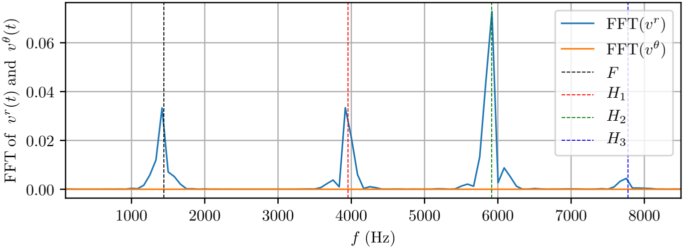
- solving the α of BU8 with initial guess: $\alpha = 1$
- $n_r \times n_\theta = 640 \times 64$
- Gauss-Seidel ($\sim 5.3 \times 10^5$) to MG (~ 50)
- O(minus) $\to$ O(ms)



Model: BU0

- $n_r \times n_\theta = 640 \times 64$
- $r = [0, 30]$, $\theta = \left[0, \frac{\pi}{2}\right]$

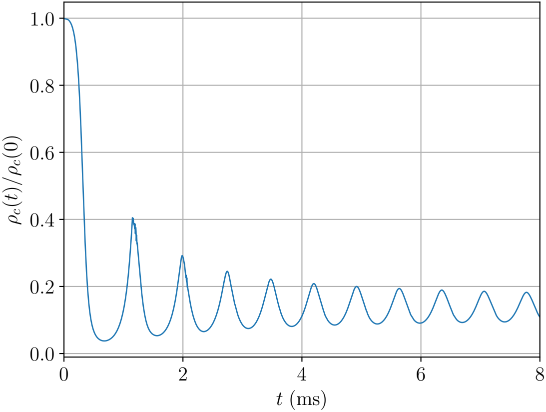




Model: SU

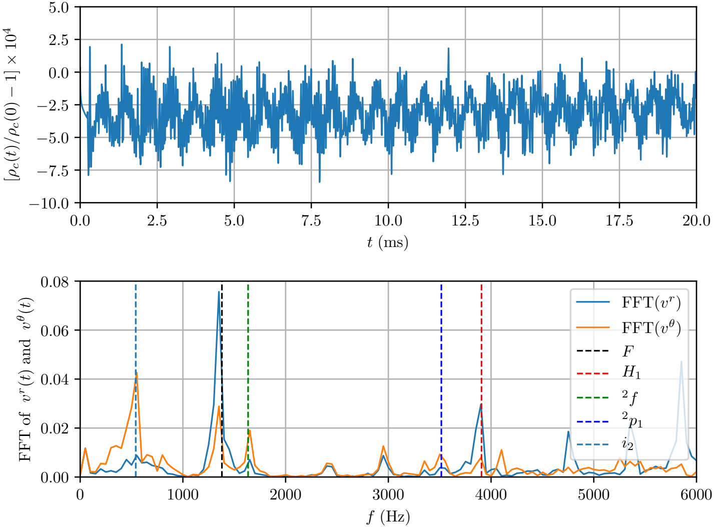
Migration test: unstable -> stable NS

- $n_r \times n_\theta = 1024 \times 1$ (1D simulation)
- $r = [0, 34]$



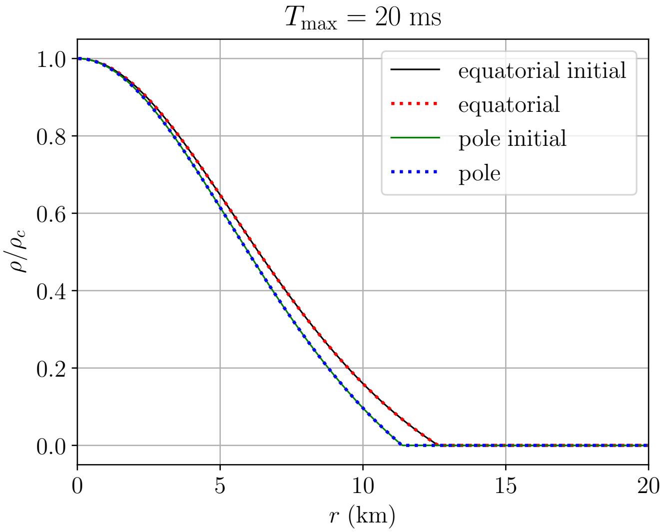
Model: BU2

- $n_r \times n_\theta = 640 \times 64$
- $r = [0, 30]$, $\theta = [0, \frac{\pi}{2}]$



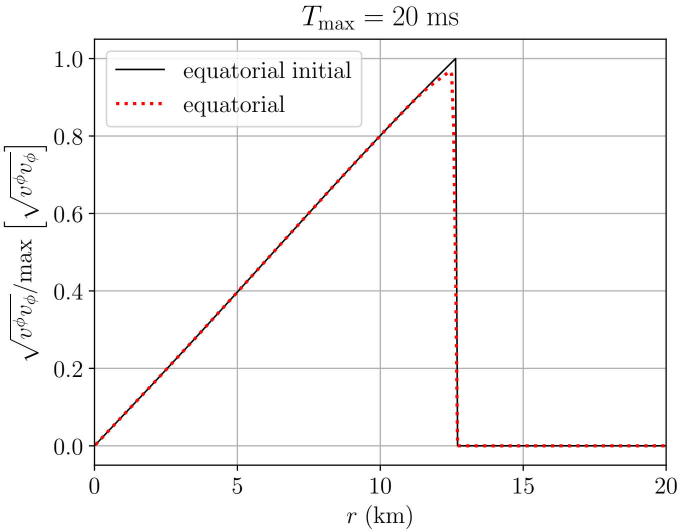
Model: BU2

- $n_r \times n_\theta = 640 \times 64$
- $r = [0, 30]$, $\theta = [0, \frac{\pi}{2}]$



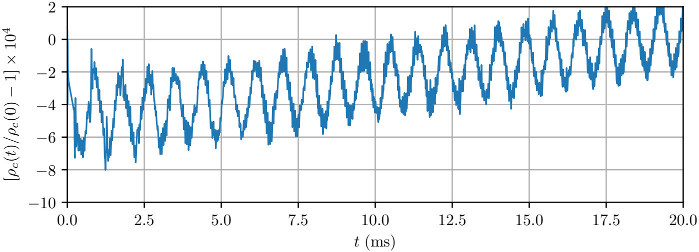
Model: BU2

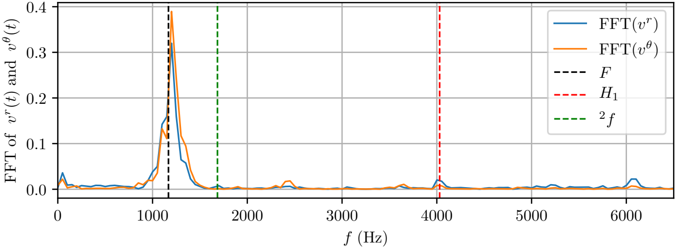
- $n_r \times n_\theta = 640 \times 64$
- $r = [0, 30]$, $\theta = [0, \frac{\pi}{2}]$



Model: BU8

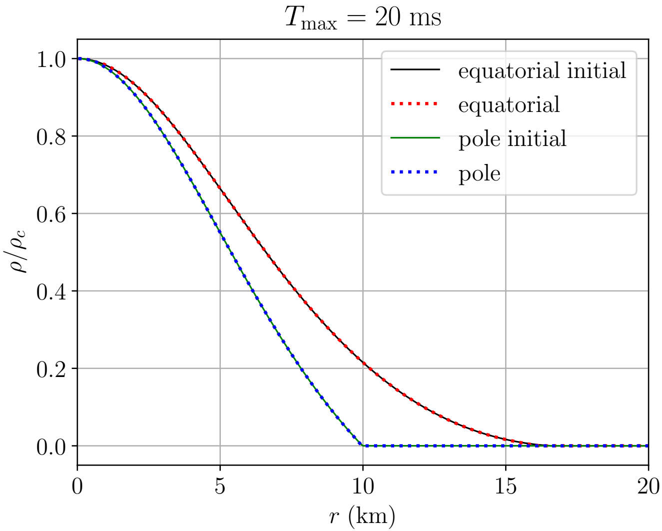
- $n_r \times n_\theta = 640 \times 64$
- $r = [0, 30]$, $\theta = [0, \frac{\pi}{2}]$





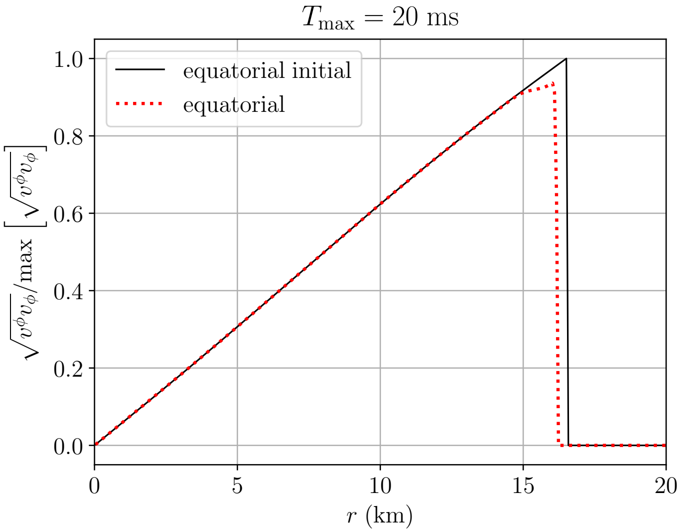
Model: BU8

- $n_r \times n_\theta = 640 \times 64$
- $r = [0, 30]$, $\theta = [0, \frac{\pi}{2}]$



Model: BU8

- $n_r \times n_\theta = 640 \times 64$
- $r = [0, 30]$, $\theta = [0, \frac{\pi}{2}]$



Conclusions

- We presented our Multigrid metirc solver
- Fast and accurate!
  - for $\epsilon \sim 10^{-7}$
  - O(minus) $\rightarrow$ O(ms) per equation
  - solving the metric $\sim$ 5 to 10 hydro steps (stable NS)
  - no need to solve the metric at every time steps
- suitable for dynamical simulations

Future direction

- extend to FCF (fully-constrained formalism)
- upgrate and parallelization
- apply to study astrophysical problems

Q&A

Thank you for your attention !!

Q & A

Backup Slides