Tolerance proportionality for parallelized adaptive Runge-Kutta methods

Kupás Vendel Péter

Department of Applied Analysis and Computational Mathematics

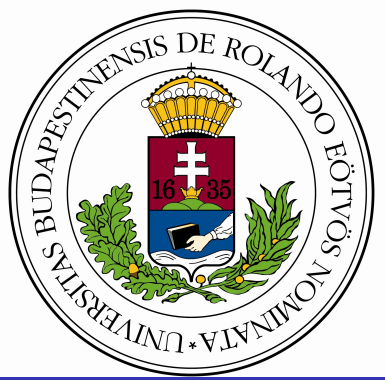
Eötvös Loránd University

Supervisors: Fekete Imre

Izsák Ferenc

Farkas Miklós seminar

2024. december 5.



Contents

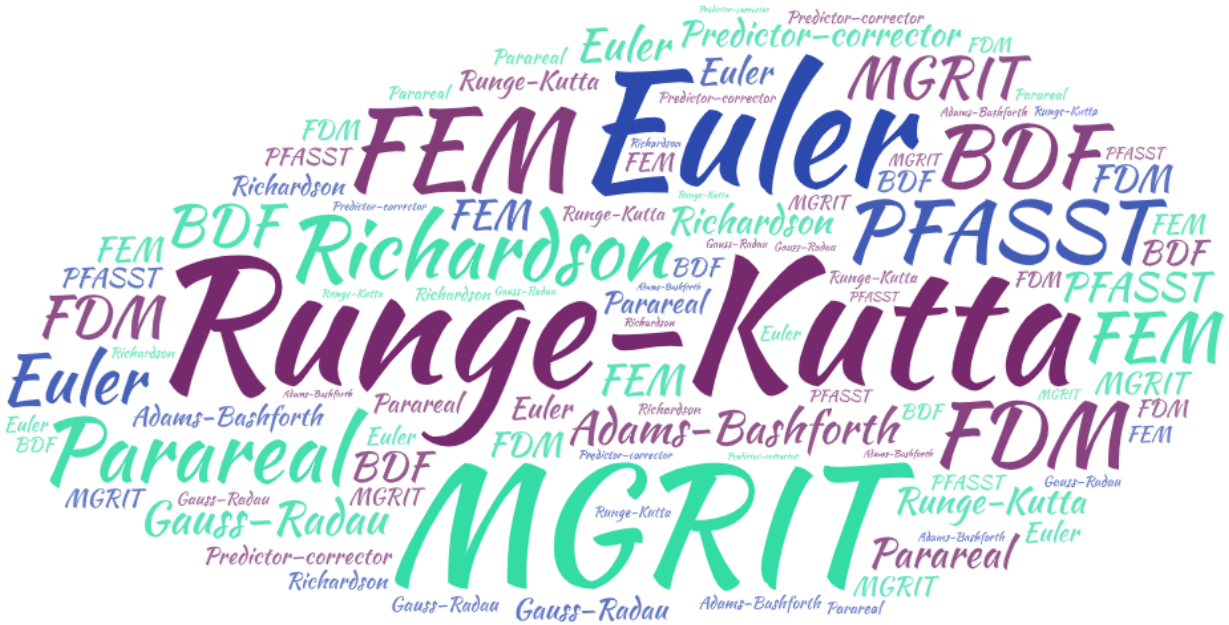
Motivation

Numerical methods

$$u'(t) = f(t, u(t)), \quad t \in [0, T],$$
$$u(0) = u_0$$

- We need to use numerical methods for solving these equations

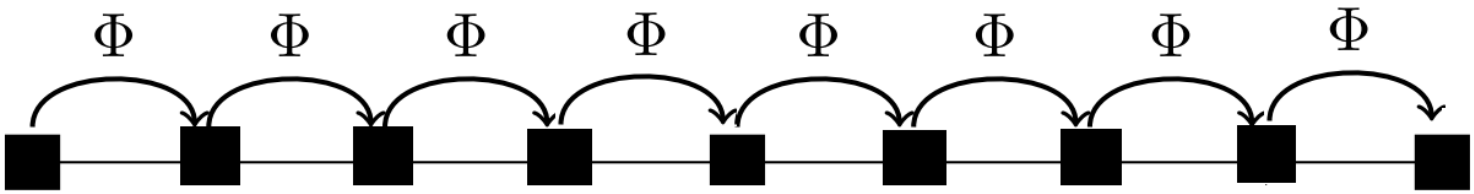
- Order
- Stability properties
- Running time



One step methods

- Let $0 = t_0 < t_1 < \ldots < t_{N-1} < t_N = T$ be a temporal mesh on the interval $[0, T]$.

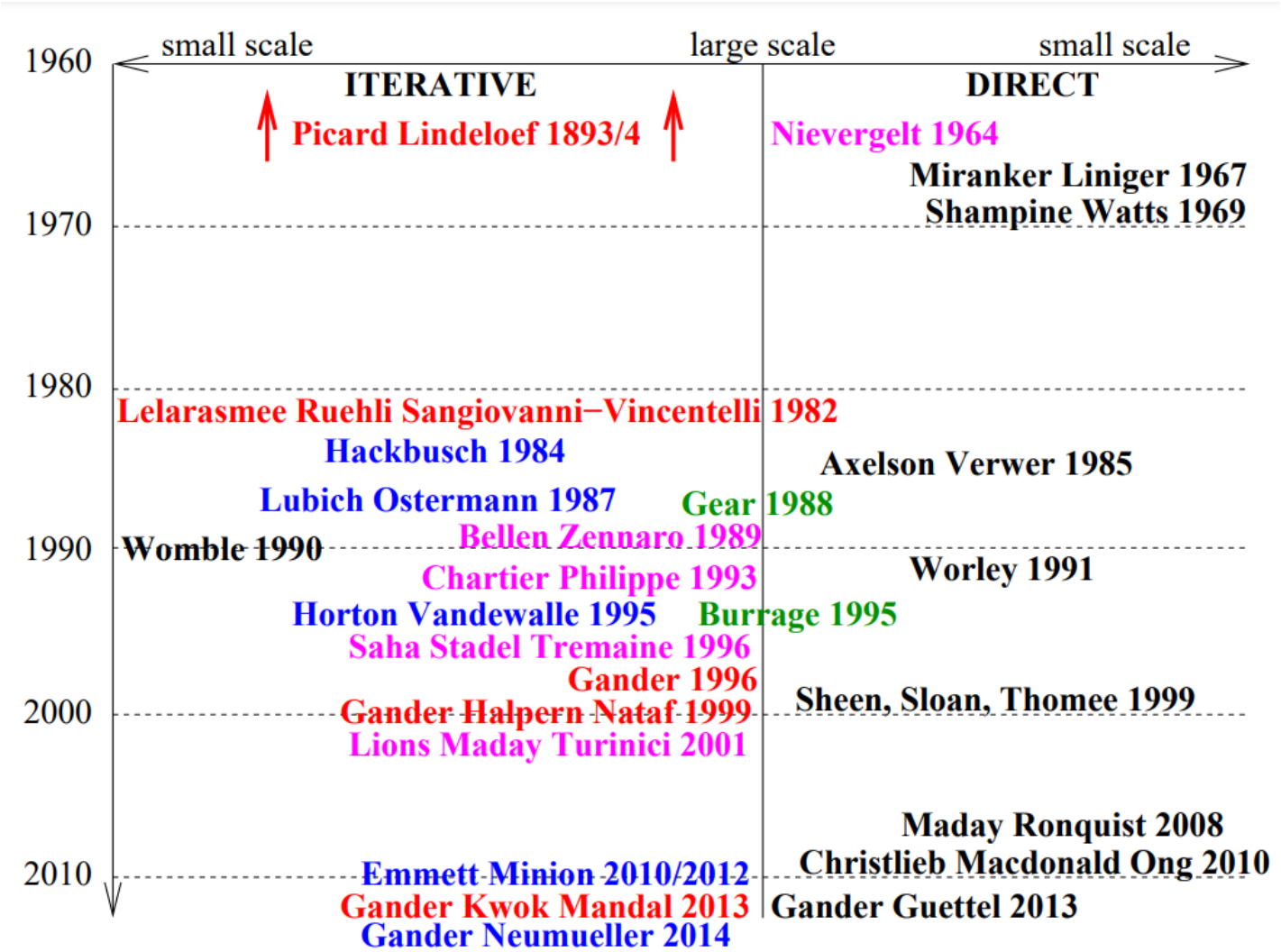
$$u_0 = u_0,$$
$$u_{i+1} = \Phi_{i+1}(u_i) + g_{i+1} \quad i = 0,1,\ldots,N-1,$$



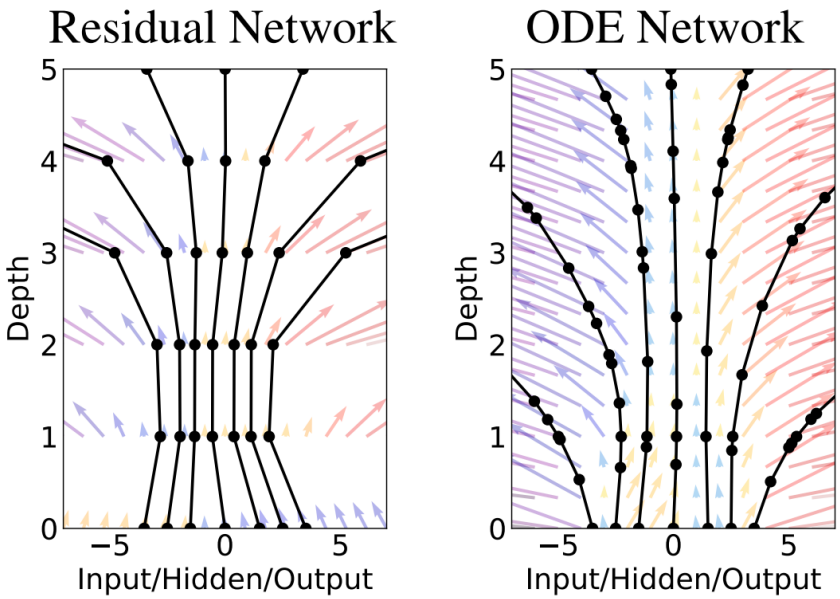
- We can assign an equation to every one-step method.

$$Au = \begin{bmatrix} I & & & \\ -\Phi_1 & I & & \\ & \ddots & \ddots & \\ & & -\Phi_N & I \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_N \end{bmatrix} = g.$$

Time parallel time integration methods



Neural ODEs



$$u'(t) = f_\theta(t, u(t))$$
$$u(0) = u_0$$

$$L(u(T)) = L(ODESolve(u_0, f, 0, T, \theta))$$

$$\frac{da(t)}{dt} = -a(t)^T \frac{\partial f_\theta(t, u(t))}{\partial z}$$

$$\frac{dL}{d\theta} = -\int_0^T a(t)^T \frac{\partial f_\theta(t, u(t))}{\partial \theta} dt$$

Multigrid Reduction in Time (MGRIT)

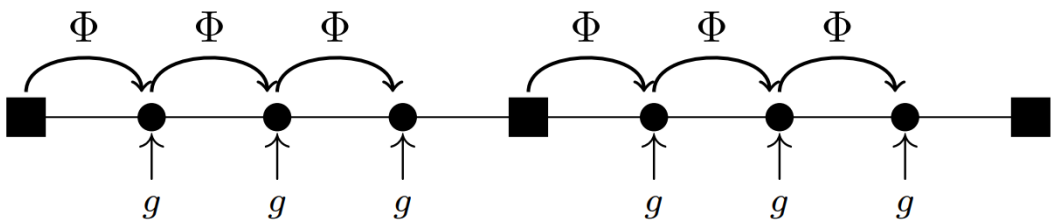
Two grid method

- The algorithm is based on a one step method
- Idea: We use coarser grids to solve the problem

- Assume that the problem is linear.
- $Au = g$ assign to the one-step method on the fine grid.
- $B_\Delta u_\Delta = g_\Delta$ assign to the one-step method on the coarse grid.

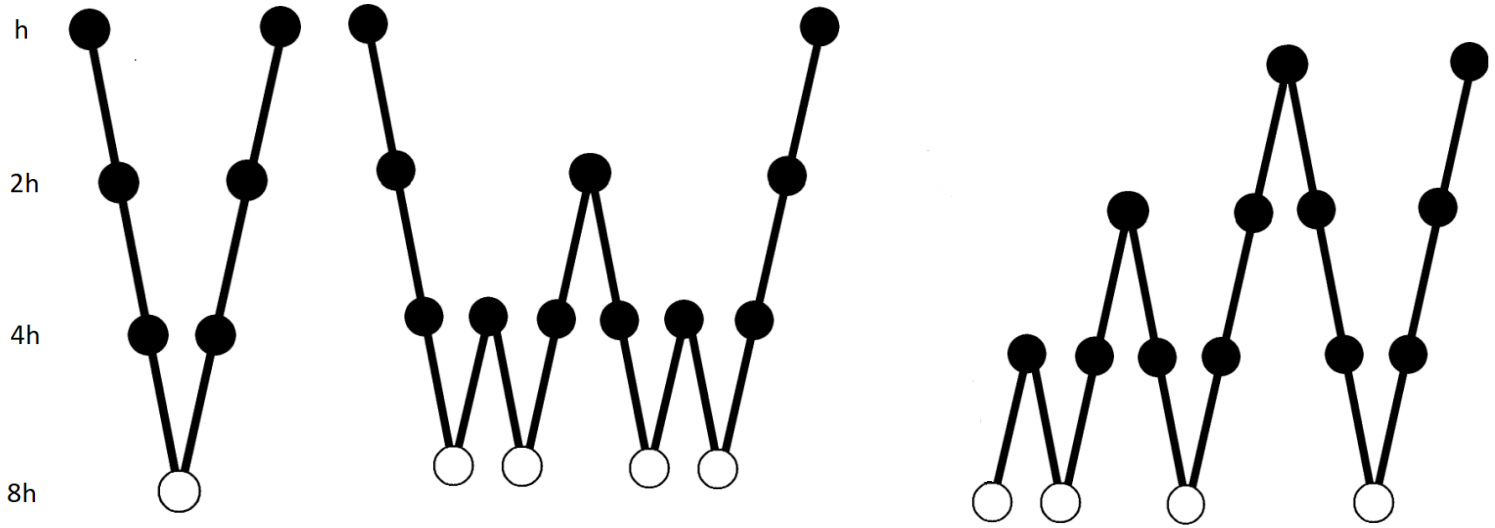
1. Algorithm Two grid method

1: Relaxation of $Au = g$ by F-relaxation (or FCF-relaxation)
2: Restriction: $r_\Delta = R_I(g - Au^k)$
3: Solve the problem $B_\Delta v = r_\Delta$ in the coarse grid
4: Prolongation: $u^{k+1} = u^k + Pv$



MGRIT

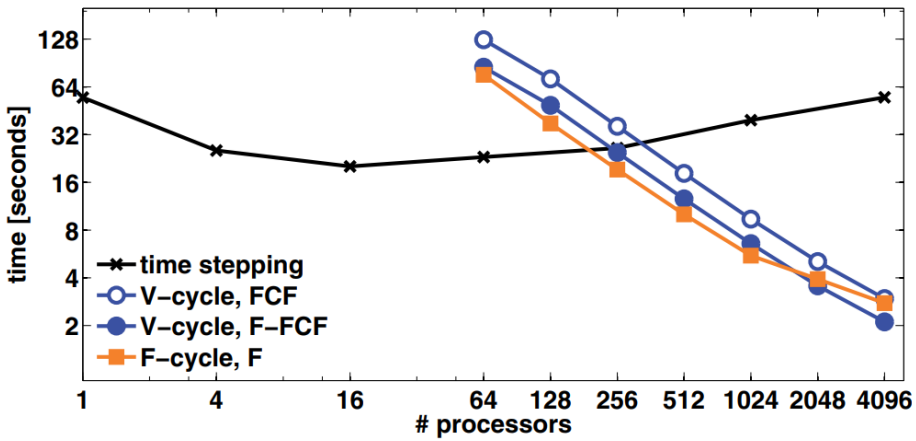
- The method can be recursively extended to more meshes.



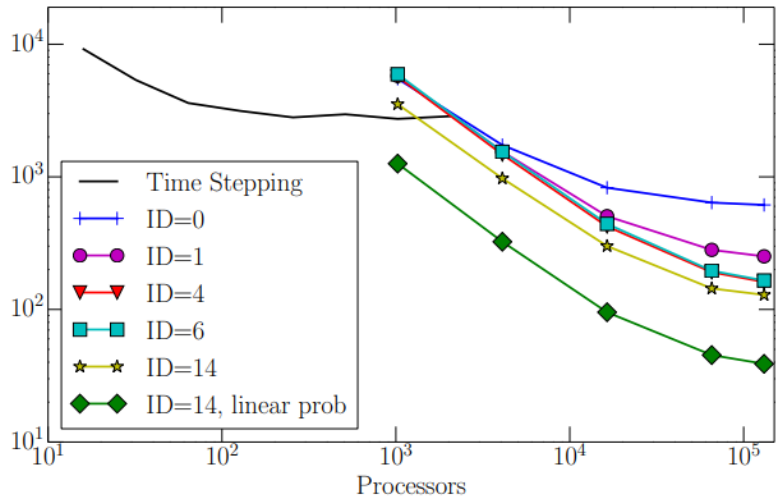
V- W- and F-cycle

- Iterative method
- Parallelizable
- Number of operation: $\mathcal{O}(N)$
- Nonlinear problems: FAS scheme

Results from the literature



Heat equation [1]



p-Laplace problem [2]

Adaptive methods

Adaptive one-step methods

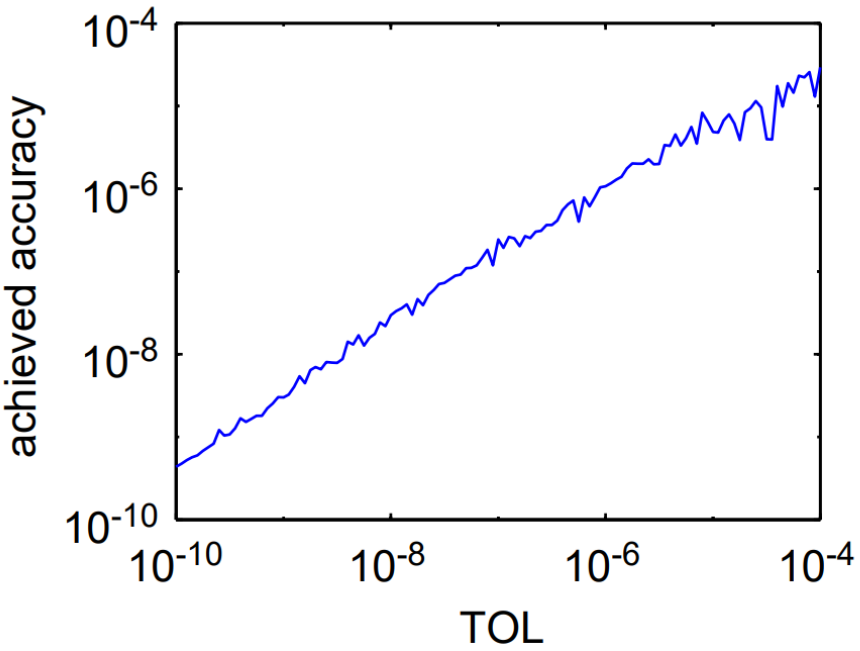
- Varying the step size based on local measurements.
- Keeping the local truncation error under a given tolerance value

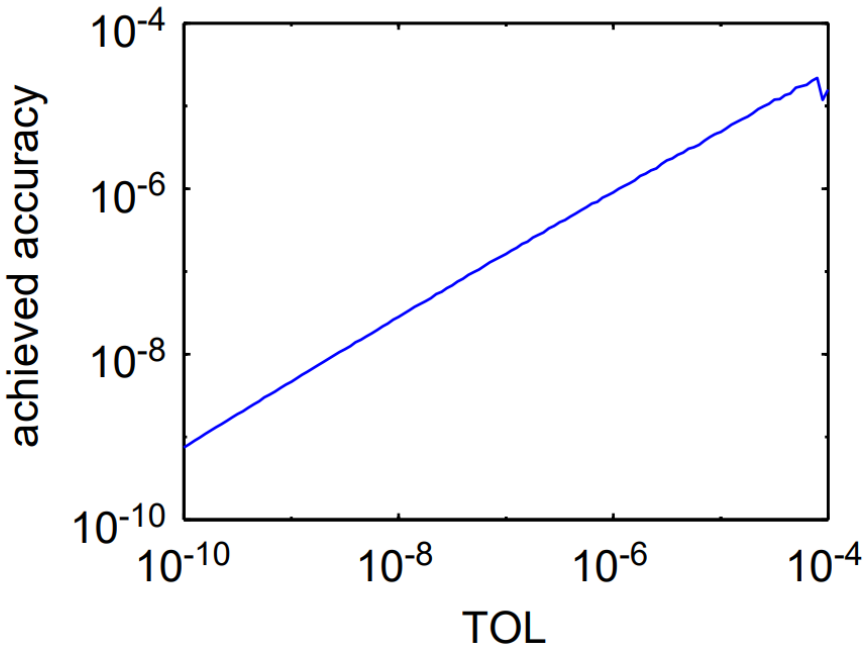
$$h_{n+1} = \rho_n h_n,$$

$$\rho_n = \left(\frac{\mathrm{TOL}}{l_n}\right)^{\beta/(p+1)} \rho_{n-1}^{\alpha_1} \rho_{n-2}^{\alpha_2}$$

Controller	β	α_1	α_2
Classic	1	0	0
Exponential forgetting	2/3	0	0
PI3333	2/3	−1/3	0
H211PI	1/6	1/6	0
H211b	1/4	1/4	1/4

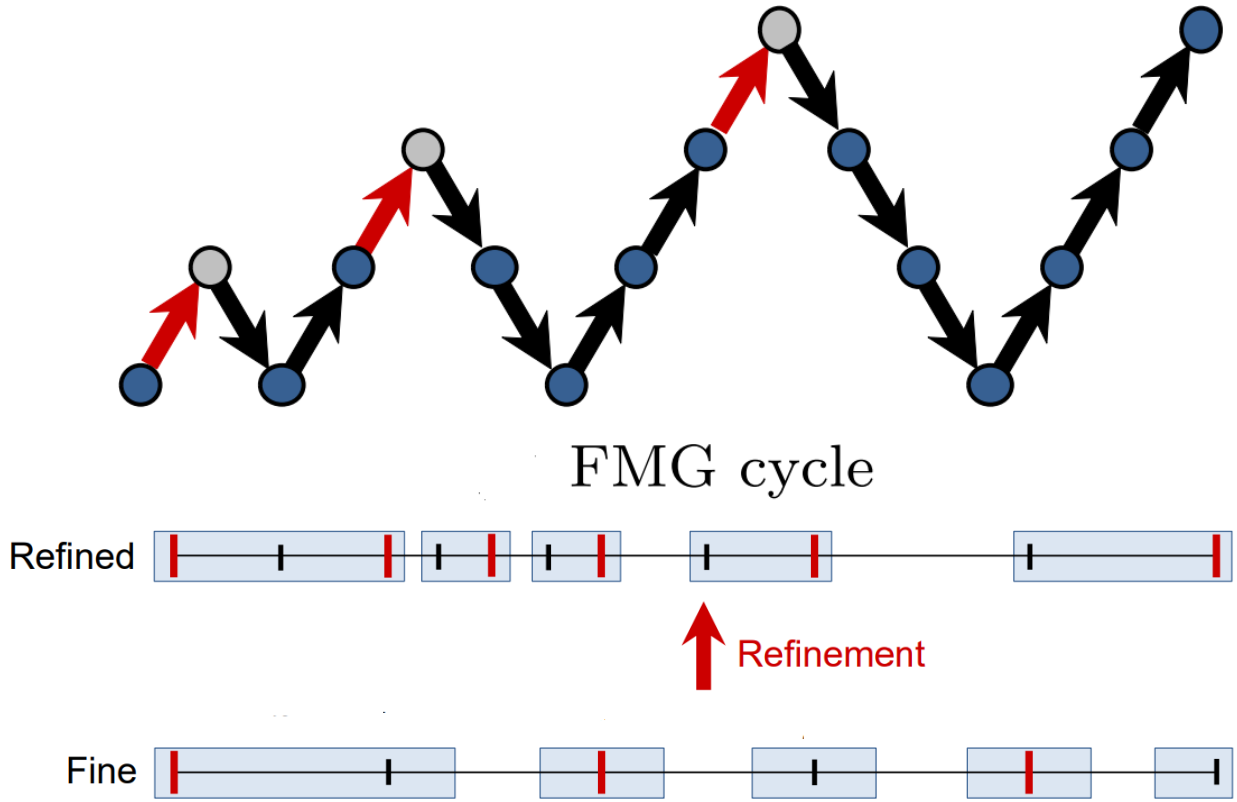
Tolerance proportionality





Tolerance proportionality in the case of the Classic and H211b controller for the stiff Chemakzo problem (Gustaf Söderlind)

Adaptive MGRIT



- Refinement factor: determines how many subintervals to divide the intervals on the current grid.

Classic controller

$$h_{n+1} = \left(\frac{TOL}{l_n}\right)^{\frac{1}{p+1}} h_n$$

$$\downarrow$$

$$rfactor = \frac{h_n}{h_{n+1}} = \left\lceil \left(\frac{l_n}{TOL}\right)^{\frac{1}{p+1}} \right\rceil$$

Results

Problems

Linear

$$y_1' = 0.2y_1$$
$$y_2' = -0.4(y_2 - y_1)$$
$$y(0) = [0.3,0],\ t \in [0,20]$$

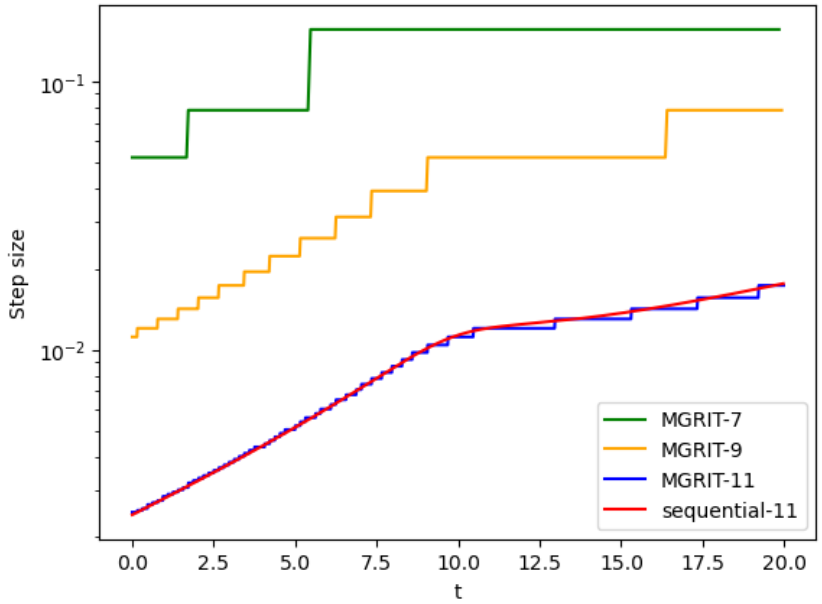
Lotka-Volterra system

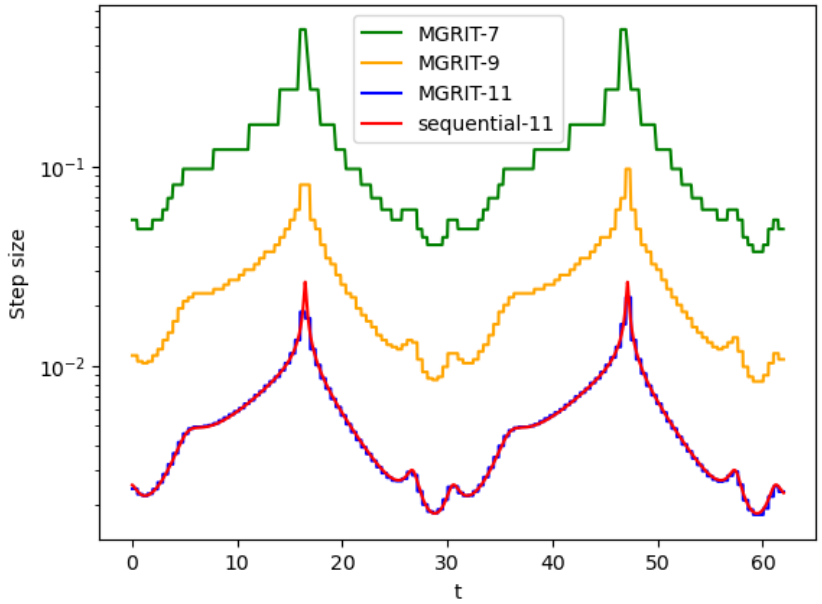
$$y_1' = 0.1y_1 - 0.3y_1y_2$$
$$y_2' = 0.5y_1y_2 - 0.5y_2$$
$$y(0) = [1,1],\ t \in [0,62]$$

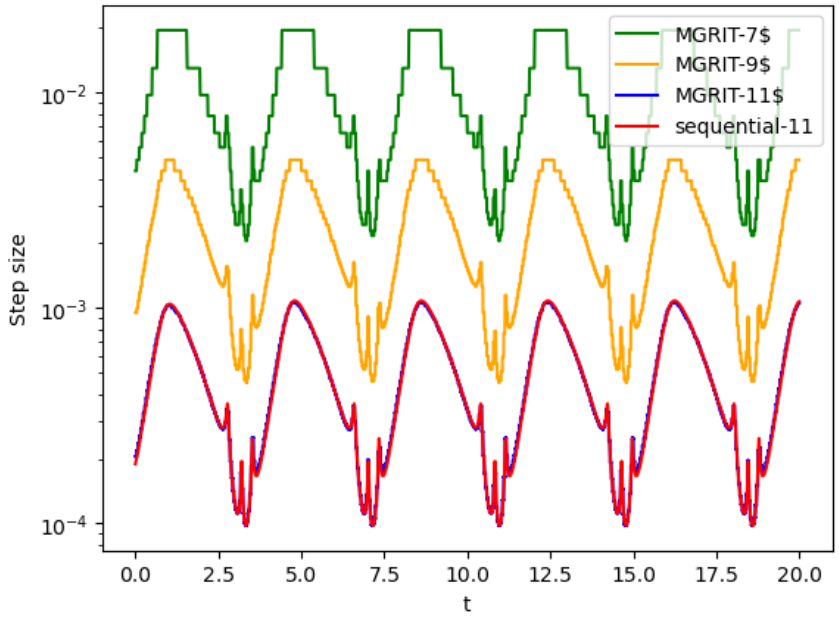
Van der Pol equation

$$y_1' = y_2$$
$$y_2' = 2(1 - y_1^2)y_2 - y_1$$
$$y(0) = [2,0],\ t \in [0,20]$$

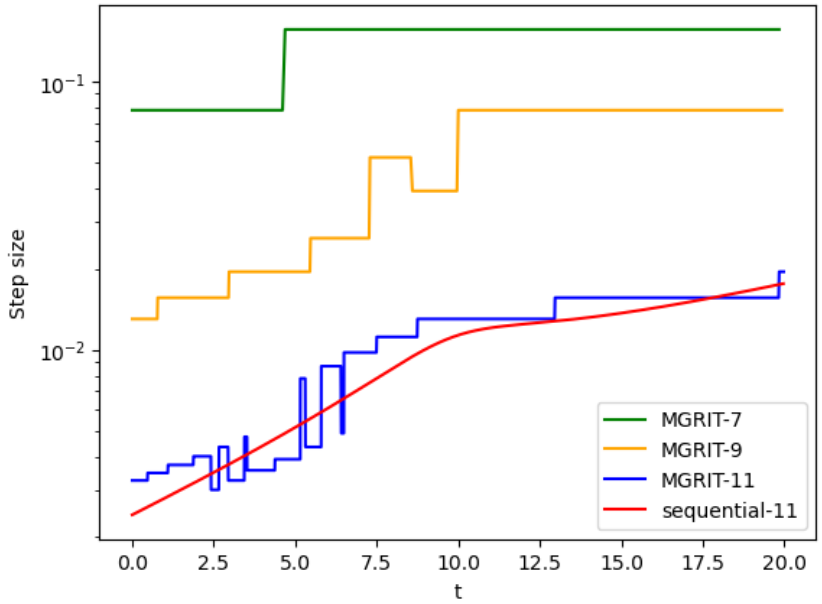
Step sizes – Classic Controler

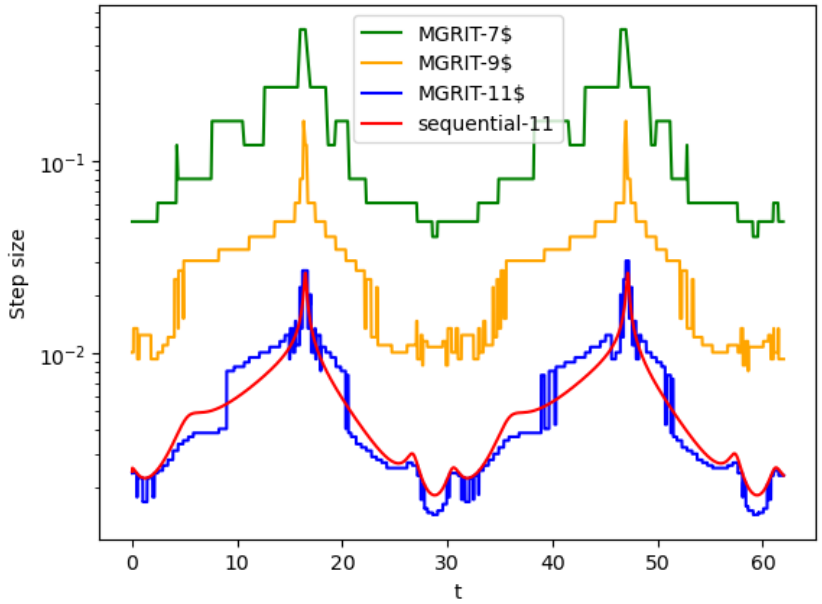


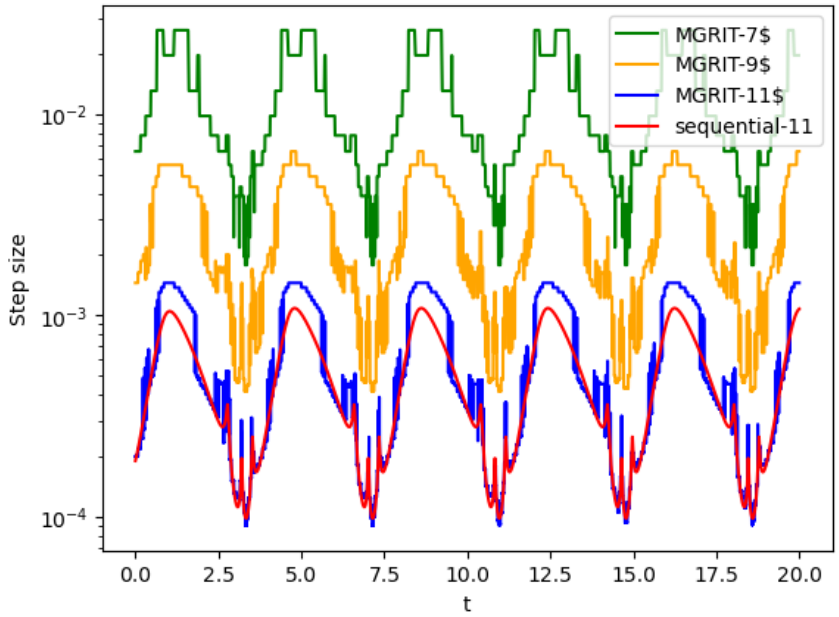




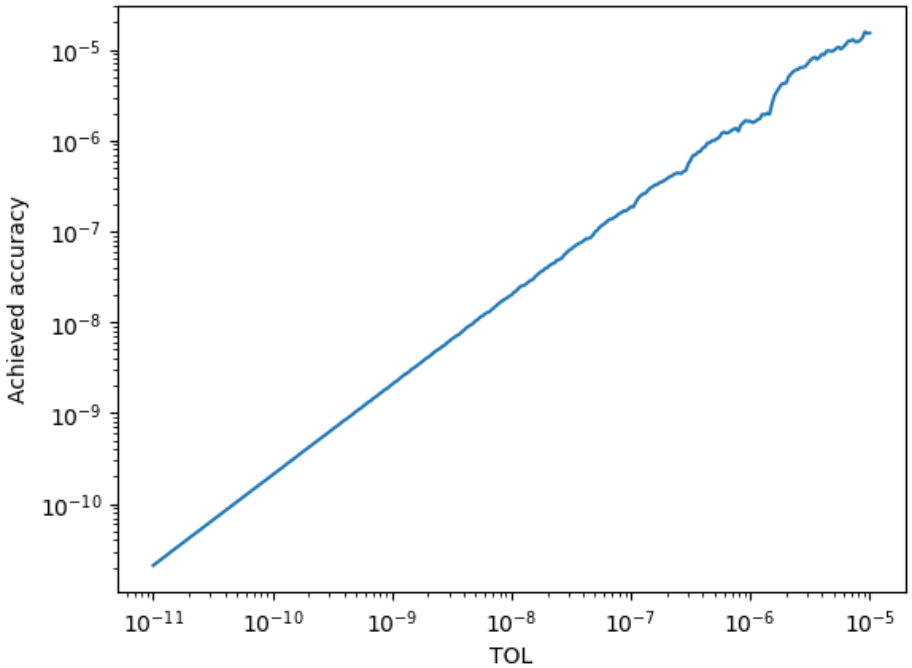
Step sizes – Exponential forgetting



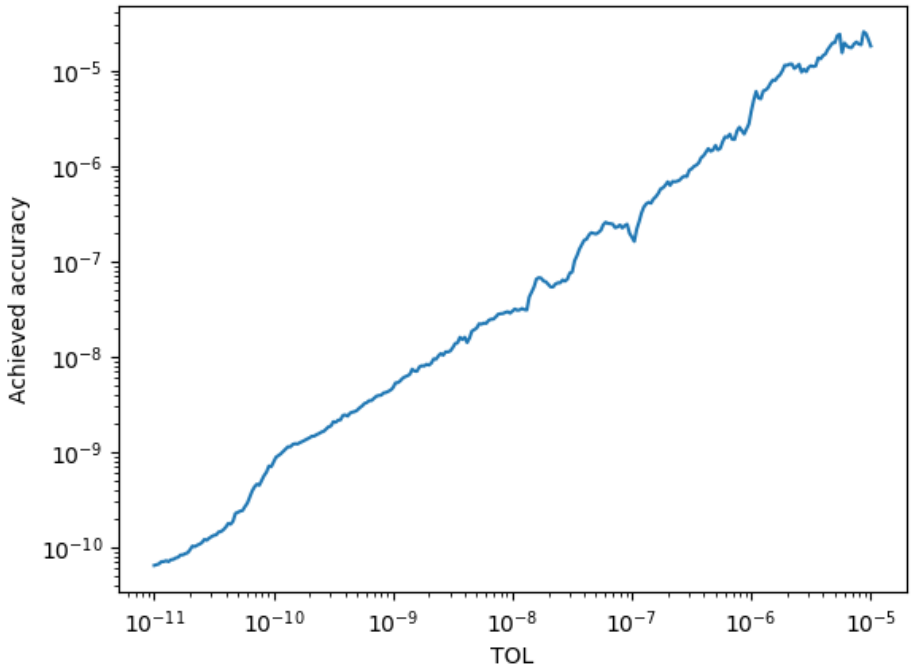




Tolerance proportionality

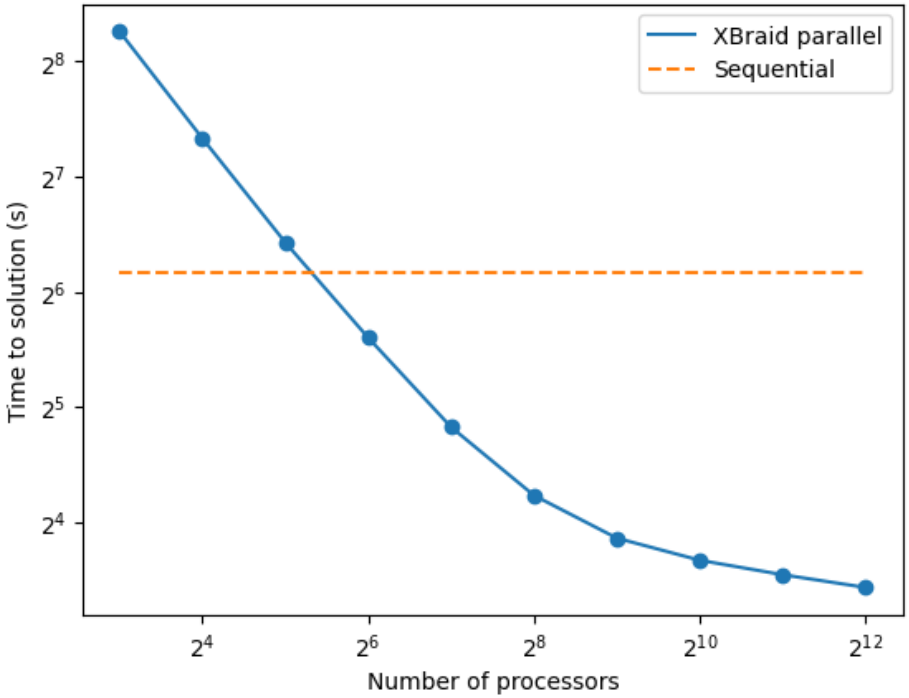


Classic controller

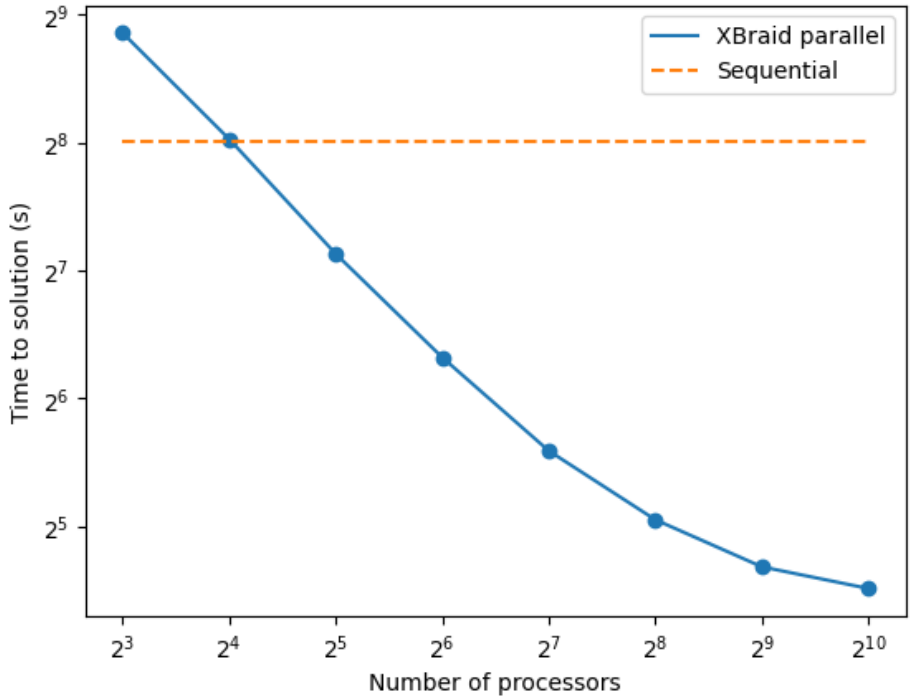


Exponential forgetting

Running time



Heat equation



Nonlinear heat (Porous medium) equation

Future plans

- Developing sophisticated controllers in parallel framework
- Testing on other partial differential equations
- Using time parallel adaptive methods to teach neural ODEs

Thank you for your attention!

Bibliography I

R. D. Falgout, S. Friedhoff, Tz. V. Kolev, S. P. MacLachlan, and J. B. Schroder: *Parallel Time Integration with Multigrid*, SIAM Journal on Scientific Computing, 36 (2014), pp.C635-C661.

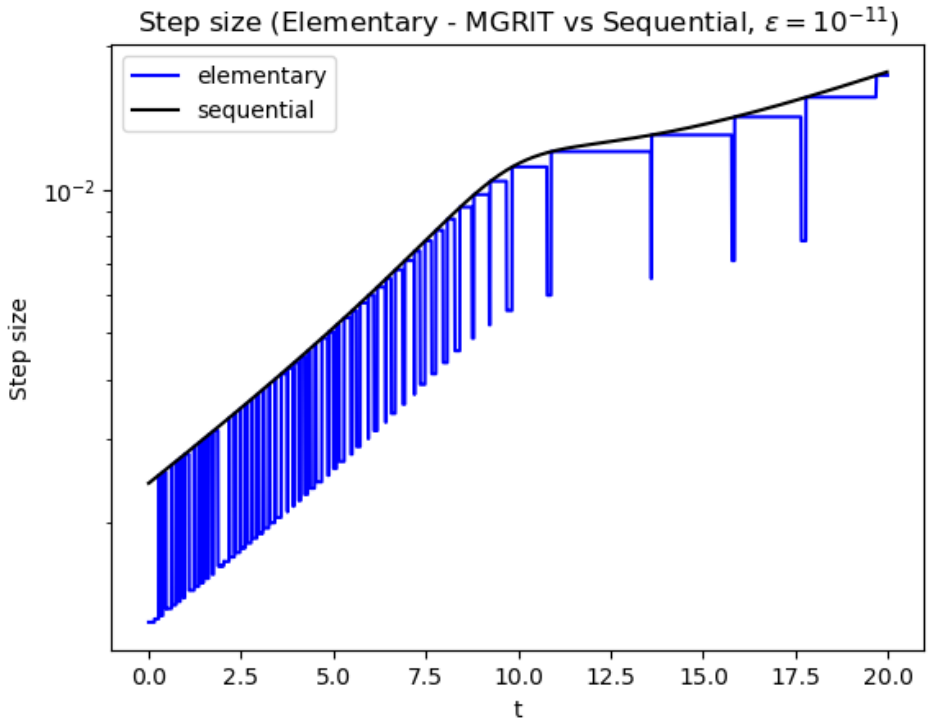
R. D. Falgout, T. A. Manteuffel, B. O'Neill, and J. B. Schroder: *Multigrid Reduction in Time for Nonlinear Parabolic Problems: A Case Study*, SIAM Journal on Scientific Computing, 39 (2017), pp 298-322.

Classic controller

$$rfactor = \frac{h_n}{h_{n+1}} = \left\lfloor \left(\frac{l_n}{TOL} \right)^{\frac{1}{p+1}} \right\rfloor$$

$$\downarrow$$

$$rfactor = \frac{h_n}{h_{n+1}} = \left\lfloor \left(\frac{l_n}{TOL} \right)^{\frac{1}{p+1}} - c \right\rfloor$$



Initial mesh