Multigrid Reduction in Time for Chaotic Dynamical Systems

David A. Vargas, Robert D. Falgout, Stefanie Günther, and Jacob B. Schroder

July 13, 2022

THE UNIVERSITY OF NEW MEXICO

Introduction and Motivation

MGRIT

solves the IVP

$$\begin{aligned} \mathbf{u}_0 &= \mathbf{f}_0 \\ \mathbf{u}_{i+1} &= \Phi(\mathbf{u}_i) + \mathbf{f}_{i+1} \qquad\qquad i = 0, 1, 2, \ldots, n \end{aligned}$$

MGRIT

solves the IVP

$$\begin{aligned} \mathbf{u}_0 &= \mathbf{f}_0 \\ \mathbf{u}_{i+1} &= \Phi(\mathbf{u}_i) + \mathbf{f}_{i+1} \qquad\qquad i = 0, 1, 2, \ldots, n \end{aligned}$$

$$A(\mathbf{u}) = \begin{bmatrix} I & & & & \\ -\Phi & I & & & \\ & -\Phi & I & & \\ & & \ddots & \ddots & \\ & & & -\Phi & I \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_n \end{bmatrix} = \begin{bmatrix} \mathbf{f}_0 \\ \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_n \end{bmatrix}$$

The two-level algorithm

- The time-domain is partitioned into C-points and F-points

The two-level algorithm

- The time-domain is partitioned into C-points and F-points

- F-relaxation sets the residual to zero at each F-point (block Jacobi)

The two-level algorithm

- The time-domain is partitioned into C-points and F-points

- F-relaxation sets the residual to zero at each F-point (block Jacobi)

- the coarse system is solved sequentially

The coarse grid equation

- The "ideal" coarse grid equation applies Φ m-times per interval

$$A_*(\mathbf{u}) = \begin{bmatrix} I & & & & \\ -\Phi^m & I & & & \\ & -\Phi^m & I & & \\ & & \ddots & \ddots & \\ & & & -\Phi^m & I \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_m \\ \mathbf{u}_{2m} \\ \vdots \\ \mathbf{u}_n \end{bmatrix} = \begin{bmatrix} \mathbf{f}_0 \\ \mathbf{f}_m \\ \mathbf{f}_{2m} \\ \vdots \\ \mathbf{f}_n \end{bmatrix}$$

The coarse grid equation

- The "ideal" coarse grid equation applies Φ m-times per interval

$$A_*(\mathbf{u}) = \begin{bmatrix} I & & & & \\ -\Phi^m & I & & & \\ & -\Phi^m & I & & \\ & & \ddots & \ddots & \\ & & & -\Phi^m & I \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_m \\ \mathbf{u}_{2m} \\ \vdots \\ \mathbf{u}_n \end{bmatrix} = \begin{bmatrix} \mathbf{f}_0 \\ \mathbf{f}_m \\ \mathbf{f}_{2m} \\ \vdots \\ \mathbf{f}_n \end{bmatrix}$$

★ This system is no less expensive to solve

The coarse grid equation

- The "ideal" coarse grid equation applies Φ m-times per interval

$$A_*(\mathbf{u}) = \begin{bmatrix} I & & & & \\ -\Phi^m & I & & & \\ & -\Phi^m & I & & \\ & & \ddots & \ddots & \\ & & & -\Phi^m & I \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_m \\ \mathbf{u}_{2m} \\ \vdots \\ \mathbf{u}_n \end{bmatrix} = \begin{bmatrix} \mathbf{f}_0 \\ \mathbf{f}_m \\ \mathbf{f}_{2m} \\ \vdots \\ \mathbf{f}_n \end{bmatrix}$$

  ★ This system is no less expensive to solve

- A_* is approximated by the coarse operator A_c, where $\Phi_c \approx \Phi^m$ (rediscretization)

The two level algorithm is a nonlinear splitting method

$$A_*(\mathbf{u}) = A_c(\mathbf{u}) - \boldsymbol{\tau}(\mathbf{u})$$

The two level algorithm is a nonlinear splitting method

$$A_*(\mathbf{u}) = A_c(\mathbf{u}) - \boldsymbol{\tau}(\mathbf{u})$$

$$\begin{aligned} A_c(\mathbf{u}^{k+1}) - \boldsymbol{\tau}(\mathbf{u}^k) &= \mathbf{f} \\ A_c(\mathbf{u}^{k+1}) &= \mathbf{f} + \boldsymbol{\tau}(\mathbf{u}^k) \end{aligned}$$

The two level algorithm is a nonlinear splitting method

$$A_*(\mathbf{u}) = A_c(\mathbf{u}) - \boldsymbol{\tau}(\mathbf{u})$$

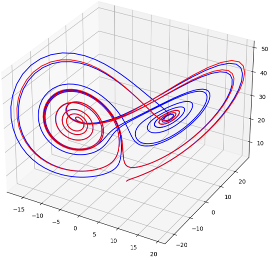
$$\begin{aligned} A_c(\mathbf{u}^{k+1}) - \boldsymbol{\tau}(\mathbf{u}^k) &= \mathbf{f} \\ A_c(\mathbf{u}^{k+1}) &= \mathbf{f} + \boldsymbol{\tau}(\mathbf{u}^k) \end{aligned}$$

$$[\boldsymbol{\tau}(\mathbf{u})]_i = \Phi^m(\mathbf{u}_{i-m}) - \Phi_c(\mathbf{u}_{i-m})$$

The MGRIT algorithm: overview

- MGRIT extends the two level algorithm to an arbitrary number of levels
  - recursive application of the two-level algorithm
  - the coarsest grid may have as few as two time-points!
- optimal scaling for parabolic problems using FCF relaxation
- becomes more efficient as total number of time-points increases

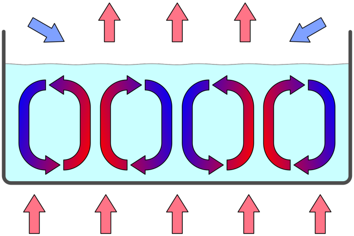
MGRIT does not converge well for chaotic systems



The Lorenz attractor

- Non-linear
- Sensitive to perturbations
- Ill-conditioned

The Lorenz system is a model problem for chaos



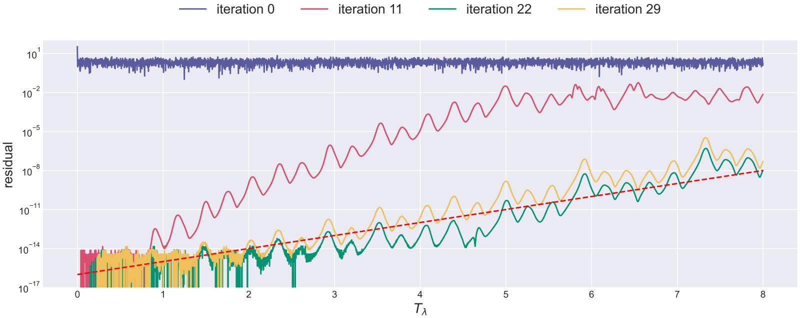
By I, Eyrian, CC BY-SA 3.0,

https://commons.wikimedia.org/w/index.php?curid=2525196

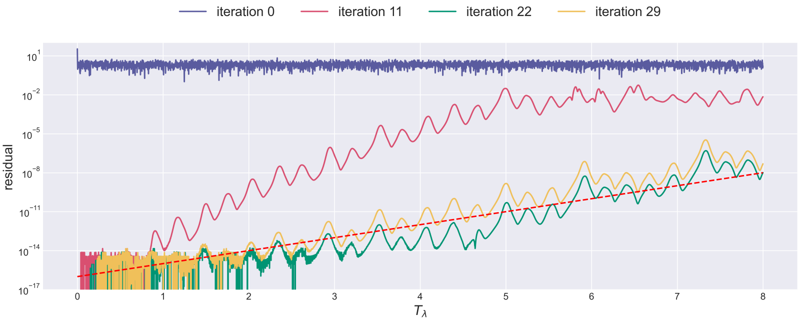
$$\begin{cases} \frac{dx}{dt} = \sigma(y - x), \\ \frac{dy}{dt} = x(\rho - z) - y, \\ \frac{dz}{dt} = xy - \beta z \end{cases}$$

- three dimensional
- nonlinear ($-xz$ and xy terms)
- derived by Lorenz as a simplified model of convection
- many chaotic PDEs can be modeled very accurately by finite dimensional ODEs

Classical MGRIT fails to converge for chaotic systems



Classical MGRIT fails to converge for chaotic systems



$\lambda \approx 0.9$

$T_\lambda = \frac{\log(10)}{\lambda}$

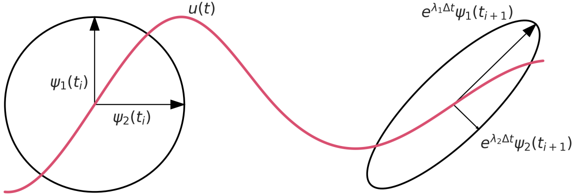
Lyapunov Exponents/Vectors

For the Lorenz system:

λ_1: Unstable- (≈ 0.9) orbits diverge exponentially

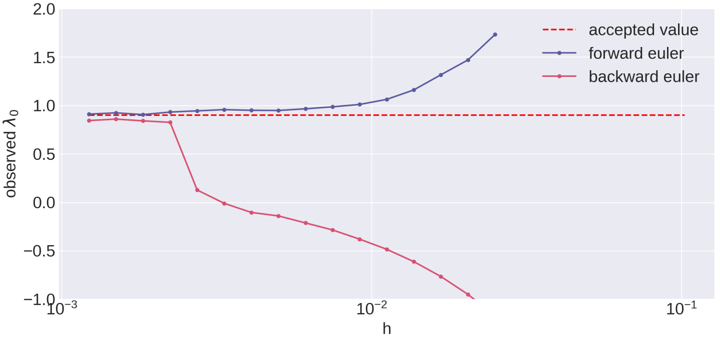
λ_2: Neutral- ($= 0$) phase difference in time

λ_3: Stable- (≈ -14) orbits exponentially approach the strange attractor



The θ Method

Explicit methods too unstable, implicit methods too dissipative



$$\mathbf{u}' = f(\mathbf{u})$$

$$\mathbf{u}_{i+i} = \mathbf{u}_i + hf(\mathbf{u}_i) \qquad \text{Forward Euler}$$

$$\mathbf{u}_{i+i} = \mathbf{u}_i + hf(\mathbf{u}_{i+1}) \qquad \text{Backward Euler}$$

The θ method splits the difference

$$\mathbf{u}' = f(\mathbf{u})$$

$$\mathbf{u}_{i+1} = \mathbf{u}_i + \theta h f(\mathbf{u}_i) + (1-\theta) h f(\mathbf{u}_{i+1}) \qquad \theta \text{ method}$$

- $\theta = 1$: forward Euler
- $\theta = 0$: backward Euler
- $\theta = 1/2$: Crank-Nicolson

The θ method splits the difference

$$\mathbf{u}' = f(\mathbf{u})$$

$$\mathbf{u}_{i+1} = \mathbf{u}_i + \theta h f(\mathbf{u}_i) + (1-\theta) h f(\mathbf{u}_{i+1}) \qquad \theta \text{ method}$$

- $\theta = 1$: forward Euler
- $\theta = 0$: backward Euler
- $\theta = 1/2$: Crank-Nicolson

- What's the best value of θ to use for a given coarsening factor, m?

Asymptotic values of θ

Since the θ method is first order for *any* value of θ, i.e.

$$\phi_\theta(0) = \phi_m(0)$$
$$\phi'_\theta(0) = \phi'_m(0)$$

we can use the extra degree of freedom to find θ such that

$$\phi''_\theta(0) = \phi''_m(0)$$

and approximate the fine-grid to *second* order in mh.

Asymptotic values of θ

Since the θ method is first order for *any* value of θ, i.e.

$$\phi_\theta(0) = \phi_m(0)$$
$$\phi'_\theta(0) = \phi'_m(0)$$

we can use the extra degree of freedom to find θ such that

$$\phi''_\theta(0) = \phi''_m(0)$$

and approximate the fine-grid to *second* order in *mh*.

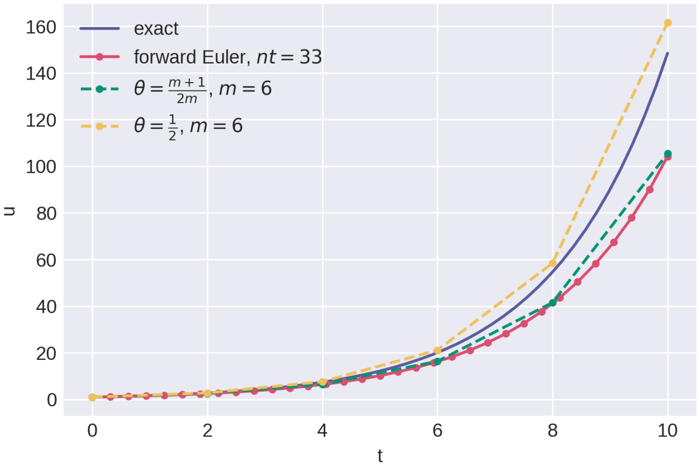
Forward Euler:

$$\theta_m = \frac{m+1}{2m}$$

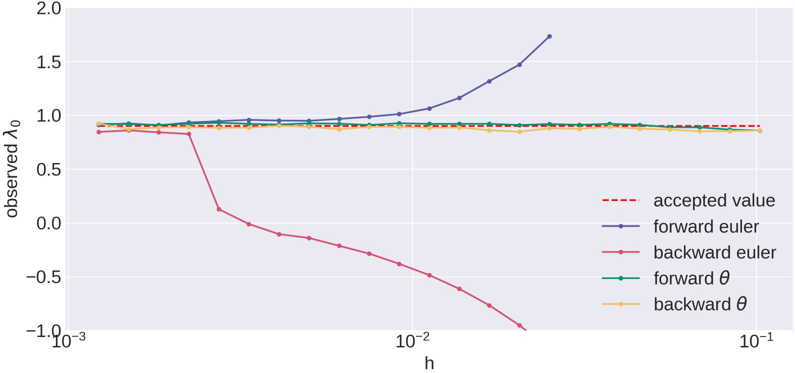
Backward Euler:

$$\theta_m = \frac{m-1}{2m}$$

The θ method approximates the fine-grid to second order, and the continuous equation to first order

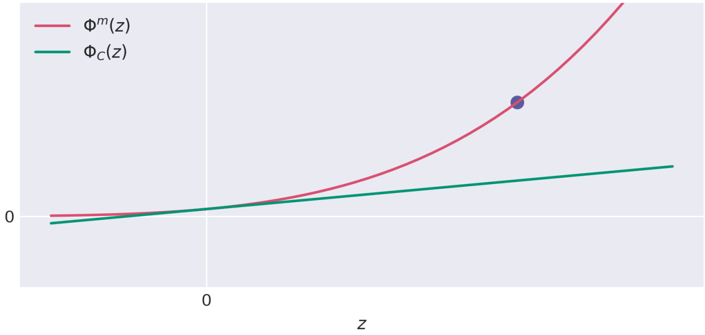


The asymptotic values of θ better preserve Lyapunov exponents on coarse grids



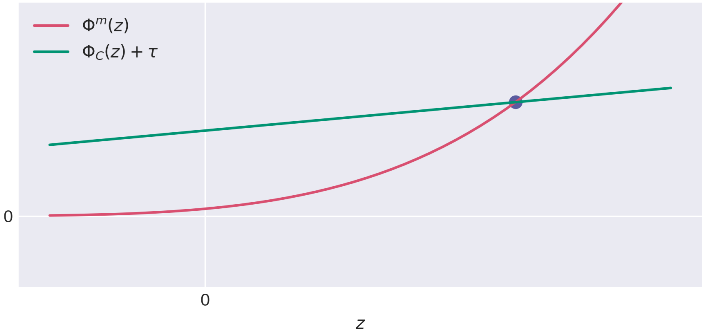
Δ correction

τ correction is a constant correction



$$A_c(\mathbf{v}^{k+1}) = \mathbf{f}$$

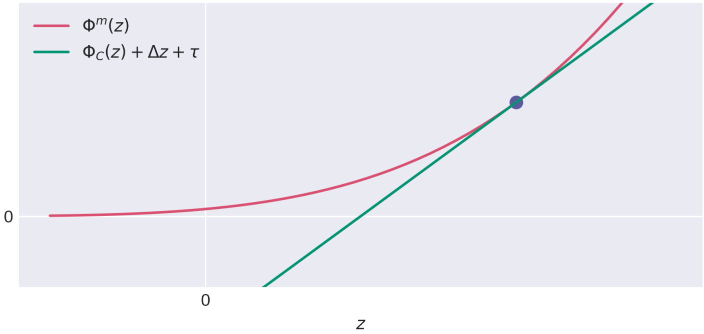
τ correction is a constant correction



$$A_c(\mathbf{v}^{k+1}) = \mathbf{f} + \boldsymbol{\tau}$$

$$\boldsymbol{\tau} = A_*(\mathbf{v}^k) - A_c(\mathbf{v}^k)$$

Δ correction is a linear correction



$$[A_c + \Delta](\mathbf{v}^{k+1}) = \mathbf{f} + \boldsymbol{\tau}$$

$$\Delta = D_{\mathrm{u}} A_*(\mathbf{v}^k) - D_{\mathrm{u}} A_c(\mathbf{v}^k)$$

$$\boldsymbol{\tau} = A_*(\mathbf{v}^k) - A_c(\mathbf{v}^k) - \Delta \mathbf{v}^k$$

Δ correction is a generalization of Newton's method

- exact method for linear systems
- quadratic convergence for nonlinear systems

$$[A_c + \Delta](\mathbf{v}^{k+1}) = \mathbf{f} + \boldsymbol{\tau}$$

Δ correction is a generalization of Newton's method

- exact method for linear systems
- quadratic convergence for nonlinear systems

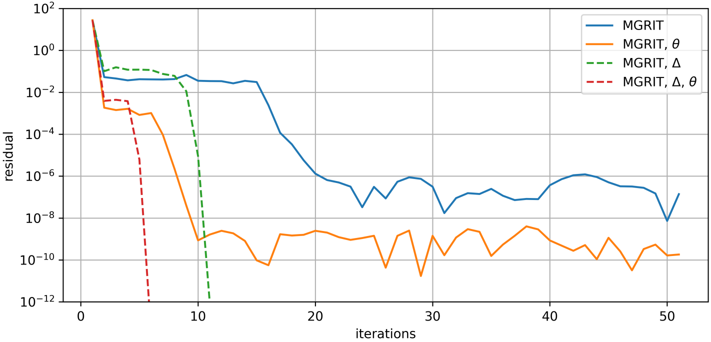
$$[A_c + \Delta](\mathbf{v}^{k+1}) = \mathbf{f} + \boldsymbol{\tau}$$

If A_c is the identity, then the coarse grid equation becomes

$$\begin{aligned}
[D_{\mathrm{u}} A_*(\mathbf{v}^k)]\mathbf{v}^{k+1} &= [D_{\mathrm{u}} A_*(\mathbf{v}^k)]\mathbf{v}^k + (\mathbf{f} - A_*(\mathbf{v}^k)) \\
\mathbf{v}^{k+1} &= \mathbf{v}^k - [D_{\mathrm{u}} A_*(\mathbf{v}^k)]^{-1}(A_*(\mathbf{v}^k) - \mathbf{f})
\end{aligned}$$

Results for Lorenz

optimal θ methods converge faster, Δ correction converges quadratically



Two-level method, Lorenz system, forward Euler fine-grid, $T_f = 8\,T_\lambda$ and $nt = 8192$

θ method improves both stability and MGRIT convergence

- Lorenz system, Forward Euler fine-grid, F-relax, $m = 2$
- T_f constant, n_t varies, iteration counts to reach residual tolerance of $1e - 10$

	T_f, n_t				
Algorithm	4, 512	4, 1024	4, 2048	4, 4096	4, 8192
MGRIT_2	*	44	22	15	12
MGRIT_2, θ	19	13	9	7	6
MGRIT_2, Δ	*	11	8	6	6
MGRIT_2, Δ, θ	8	6	5	4	4

Δ correction extends the length of the time-domain

- Lorenz system, Forward Euler fine-grid, $m = 2$
- T_f, n_t vary, iteration counts to reach residual tolerance of $1e-10$

	T_f, n_t			
Algorithm	2, 4096	4, 8192	8, 16384	12, 24576
MGRIT_2	10	13	64	-
MGRIT_2, θ	4	5	7	-
MGRIT_2, Δ	5	6	8	94
MGRIT_2, Δ, θ	3	4	5	48

RK4-θ method allows us to reach very coarse grids for Lorenz

- 4-level method, RK4 fine-grid, 4th order θ method coarse grid, $m = 4$
- T_f, n_t vary, iteration counts to reach residual tolerance of $1e - 10$

	T_f, n_t			
Algorithm	2, 2048	4, 4096	8, 8192	12, 12288
MGRIT_4	*	*	*	*
MGRIT_4, θ	15	29	-	-
MGRIT_4, Δ	*	*	*	*
MGRIT_4, Δ, θ	6	9	18	51

RK4-θ method allows us to reach very coarse grids for Lorenz

- 4-level method, RK4 fine-grid, 4th order θ method coarse grid, $m = 4$
- T_f, n_t vary, iteration counts to reach residual tolerance of $1e-10$

	T_f, n_t			
Algorithm	2, 2048	4, 4096	8, 8192	12, 12288
MGRIT_4	$*$	$*$	$*$	$*$
MGRIT_4, θ	15	29	-	-
MGRIT_4, Δ	$*$	$*$	$*$	$*$
MGRIT_4, Δ, θ	6	9	18	51

coarsest grids have 32 and 64 time-points, respectively!

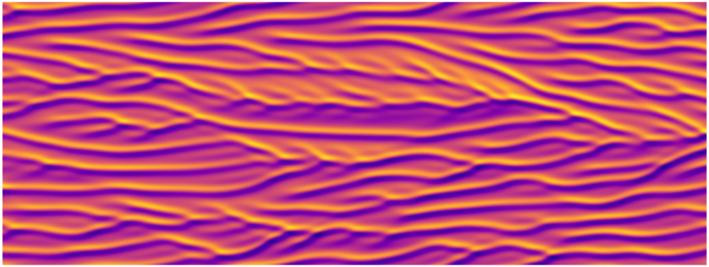
Low rank Δ correction for chaotic PDEs

Δ correction is too expensive for PDEs

- For Lorenz, the Δ_i are 3×3 matrices
- For a 1D PDE, they are $n_x \times n_x$
- For a 2D PDE...
- Jacobians might not be available

The Kuramoto-Sivashinsky (KS) equation

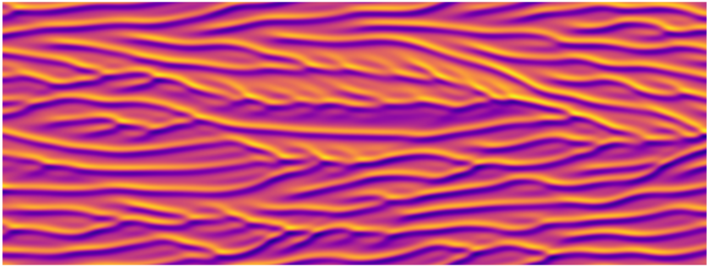
$$\mathbf{u}_t = -\mathbf{u}_{xx} - \mathbf{u}_{xxxx} - \mathbf{u}\mathbf{u}_x$$
$$\mathbf{u}(t, 0) = \mathbf{u}(t, L)$$



The Kuramoto-Sivashinsky (KS) equation

$$\mathbf{u}_t = -\mathbf{u}_{xx} - \mathbf{u}_{xxxx} - \mathbf{u}\mathbf{u}_x$$

$$\mathbf{u}(t, 0) = \mathbf{u}(t, L)$$



- Infinitely many Lyapunov exponents, λ
- Finitely many $\lambda > 0$
- Why correct for modes which already converge well?

Low-rank Δ correction

Given a rank k orthogonal basis for the unstable manifold: Ψ_i

$$\hat{\Delta}_i = (\Delta_i \Psi_i) \Psi_i^T$$

Low-rank Δ correction

Given a rank k orthogonal basis for the unstable manifold: Ψ_i

$$\hat{\Delta}_i = (\Delta_i \Psi_i)\Psi_i^T$$

- Only have to store the factors Ψ_i and $\Delta_i \Psi_i$
- The columns of $\Delta_i \Psi_i$ are directional derivatives (matrix free)

Backward Lyapunov Vectors

Given a trajectory $\{\mathbf{u}_i\}$ satisfying

$$\begin{aligned} \mathbf{u}_0 &= \mathbf{f}_0 \\ \mathbf{u}_{i+1} &= \Phi(\mathbf{u}_i) + \mathbf{f}_{i+1} \qquad\qquad i = 0, 1, 2, \ldots, n, \end{aligned}$$

the backward Lyapunov Vectors are an orthonormal set satisfying the tangent equation

$$\Psi_{i+1} R_{i+1} = [D_u \Phi(\mathbf{u}_i)] \Psi_i$$

Low-rank Δ correction solves state and tangent equations simultaneously

- Given a trajectory $\mathbf{u}$ we can find the Lyapunov Vectors Ψ
- Given the Lyapunov Vectors Ψ, we can accelerate the solution of $\mathbf{u}$

Low-rank Δ correction solves state and tangent equations simultaneously

- Given a trajectory $\mathbf{u}$ we can find the Lyapunov Vectors Ψ
- Given the Lyapunov Vectors Ψ, we can accelerate the solution of $\mathbf{u}$
- So low-rank Δ correction uses the same MGRIT cycle to solve for both iteratively

Results for the KS equation

Discretization

- 4th order finite-differencing in space
- 2nd order Lobatto IIIC method in time (Stiffly accurate 2nd order Runge Kutta method)

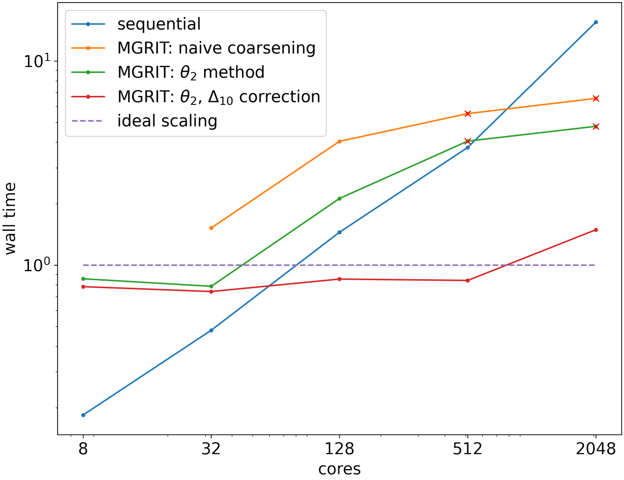
$$\begin{array}{c|cc} 0 & \frac{1}{2} & -\frac{1}{2} \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

- second order θ method constructed from combination of second order Lobatto methods, along with stiff constraint

$$\lim_{z \to -\infty} \phi(z) = 0$$

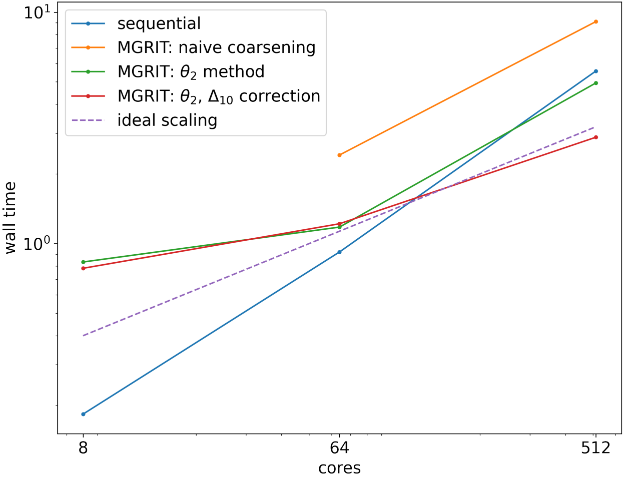
Weak scaling

Figure 1: $T_f = 4T_\lambda$, $n_x = 128$, $n_t = 128$, refinement in time only



Weak scaling

$T_f = 4T_\lambda$, $n_x = 128$, $n_t = 128$, refinement in both time and space (parallel in time only)



Future work

- space-time parallel for KS equation
- extend θ methods to wide range of applications
- convergence on arbitrary time-domains