High-Order Multi-Grid-In-Time

Benjamin Ong

Michigan Technological University
ongbw@mtu.edu

May 19, 2026

Michigan Tech
Create the Future

Michigan Technological University

Conference in honor of Bengt Fornberg's 80th birthday
June 8-10 2026, https://conferences.math.mtu.edu

Michigan Tech
Create the Future

Michigan Tech
Create the Future

Problem Specification

Seek: parallel solution of:

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

- IVP often arises as a Method-Of-Lines discretization of a PDE
- $f : \mathbb{R}^{m+1} \to R^m$
- Time is "causal": employ a sequential time integrator:

$$u_n = \phi_h(t_{n-1}, u_{n-1}), \quad n = 1, \ldots, N \tag{1}$$

- Classically: parallelism means solve eq. (1) in parallel
- Sequential time-stepping can still be bottleneck

Problem Specification

Seek: parallel solution of:

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

- IVP often arises as a Method-Of-Lines discretization of a PDE
- $f : \mathbb{R}^{m+1} \to R^m$
- Time is "causal": employ a sequential time integrator:

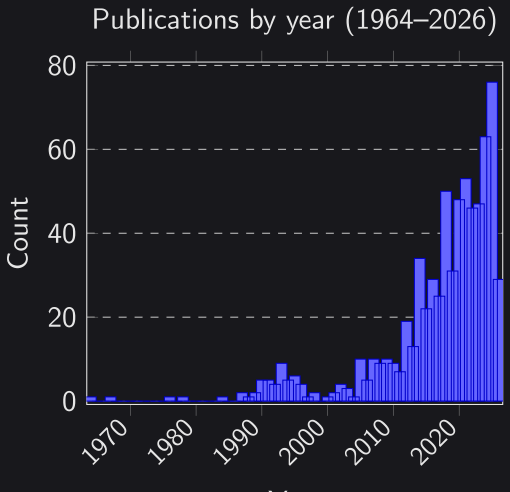
$$u_n = \phi_h(t_{n-1}, u_{n-1}), \quad n = 1, \ldots, N \tag{1}$$

- Classically: parallelism means solve eq. (1) in parallel
- Sequential time-stepping can still be bottleneck

Michigan Tech
Create the Future

PinT Research Activity

https://parallel-in-time.org



New-ish book on time-parallelism (Gander & Lunet, 2024)

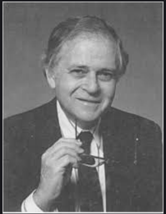
Survey articles: (Gander, 2015), (Ong & Schroder, 2022)

Michigan Tech
Create the Future

Parallel Time Integration: http://parallel-in-time.org

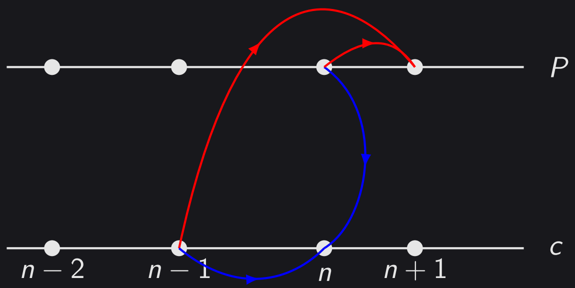
Parallel time integration?

- One approach: identify task parallelism
  Example: Diagonal / block Diagonal RK methods:
  Concurrently evaluate stages.
- Another example (less obvious)
  predictor–corrector schemes (Miranker, Liniger 1967)



Michigan Tech
Create the Future

Example: Miranker and Liniger



$$u^p_{n+1} = u^c_{n-1} + 2h\, f(t_n, u^p_n)$$

$$u^c_n = u^c_{n-1} + \frac{h}{2}\left(f(t_n, u^p_n) + f(t_{n-1}, u^c_{n-1})\right)$$

Michigan Tech
Create the Future

Table Activity

Switching gears:

- What is a multigrid method for solving $Ax = b$?
- How/why does it work?

Table Activity

Ben's definition:

- A multigrid method seeks to accelerate iterative approaches for solving $Ax = b$.
- Key idea: iterative methods (often) remove high-frequency, so
  - use a hierarchy of grids: map the low-frequency error to a coarser grid (low-frequency error looks like high-frequency on coarse grid)
  - solve for a correction on a coarse grid
  - interpolate and correct fine-grid solution.

MultiGrid Reduction In Time (MGRIT)

~~Identify potential task parallelism~~

Build in parallelism capability by using parallel smoothers + multigrid

Assemble all-at-once system (using operators)

$$Au = \begin{bmatrix} I & & & \\ -\Phi & I & & \\ & \ddots & \ddots & \\ & & -\Phi & I \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix} = f$$

MultiGrid Reduction In Time (MGRIT)

~~Identify potential task parallelism~~

Build in parallelism capability by using parallel smoothers + multigrid

Assemble all-at-once system (using operators)

$$Au = \begin{bmatrix} I & & & \\ -\Phi & I & & \\ & \ddots & \ddots & \\ & & -\Phi & I \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix} = f$$

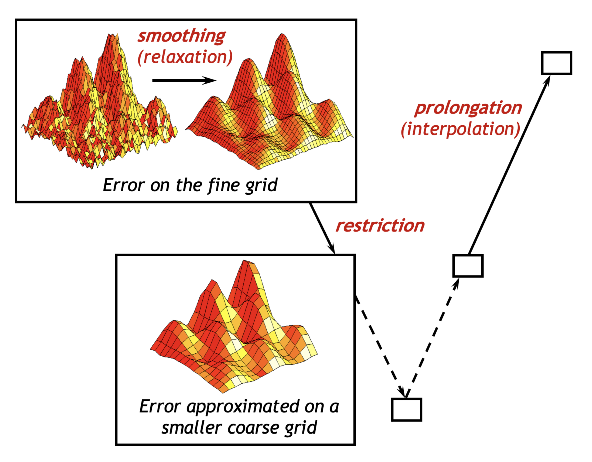
Michigan Tech
Create the Future

(Geometric) Multigrid

Idea:

- given: (iterative) smoother that reduces high-frequency error
- restrict residual to a coarser mesh
  (low-frequency error looks like high frequency error)
- apply smoother to estimate error on coarse mesh
- correct solution on fine mesh.

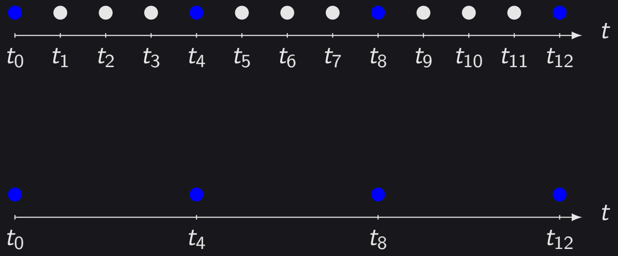
Cartoon


smoothing
(relaxation)
Error on the fine grid
prolongation
(interpolation)
restriction
Error approximated on a
smaller coarse grid

Two-level Reduction Scheme

Split variables into two sets:

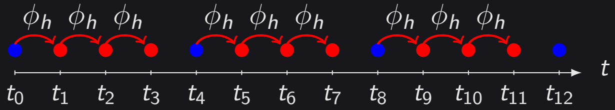
- blue: variables on coarse grid
- white: variables on fine grid



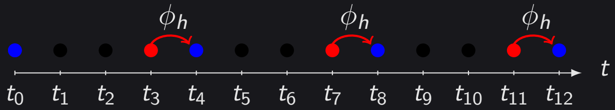
Two-level Reduction Scheme

Smoother.

Simplest: F-relaxation:

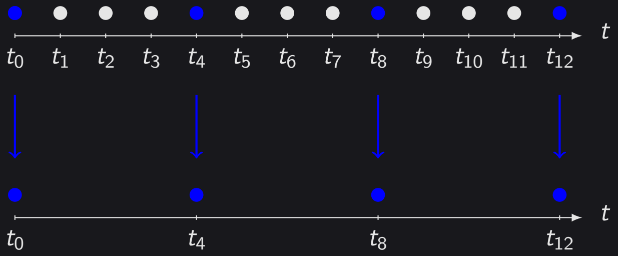


Also popular: $F(CF)^{\nu}$, where
C-relaxation:

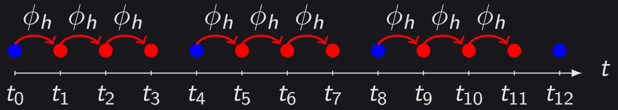


Two-level Reduction Scheme

Restriction: use injection



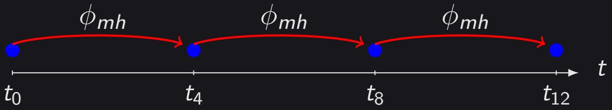
Prolongation (F-relaxation)



Two-level Reduction Scheme

Coarse Problem

Rediscretize: solve original/error ODE with same integrator, larger time step mh.

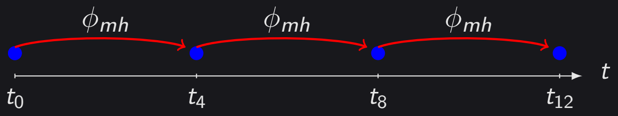


- Surprisingly: active research to specify coarse problem
- Reason: for convection dominated problems, MGRIT convergence degrades (even diverges!)
- My reason: improve accuracy

Two-level Reduction Scheme

Coarse Problem

Rediscretize: solve original/error ODE with same integrator, larger time step mh.

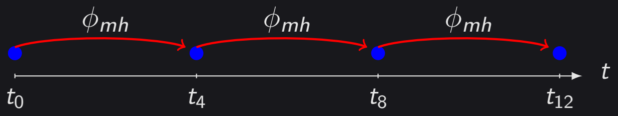


- Surprisingly: active research to specify coarse problem
- Reason: for convection dominated problems, MGRIT convergence degrades (even diverges!)
- My reason: improve accuracy

Two-level Reduction Scheme

Coarse Problem

Rediscretize: solve original/error ODE with same integrator, larger time step mh.



- Surprisingly: active research to specify coarse problem
- Reason: for convection dominated problems, MGRIT convergence degrades (even diverges!)
- My reason: improve accuracy

Michigan Tech
Create the Future

Multi-Grid Reduction In Time (MGRIT)

MGRIT:

- Develop multilevel algorithm from this viewpoint
- Fridhoff, Falgout, Kolev, MacLachlan, Schroder (2013)
- Space–time coarsening may be needed to satisfy CFL
- Time Parallelism gives multiplicative scalability
- Established software package (XBraid) designed to work with PETSc and Sundials
- Very elegant: user-specified sequential integrator, Φ
  XBraid wraps multigrid structure $\rightarrow$ PinT

Multi-Grid Reduction In Time (MGRIT)

But . . .

- Although excellent parallel scalability (with minimal communication), generally low efficiency.
- e.g., 100x processors for time parallelization
  converges to sequential time integrator solution ∼5x walltime speedup
- probably best used to extend scalability once spatial parallelism is exhausted

Our Goal:

- provide additional motivation to consider MGRIT
- modify error/residual equation to obtain parallel high-order solution

Michigan Tech
Create the Future

Multi-Grid Reduction In Time (MGRIT)

But . . .

- Although excellent parallel scalability (with minimal communication), generally low efficiency.
- e.g., 100x processors for time parallelization
  converges to sequential time integrator solution ∼5x walltime speedup
- probably best used to extend scalability once spatial parallelism is exhausted

Our Goal:

- provide additional motivation to consider MGRIT
- modify error/residual equation to obtain parallel high-order solution

Deferred Correction

Idea: improve accuracy of inaccurate discrete approximation. (Ong & Spiteri, 2020)

$$\{u_n\}, \quad n = 1, \ldots N$$

Let

- $u(t)$: continuous extension of $\{u_n\}$,
- $U(t)$: true (unknown) solution to DE; $U'(t) = f(t, U)$

Want: approximation to error, $e(t)$,

$$e(t) = U(t) - u(t)$$

Differentiating both sides:

$$\begin{aligned} e'(t) &= U'(t) - u'(t) \\ &= f(t, U) - u'(t) \\ &= f(t, u + e) - u'(t) \end{aligned}$$

Deferred Correction

Define residual:

$$r(t) = u'(t) - f(t, u)$$

(Differential form) Error IVP:

$$e'(t) = f(t, u + e) - f(t, u) - r(t), \qquad e(0) = 0$$

(Integral form) Error IVP:

$$\left[e(t) + \int_0^t r(\tau)\, d\tau \right]' = f(t, u + e) - f(t, u), \qquad e(0) = 0$$

- Discretize using user-provided integrator, Φ + quadrature/interpolation using $\{u_n\}$
- Notation: $\psi \leftarrow \Phi$ + quadrature/interpolation

MGRIT-DC method

Our algorithm:

- On finest level, relax IVP, à la MGRIT (using Φ)
- On coarsest level, solve error equation (using ψ)
- On intermediate levels, relax error equation (using ψ)
- Prolongation operator (F-relaxation) replaced

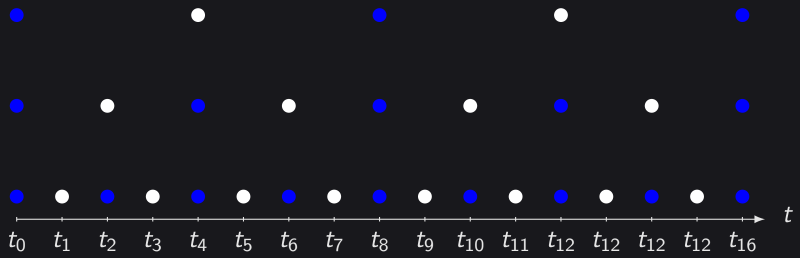
Note:

- couples multiple levels together: (approximate integral of residual from finer level)
- Increased communication

Michigan Tech
Create the Future

Three-level cartoon

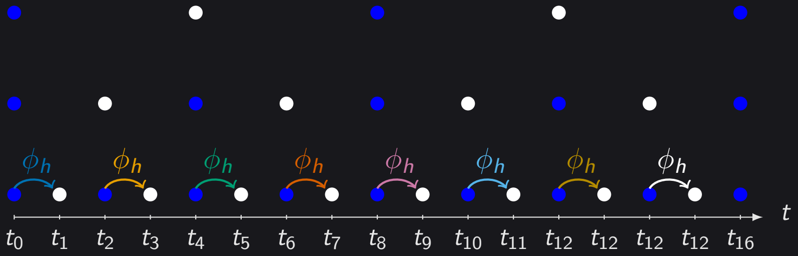
Grid Hierarchy (coarsening factor, $m = 2$)



Michigan Tech
Create the Future

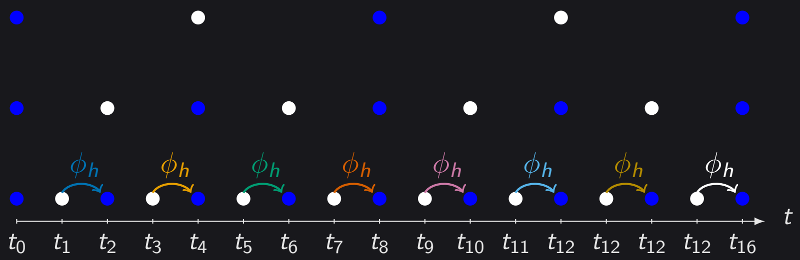
Three-level cartoon

Relax ODE on Fine Grid (F-relaxation, in parallel)



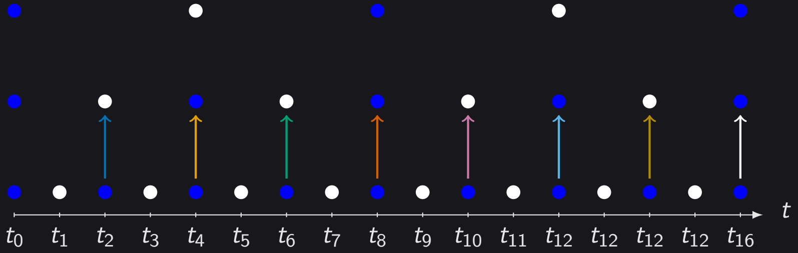
Three-level cartoon

Compute residual (in parallel)



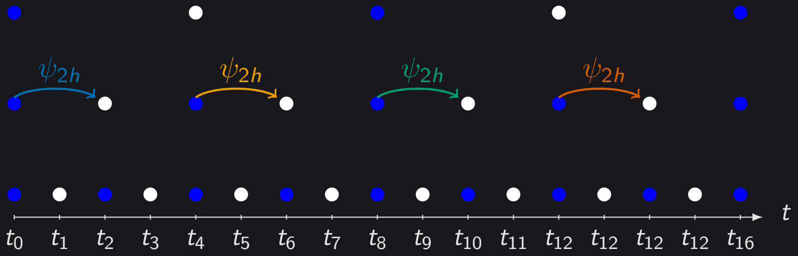
Three-level cartoon

Restrict to coarser mesh



Three-level cartoon

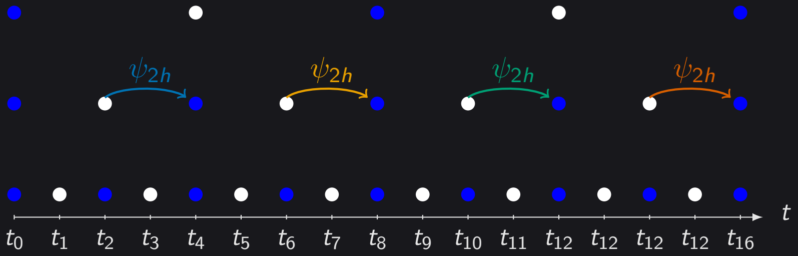
Relax Error IVP on Coarser Grid (F-relaxation, in parallel)



Michigan Tech
Create the Future

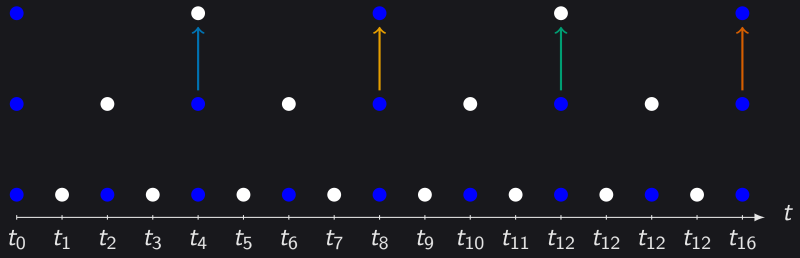
Three-level cartoon

Compute residual (in parallel)



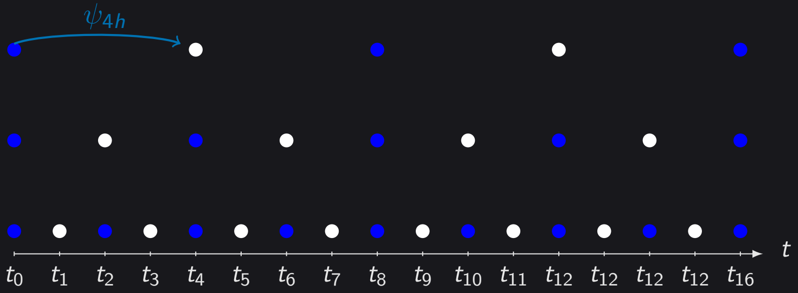
Three-level cartoon

Restrict to coarsest mesh



Three-level cartoon

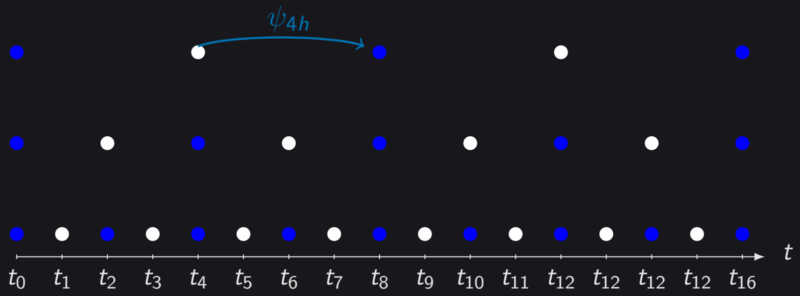
(Sequential) Solve on coarsest mesh



Michigan Tech
Create the Future

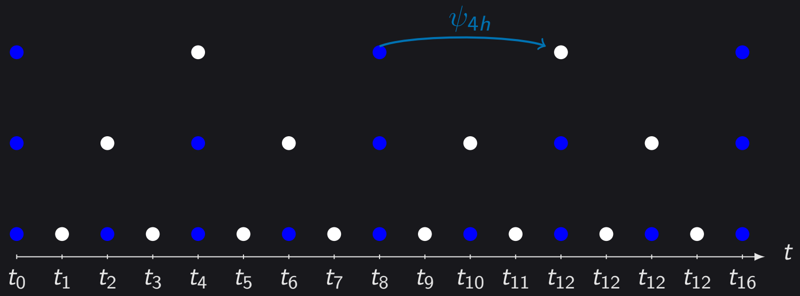
Three-level cartoon

(Sequential) Solve on coarsest mesh



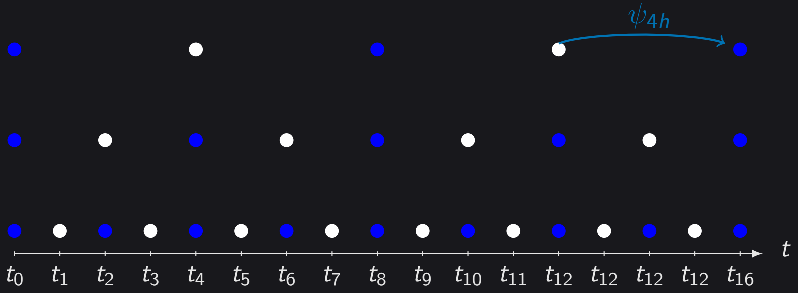
Three-level cartoon

(Sequential) Solve on coarsest mesh



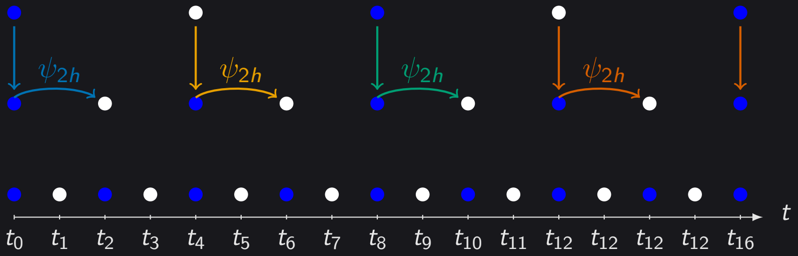
Three-level cartoon

(Sequential) Solve on coarsest mesh



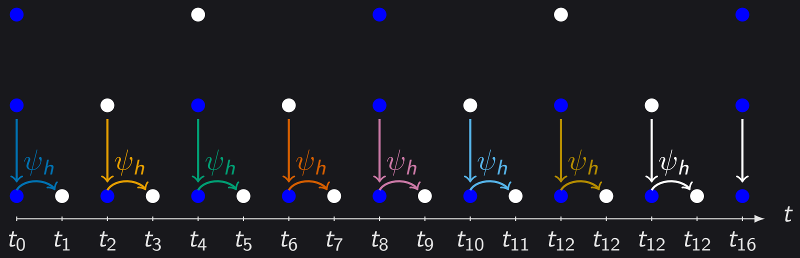
Three-level cartoon

Prolongate (in parallel) to finer mesh



Three-level cartoon

Prolongate (in parallel) to finest mesh



- This step is not needed for MGRIT iteration
- Needed only if high-order fine-solution needed at end of MGRIT iteration

Advection–Diffusion

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x} + \alpha \frac{\partial^2 u}{\partial x^2}$$
$$u(t,0) = u(t,1)$$

- Method of lines:
  - upwind first-order differencing for advective term
  - centered second-order differencing for diffusive term
- first-order IMEX scheme

$$u^{n+1} = u^n + h\left(D_x u^n + \alpha D_{xx} u^{n+1}\right)$$

Michigan Tech
Create the Future

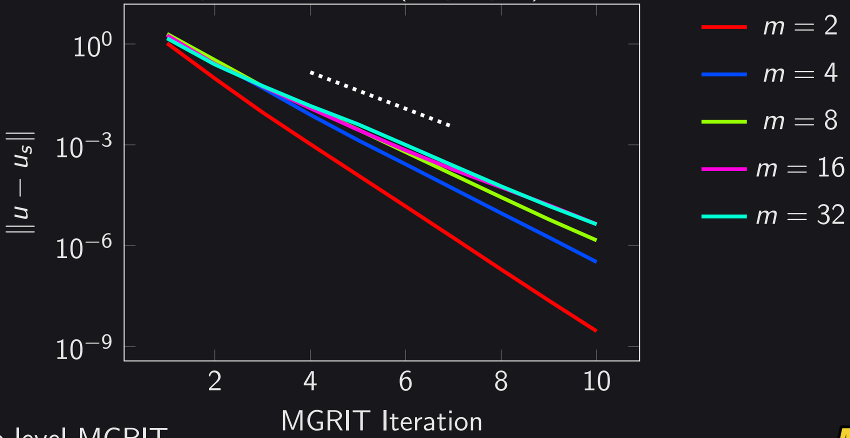
Notation

- U: analytic solution
- u_s: single grid (sequential) integrator (Φ)
- u: multigrid solution

Michigan Tech
Create the Future

Advection–Diffusion

Classical MGRIT converges to reference (single grid) solution



- two-level MGRIT
- Here, $N = 2^{10}$

Michigan Tech
Create the Future

Rate Estimates

Theoretical Bound (Doborev et. al, 2016)

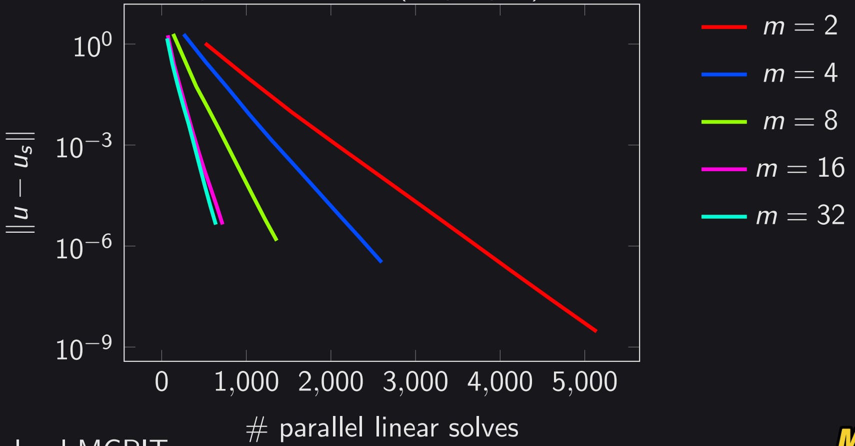
$$\rho \leq \max_{\omega} |\lambda_\omega^m - \mu_\omega| \left(\frac{1 - |\mu_\omega|^{N/m}}{1 - |\mu_\omega|} \right),$$

- m: coarsening factor
- λ_ω: eigenvalues of Φ_h
- μ_ω: eigenvalues of Φ_{mh}

Michigan Tech
Create the Future

Advection–Diffusion

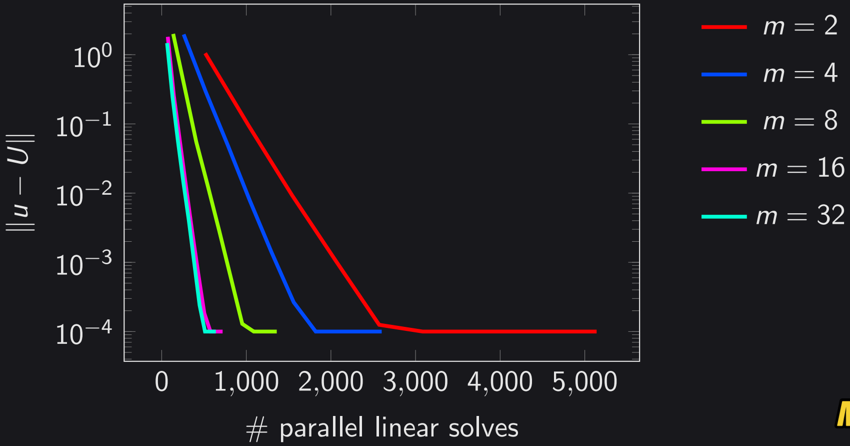
Classical MGRIT converges to reference (single grid) solution



- two-level MGRIT
- Here, $N = 2^{10}$

Advection–Diffusion

Computing error wrt analytic solution ...



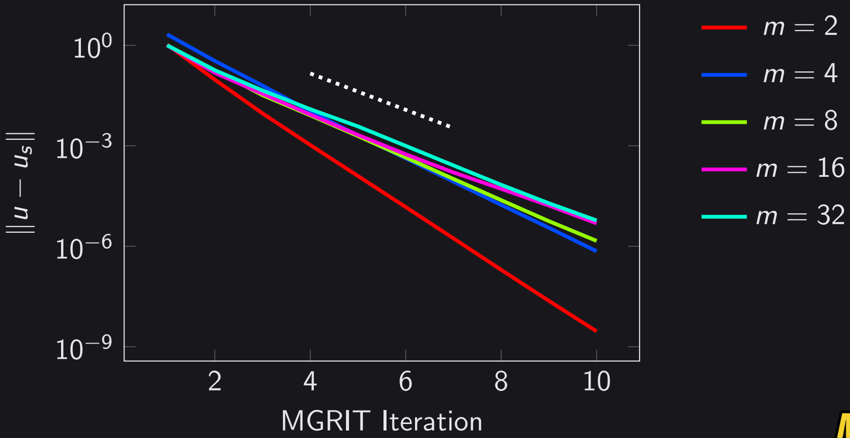
Notation

To discuss MGRIT-DC

- U: analytic solution
- u_s: single grid (sequential) integrator (Φ)
- $\hat{u}_s$: single grid (sequential) DC integrator
- u: multigrid solution

Advection–Diffusion

There is a slight degradation in convergence rate



Michigan Tech
Create the Future

Rate Estimates

Implementation details not discussed (e.g., estimating quadrature)

If Newton–Cotes used for quadature, then theoretical bound:

$$\rho \leq \max_{\omega} \left| \mu_\omega \, \gamma_\omega \left(\sum_{i=0}^{m-1} w_i \lambda_\omega^i + \widetilde{w}_m \lambda_\omega^m \right) \right| \left(\frac{1 - |\mu_\omega|^{N/m}}{1 - |\mu_\omega|} \right).$$

Interesting side project that arose from implementation exploration: quadrature rules from least-squares polynomial fits.

Rate Estimates

Implementation details not discussed (e.g., estimating quadrature)

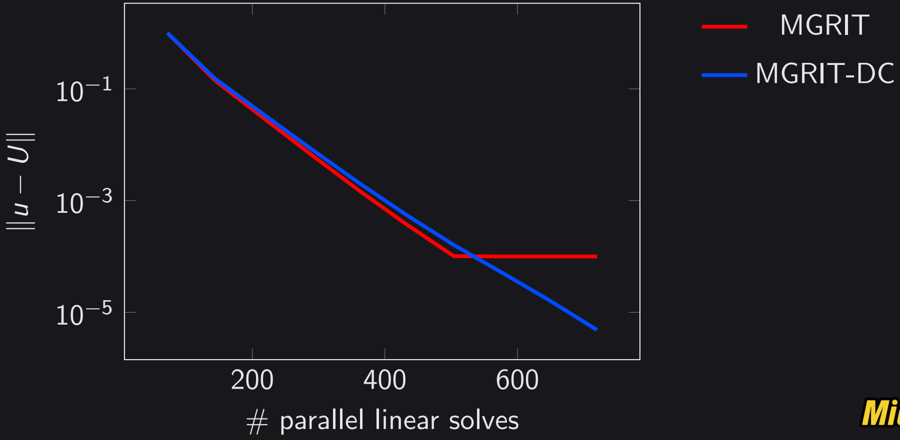
If Newton–Cotes used for quadature, then theoretical bound:

$$\rho \leq \max_{\omega} \left| \mu_\omega \, \gamma_\omega \left(\sum_{i=0}^{m-1} w_i \lambda_\omega^i + \widetilde{w}_m \lambda_\omega^m \right) \right| \left(\frac{1 - |\mu_\omega|^{N/m}}{1 - |\mu_\omega|} \right) .$$

Interesting side project that arose from implementation exploration: quadrature rules from least-squares polynomial fits.

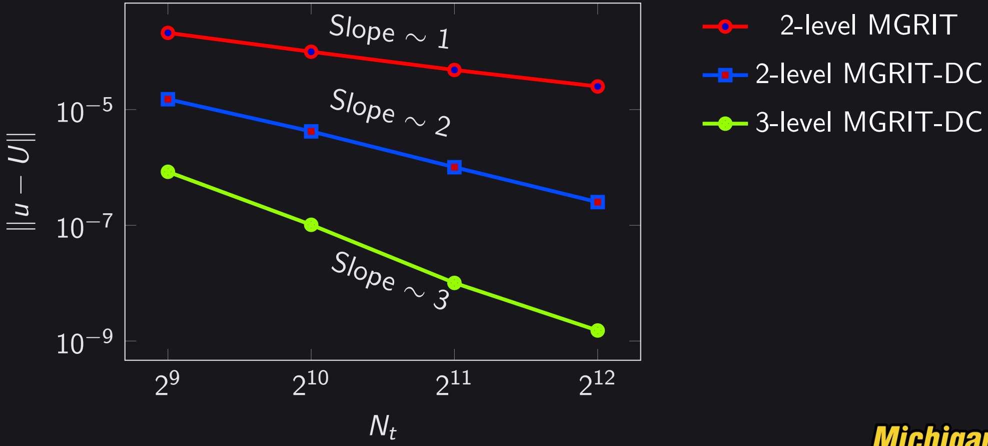
MGRIT-DC vs MGRIT

- $m = 16$



MGRIT-DC vs MGRIT

- Can bootstrap: each level adds improves order by one



Michigan Tech
Create the Future

Future Work

- IDC analysis extends to justify increase in order of accuracy
- Didn't do an work–precision plot yet.
  Not sure how to fairly stop MGRIT iterations if U unknown.
- Didn't compare MGRIT-DC to a high-order split scheme.
- In a non-MGRIT setting, compared
  - a sequential Yoshida 9-6 scheme w/ 6th order SDIRK scheme
  - a 6th order RIDC–IMEX scheme,

  gives $\sim 100\times$ speedup. (Joint work with H. Johansen, LBNL)

Michigan Tech
Create the Future