

University of New Mexico

UNM Digital Repository

Mathematics & Statistics ETDs

Electronic Theses and Dissertations

Spring 5-15-2020

Multilevel Asymptotic Parallel-in-Time Techniques For Temporally Oscillatory PDEs

Nicholas Abel

University of New Mexico - Main Campus

Follow this and additional works at: https://digitalrepository.unm.edu/math_etds



Part of the [Applied Mathematics Commons](#), [Mathematics Commons](#), and the [Statistics and Probability Commons](#)

Recommended Citation

Abel, Nicholas. "Multilevel Asymptotic Parallel-in-Time Techniques For Temporally Oscillatory PDEs." (2020). https://digitalrepository.unm.edu/math_etds/173

This Thesis is brought to you for free and open access by the Electronic Theses and Dissertations at UNM Digital Repository. It has been accepted for inclusion in Mathematics & Statistics ETDs by an authorized administrator of UNM Digital Repository. For more information, please contact disc@unm.edu.

Nicholas Abel

Candidate

Mathematics and Statistics

Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

Jacob Bayer Schroder

, Chairperson

Jehanzeb Hameed Chaudhry

Rob Falgout

Deborah Sulsky

Multilevel Asymptotic Parallel-in-Time Techniques For Temporally Oscillatory PDEs

by

Nicholas Theodore Abel

B.S., Mathematics, University of New Mexico, 2017

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Mathematics

The University of New Mexico

Albuquerque, New Mexico

July 2020

Dedication

*This one is dedicated to my family, my friends, and my pets, who I am indebted to
for their support during this wild journey.*

Acknowledgments

I would like to express deep gratitude to the people who have helped me along the way and extended me opportunities to work as a researcher: Jacob Schroder, Jehanzeb Chaudhry, Rob Falgout, Deborah Sulsky, Patrick Bridges, Bob Robey, Jon Reisner, and others.

Many thanks to Adam Peddle, Terry Haut, Beth Wingate for their helpful correspondence, and to my fellow students Quincy Wofford, Ari Rappaport, and others for their feedback and conversations that helped me in the process of research and thesis writing.

Multilevel Asymptotic Parallel-in-Time Techniques For Temporally Oscillatory PDEs

by

Nicholas Theodore Abel

B.S., Mathematics, University of New Mexico, 2017

M.S., Mathematics, University of New Mexico, 2020

Abstract

As the clock speeds of individual processors level off and the amount of parallel resources continue to increase rapidly, further exploitation of parallelism is necessary to improve compute times. For time-dependent differential equations, the serial computation of time-stepping presents a bottleneck, but parallel-in-time integration methods offer a way to compute the solution in parallel along the time domain. Parallel-in-time methods have been successful in achieving speedup when computing solutions for parabolic problems; however, for problems with large hyperbolic terms and no strong diffusivity, parallel-in-time methods have traditionally struggled to offer speedup. While work has been done to understand why parallel-in-time methods struggle to converge quickly for hyperbolic problems, a few parallel-in-time techniques have been demonstrated to achieve speedup for certain hyperbolic problems. We consider a previously proposed technique based on parareal, which is a general parallel-in-time method that uses a relatively cheap coarse-grid approximation to compute error corrections to accelerate the solution of a fine-grid time-marching

problem. In particular, we look at a method which constructs an asymptotically time-averaged approximation on the parareal coarse grid, which has been shown to work well when solving hyperbolic problems whose solutions exhibit fast oscillations in the time dimension. Using the generalizability of the parareal method into the multigrid-reduction-in-time (MGRIT) algorithm, we investigate the expansion of the two-grid asymptotic parareal method to a multilevel MGRIT setting. In particular, we research runtime improvements when rapid oscillations are present by using the multilevel capabilities and FCF-relaxation smoothing aspects of MGRIT. Methods to improve compute speed in flow regimes without fast temporal oscillations are also examined.

Contents

List of Figures	ix
List of Tables	xii
Glossary	xv
1 Introduction	1
1.1 Overview	1
1.2 Summary of Contributions	4
2 MGRIT algorithm	6
3 Asymptotic Parallel-in-Time Methods	12
4 Test Problem - Rotating Shallow Water Equations	17
4.1 Implementation	22
5 Research Objectives	23

Contents

6	Numerical Results	25
6.1	Problem Setup	25
6.2	Code Verification	27
6.3	Multilevel Asymptotic MGRIT	30
6.4	Asymptotic MGRIT: Two-Level With FCF-Relaxation	34
6.4.1	Investigating the Wall-Time Discrepancy for FCF-relaxation	35
6.5	Accelerating $\bar{\mathbf{u}}$ Computations by Reusing Values	38
6.5.1	Reuse of Stale Coarse Grid Solutions $\bar{\mathbf{u}}$	39
6.5.2	Computation of $\bar{\mathbf{u}}$ Based on Stale Nonlinear Quantities $\bar{\mathcal{N}}()$	41
6.6	Three-Scale RSWE	44
6.7	The $\epsilon = 1$ Case: On the Use of CF-relaxation Sweeps and Larger Averaging Values η to Allow For More Efficient Coarse Grids	51
7	Conclusions	57
8	Future Work	61
	References	63

List of Figures

2.1	Illustration of a temporal grid decomposed into a fine and coarse grid, uniformly spaced. F-points are only present on the fine grid, and C-points are present on both the fine and coarse grid.	9
2.2	Illustration of F- and C-relaxation. C-points are denoted with T_j, T_{j+1} and F-points are denoted with t_{i+1}, t_{i+2}, \dots	10
4.1	Wave-height solution profile of h in a space-time domain $X = [0, 2\pi], T = [0, 5]$ with $N_x = 64$ Fourier modes in space, $N_\Delta = 50$ time points, various ϵ values, and a fixed $\mu = .0001$. Here, time evolves in the x -axis and the y -axis is the spatial dimension. For small values of ϵ , the magnitude of \mathcal{L} is large and oscillations in the direction of the time domain are rapid. As ϵ becomes large, the magnitude of the $1/\epsilon \mathcal{L}$ term shrinks and oscillations occur over much larger timescale. In particular, when $\epsilon = 10$ and $\epsilon = 100$, a complete temporal oscillation is no longer present because oscillations occur on an $\mathcal{O}(\epsilon)$ timescale. See subfigure titles for specific ϵ, μ values.	20

List of Figures

4.2 Wave-height solution profile of h in a space-time domain $X = [0, 2\pi], T = [0, 5]$ with $N_x = 64$ Fourier modes in space, $N_\Delta = 50$ time points, various μ values, and a fixed $\epsilon = .1$. Here, time evolves in the x -axis and the y -axis is the spatial dimension. As μ increases, so does the magnitude of the dissipation term. The result is that we see the effects of dissipation become prominent at ever earlier times. See subfigure titles for specific ϵ, μ values. 21

6.1 Comparison of residual histories for two-level asymptotic MGRIT solves using various schemes to reuse the slowly-varying coarse grid solution $\bar{\mathbf{u}}$ where $\epsilon = 0.01$. A coarse grid of $N_\Delta = 50$, a fine grid of $N = 50,000$, final time $t_f = 5.0$ and F-relaxation are used. 40

6.2 Comparison of residual histories for two-level asymptotic MGRIT solves using various schemes to reuse the slowly-varying coarse grid solution $\bar{\mathbf{u}}$ where $\epsilon = 1$. A coarse grid of $N_\Delta = 50$, a fine grid of $N = 50,000$, final time $t_f = 5.0$ and F-relaxation are used. 41

6.3 Comparison of residual norm histories for two-level asymptotic MGRIT solves using various techniques to recompute the coarse grid solution $\bar{\mathbf{u}}$ based on stale nonlinear quantities $\bar{\mathcal{N}}()$, where $\epsilon = 1$. A coarse grid of $N_\Delta = 50$, a fine grid of $N = 50,000$, final time $t_f = 5.0$ and F-relaxation are used. 42

6.4 Comparison of residual norm histories for two-level asymptotic MGRIT solves using various techniques to recompute the coarse grid solution $\bar{\mathbf{u}}$ based on stale nonlinear quantities $\bar{\mathcal{N}}()$, where $\epsilon = 0.01$. A coarse grid of $N_\Delta = 50$, a fine grid of $N = 50,000$, final time $t_f = 5.0$ and F-relaxation are used. 43

6.5 $F = 10^4, \epsilon = 1, T = [0, 5],$ fine grid $N = 50000,$ coarse grid $N_\Delta = 50$ 46

List of Figures

6.6	$F = 1, \epsilon = 1, T = [0, 5]$, fine grid $N = 50000$, coarse grid $N_{\Delta} = 50$.	47
6.7	Braid residual after 2, 3, 4, and 5 iterations for $\epsilon = 1, F = 10^{-4}$, and F-relaxation. Choice of η is given on the x -axis and residual norm measured after the indicated number of iterations is given on the y -axis. Note that the coarse grid $\Delta T = 0.1$	48
6.8	Braid residual after 2, 3, 4, and 5 iterations for $\epsilon = 1, F = 10^{-4}$, and FCF-relaxation. Choice of η is given on the x -axis and residual norm measured after the indicated number of iterations is given on the y -axis. Note that the coarse grid $\Delta T = 0.1$	49

List of Tables

6.1	Comparison of per-iteration histories of the XBraid residual norm and the L^∞ -norms when performing asymptotic parareal solves with XBraid and Cyclops respectively, for the 1-dimensional RSWE on a time domain of $[0, 1]$	28
6.2	Comparison of per-iteration histories of the XBraid residual norm and the L^∞ -norms when performing asymptotic parareal solves with XBraid and Cyclops respectively, for the 1-dimensional RSWE on a time domain of $[0, 3]$	29
6.3	$\epsilon = 1$ comparison of three-level MGRIT solvers using either a “Strang-asymptotic-asymptotic” (SAA) or “Strang-asymptotic-Strang” (SAS) method of adding a third grid to a two-level asymptotic MGRIT solve with the structure $\{4096_S, 64_A\}$. F-relaxation is used. Runs that fail to converge are marked with “**”. Columns are ordered by the number of coarse points N_3 on the third grid, except for the rightmost column which provides the reference iteration count and wall time for the two-level $\{4096_S, 64_A\}$ solve.	33

List of Tables

6.4 $\epsilon = .01$ comparison of three-level MGRIT solvers using either a “Strang-asymptotic-asymptotic” (SAA) or “Strang-asymptotic-Strang” (SAS) method of adding a third grid to a two-level asymptotic MGRIT solve with the structure $\{4096_S, 64_A\}$. F-relaxation is used. Runs that fail to converge are marked with “***”. Columns are ordered by the number of coarse points N_3 on the third grid, except for the rightmost column which provides the reference iteration count and wall time for the two-level $\{4096_S, 64_A\}$ solve. 33

6.5 Iteration counts for 2-level asymptotic MGRIT runs, $\epsilon = .01$, using the coarsening factors m and relaxation schemes provided in the top row of each column. These results are “strange” when compared with the wall times in Table 6.6, in that we do not generally expect improvements in wall time from FCF-relaxation unless we are cutting down on iteration counts far more than in half when compared to F-relaxation iteration counts. Runs that fail to converge are marked with “***”. 35

6.6 Wall times (using 16 cores) for asymptotic MGRIT runs, $\epsilon = .01$, using the coarsening factors m and relaxation schemes provided in the top row of each column. These results are “strange” when compared with the iteration counts in Table 6.5, in that we do not generally expect improvements in wall time from FCF-relaxation, unless we are cutting down on iteration counts far more than 50% when compared to F-relaxation iteration counts. Runs that fail to converge are marked with “***”. 35

List of Tables

6.7 Average wall times on levels 0, 1 for $\overline{\mathcal{N}}()$ reuse. The reuse pattern follows using $\overline{\mathcal{N}}()$ terms from the prior iteration at iterations $j = \{1, 3, 5, 7, 9\}$ over a Braid run with iterations $j = 0, 1, 2, \dots, 10$. As with the other experiments in this section, F-relaxation is used as indicated by the f subscripts. Here, $\mathcal{N}_\Delta = 50$ and $t_f = 5.0$ 45

6.8 Braid residuals after 4 and 8 iterations for various two- and three-grid techniques when $F = 10^{-4}$ and $\epsilon = 1$ 50

6.9 Iteration counts for two-level asymptotic MGRIT solves of the RSWE. Here, the averaging window η and quadrature points are chosen using Equations (6.1) and (6.2), where α is given in the upper-left cell of each table and $\beta = 1$. The value $\epsilon = 1$ is used corresponding to no scale separation and the time grid size is fixed at $N_\Delta = 50$, $N = 50000$. Columns are ordered as the final time t_f varies from 1.0 to 20.0 as given by the top row. Rows are ordered by number of CF-relaxations applied in smoothing i.e., $F(CF)^n$ -relaxation where the left-most column gives n for each row. Runs that fail to converge are marked with “**”. 55

6.10 MPI 16-core wall clock times for two-level asymptotic MGRIT solves of the RSWE. Here, the averaging window η and quadrature points are chosen using Equations (6.1) and (6.2), where α is given in the upper-left cell of each table and $\beta = 1$. The value $\epsilon = 1$ is used corresponding to no scale separation and the time grid size is fixed at $N_\Delta = 50$, $N = 50000$. Columns are ordered as the final time t_f varies from 1.0 to 20.0 as given by the top row. Rows are ordered by number of CF-relaxations applied in smoothing i.e., $F(CF)^n$ -relaxation where the left-most column gives n for each row. Runs that fail to converge are marked with “**”. 56

Glossary

APinT	Asymptotic Parallel-in-Time; parallel-in-time solver that uses a solution to a slowly-varying asymptotic approximation to the problem on the coarse grid(s).
HMM	Heterogeneous Multiscale Method; APinT uses a type of HMM to obtain the slowly-varying coarse-grid equations.
MGRIT	The multigrid reduction-in-time method.
η	The size of the time-averaging window.
ϵ	The Rossby number, representing the ratio of gravity wave timescale to Rossby wave timescale.
Fr	The Froude number, defined as $F^{1/2}\epsilon$, representing the ratio of fluid characteristic advection velocity to gravity wave speed.
μ	The coefficient for the dissipation operator. A larger μ encodes stronger dissipation in the problem.

Chapter 1

Introduction

1.1 Overview

In recent years, the increase in the clock speeds of individual processors has begun to slow to a halt. As this is due to physical constraints, this is a trend that we can expect will continue in the future [22]. Moreover, the available parallelism continues to rapidly increase. This means that in order to continue making improvements on computation speeds, exploiting parallelism is necessary.

In the instance of computing solutions to time-dependent PDEs, parallelism in the spatial component of the problem is already widespread and for many applications, the amount of parallel computing resources available has led to a saturation of parallel speedup available from spatial parallelism [15]. Parallel-in-time methods are a class of numerical techniques that enable parallelism in the time dimension, addressing the issue of new sources of parallelism being needed. A historical overview of parallel-in-time methods is discussed in [15], and a list of many recent application areas is discussed in [37], e.g., parabolic problems, Navier-Stokes, power-grid, machine learning, and so on.

Chapter 1. Introduction

Here, the parallel-in-time algorithms we consider are the multigrid-reduction-in-time (MGRIT) method [13, 10, 5] and the related parareal method [24, 16]. Parareal and MGRIT are multilevel solvers that decompose the time domain of a problem into subdomains over which the solution is iteratively improved using error correction terms which come from inexpensive coarse grid approximations to the problem. Each subdomain can be updated using a local coarse grid correction term simultaneously, hence they are parallel-in-time methods. In contrast to serial timestepping methods which solve the space-time equations from an initial time to a desired final time in serial, parareal and MGRIT improve an initial solution guess across the entire space-time domain at each iteration.

Parallel-in-time techniques have been very successful in speeding up the computation of parabolic problems; however, for hyperbolic problems that are advection-dominated, the performance of parallel-in-time methods deteriorates greatly when using traditional choices for the coarse-grid approximation. This phenomenon is documented in the parareal setting for the linear advection equation and second-order wave equation [14, 34], as well as the Navier-Stokes equation, where convergence improves as the Reynolds number becomes small, making the problem more diffusive [38]. Similarly, MGRIT has been shown to struggle with the linear advection and inviscid Burgers equations, likewise improving with the introduction of diffusivity terms [20, 5, 4, 39].

Recently, mathematical analysis of why hyperbolic problems are difficult for parallel time integration using standard coarse-grid techniques has been explored. Ruprecht shows in [34] that under the standard parareal method, instabilities for the linear advection equation are due to parareal overestimating wave amplitudes associated with middle to high wave number modes, which is due to phase errors in the coarse grid representation. While this behavior is also present when solving strongly parabolic problems in parareal, the introduction of a diffusive term into

Chapter 1. Introduction

the linear advection equation prevents these instabilities by strongly damping the amplitudes of higher wave numbers. This allows parareal room to overshoot the correct amplitudes without causing instabilities. Furthermore, it is suggested that the construction of coarse grid operators designed such that they do not contribute to overamplification of high wave modes may be a viable technique for achieving speedup for hyperbolic problems by parallel-in-time integration.

Much work has been done investigating alternative coarse grid discretizations for hyperbolic problems for use with parareal and MGRIT. In [39, 21], a technique that constructs a coarse grid propagator by approximating the spectrum of the ideal MGRIT coarse grid operator using a weighted least squares approximation is proposed and investigated for the linear advection equation. The use of a semi-Lagrangian scheme as a parareal coarse operator is shown in [36] to converge for the viscous Burgers equation and offer theoretical speedup as viscosity goes to 0. Nielsen et al. [28] use Roe’s method, an approximate Riemann solver, on the coarse grid combined with a third order WENO scheme on the fine grid for the spatial discretization and third-order Strong-Stability-Preserving Explicit Runge-Kutta time integration to achieve convergence for the two dimensional shallow water equations. However, in all these cases, the convergence of parareal and MGRIT is either slow for strongly hyperbolic problems, or the method is yet to be generalized to variable coefficient of nonlinear problems [39, 21].

In contrast, asymptotic parareal [18, 32, 31] is a parallel-in-time method designed for strongly hyperbolic problems with fast temporal oscillations due to a linear term and slow characteristic advection due to a nonlinear term, a kind of problem common in geophysical fluid dynamics. Asymptotic parareal achieves superior convergence to standard parareal in highly oscillatory flow regimes and is roughly equivalent in performance to standard parareal in non-oscillatory flow regimes. In this work, we research improvements to the asymptotic parareal method when used with the

MGRIT algorithm.

1.2 Summary of Contributions

Here, I give a summary of my contributions pertaining to the work in this paper.

Our main research focus is to investigate asymptotic MGRIT, which is the asymptotic parareal method generalized to the MGRIT setting. Use of FCF-relaxation and multilevel schemes are examined. In particular, FCF-relaxation is found to improve performance for solves where the coarse grid timestep ΔT is relatively large. We study the optimal number of FCF-relaxations in this setting, noting that FCF-relaxation has not been used before with asymptotic parareal-based methods, and that in some cases, the savings are substantial.

Regarding multilevel schemes, we seek an answer to the question, whether additional levels in the asymptotic parareal method can be beneficial. In other words, can we target different scales of the problem on specific coarse grids, with the result that a problem with multiple scales will correspond to a solver with multiple levels. The results here are mixed, with only some evidence that multilevel can be beneficial.

Our last research objective is to explore data reuse in the asymptotic MGRIT method, where the dominant cost of computing a time-average allows for data reuse in some MGRIT iterations. That is, can we use time-averaged information from iteration k in iteration $k + 1$, and can we use time-averaged information from level ℓ on the next coarser level $\ell + 1$?

There are also some other worthwhile, but more minor discoveries discussed. In particular, we study how FCF-relaxation and the use of longer time-average windows (which is a part of the asymptotic parareal time-stepping process) impact the ability to coarsen in time.

Chapter 1. Introduction

Regarding software contributions, a Cython interface between XBraid, our chosen MGRIT software, and Cyclops, a Python numerical code implementing asymptotic parareal with a Fourier spectral discretization in space, was developed under the working name CycloBraid. This Cython interface also now serves as a template for coupling XBraid and other Python codes. For instance, the interface is already impacting progress in machine learning research at Sandia National Laboratories where it has been repurposed to create an interface between PyTorch [29] and XBraid to study layer-parallel training of neural networks. More information on layer-parallel neural network training can be found in [17].

For more details on the research objectives, see Chapter 5.

Chapter 2

MGRIT algorithm

MGRIT is an algorithm for parallel time integration that applies the technique of multigrid reduction (MGR) [33] to iteratively compute the solution of a time-dependent differential equation.

It is intuitive to introduce MGRIT in the context of a linear system of ODEs, which we do here in a two-level setting. For compatibility with general nonlinear problems, MGRIT uses nonlinear full approximation scheme (FAS) [2] multigrid to restructure the coarse grid. To generalize MGRIT to multilevel, apply the following two-level MGRIT process recursively.

Consider an ODE system

$$\mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)), \mathbf{u}(0) = \mathbf{u}_0, t \in [0, T], \quad (2.1)$$

with a time discretization represented by the points

$$t_i = i\Delta t, \quad i = 0, 1, \dots, N, \quad (2.2)$$

where $\Delta t = \frac{T}{N}$, and $u_i \approx \mathbf{u}(t_i)$. Using a one-step method, the discretization takes

Chapter 2. MGRIT algorithm

the form

$$\begin{aligned} u_0 &= g_0, \\ u_i &= \Phi_i(u_{i-1}) + g_i, \quad i = 1, 2, \dots, N, \end{aligned} \tag{2.3}$$

where g_i is a forcing term evaluated at timestep i and Φ_i is the timestepping operator which evolves the state variable u from timestep $i - 1$ to timestep i .

For simplicity, we assume Φ_i is a linear operator. So, we can write the space-time discretization as a linear system

$$Au = \begin{pmatrix} I & & & & \\ -\Phi_1 & I & & & \\ & & \ddots & \ddots & \\ & & & -\Phi_N & I \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_N \end{pmatrix} = \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_N \end{pmatrix} = g. \tag{2.4}$$

Sequential time integration of (2.3) is equivalent to solving the block lower bidiagonal system (2.4), whereas using the MGRIT algorithm is equivalent to solving Equation (2.4) iteratively, using block-Jacobi relaxation with error corrections obtained from coarse approximations to the system.

For simplicity, we assume here that the timestepping operator Φ is constant, so that $\Phi_i = \Phi_j$, $i, j \in [1, 2, \dots, N]$. To create a coarse approximation to A for the purposes of computing error correction terms, we construct a coarse time grid consisting of $N_\Delta + 1 = \frac{N}{m} + 1$ time points with a correspondingly larger timestep $\Delta T = m\Delta t$, where m is an integer called the **coarsening factor**. Without loss of generality, we assume that N is divisible by m . The coarse time grid is then represented by the points

$$T_i = i\Delta T = i(m\Delta t) = t_{im}, \quad i = 0, 1, \dots, N_\Delta. \tag{2.5}$$

An illustration of the decomposition of the time grid into coarse and fine grids is given in Figure 2.1.

Then, we define our coarse grid timestepping operator Φ_Δ . Ideally, one would apply the timestepping operator Φ m -times over a coarse time step, i.e., apply Φ^m

Chapter 2. MGRIT algorithm

over $[T_{i-1}, T_i]$. The corresponding discretized initial value problem on the coarse time grid, equivalent to Equation (2.3), is given by

$$\begin{aligned} u_0 &= g_0, \\ u_{km} &= \Phi^m(u_{(k-1)m}) + \tilde{g}_{km}, \quad k = 1, 2, \dots, N_\Delta, \end{aligned} \tag{2.6}$$

where

$$\tilde{g}_{km} = \sum_{j=0}^{m-1} \Phi^j g_{km-j}.$$

The system (2.6) can be written as a linear system as well, which takes the form

$$A_\Delta u_\Delta = \begin{pmatrix} I & & & & \\ -\Phi^m & I & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & -\Phi^m & I \end{pmatrix} \begin{pmatrix} u_0 \\ u_m \\ u_{2m} \\ \vdots \\ u_{N_\Delta m} \end{pmatrix} = \begin{pmatrix} g_0 \\ \tilde{g}_m \\ \tilde{g}_{2m} \\ \vdots \\ \tilde{g}_{N_\Delta m} \end{pmatrix} = \tilde{g}_\Delta, \tag{2.7}$$

where A_Δ is a block lower bidiagonal matrix with $N_\Delta + 1 = \frac{N}{m} + 1$ total block rows and columns. Since the ideal coarse grid timestepping operator Φ^m is as expensive to compute on the coarse grid as m applications of the fine grid operator Φ , inverting A_Δ is as expensive as inverting A . Thus, a cheap approximation $\Phi_\Delta \approx \Phi^m$ is chosen. A typical choice of Φ_Δ for a parabolic problem is the fine grid operator Φ with a timestep size of $\Delta T = m\Delta t$, instead of the fine grid step size Δt . For instance, if Φ is backward Euler with a step size of Δt , Φ_Δ would then be backward Euler with a step size $\Delta T = m\Delta t$. However for advection-dominated problems without strong diffusive terms, this has been shown not to be a good choice of Φ_Δ as discussed in Section 1.1. Most research for solving these problems with MGRIT and parareal is focused on choosing Φ_Δ that work well, and this is what we address in Chapter 3.

Then the coarse grid approximation takes the form $B_\Delta u_\Delta = \tilde{g}_\Delta$ where $B_\Delta \approx A_\Delta$,

and

$$B_{\Delta} = \begin{pmatrix} I & & & & \\ -\Phi_{\Delta} & I & & & \\ & \ddots & \ddots & & \\ & & & -\Phi_{\Delta} & I \end{pmatrix}. \quad (2.8)$$

Equations (2.2) and (2.5) give rise to a partitioning of the time grid into **F-points** denoted F_p and **C-points** denoted by C_p , which are defined as

$$\begin{aligned} F_p &= \{t_i\} \setminus \{T_i\}, \\ C_p &= \{T_i\}. \end{aligned} \quad (2.9)$$

The partition is depicted in Figure 2.1.

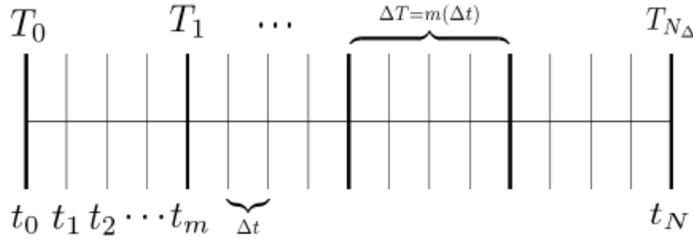


Figure 2.1: Illustration of a temporal grid decomposed into a fine and coarse grid, uniformly spaced. F-points are only present on the fine grid, and C-points are present on both the fine and coarse grid.

The partitioning in Equation (2.9) gives rise to the two fundamental types of relaxation in the MGRIT algorithm, **F-relaxation** and **C-relaxation**. F-relaxation consists of applying Φ within each interval of F-points located between two C-points. C-relaxation applies Φ once per C-point, evolving from the F-point immediately preceding the C-point. This is depicted in Figure 2.2. As shown in Algorithms 1 and 2, the F- and C-relaxation algorithms can be performed in each of the N_{Δ} coarse intervals in parallel.

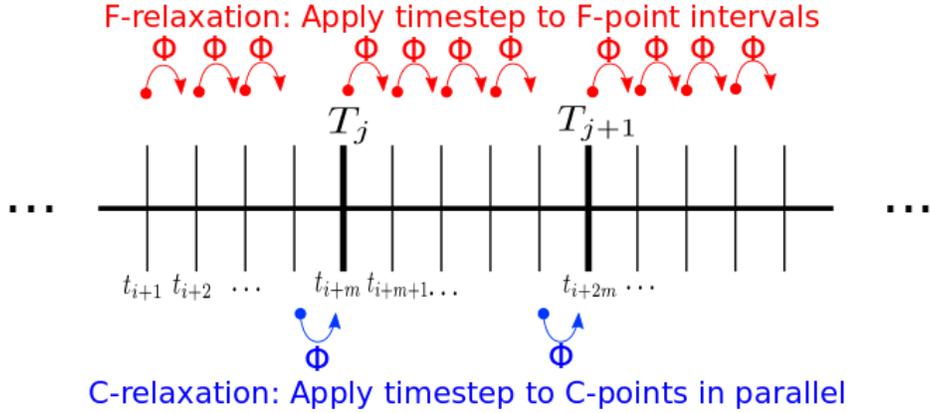


Figure 2.2: Illustration of F- and C-relaxation. C-points are denoted with T_j, T_{j+1} and F-points are denoted with t_{i+1}, t_{i+2}, \dots .

Algorithm 1 F-relaxation

- 1: **for** $k = 0, 1, \dots, N_\Delta$ **do** {Parallel for-loop over k }
 - 2: **for** $i = 1, 2, \dots, m - 1$ **do** {Sequential for-loop over i }
 - 3: $u_{km+i} \leftarrow \Phi(u_{km+i-1}) + g_{km+i-1}$
 - 4: **end for**
 - 5: **end for**
-

Algorithm 2 C-relaxation

- 1: **for** $k = 0, 1, \dots, N_\Delta$ **do** {Parallel for-loop over k }
 - 2: $u_{km} \leftarrow \Phi(u_{km-1}) + g_{km-1}$
 - 3: **end for**
-

An application of F-relaxation, followed by C-relaxation, is equivalent to applying Φ^m to u_{km} for $k = 0, 1, \dots, N_\Delta$. This FC-relaxation sweep is equivalent to an application of block-Jacobi relaxation on the coarse grid system in Equation (2.7). MGRIT allows for the use of F-relaxation or FCF-relaxation smoothing schemes, where FCF-relaxation is an application of an FC-sweep followed by another application of F-relaxation.

Chapter 2. MGRIT algorithm

For restriction to a coarse grid, MGRIT uses injection at the C-points, $u_k^c \leftarrow u_{km}$, where u^c is the coarse grid vector. Interpolation is injection from the coarse to fine grid, $u_k^c \rightarrow u_{km}$, followed by an F-relaxation. The F-relaxation guarantees that if the solution is exact (i.e., equal to the solution one would obtain from a forward solve of Equation 2.4) at the C-points u_{km} , then the solution will also be exact at F-points, thus it is called ideal interpolation.

Putting these pieces together, MGRIT first carries out an F- or FCF-relaxation, followed by restriction of the problem and residual to the coarse-grid. There, the error correction is computed according to the FAS coarse-grid formulation using B_Δ . Lastly, this error correction is interpolated back to the fine grid and added to the current solution guess u and the algorithm repeats iteratively until the 2-norm of the coarse grid residual r_Δ is smaller than a given halting tolerance tol . Pseudocode for the two-grid FAS MGRIT algorithm is given in Algorithm 3.

Algorithm 3 Two-Grid FAS MGRIT($\Phi, \Phi_\Delta, u, g, tol$)

- 1: Apply F- or FCF-relaxation to $Au = g$
 - 2: Compute and restrict fine grid approximation and its residual to the coarse grid
via injection: $u_{\Delta,i} \leftarrow u_{mi}$, $r_{\Delta,i} \leftarrow g_{mi} - (Au)_{mi}$
 - 3: Solve $B_\Delta v_\Delta = B_\Delta u_\Delta + r_\Delta$
 - 4: Compute coarse grid error correction: $e_\Delta \leftarrow v_\Delta - u_\Delta$
 - 5: Correct u at C-points: $u_{mi} \leftarrow u_{mi} + e_{\Delta,i}$
 - 6: If $\|r_\Delta\|_2 < tol$, apply F-relaxation to $Au = g$ and terminate
 - 7: Else, go to step 1
-

Chapter 3

Asymptotic Parallel-in-Time Methods

The parareal algorithm [24] is a parallel-in-time method which is equivalent to a two-level MGRIT solve using F-relaxation [10].

Here, we consider the so-called asymptotic parareal method, which is designed for hyperbolic problems which take the form

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{1}{\epsilon} \mathcal{L} \mathbf{u} = \mathcal{N}(\mathbf{u}) + \mathcal{D} \mathbf{u}, \quad \mathbf{u}(0) = \mathbf{u}_0, \quad (3.1)$$

where \mathcal{L} is a linear operator with purely imaginary eigenvalues, $\mathcal{N}(\mathbf{u})$ is a nonlinear quadratic polynomial, \mathcal{D} is a dissipation operator, and ϵ is a nondimensional parameter which determines the time scale separation between \mathcal{L} and \mathcal{N} . In particular, when ϵ is small enough, the eigenvalues of the linear operator \mathcal{L} inhabit an area of the complex plane distinctly separate from where the eigenvalues of \mathcal{N} and \mathcal{D} are located. As ϵ becomes larger, the eigenvalues of \mathcal{L} , \mathcal{N} and \mathcal{D} become collocated on the complex plane, which corresponds to a lack of scale separation. Here, the oscillations from \mathcal{L} mix with the diffusion and nonlinear advection of \mathcal{D} and \mathcal{N} .

Chapter 3. Asymptotic Parallel-in-Time Methods

Rapid oscillations due to the linear term $\frac{1}{\epsilon}\mathcal{L}\mathbf{u}$ occur on a timescale of $\mathcal{O}(\epsilon)$. In order to resolve these oscillations, it is necessary to take timesteps of size $\mathcal{O}(\epsilon)$ when using traditional time-stepping methods. Thus, the coarse grid approximation in parareal or MGRIT would also require a timestep size of $\mathcal{O}(\epsilon)$, which would require too fine a coarse grid representation for parareal or MGRIT to be efficient [18]. In summary, using a coarse timestep size in parareal or MGRIT that is $\gg \mathcal{O}(\epsilon)$ with standard choices of coarse grid operators will violate the Nyquist sampling rate of the temporal oscillations. This will alias the oscillations on the coarse grid and lead to inaccurate time derivatives. For this reason, problems of the form given by Equation (3.1) are difficult for parareal and MGRIT to solve efficiently when using traditional coarse grid approximations.

We investigate a coarse grid propagator that is based on the observation [26] that the solution $\mathbf{u}(t)$ to Equation (3.1) has the asymptotic approximation

$$\mathbf{u}(t) = e^{(-t/\epsilon)\mathcal{L}}\bar{\mathbf{u}}(t) + \mathcal{O}(\epsilon), \quad (3.2)$$

where the asymptotic solution $\bar{\mathbf{u}}(t)$ varies slowly in time and satisfies the equation

$$\frac{\partial \bar{\mathbf{u}}}{\partial t} = \bar{\mathcal{N}}(\bar{\mathbf{u}}) + \bar{\mathcal{D}}\bar{\mathbf{u}}, \quad \bar{\mathbf{u}}(0) = \mathbf{u}_0, \quad (3.3)$$

and the nonlinear term $\bar{\mathcal{N}}$ is given by the time-averaged quantity

$$\bar{\mathcal{N}}(\bar{\mathbf{u}}(t)) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T e^{s\mathcal{L}} \mathcal{N}(e^{-s\mathcal{L}}\bar{\mathbf{u}}(t)) ds. \quad (3.4)$$

The operator $\bar{\mathcal{D}}$ is time-averaged in the same fashion, but note that if D is a linear operator, then $\bar{\mathcal{D}} = \mathcal{D}$. Also note that as opposed to being evolved over the entire time domain $[0, T]$, the coarse grid operator only timesteps $\bar{\mathbf{u}}(t)$ over a coarse time interval $[(j-1)\Delta T, j\Delta T]$. An approximation to $\mathbf{u}(t)$ is then obtained by applying $e^{(-t/\epsilon)\mathcal{L}}$ to $\bar{\mathbf{u}}(t)$.

As a result, time integration for the time-averaged solution $\bar{\mathbf{u}}(t)$ can be done without evolving the oscillatory $e^{(1/\epsilon)\mathcal{L}}$ term. This allows coarse grid timesteps

$\Delta T \gg \epsilon$ to be taken without violating the Nyquist sampling rate of the rapid temporal oscillations.

A finite approximation to Equation (3.4) is necessary for numerical evaluation. The finite time average is formally introduced as

$$\bar{\mathcal{N}}(\bar{\mathbf{u}}(t)) \approx \frac{1}{\eta} \int_0^\eta \rho(s/\eta) e^{s\mathcal{L}} \mathcal{N}(e^{-s\mathcal{L}} \bar{\mathbf{u}}(t)) ds, \quad (3.5)$$

where η is referred to as the **averaging window**, and $\rho(s)$ is referred to as the **integrating kernel**. The choice of $\rho(s)$ is investigated thoroughly in [18, 32, 31]. In particular, $\rho(s)$ is chosen to be a Gaussian bump function with compact support in $(0, 1)$ and an integral of unity; i.e:

$$\int_0^1 \rho(s) ds = 1.$$

We compute $\rho(s)$ using the approximation (from [30])

$$\rho(s) \approx \exp(-50 * (s - 0.5)^2).$$

The purpose of introducing the integrating kernel is to increase the accuracy of the approximation to $\bar{\mathcal{N}}(\bar{\mathbf{u}}(t))$, as well as to prevent discontinuities in the solution profile $\bar{\mathbf{u}}(t)$. The use of integrating kernels for these purposes is a standard technique in averaging computations for multiscale methods [9, 8]. This time-averaging process over \bar{M} quadrature points for computing (3.3) is described in Algorithm 4.

The convergence rate of asymptotic parareal depends on the choice of η . The optimal choice of η , in the asymptotic parareal setting, is shown in [31] to depend most heavily on the coarse grid step size ΔT and scale separation parameter ϵ . In particular for the case where $\Delta T = 0.1$, the optimal choice of η grows as $\epsilon \rightarrow 0$, but then stabilizes around a choice of $\eta = \Delta T$ for values of $\epsilon = 0.1$ and larger.

More generally as $\eta \rightarrow 0$, no time averaging is being performed and the error due to averaging goes to 0 as well. At the same time, the error due to timestepping

increases as $\eta \rightarrow 0$. As η becomes larger, the timestepping error goes to 0 and the averaging error increases. The optimal η for a given ΔT and ϵ is the solution of an optimization problem where the sum of the averaging error and timestepping error are minimized.

Algorithm 4 Evaluate time average (in parallel):

- 1: **for** $j = 1, \dots, \overline{M} - 1$ **do** {Parallel for-loop over j }
 - 2: $s_m = \eta m / \overline{M}$
 - 3: $\bar{\mathbf{u}}_m \leftarrow \rho(s_m / \eta) e^{s_m \mathcal{L}} \mathcal{N}(e^{-s_m \mathcal{L}} \bar{\mathbf{u}}_0)$
 - 4: **end for**
 - 5: **return** $(1 / \overline{M}) * \text{Sum}(\bar{\mathbf{u}}_1, \dots, \bar{\mathbf{u}}_{\overline{M}})$
-

The fine grid timestepping operator Φ for asymptotic parareal evolves \mathbf{u} over the fine timestep Δt using a three-step Strang splitting method [3] which first advances the linear terms $\epsilon^{-1} \mathcal{L} + \mathcal{D}$ by a half timestep $\Delta t / 2$ using an exponential integrator, then advances the nonlinear term \mathcal{N} by a full timestep Δt using midpoint quadrature, and finally takes another $\Delta t / 2$ half step for $\epsilon^{-1} \mathcal{L} + \mathcal{D}$ with the exponential integrator. The pseudocode for the fine propagator is given in Algorithm 5.

The coarse grid operator Φ_Δ used in asymptotic parareal also uses the Strang splitting scheme, integrating over the coarse interval ΔT instead. The exponential integrator only acts on the dissipative operator \mathcal{D} , and the averaged nonlinear term $\overline{\mathcal{N}}$ is evaluated using the midpoint rule. Moreover, the slowly-varying $\bar{\mathbf{u}}$ is transformed back to an approximation to \mathbf{u} by applying the $e^{(-t/\epsilon)\mathcal{L}}$ operator, as shown in Equation (3.2). Pseudocode for the coarse solver is shown in Algorithm 6.

Algorithm 5 fine solver Φ : Fine_Solver($\mathbf{u}_0, \Delta t, \Delta T$)

$$M = \Delta T / \Delta t$$

- 1: **for** $m = 1, \dots, M$ **do**
 - 2: Take a $\Delta t/2$ time-step for the linear term:

$$\hat{\mathbf{v}} \leftarrow e^{(\Delta t/2)(\epsilon^{-1}\mathcal{L}+\mathcal{D})}\mathbf{u}_m \text{ \{ } \Delta t/2 \text{ step on } \mathbf{u}' = (\epsilon^{-1}\mathcal{L} + \mathcal{D})\mathbf{u} \}$$
 - 3: Take a Δt time-step for the nonlinear term:

$$\mathbf{v} \leftarrow \mathcal{N}(\hat{\mathbf{v}}),$$

$$\mathbf{v} \leftarrow \hat{\mathbf{v}} + \Delta t \mathcal{N} \left(\hat{\mathbf{v}} + \frac{\Delta t}{2} \mathbf{v} \right) \text{ \{ } \Delta t \text{ step on } \mathbf{u}' = \mathcal{N}(\mathbf{u}) \text{ with mid-point} \}$$
 - 4: Take a $\Delta t/2$ time-step for the linear term:

$$\mathbf{u}_{m+1} \leftarrow e^{(\Delta t/2)(\epsilon^{-1}\mathcal{L}+\mathcal{D})}\mathbf{v} \text{ \{ } \Delta t/2 \text{ step on } \mathbf{u}' = (\epsilon^{-1}\mathcal{L} + \mathcal{D})\mathbf{u} \}$$
 - 5: **end for**
 - 6: **return** \mathbf{u}_M
-

Algorithm 6 asymptotic slow solver Φ_Δ : Coarse_Solver($\mathbf{u}_0, \Delta T$)

- 1: Take a $\Delta T/2$ time-step for the linear dissipative term:

$$\hat{\mathbf{v}} \leftarrow e^{(\Delta T/2)\mathcal{D}}\mathbf{u}_0 \text{ \{ } \Delta T/2 \text{ step on } \mathbf{u}' = \mathcal{D}\mathbf{u} \}$$
 - 2: Take a ΔT time-step for the averaged nonlinear term:

$$\mathbf{v} \leftarrow \overline{\mathcal{N}}(\hat{\mathbf{v}}),$$

$$\mathbf{v} \leftarrow \hat{\mathbf{v}} + \Delta T \overline{\mathcal{N}} \left(\hat{\mathbf{v}} + \frac{\Delta T}{2} \mathbf{v} \right) \text{ \{ } \Delta T \text{ step on } \mathbf{u}' = \overline{\mathcal{N}}(\mathbf{u}) \text{ with midpoint formula} \}$$
 - 3: Take a $\Delta T/2$ time-step for the linear dissipative term:

$$\mathbf{v} \leftarrow e^{(\Delta T/2)\mathcal{D}}\mathbf{v} \text{ \{ } \Delta T/2 \text{ step on } \mathbf{u}' = \mathcal{D}\mathbf{u} \}$$
 - 4: Transform back to the fast time coordinate: $\mathbf{u}_1 = e^{(\Delta T/\epsilon)\mathcal{L}}\mathbf{v}$
 - 5: **return** \mathbf{u}_1
-

Chapter 4

Test Problem - Rotating Shallow Water Equations

The test problem that we consider is the one-dimensional rotating shallow water equations (RSWE). The RSWE is an important set of equations in geophysical fluid dynamics that is of particular interest in the weather and climate modeling community [18, 7, 26, 25]. The RSWE in the one-dimensional form is given as

$$\begin{aligned} \frac{\partial v_1}{\partial t} + \frac{1}{\epsilon} \left(-v_2 + F^{-1/2} \frac{\partial h}{\partial x} \right) + v_1 \frac{\partial v_1}{\partial x} &= \mu \partial_x^4 v_1, \\ \frac{\partial v_2}{\partial t} + \frac{1}{\epsilon} v_1 + v_1 \frac{\partial v_2}{\partial x} &= \mu \partial_x^4 v_2, \\ \frac{\partial h}{\partial t} + \frac{F^{-1/2}}{\epsilon} \frac{\partial v_1}{\partial x} + \frac{\partial}{\partial x} (h v_1) &= \mu \partial_x^4 h, \end{aligned} \tag{4.1}$$

where $h(x, t)$ denotes the surface height of the fluid, and $v_1(x, t)$ and $v_2(x, t)$ denote the horizontal fluid velocities. The quantity v_1 denotes horizontal velocity in the x -direction (along the direction of the spatial domain,) and v_2 denotes horizontal velocity in the y -direction.

The non-dimensional parameter ϵ is the **Rossby number**, which represents the ratio of the rotation timescale to the timescale of horizontal advection. Recall

Chapter 4. Test Problem - Rotating Shallow Water Equations

that temporal oscillations occur on a $\mathcal{O}(\epsilon)$ timescale, so as $\epsilon \rightarrow 0$, the frequency of oscillations increases accordingly. In realistic geophysical flows, ϵ is typically small, and a reasonable choice for testing the handling of fast gravity waves is $\epsilon = \mathcal{O}(10^{-2})$. The value μ is the coefficient to a fourth-order hyperviscosity term which is introduced for stability, and is taken here to be 10^{-4} unless otherwise noted. As μ becomes larger, the magnitudes of temporal oscillations decrease as the problem evolves in time and the problem becomes more diffusion-dominated. Figures 4.1 and 4.2 depict the effect that different choices of ϵ and μ have on the exact space-time solution.

The quantity $Fr := F^{1/2}\epsilon$ is the **Froude number**, representing the ratio of fluid velocity to gravity wave speed. F is the **Rossby deformation radius**, which is the distance a wave travels before being significantly effected by rotation. Except where otherwise specified, we examine the case where $F = 1$ and hence $Fr = \epsilon$.

These equations are given in a two-dimensional form by Embid and Majda in [7]. The simplified one-dimensional form in Equation (4.1) is given by Haut and Wingate in [18], and is obtained from the two-dimensional form by assuming that all derivatives in the y -direction are zero. This is done in order to preserve terms which take rotation into account, while still having an essentially 1-dimensional model problem.

We can write Equation (4.1) in the general form of Equation (3.1) by letting the state variable

$$\mathbf{u}(t, \mathbf{x}) = \begin{pmatrix} v_1(t, x) \\ v_2(t, x) \\ h(t, x) \end{pmatrix}. \quad (4.2)$$

Chapter 4. Test Problem - Rotating Shallow Water Equations

Then, our operators from Equation ((3.1)) are

$$\mathcal{L} = \begin{pmatrix} 0 & -1 & F^{-1/2}\partial_x \\ 1 & 0 & 0 \\ F^{-1/2}\partial_x & 0 & 0 \end{pmatrix}, \quad \mathcal{D} = \mu\partial_x^4 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathcal{N}(\mathbf{u}) = \begin{pmatrix} v_1(v_1)_x \\ v_1(v_2)_x \\ (hv_1)_x \end{pmatrix}. \quad (4.3)$$

Indeed, \mathcal{L} is skew-hermitian, i.e., $-\mathcal{L} = (\mathcal{L}^T)^*$ and has purely imaginary eigenvalues with an orthonormal basis of eigenvectors. Thus, \mathcal{L} gives rise to oscillations. Furthermore, we see that \mathcal{D} is a standard hyperviscosity operator and $\mathcal{N}()$ is a nonlinear quadratic term.

Chapter 4. Test Problem - Rotating Shallow Water Equations

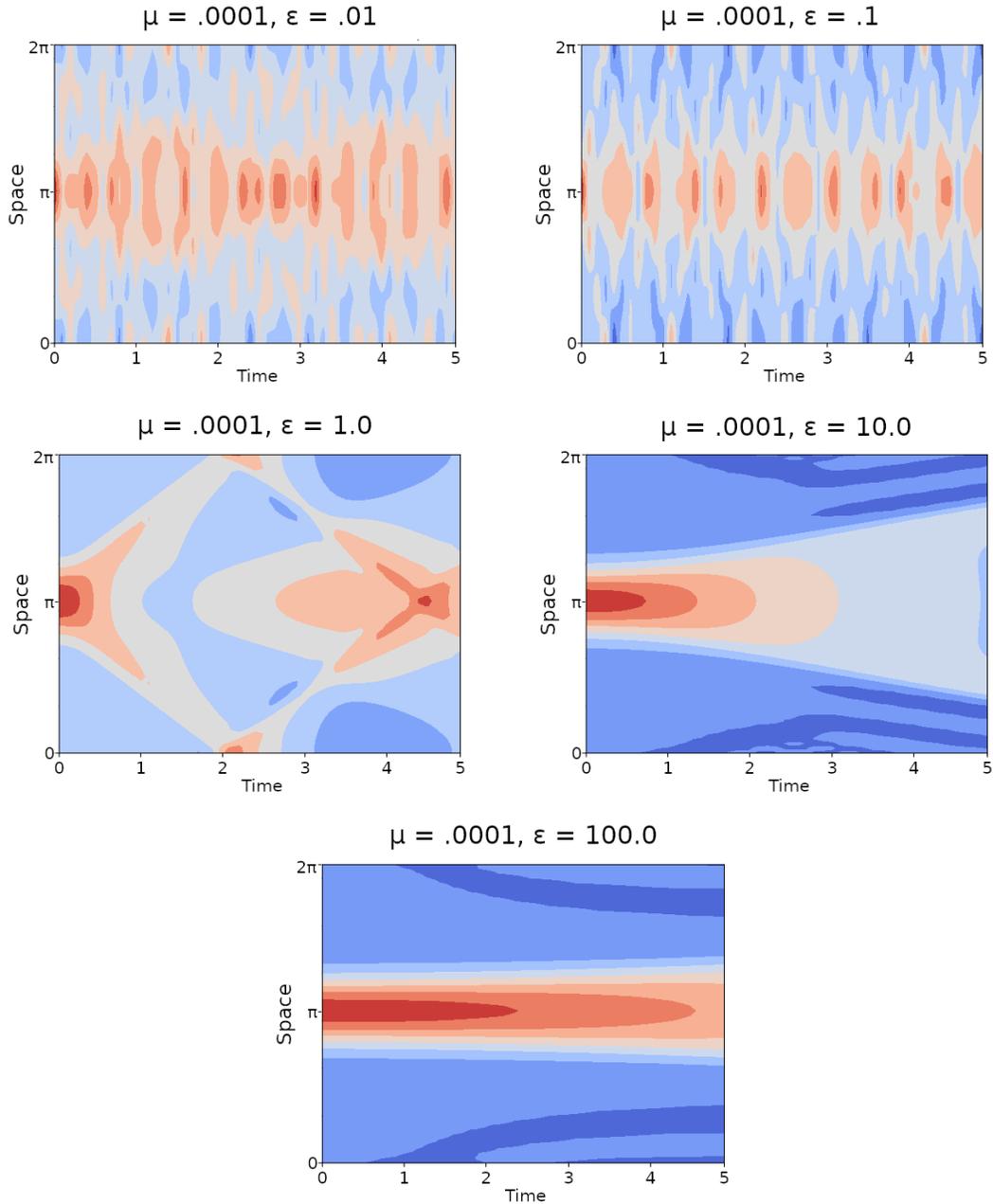


Figure 4.1: Wave-height solution profile of h in a space-time domain $X = [0, 2\pi], T = [0, 5]$ with $N_x = 64$ Fourier modes in space, $N_\Delta = 50$ time points, various ϵ values, and a fixed $\mu = .0001$. Here, time evolves in the x -axis and the y -axis is the spatial dimension. For small values of ϵ , the magnitude of \mathcal{L} is large and oscillations in the direction of the time domain are rapid. As ϵ becomes large, the magnitude of the $1/\epsilon \mathcal{L}$ term shrinks and oscillations occur over much larger timescale. In particular, when $\epsilon = 10$ and $\epsilon = 100$, a complete temporal oscillation is no longer present because oscillations occur on an $\mathcal{O}(\epsilon)$ timescale. See subfigure titles for specific ϵ, μ values.

Chapter 4. Test Problem - Rotating Shallow Water Equations

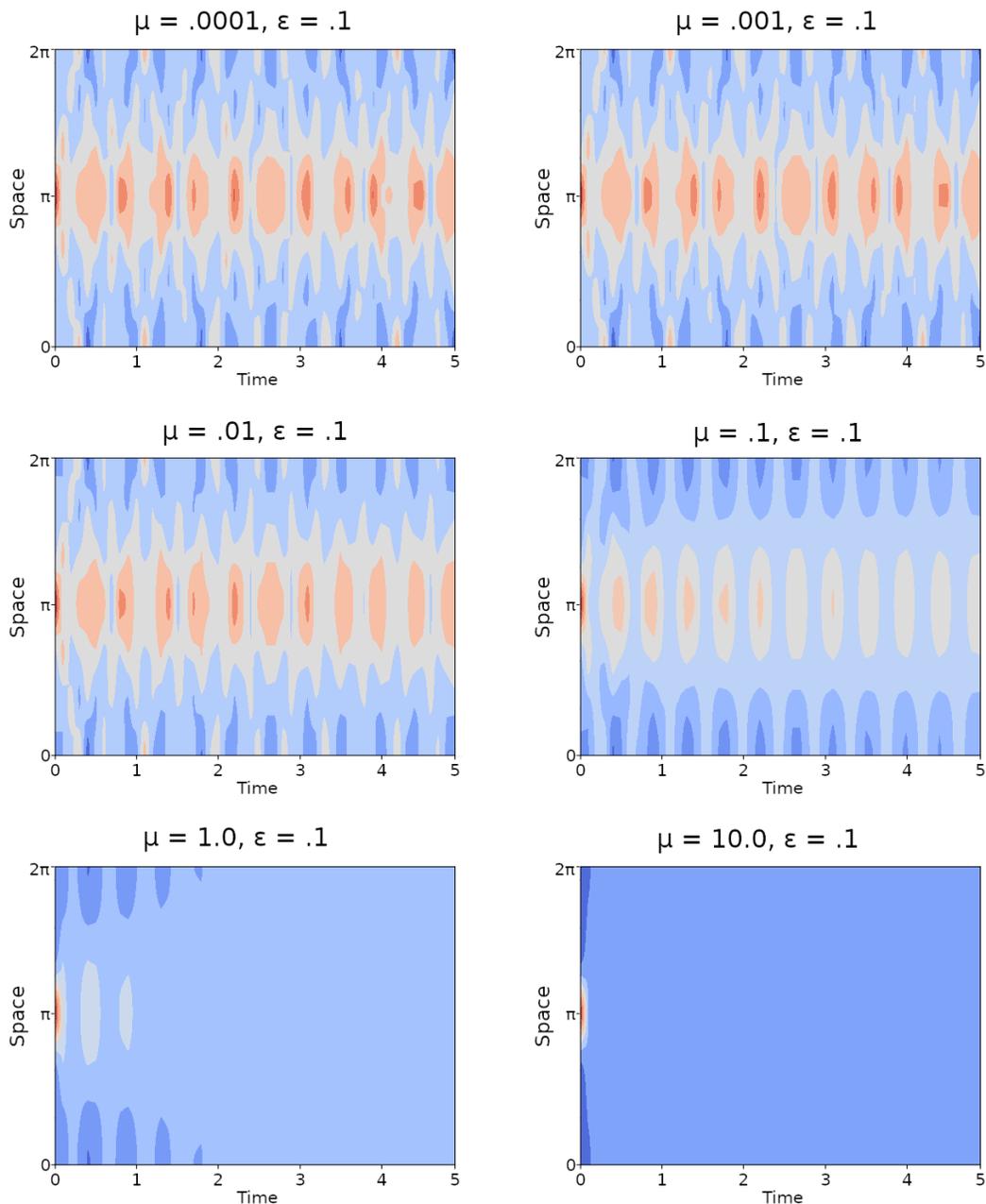


Figure 4.2: Wave-height solution profile of h in a space-time domain $X = [0, 2\pi], T = [0, 5]$ with $N_x = 64$ Fourier modes in space, $N_\Delta = 50$ time points, various μ values, and a fixed $\epsilon = .1$. Here, time evolves in the x -axis and the y -axis is the spatial dimension. As μ increases, so does the magnitude of the dissipation term. The result is that we see the effects of dissipation become prominent at ever earlier times. See subfigure titles for specific ϵ, μ values.

4.1 Implementation

A Python implementation of an asymptotic parareal solver for the 1-dimensional RSWE, titled Cyclops-Lite, is publicly available from Wingate’s research group [30]. Following the software structure for implementing parareal in a decentralized fashion described in [35], Cyclops was written by Adam Peddle for his PhD studies on asymptotic parareal. In Cyclops, the RSWE are semi-discretized yielding a system of ODEs like in Equation (2.3), and a Fourier spectral method is used as a spatial discretization.

Cyclops is written in Python and our chosen MGRIT implementation, XBraid [1], is written in C/C++. We have developed an interface written in the Cython language between Cyclops and XBraid to extend the asymptotic parareal method to asymptotic MGRIT, titled CycloBraid. CycloBraid allows XBraid to call Cyclops’ routines for handling Fourier spectral discretizations as well as functions for time-averaging and time-integration (refer to Algorithms 4, 5, 6). XBraid then uses these wrapped Cyclops functions to define Φ on each level and carry out the MGRIT cycling. Some other wrapper functions were required, e.g., taking a norm, performing vector addition, and MPI buffer packing and unpacking.

CycloBraid also serves as a template for interfacing XBraid with other numerical codebases written in Python. This aspect of the CycloBraid contribution is already having impact at Sandia National Laboratories, where it is being used to pair PyTorch [29] with XBraid for research on layer-parallel training of residual neural networks [17].

Chapter 5

Research Objectives

Our primary objective is to determine if we can create an asymptotic MGRIT algorithm that outperforms asymptotic parareal by using the generalizability of the parareal algorithm to the MGRIT algorithm. That is, we seek to create an asymptotic MGRIT algorithm that is more robust with respect to problem parameters such as ϵ or F , and that converges faster and more efficiently.

We first consider the recursive multilevel capability of MGRIT when using the asymptotic coarse grid technique described in Chapter 3. We consider the effect of adding a novel coarser third level of various sizes to a two-level asymptotic MGRIT solve, thereby allowing the application of the asymptotic coarse grid operator Φ_Δ on the first coarse level to be performed in parallel with relaxation. For the second coarse level, we consider both the use of the asymptotic coarse grid timestepping operator Φ_Δ , as well as the use of the Strang splitting operator Φ applied to a coarsened representation of the slowly-varying solution $\bar{\mathbf{u}}$ from the first coarse level. We also propose an investigation of alternative time integration techniques for use exclusively on the third level.

We investigate the effect of $F(\text{CF})^n$ -relaxation, where there are n applications of

Chapter 5. Research Objectives

FC-relaxation sweeps. We note that only F-relaxation has been considered previously for asymptotic parareal. In Section 6.4, we examine the case of $\epsilon = .01$, and compare the use of F-relaxation and FCF-relaxation in two-level MGRIT solves on various coarse grid sizes and time domains. FCF-relaxation turns out to offer significant performance gains in some cases. Furthermore, we look at the case where $\epsilon = 1.0$ in Section 6.7 and examine how additional CF-sweeps in the relaxation stage, combined with an increased averaging window size η , can be used to allow for more efficient coarse grids. That is, these strategies allow for more aggressive coarsening in time.

The computational cost of the coarse grid operator Φ_Δ is discussed in Section 6.4.1, where we find that the cost of computing the time-averaged quantity $\overline{\mathcal{N}}$ dominates an entire MGRIT cycle. Thus, we examine in Section 6.5 techniques for reusing the slowly-varying coarse grid solution $\overline{\mathbf{u}}$ from previous MGRIT iterations, as well as recomputing $\overline{\mathbf{u}}$ based on stale $\overline{\mathcal{N}}$ values from previous MGRIT iterations. The goal is to save computational work.

As a final objective, we consider the fact that asymptotic parareal for the RSWE has primarily focused on the case where the Rossby radius of deformation $F = 1$, and the timescales of rotational and gravitational effects are equal. In [31], Peddle discusses how the timescales of rotation and gravitation become separated when F becomes small. In the RSWE as given in Equation (4.1), the effects of rotation and gravitation are both represented in the oscillatory linear term \mathcal{L} , but \mathcal{L} can be decomposed into a rotational operator \mathcal{R} and gravitational operator \mathcal{G} [40]. We consider some MGRIT techniques for solving the RSWE at $F = 10^{-4}$ where rotational effects are two orders of magnitude larger than the gravitational effects. In particular, we consider techniques to use two different time averaging windows η to target accuracy separately for \mathcal{R} and \mathcal{G} .

Chapter 6

Numerical Results

6.1 Problem Setup

Here we describe how we choose the problem parameters for the RSWE coefficients and space-time discretization, as well as the specifications for the machine on which the experiments are run. Our choices of problem parameters follow general conventions for asymptotic parareal studies.

Our experiments are run on a machine with two Intel Xeon Silver 4110 2.10GHz processors, based on the Skylake x86 architecture. Each processor has 8 cores and 16 threads per processor, giving a total of 16 cores and 32 threads.

In all experiments, the RSWE are semi-discretized to a system of ODEs, like in Equation (2.3). The size of the time domain $T = [0, t_f]$ and the number of points in the time grids vary between experiments, but are always given. Note that the initial time of the time domain is always 0.

In space, we use a spectral discretization as discussed in Section 4.1 of 64 Fourier modes on a spatial domain with size $[0, 2\pi]$. The initial conditions are set up in

Chapter 6. Numerical Results

real space such that velocity profiles v_1, v_2 are identically zero and the wave height profile h is a Gaussian bump with a maximum height of 1 at the center of the spatial domain.

The hyperviscosity coefficient μ is set to 10^{-4} in all experiments, which provides stability for more oscillatory Fourier modes while still allowing the problem to be advection-dominated.

We choose the Froude number $Fr = F^{1/2}\epsilon$ with $F = 1$ and hence $Fr = \epsilon$. We will focus on two choices of the scale separation parameter ϵ : $\epsilon = 1.0$ (where no separation between timescales occurs and temporal oscillations are not rapid,) and $\epsilon = 10^{-2}$ (where the timescale of gravity waves is 100 times smaller than the timescale of Rossby waves and fast temporal oscillations occur.)

The time averaging window η is computed using the formula (from [30])

$$\eta = \alpha \frac{dT}{\epsilon^{0.2}}, \quad (6.1)$$

where α is a coefficient that is equal to 1 unless otherwise noted and dT is the timestep size on the level where time averaging is being done. The number of quadrature points \bar{M} over which the time average is done is computed by the formula (from [30])

$$\bar{M} = \max\left\{25, \frac{80}{\beta}\eta\right\}, \quad (6.2)$$

where β is a nonzero coefficient that is equal to 1 unless otherwise noted. Equation (6.1) is derived experimentally in [31].

The algorithms for MGRIT are given in Chapter 2, and the time-stepping algorithms are given in Chapter 3.

6.2 Code Verification

When XBraid is coupled to a new numerical code (here Cyclops), code verification is of critical importance and must be documented. The literature on the asymptotic parareal method uses an L^∞ -error estimate, based on the L^∞ -error at the current and prior parareal iteration, as the stopping criteria, as opposed to Braid’s use of the residual norm at C-points. The L^∞ error for an iteration is the greatest L^∞ error entry-wise over the whole space-time solution at that iteration, and is estimated with

$$L^\infty = \max_i \left| \frac{U_{new,i} - U_{old,i}}{U_{old,i}} \right|,$$

where the max is taken entry-wise over all vector entries i , U_{new} is the coarse grid space-time state vector at the current MGRIT iteration, and U_{old} is the coarse grid space-time state vector at the previous iteration. In Cyclops [30], the errors are computed in Fourier space and returned in real space.

The motivation for obtaining a history of L^∞ errors from the Braid computation is to compare it with the L^∞ error history from Cyclops, so that we can verify that the Braid implementation of the asymptotic parareal method is correct. By inspection of the Cyclops code, it should be the case that one iteration of Cyclops is equivalent to one iteration of the Braid implementation in a two-level setting with the use of F-relaxation and the “skip option” that omits the initial smoothing during the first iteration prior to the first down-cycle. (Remember that two-level MGRIT with F-relaxation is equivalent to parareal [16].) Due to this, we expect that the L^∞ error histories should match. These results will also be used to compare and contrast the Braid residual history with the L^∞ error history, and verify that they “mirror” each other.

To investigate L^∞ error agreement between the codes and compare L^∞ errors to the Braid residual, we consider two final times, $t_f = 1.0$ and $t_f = 3.0$. We use a time discretization of $N_\Delta = 50$ coarse grid time points and $N = 500$ fine grid time points,

Chapter 6. Numerical Results

hence we have a coarsening factor of $m = 10$. The scale separation parameter $\epsilon = .01$ is used. In both the Cyclops and XBraid codes, the L^∞ errors are computed at each iteration after the coarse grid error correction has been applied (i.e., after Step 5 in Algorithm 3 in Chapter 2.)

Tables 6.1 and 6.2 demonstrate the per-iteration histories of the L^∞ errors obtained from Cyclops and two-level Braid using the skip option, F-relaxation, and the asymptotic coarse solver. Also included is the XBraid residual history. The first iterations at which the L^∞ errors do not match to at least seven digits of accuracy are bold and highlighted in red. We see that Braid and Cyclops do appear to be carrying out the same algorithm, because the L^∞ histories match up well. And we see that the Braid residual history is as good of a convergence indicator as the L^∞ estimates.

Iteration	Cyclops L^∞	Braid L^∞	Braid Residual
1	1.319788e-01	1.319788e-01	5.833705e-03
2	4.275742e-03	4.275742e-03	1.354342e-04
3	1.786624e-04	1.786624e-04	7.120944e-06
4	1.251012e-05	1.251012e-05	2.566034e-07
5	8.747519e-07	8.747519e-07	1.131633e-08
6	6.769316e-08	6.769315e-08	6.109124e-10
7	5.649629e-09	5.649619e-09	4.439781e-11
8	4.349574e-10	4.349486e-10	3.147370e-12

Table 6.1: Comparison of per-iteration histories of the XBraid residual norm and the L^∞ -norms when performing asymptotic parareal solves with XBraid and Cyclops respectively, for the 1-dimensional RSWE on a time domain of $[0, 1]$.

Chapter 6. Numerical Results

Iteration	Cyclops L^∞	Braid L^∞	Braid Residual
1	8.694573e-02	8.694573e-02	2.812960e-02
2	4.297909e-02	4.297909e-02	3.356759e-03
3	7.777487e-03	7.777487e-03	6.762658e-04
4	1.371985e-03	1.371985e-03	9.505859e-05
5	2.932189e-04	2.932189e-04	1.740498e-05
6	4.694276e-05	4.694276e-05	3.247782e-06
7	9.274929e-06	9.274929e-06	7.799707e-07
8	1.703251e-06	1.703251e-06	1.503292e-07
9	3.303419e-07	3.303419e-07	3.176993e-08
10	5.869087e-08	5.869087e-08	5.248516e-09
11	8.914748e-09	8.914733e-09	9.500381e-10
12	1.310012e-09	1.415601e-09	1.395190e-10
13	2.239311e-10	2.239173e-10	2.362944e-11

Table 6.2: Comparison of per-iteration histories of the XBraid residual norm and the L^∞ -norms when performing asymptotic parareal solves with XBraid and Cyclops respectively, for the 1-dimensional RSWE on a time domain of $[0, 3]$.

6.3 Multilevel Asymptotic MGRIT

In this section, we research the construction of a third MGRIT level and examine performance effects. Given a two-level asymptotic MGRIT solve, we introduce a coarser grid as a third level, which has not been studied before for asymptotic parareal. In doing so, we introduce parallelism to the solver on the second level (i.e., first coarse level) as the second level will no longer be solved exactly but instead with parallel block-Jacobi relaxation. Thus, if the third grid does not incur a significant penalty in convergence and the solution on the third grid itself can be computed cheaply, then the introduction of a third grid can lead to performance gains in parallel.

Once the oscillations are removed on level two through the asymptotic time-averaging, it is possible that using the non-asymptotic time-stepper Φ described in Algorithm 5 would make for a good time propagator on the third grid. To do this, the transform back to the fast time coordinate operation $e^{-(t/\epsilon)L}$ (see Equation (3.2)) must not be used to add the fast temporal oscillations back into the space-time solution guess on the second level before restriction is performed to transfer information to the third level. In this case where there are now no rapid oscillations to alias on the third grid, the relatively cheap Φ operator could be a good option on the third level, if using it does not incur a penalty in convergence.

Following this, we consider two techniques for timestepping on the coarsest grid. One we will refer to as “Strang-asymptotic-asymptotic MGRIT”, or SAA-MGRIT, which uses the time-averaged coarse grid operator Φ_Δ on the second and third levels. This approach applies the $e^{-(t/\epsilon)L}$ transform to reintroduce the rapid temporal oscillations before performing restriction and interpolation between the second and third levels, as well as when interpolating from the second level to the fine level. The other we refer to as “Strang-asymptotic-Strang MGRIT”, or SAS-MGRIT, which uses the time-averaged operator Φ_Δ on the second level and the standard Strang

Chapter 6. Numerical Results

splitting operator Φ described in Algorithm 5 on the coarsest level. In SAS-MGRIT, the transformation $e^{-(t/\epsilon)L}$ is only applied when interpolating from the second level to the fine level.

For convenience, we introduce the following notation to denote a general multilevel MGRIT time grid hierarchy. Let $N_{t,0} \geq N_{t,1} \geq \dots \geq N_{t,k-1}$ be the numbers of time points on a k -level MGRIT time grid hierarchy, where level 0 corresponds to the finest grid and level $k - 1$ corresponds to the coarsest grid. Then, the notation $\{N_{t,0}, N_{t,1}, \dots, N_{t,k-1}\}$ denotes the number of time points on each level in an MGRIT hierarchy. We will use subscript A to denote the use of the asymptotic coarse grid operator Φ_{Δ} and subscript S to denote the use of the Strang splitting operator Φ on the given level. For instance, $\{4096_S, 64_A\}$ denotes a two-level asymptotic MGRIT solve with 4096 time points and Strang splitting Φ on the fine grid and 64 time points with the asymptotic time integrator Φ_{Δ} on the coarse grid. We also say $\{4096_S, 64_S\}$ to denote a non-asymptotic MGRIT solve with the same two-level time grid hierarchy, and $\{4096_S, 64_A, 8_S\}$ denotes a three-level SAS-MGRIT solve with 4096 time points on the fine level, 64 time points on the second level, and 8 time points on the third level. Likewise, $\{4096_S, 64_A, 8_A\}$ denotes a three-level SAA-MGRIT solve with 4096 time points on the fine level, 64 time points on the second level, and 8 time points on the third level.

In Tables 6.3 and 6.4, we compare the iteration counts and wall times of a two-grid asymptotic MGRIT solve $\{4096_S, 64_A\}$ with three-level SAA- and SAS-MGRIT solves $\{4096_S, 64_A, x_A\}$ and $\{4096_S, 64_A, x_S\}$ where $x = \{2, 4, 8, 16, 32\}$. In other words, given a two-grid asymptotic MGRIT solver, we consider the performance effects of adding a third coarser grid under the SAA- and SAS-MGRIT techniques. The two-level solver is given 64 time points on the coarse level because this is comparable to the coarse grid sizes used in studies of asymptotic parareal for the time domain sizes considered [18, 32, 31] and because this coarse grid size is required for MGRIT

Chapter 6. Numerical Results

convergence in the case of $\epsilon = 1.0$. In particular, time domains considered are $T = [0, t_f]$ where $t_f = \{1, 5, 10\}$. F-relaxation is used, and both $\epsilon = .01$ and $\epsilon = 1.0$ are considered. The halting tolerance is set to $tol = 1.0e-5$. Averaging windows η and quadrature points \overline{M} are computed at each level using Equations (6.1) and (6.2) and the ΔT of that level.

For $t_f = 1.0$, SAA- and SAS-MGRIT can perform as well or better than two-level asymptotic MGRIT for both $\epsilon = .01$ and $\epsilon = 1.0$. In particular, we see in Table 6.3 for $t_f = 1.0$, the described multilevel MGRIT techniques are able to offer a 36% improvement in compute time over the two-grid method for $\{4096_S, 64_A, 32_S\}$ and a 43% improvement in compute time for $\{4096_S, 64_A, 16_A\}$ when $\epsilon = 1$. We see similar improvements in compute time when $\epsilon = .01$ in Table 6.4, where $\{4096_S, 64_A, 32_S\}$ reduces compute time by 47% and $\{4096_S, 64_A, 16_A\}$ reduces compute time by 20% when $t_f = 1.0$. Furthermore, when $t_f = 5.0$ and $\epsilon = .01$, we also observe speedup; $\{4096_S, 64_A, 16_A\}$ yields a 16% reduction in wall time and $\{4096_S, 64_A, 16_A\}$ gives a 21% reduction in wall time.

For larger time domains where $\epsilon = 1.0$, SAA- and SAS-MGRIT both fail to converge. This is not attributable to the SAA- or SAS-MGRIT methods, but it is instead attributable to the “coarse-grid time domain limit” phenomenon for $\epsilon = 1.0$ which also causes two-grid MGRIT time grid hierarchies (i.e., $\{4096_S, x_A\}$ where $x = \{2, 4, 8, 16\}$ for $t_f = 5.0$ and $x = \{2, 4, 8, 16, 32\}$ for $t_f = 10.0$) to diverge. This phenomenon is addressed in detail in Section 6.7.

When $\epsilon = .01$, SAS-MGRIT is stable for smaller time domains but becomes unstable for larger time domains, whereas SAA-MGRIT is convergent for every examined time grid hierarchy. It is conceivable that using a time integration technique other than the asymptotic solver Φ_Δ or the Strang splitting solver Φ on the third grid would yield convergent behavior for larger time domains and $\epsilon = .01$ under the given time grids. An interesting future direction would be the investigation

Chapter 6. Numerical Results

of a third-grid-specific coarse grid operator that enables stability for longer time steps. We denote this yet-undetermined three-level asymptotic MGRIT method “Strang-asymptotic-X MGRIT”, or SAX-MGRIT, where X is some to-be-discovered suitable time integration method for the coarsest time grid.

	{4096, 64, 2}	{4096, 64, 4}	{4096, 64, 16}	{4096, 64, 32}	(ref) {4096 _S , 64 _A }
$t_f = 1.0$	SAA: 4 (5.2s) SAS: 4 (5.0s)	SAA: 6 (5.4s) SAS: 6 (5.1s)	SAA: 3 (2.3s) SAS: 5 (2.7s)	SAA: 3 (3.0s) SAS: 5 (2.6s)	3 (4.1s)
$t_f = 5.0$	**	**	SAA: 14 (10.3s) SAS: 15 (8.2s)	SAA: 12 (12.3s) SAS: 16 (8.7s)	5 (7.1s)
$t_f = 10.0$	**	**	**	**	12 (16.9s)

Table 6.3: $\epsilon = 1$ comparison of three-level MGRIT solvers using either a “Strang-asymptotic-asymptotic” (SAA) or “Strang-asymptotic-Strang” (SAS) method of adding a third grid to a two-level asymptotic MGRIT solve with the structure {4096_S, 64_A}. F-relaxation is used. Runs that fail to converge are marked with “**”. Columns are ordered by the number of coarse points N_3 on the third grid, except for the rightmost column which provides the reference iteration count and wall time for the two-level {4096_S, 64_A} solve.

	{4096, 64, 2}	{4096, 64, 4}	{4096, 64, 16}	{4096, 64, 32}	(ref) {4096 _S , 64 _A }
$t_f = 1.0$	SAA: 4 (5.5s) SAS: 5 (6.6s)	SAA: 5 (4.7s) SAS: 6 (5.2s)	SAA: 6 (4.6s) SAS: 7 (3.7s)	SAA: 5 (5.0s) SAS: 6 (3.0s)	4 (5.7s)
$t_f = 5.0$	SAA: 11 (24.5s) SAS: 16 (22.6s)	SAA: 11 (17.5s) SAS: **	SAA: 11 (13.5s) SAS: **	SAA: 16 (18.5s) SAS: 23 (12.6s)	11 (16.1s)
$t_f = 10.0$	SAA: 21 (90.5s) SAS: **	SAA: 21 (60.8s) SAS: **	SAA: 11 (22.1s) SAS: **	SAA: 10 (19.1s) SAS: **	21 (38.2s)

Table 6.4: $\epsilon = .01$ comparison of three-level MGRIT solvers using either a “Strang-asymptotic-asymptotic” (SAA) or “Strang-asymptotic-Strang” (SAS) method of adding a third grid to a two-level asymptotic MGRIT solve with the structure {4096_S, 64_A}. F-relaxation is used. Runs that fail to converge are marked with “**”. Columns are ordered by the number of coarse points N_3 on the third grid, except for the rightmost column which provides the reference iteration count and wall time for the two-level {4096_S, 64_A} solve.

6.4 Asymptotic MGRIT: Two-Level With FCF-Relaxation

Here, we investigate whether FCF-relaxation can outperform F-relaxation in a two-grid asymptotic MGRIT solve. FCF-relaxation has previously shown a benefit (sometimes significant) over F-relaxation, e.g., in [10, 19, 20], with the paper [5] giving a theoretical justification for this benefit. Essentially, FCF-relaxation effectively damps all error modes that correspond to an eigenvalue of the error propagator that is significantly less than 1.

The setup for the experiments in this section is as follows. We consider $\epsilon = .01$ and time domains of the size $[0, t_f] \in \{[0.0, 10.0], [0.0, 15.0], [0.0, 20.0], [0.0, 25.0]\}$. We consider the two-level case with $N_\Delta = 16$ time points on the coarsest grid, and coarsening factors of $m = \{16, 128, 256\}$, which yields $N = \{256, 2048, 4096\}$ time points on the fine grid. We choose these coarse grid sizes and coarsening factors to follow the choices of coarse grid sizes used in previous studies of asymptotic parareal [18, 32, 31].

The residual halting tolerance is chosen to be $(1.0e-8)/\sqrt{\Delta t}$, and we note that standard MGRIT (i.e., with the use of Φ instead of Φ_Δ on the coarse grid) fails in every run to achieve convergence for this battery of tests and thus these results are omitted. The iteration counts for the various runs are shown in Table 6.5. Corresponding wall times are given in Table 6.6. The wall times are much faster for FCF-relaxation, which is a surprise given the iteration counts. Generally, FCF-relaxation is twice as expensive as F-relaxation. Thus, one would expect that an over 50% savings in iterations by FCF-relaxation would be required to see a runtime benefit. Thus, these results are unusual for MGRIT, and the reason for this wall time discrepancy is explained in Section 6.4.1.

Chapter 6. Numerical Results

t_f	F, $m = 16$	FCF, $m = 16$	F, $m = 128$	FCF, $m = 128$	F, $m = 256$	FCF, $m = 256$
10	14	10	10	7	10	7
15	15	9	16	8	15	8
20	18	10	17	8	17	8
25	**	**	14	9	13	9

Table 6.5: Iteration counts for 2-level asymptotic MGRIT runs, $\epsilon = .01$, using the coarsening factors m and relaxation schemes provided in the top row of each column. These results are “strange” when compared with the wall times in Table 6.6, in that we do not generally expect improvements in wall time from FCF-relaxation unless we are cutting down on iteration counts far more than in half when compared to F-relaxation iteration counts. Runs that fail to converge are marked with “**”.

t_f	F, $m = 16$	FCF, $m = 16$	F, $m = 128$	FCF, $m = 128$	F, $m = 256$	FCF, $m = 256$
10	22	16	18	13	19	15
15	36	22	42	21	43	23
20	61	34	60	29	62	31
25	**	**	63	42	64	44

Table 6.6: Wall times (using 16 cores) for asymptotic MGRIT runs, $\epsilon = .01$, using the coarsening factors m and relaxation schemes provided in the top row of each column. These results are “strange” when compared with the iteration counts in Table 6.5, in that we do not generally expect improvements in wall time from FCF-relaxation, unless we are cutting down on iteration counts far more than 50% when compared to F-relaxation iteration counts. Runs that fail to converge are marked with “**”.

6.4.1 Investigating the Wall-Time Discrepancy for FCF-relaxation

When the coarse grid operator Φ_Δ is computationally cheap, the dominant cost of one Braid iteration is the application of F-relaxation. In such a case, it is reasonable to assume that one Braid iteration using FCF-relaxation will be roughly twice as expensive as one Braid iteration using F-relaxation, as FCF-relaxation performs two sweeps of F-relaxation. However, the wall times in Table 6.6 indicate that asymptotic MGRIT wall times are primarily a function of the iteration counts seen in Table 6.5, regardless of the relaxation strategy. In particular, we see that in the problems where FCF-relaxation halves the iteration count, the wall times are roughly halved. Here,

Chapter 6. Numerical Results

we show the results of a performance study that explain the observations in Table 6.6.

First, we use the Python function `time.perf_counter()` to measure the wall time taken to execute the `my_Step()` function (i.e the Φ operator) in Cyclobraid. Since F- and FCF-relaxation runs for the same problem require different numbers of `my_Step()` calls, we first look at an average of the wall clock times for the `my_Step()` calls between the F- and FCF-relaxation runs.

Let s_f and s_{fcf} be the total number of `my_Step()` calls for an F- and FCF-relaxation Braid run, respectively. Then, let the wall clock times for the k th `my_Step()` call during F- and FCF-relaxation runs, respectively, be w_f^k , for $k = 1, 2, \dots, s_f$ and w_{fcf}^k , $k = 1, 2, \dots, s_{fcf}$. Then, let the sums of the wall clock times for the `my_Step()` calls be:

$$\omega_f = \sum_{k=1}^{s_f} w_f^k \quad \text{and} \quad \omega_{fcf} = \sum_{k=1}^{s_{fcf}} w_{fcf}^k. \quad (6.3)$$

Finally, the averages of the wall clock times for `my_Step()` are given by:

$$a_f = \frac{\omega_f}{s_f} \quad \text{and} \quad a_{fcf} = \frac{\omega_{fcf}}{s_{fcf}}. \quad (6.4)$$

These quantities are all measured by `time.perf_counter()`.

We first consider a problem with $\Delta t = \frac{1}{2048}$ on the fine grid, a coarsening factor $m = 2$, and a total of 8 time points on the fine grid. Our goal is to understand if the cost of Φ_Δ is negligible, i.e., whether F-/FCF-relaxation is truly the dominant computational cost here. For our first experiment, we therefore avoid using the asymptotic coarse grid Φ_Δ , and instead use the fine-grid Φ (Algorithm 5) for Φ_Δ . In this setting, the timings yield almost identical averages,

$$s_f = 52, \quad \omega_f = .034, \quad a_f = .00060, \quad \text{and}$$

$$s_{fcf} = 68, \quad \omega_{fcf} = .041, \quad a_{fcf} = .00061.$$

Next for comparison, we use asymptotic MGRIT, i.e., Algorithm 6 on the coarse grid.

Chapter 6. Numerical Results

In this setting, the timings yield rather different average timings,

$$s_f = 72, \omega_f = .38, a_f = .0053, \text{ and}$$

$$s_{fcf} = 96, \omega_{fcf} = .39, a_{fcf} = .0044.$$

The average wall clock times a_f and a_{fcf} indicate that the cost of Algorithm 6 on the coarse grid is not negligible. Thus, we investigate further.

Next, we let $\omega_{0,f}, \omega_{1,f}$ be the sum of the wall clock times for `my_Step()` only on Braid levels 0 and 1 (fine and coarse), $s_{0,f}, s_{1,f}$ be the total number of calls made to `my_Step()` on Braid levels 0 and 1, and $a_{0,f}, a_{1,f}$ be the average wall clock time for a `my_Step()` call on Braid levels 0 and 1, respectively, when using F-relaxation. The same notation applies to Braid runs with FCF-relaxation. We indeed see that the `my_Step()` calls on level 1 (i.e, when using the coarse propagator Φ_Δ) are roughly two orders of magnitude more expensive than `my_Step()` calls on level 0:

$$\omega_{0,f} = .022, \quad s_{0,f} = 44, \quad a_{0,f} = .0005$$

$$\omega_{1,f} = .363, \quad s_{1,f} = 28, \quad a_{1,f} = .013$$

$$\omega_{0,fcf} = .032, \quad s_{0,fcf} = 68, \quad a_{0,fcf} = .0005$$

$$\omega_{1,fcf} = .355, \quad s_{1,fcf} = 28, \quad a_{1,fcf} = .013$$

The dominant computational cost within a call to `my_Step()` on level 1 when using Algorithm 6 is the call to `RSWE_Direct.compute_average_force()`, which evaluates the integral for $\bar{\mathcal{N}}$ (see Equation (3.4)). Letting $a_{\bar{\mathcal{N}}}$ be the average wall time to call `RSWE_Direct.compute_average_force()`, our timings reported that $a_{\bar{\mathcal{N}}} = .006$. Since a call `my_Step()` (i.e., Φ_Δ) requires two calls to `compute_average_force()`, we have that $2a_{\bar{\mathcal{N}}} = .012$, and $2\frac{a_{\bar{\mathcal{N}}}}{a_{1,f}/fcf} = \frac{.012}{.013} \approx .92$. That is, roughly 92% of the run-time for `my_Step()` on the coarse level is attributable to this computation, making

this time-averaging computation the dominant cost overall for an asymptotic MGRIT iteration.

One can see why this computation is so expensive. The dominant cost of a call to `RSWE_Direct.compute_average_force()` is a call to the nonlinear multiplication routine. In order to compute the nonlinear terms for the time-averaged quantity in Line 3 of Algorithm 4, it is necessary to perform the multiplications to find the terms of $\mathbf{N}(\mathbf{u})$ in real space. So, one inverse Fourier transform and one forward Fourier transform is necessary per quadrature point when computing $\overline{\mathcal{N}}$. With M total quadrature points, $2M$ Fourier transforms are necessary per call to `compute_average_force()`.

This cost can be mitigated, however. The `for` loop in Algorithm 4 can be parallelized, so if at least M idle processors are available, then a speedup of roughly M would be available when executing Algorithm 6 (i.e., Φ_Δ). In this case, the cost of Φ_Δ , F-, and FCF-relaxation would be more balanced, and further performance analysis would be required. However in our case, compute time is primarily a function of the iteration count, and the use of FCF-relaxation is advised given the large savings in wall time. We note that this preference for FCF is novel and should be reflected often in practice, as idle processors are not always available.

6.5 Accelerating $\bar{\mathbf{u}}$ Computations by Reusing Values

In instances where Φ or Φ_Δ is computationally expensive, one technique is to reuse information computed from previous MGRIT iterations or from previous MGRIT levels, see [11, 20]. As shown in Section 6.4.1, the dominant cost of an asymptotic MGRIT iteration is the application of the coarse grid operator Φ_Δ , particularly the forward and inverse Fourier transforms used to compute nonlinear quantities

in Algorithm 4. Here, we examine strategies for reusing the time-averaged coarse grid solution $\bar{\mathbf{u}}$, as well as recomputing $\bar{\mathbf{u}}$ using stale values of the computationally expensive nonlinear quantities $\bar{\mathcal{N}}$ at certain iterations in asymptotic MGRIT.

We fix a coarse time grid of $N_\Delta = 50$ time points, a fine grid of $N = 50,000$ time points, time domain $T = [0, 5]$, and consider $\epsilon = 0.01$ as well as $\epsilon = 1$. F-relaxation is used.

6.5.1 Reuse of Stale Coarse Grid Solutions $\bar{\mathbf{u}}$

We examine the residual histories over 10 Braid iterations and compare reuse of $\bar{\mathbf{u}}$ to a baseline computation where no re-use is performed. We look at four schemes for reusing $\bar{\mathbf{u}}$ from the previous iteration:

1. At iteration $j = 9$, reuse $\bar{\mathbf{u}}$ computed at iteration $j = 8$. All other iterations are computed without reuse.
2. At iterations $j = \{8, 9\}$ reuse $\bar{\mathbf{u}}$ computed at iteration $j = 7$. All other iterations are computed without reuse.
3. At all odd iterations $j = \{1, 3, 5, 7, 9\}$, reuse $\bar{\mathbf{u}}$ computed at the respective prior iterations $j = \{0, 2, 4, 6, 8\}$. All other iterations are computed without reuse.
4. At odd iterations $j = \{5, 7, 9\}$ reuse $\bar{\mathbf{u}}$ computed at respective prior iterations $j = \{4, 6, 8\}$. All other iterations are computed without reuse.

The residual histories shown in Figures 6.1 and 6.2 indicate that, while reuse of $\bar{\mathbf{u}}$ computed at previous iterations never causes divergent behavior or causes the residual to become larger, the convergence rate at the iterations associated with the reuse of $\bar{\mathbf{u}}$ is very close to 1 (i.e., almost nonconvergent). As a result, even though we skip the computation of $\bar{\mathbf{u}}$, the number of $\bar{\mathbf{u}}$ computations required to reach a given residual

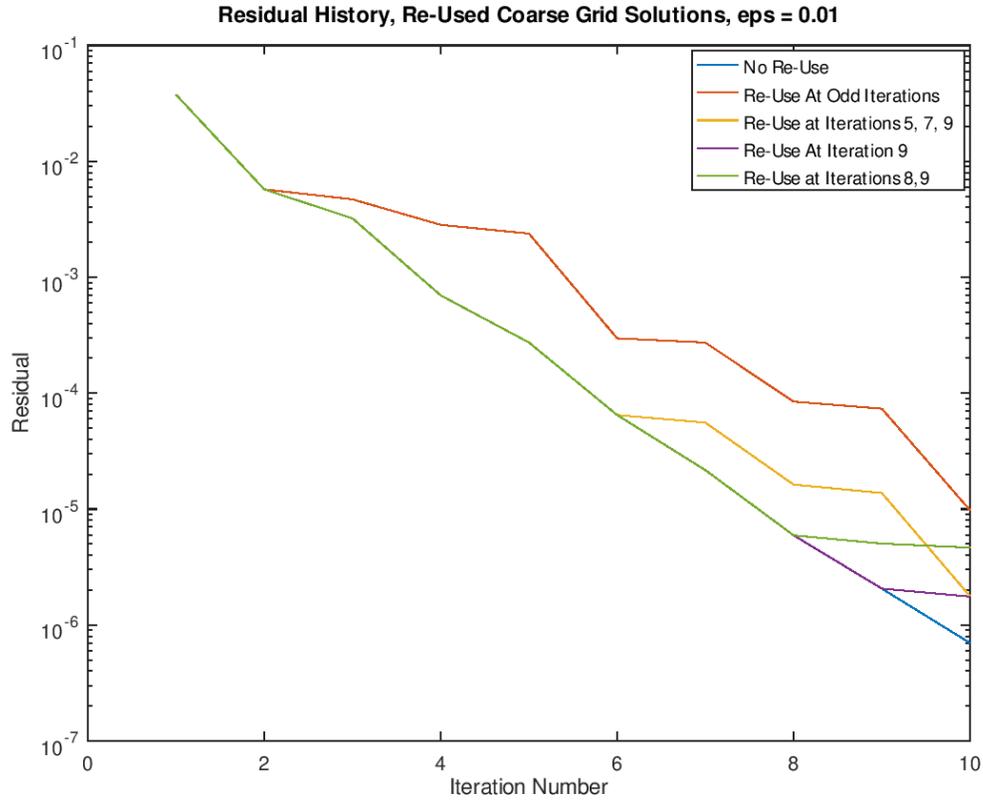


Figure 6.1: Comparison of residual histories for two-level asymptotic MGRIT solves using various schemes to reuse the slowly-varying coarse grid solution $\bar{\mathbf{u}}$ where $\epsilon = 0.01$. A coarse grid of $N_{\Delta} = 50$, a fine grid of $N = 50,000$, final time $t_f = 5.0$ and F-relaxation are used.

norm is equal to the number of $\bar{\mathbf{u}}$ computations required to reach that norm without skipping $\bar{\mathbf{u}}$ computations. Hence, the reuse of $\bar{\mathbf{u}}$ from previous iterations does not offer any speedup to asymptotic MGRIT. In fact, the work from extra iterations when reusing $\bar{\mathbf{u}}$ on all odd iterations (18.1s wall time) or odd iterations 5, 7, and 9 (16.2s wall time) negatively impacted the wall time to reach a halting tolerance of $2.0e-5$ when compared to the reference solver that did not reuse information (8.6s wall time) on a 16-core MPI run.

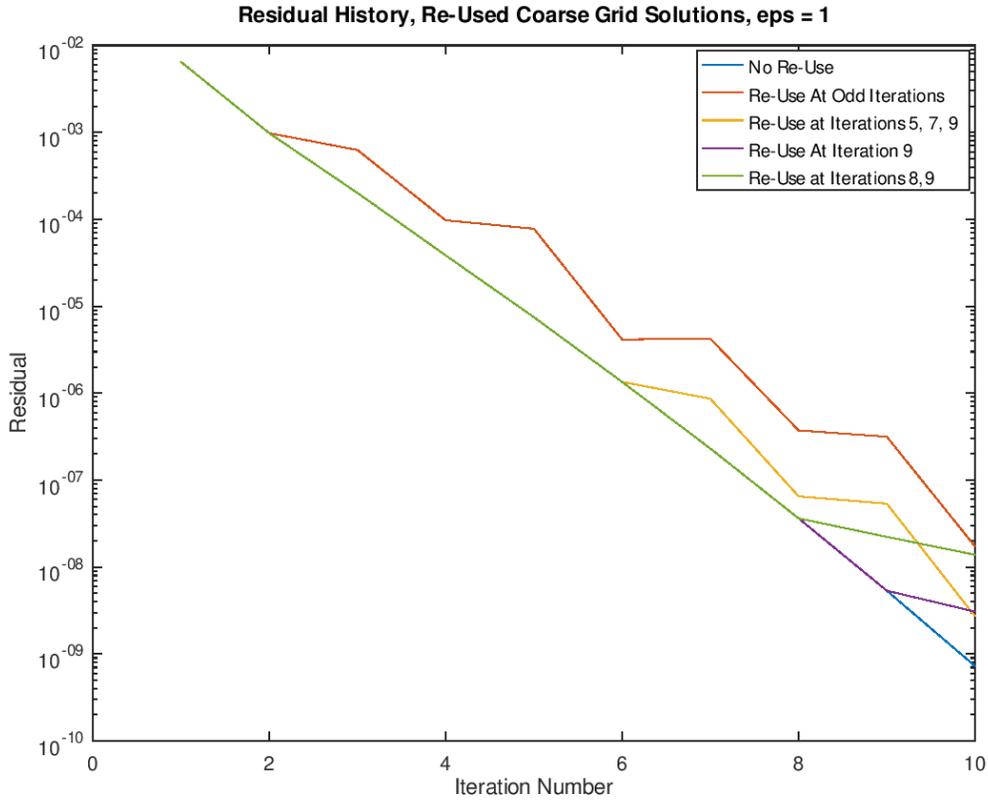


Figure 6.2: Comparison of residual histories for two-level asymptotic MGRIT solves using various schemes to reuse the slowly-varying coarse grid solution $\bar{\mathbf{u}}$ where $\epsilon = 1$. A coarse grid of $N_{\Delta} = 50$, a fine grid of $N = 50,000$, final time $t_f = 5.0$ and F-relaxation are used.

6.5.2 Computation of $\bar{\mathbf{u}}$ Based on Stale Nonlinear Quantities

$$\bar{\mathcal{N}}()$$

The routine used to integrate the coarse grid approximation $\bar{\mathbf{u}}$ is described in Algorithm 6 in Chapter 3. As discussed in Section 6.4.1, the majority of the cost of computing $\bar{\mathbf{u}}$ comes from the two $\bar{\mathcal{N}}()$ evaluations in Step 2 of Algorithm 6; in contrast, the exponential integration calls in Step 1 and Step 3 of Algorithm 6 are relatively cheap. Here, we consider a reuse scheme which uses stale values of the

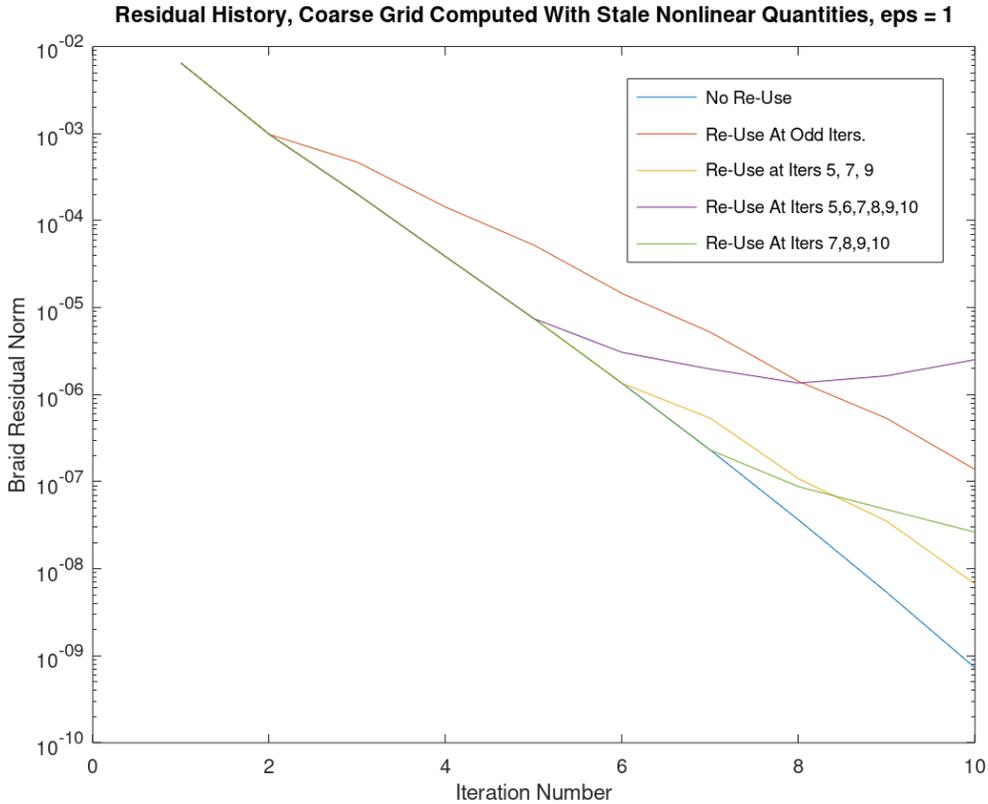


Figure 6.3: Comparison of residual norm histories for two-level asymptotic MGRIT solves using various techniques to recompute the coarse grid solution $\bar{\mathbf{u}}$ based on stale nonlinear quantities $\bar{\mathcal{N}}()$, where $\epsilon = 1$. A coarse grid of $N_{\Delta} = 50$, a fine grid of $N = 50,000$, final time $t_f = 5.0$ and F-relaxation are used.

expensive $\bar{\mathcal{N}}(\hat{\mathbf{v}})$ and $\bar{\mathcal{N}}(\hat{\mathbf{v}} + \frac{\Delta T}{2}\mathbf{v})$ computations in Step 2 at certain iterations, but otherwise performing the time integration on the coarse grid as laid out in Algorithm 6.

To accomplish this, we store computed values of both $\bar{\mathcal{N}}(\hat{\mathbf{v}})$ and $\bar{\mathcal{N}}(\hat{\mathbf{v}} + \frac{\Delta T}{2}\mathbf{v})$ in a Python dictionary, indexed by the Braid iteration number and coarse time point index. This way, when performing coarse grid time integration with re-use, the previously computed $\bar{\mathcal{N}}()$ values from a previous iteration can be accessed and substituted in place of the calls to the $\bar{\mathcal{N}}()$ time-averaging in Step 2. We consider

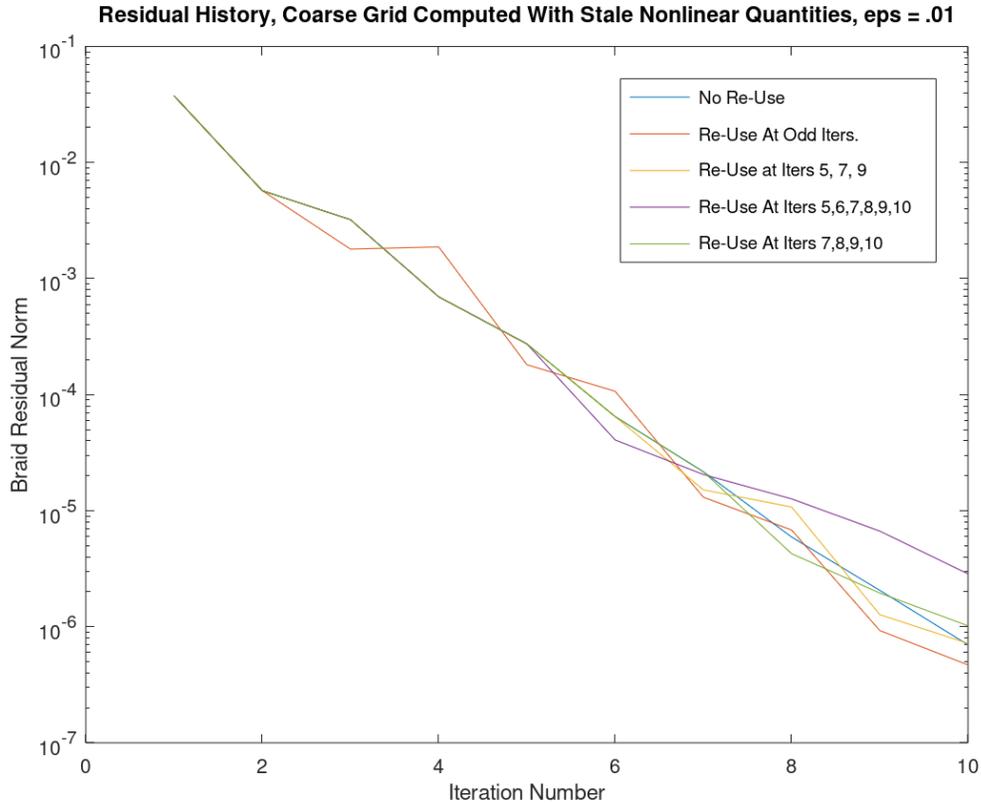


Figure 6.4: Comparison of residual norm histories for two-level asymptotic MGRIT solves using various techniques to recompute the coarse grid solution $\bar{\mathbf{u}}$ based on stale nonlinear quantities $\bar{\mathcal{N}}()$, where $\epsilon = 0.01$. A coarse grid of $N_{\Delta} = 50$, a fine grid of $N = 50,000$, final time $t_f = 5.0$ and F-relaxation are used.

four reuse patterns and compare them to a baseline solve without reuse, over Braid iterations $j = \{0, 1, 2, \dots, 9, 10\}$.

1. At iterations $j = \{7, 8, 9, 10\}$, recompute $\bar{\mathbf{u}}$ by reusing nonlinear terms computed at iteration $j = 6$. All other iterations are computed without reuse.
2. At iterations $j = \{5, 6, 7, 8, 9, 10\}$ recompute $\bar{\mathbf{u}}$ by reusing nonlinear terms computed at iteration $j = 4$. All other iterations are computed without reuse.
3. At all odd iterations $j = \{1, 3, 5, 7, 9\}$, recompute $\bar{\mathbf{u}}$ by reusing nonlinear terms

computed at the respective prior iterations $j = \{0, 2, 4, 6, 8\}$. All other iterations are computed without reuse.

4. At odd iterations $j = \{5, 7, 9\}$ recompute $\bar{\mathbf{u}}$ by reusing nonlinear terms computed at respective prior iterations $j = \{4, 6, 8\}$. All other iterations are computed without reuse.

In Figure 6.4, we find that for $\epsilon = .01$, all of the $\bar{\mathcal{N}}()$ reuse patterns we consider do not incur a convergence penalty of more than one order of magnitude when compared to reference solution. More strikingly, we see that for the schemes that perform $\bar{\mathcal{N}}()$ reuse at all odd iterations, the residual norm at iteration 10 is marginally smaller than the reference solution.

However, for $\epsilon = 1$, the convergence penalty for reusing prior $\bar{\mathcal{N}}()$ terms is generally steeper. In this case, the slowing of convergence results in all cases of reuse leading to an order of magnitude or greater penalty in residual norm convergence at iteration 10, when compared with the reference solution.

In Table 6.7, we see that the average wall time of a call to the timestepping routine is roughly halved when $\bar{\mathcal{N}}()$ terms are reused in accordance with the even-odd alternating $\bar{\mathcal{N}}()$ reuse pattern, where $\bar{\mathcal{N}}()$ terms are reused from the prior iteration at iterations $j = \{1, 3, 5, 7, 9\}$ over a Braid run with iterations $j = 0, 1, 2, \dots, 10$ (i.e., 5 of 11 iterations reuse $\bar{\mathcal{N}}()$ terms for the coarse grid computation.) This holds true for both $\epsilon = .01$ and $\epsilon = 1$.

6.6 Three-Scale RSWE

Up to this point, our research has considered the case where the Rossby radius of deformation $F = 1$. Remember $\epsilon = Ro = F^{-1/2}Fr$, so with $\epsilon = 1$, we get $Ro = Fr$ (Ro is the Rossby number and Fr is the Froude number). In this instance,

Chapter 6. Numerical Results

	$\epsilon = .01$	$\epsilon = 1$
Reuse	$a_{f,0} = .00045$ $a_{f,1} = .00682$	$a_{f,0} = .00045$ $a_{f,1} = .00676$
No Reuse	$a_{f,0} = .00047$ $a_{f,1} = .01335$	$a_{f,0} = .00047$ $a_{f,1} = .01273$

Table 6.7: Average wall times on levels 0, 1 for $\overline{\mathcal{N}}()$ reuse. The reuse pattern follows using $\overline{\mathcal{N}}()$ terms from the prior iteration at iterations $j = \{1, 3, 5, 7, 9\}$ over a Braid run with iterations $j = 0, 1, 2, \dots, 10$. As with the other experiments in this section, F-relaxation is used as indicated by the f subscripts. Here, $\mathcal{N}_\Delta = 50$ and $t_f = 5.0$.

the timescales of the rotational and gravitational effects are the same. This is not typically true for geophysical applications, and deformation radii F vary dramatically in oceanic flows depending on parameters, including but not limited to, latitude, density stratification and seabed depth. Thus, we are also interested in cases where $F \neq 1$.

To explore these cases, Peddle demonstrates in [31] that the linear term in Equation (3.1) can be written as

$$\frac{1}{\epsilon} \mathcal{L}\mathbf{u} = \frac{1}{Ro} \mathcal{R}\mathbf{u} + \frac{1}{Fr} \mathcal{G}\mathbf{u}, \quad (6.5)$$

and the governing Equation (3.1) can be rewritten in the form

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{1}{Ro} \mathcal{R}\mathbf{u} + \frac{1}{Fr} \mathcal{G}\mathbf{u} = \mathcal{N}(\mathbf{u}) + \mathcal{D}\mathbf{u}. \quad (6.6)$$

As a result, there are three timescales, two fast linear scales (\mathcal{R} and \mathcal{G}) and one slow nonlinear scale ($\mathcal{N}(\mathbf{u})$). Peddle conjectures that in the instance where $F = 10^{-4}$, corresponding to rotational effects being two orders of magnitude larger than gravitational effects, there are two optimal choices for η in Algorithm 4. The two choices correspond to optimal averaging for the gravitational operator \mathcal{G} and the rotational operator \mathcal{R} . In contrast, when $F = 1$ or $F = 10^4$, Figures 6.5 and 6.6 demonstrate that the ideal averaging window for MGRIT convergence is clearly unique at $\eta = \Delta T$. Interestingly, the case where $F = 10^4$ does not have two minima despite the timescales of \mathcal{G} and \mathcal{R} being separated, which is unexplained at this time.

Chapter 6. Numerical Results

The asymptotic parareal method is designed for problems where there is only one fast timescale, meaning that the algorithm uses only one averaging window η . Here, we give the result of experiments that determine optimal η with respect to MGRIT convergence for a problem where $F = 10^{-4}$ and look at possible techniques to handle three scales in asymptotic MGRIT. In particular, we investigate whether multiple coarse grids in MGRIT are beneficial for capturing these two separate fast scales.

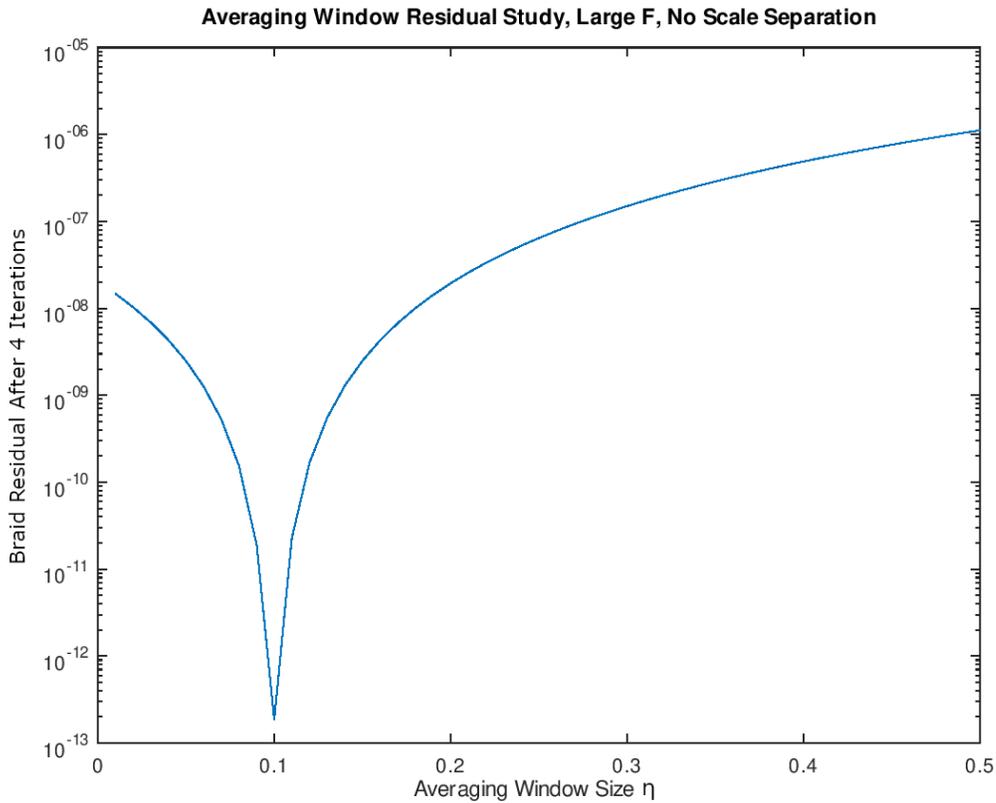


Figure 6.5: $F = 10^4$, $\epsilon = 1$, $T = [0, 5]$, fine grid $N = 50000$, coarse grid $N_\Delta = 50$

First, we determine more precisely the two optimal choices of η for a given model problem where $\epsilon = 1$ and $F = 10^{-4}$. We use a time domain of $[0, 5]$ with a time discretization of $N = 50000$ fine grid time points and $N_\Delta = 50$ coarse grid time points. We also test whether the optimal choices of η are sensitive to the current XBraid

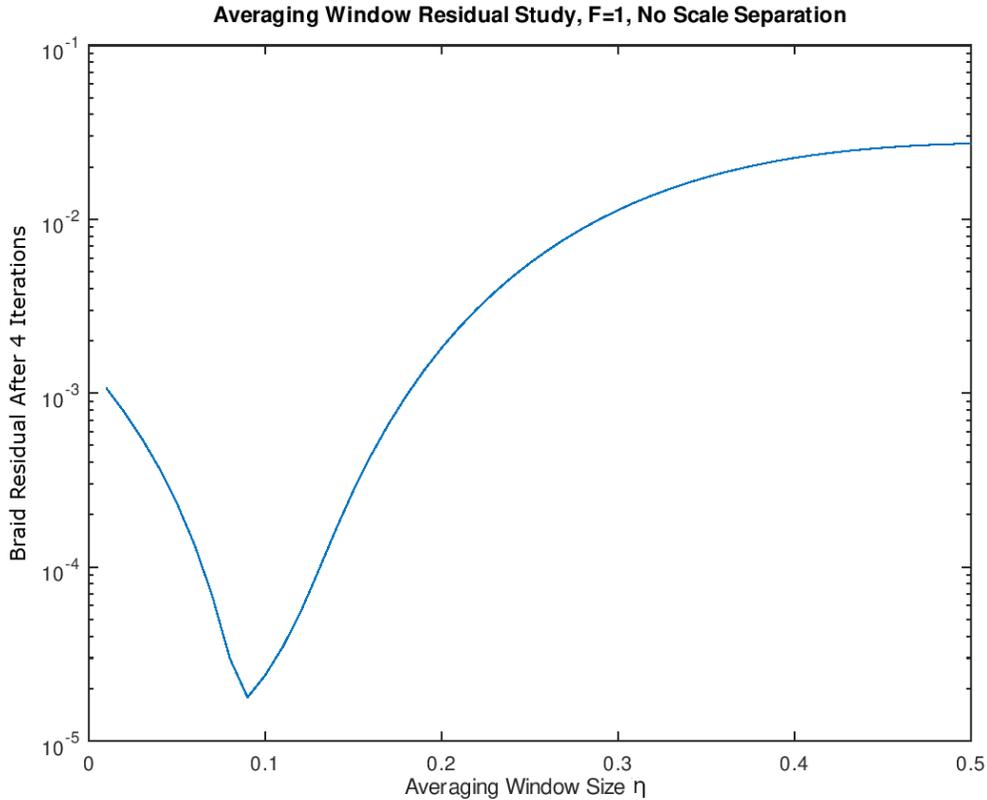


Figure 6.6: $F = 1$, $\epsilon = 1$, $T = [0, 5]$, fine grid $N = 50000$, coarse grid $N_{\Delta} = 50$

iteration and choice of F-relaxation or FCF-relaxation. To examine this, Figures 6.7 and 6.8 depict the Braid residual magnitude after a fixed number of iterations as η varies for F- and FCF-relaxation, respectively. More effective η values are expected to result in larger drops in the Braid residual size. The dips in measured residual at $\eta = .15, \eta = .32$ are more pronounced at later iterations and do not shift as a consequence of which relaxation scheme is chosen. This is indicative of possible η values for use in practice.

Assuming that the two optimal $\eta = .15$ and $\eta = .32$ correspond to optimal averaging values for the gravitational operator \mathcal{G} and rotational operator \mathcal{R} , we consider the following three-scale asymptotic MGRIT schemes.

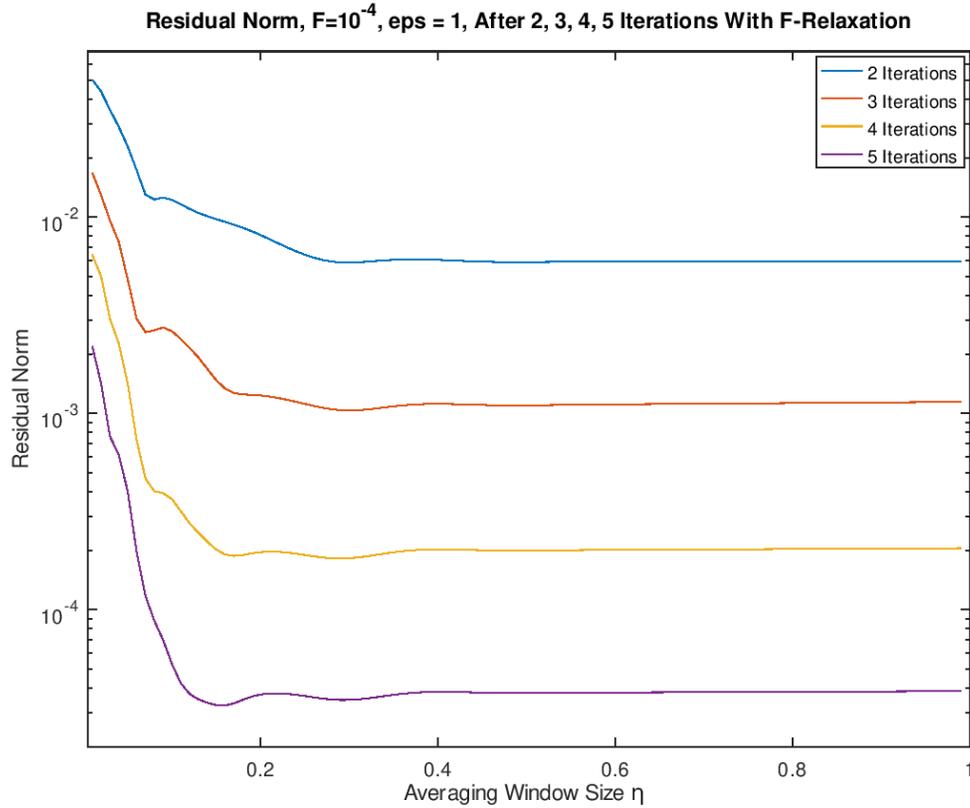


Figure 6.7: Braid residual after 2, 3, 4, and 5 iterations for $\epsilon = 1$, $F = 10^{-4}$, and F-relaxation. Choice of η is given on the x -axis and residual norm measured after the indicated number of iterations is given on the y -axis. Note that the coarse grid $\Delta T = 0.1$.

1. Two-level solver that alternates between averaging using $\eta = .15$ on odd iterations and $\eta = .32$ on even iterations.
2. Three-level solver where the two coarse grids have equal size (e.g., level 0 has 50000 time-points and levels 1 and 2 both have 50 time-points). We use $\eta = .15$ on level 1 and $\eta = .32$ on level 2.

In Table 6.8 we see that alternating η to be .15 on even iterations and .32 on odd iterations yields a marginally smaller residual after both 4 and 8 iterations than

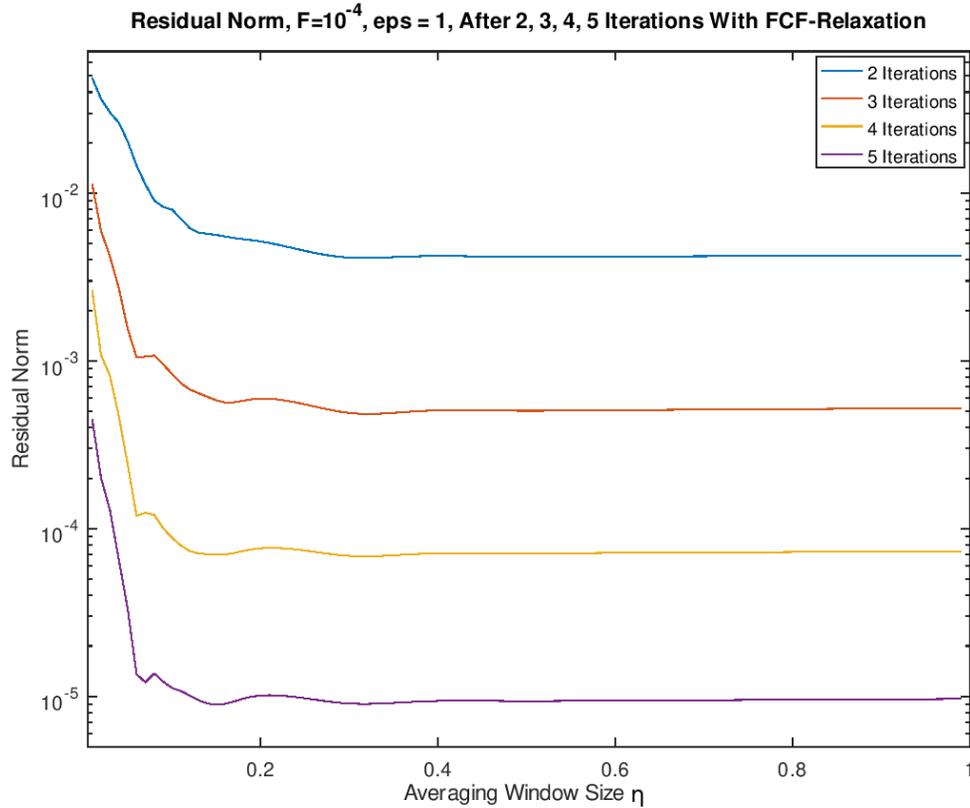


Figure 6.8: Braid residual after 2, 3, 4, and 5 iterations for $\epsilon = 1$, $F = 10^{-4}$, and FCF-relaxation. Choice of η is given on the x -axis and residual norm measured after the indicated number of iterations is given on the y -axis. Note that the coarse grid $\Delta T = 0.1$.

using either $\eta = .15$ or $\eta = .32$ exclusively. On the other hand, using $\eta = .32$ on even iterations and $\eta = .15$ on odd iterations is slightly superior to using $\eta = .32$ exclusively but not to using $\eta = .15$ exclusively. Furthermore, the three-grid experiments have residual histories that comport exactly to our two-grid experiments, depending on which value of η was chosen on level 2. The three-grid solve that uses $\eta = .32$ on level 2 yields an identical result to the two-grid solve that uses $\eta = .32$ on level 1. Likewise, the three-grid solve that uses $\eta = .15$ on level 2 yields an identical result to the two-grid solve that uses $\eta = .15$ on level 1. This could be due to the work done

Solving Technique	Residual After 4 Iterations	Residual After 8 Iterations
Two-Grid, $\eta = .15$ on all iterations	7.156544e-03	1.570172e-04
Two-Grid, $\eta = .32$ on all iterations	9.273075e-03	2.720488e-04
Two-Grid, $\eta = .15$ on even iterations, $\eta = .32$ on odd iterations	7.018559e-03	1.508769e-04
Two-Grid, $\eta = .15$ on odd iterations, $\eta = .32$ on even iterations	8.928996e-03	2.389517e-04
Three-Grid, level 1: $N_2 = 50$, $\eta = .15$ level 2: $N_3 = 50$, $\eta = .15$	7.156544e-03	1.570172e-04
Three-Grid, level 1: $N_2 = 50$, $\eta = .32$ level 2: $N_3 = 50$, $\eta = .32$	9.273075e-03	2.720488e-04
Three-Grid, level 1: $N_2 = 50$, $\eta = .15$ level 2: $N_3 = 50$, $\eta = .32$	9.273075e-03	2.720488e-04
Three-Grid, level 1: $N_2 = 50$, $\eta = .32$ level 2: $N_3 = 50$, $\eta = .15$	7.156544e-03	1.570172e-04

Table 6.8: Braid residuals after 4 and 8 iterations for various two- and three-grid techniques when $F = 10^{-4}$ and $\epsilon = 1$.

on level 1 in the three-grid case essentially being overwritten by the information from level 2 as relaxation is applied after interpolating from level 1 to level 2 during the up-cycle of MGRIT.

While the two-grid method that uses $\eta = .15$ on even iterations and $\eta = .32$ on odd iterations performs slightly better than any other method investigated here, the magnitude of the dips about the minima seen in Tables 6.7 and 6.8 are very small. Thus, it is not surprising that the MGRIT performance gains are also very small. In [31], Peddle proposes that an alternative wave-averaging method would be a necessary component to constructing a solver specifically tailored to the three-scale RSWE. If such an averaging method is found, its use in a multilevel MGRIT setting would bear

investigation.

6.7 The $\epsilon = 1$ Case: On the Use of CF-relaxation Sweeps and Larger Averaging Values η to Allow For More Efficient Coarse Grids

For the rotating shallow water equations, the case where $\epsilon = \mathcal{O}(1)$ corresponds to regimes in geophysical flows where gravity waves are not prominent and the behavior of the flow is influenced instead by slow Rossby waves. Thus, we want to improve the solver for this physically relevant case. In [18, 31, 32], attention is paid to the case where $\epsilon = 1$. In particular, it is shown in numerical experiments that the performance of asymptotic parareal is roughly equivalent, but not superior, to standard parareal for the rotating shallow water equations in terms of convergence. Thus, this is a difficult problem for asymptotic parareal. We demonstrate in this section that two-level asymptotic MGRIT can achieve convergence where standard and asymptotic parareal cannot when using an averaging window $\eta > \Delta T$ and an increased number of CF-sweeps in the relaxation scheme.

We choose the following parameters similar to [18, 32, 31], in particular the coarse and fine grid sizes are the same as found in much of [31]. The smallest time domain sizes of $t_f = \{1, 4\}$ are among the time domain sizes used in the previous literature to show convergence, and we introduce larger time domains thereafter to demonstrate where divergence occurs. Using $\epsilon = 1$ and a fixed coarse grid of $N_\Delta = 50$ time points and fine grid of $N = 50000$ time points, we vary the time domain $T = [0, t_f]$ where $t_f = \{1, 4, 8, 12, 16, 20\}$. We also vary the size of the averaging window, using Equation (6.1) and choosing the coefficient $\alpha = \{1, 4, 12, 16, 20\}$. Thus as α increases, the size of the averaging window also increases commensurately. Equation (6.2) is used with

Chapter 6. Numerical Results

$\beta = 1$ to determine the number of quadrature points \overline{M} used in the computation of the slowly-varying term $\overline{\mathcal{N}}$ in Algorithm 4. The result is that the number of quadrature points scales proportionally with the size of the averaging window. In Tables 6.9 and 6.10, iteration counts to reach a halting tolerance of $1.0\text{e-}6$ and associated wall times are shown for time domains of $T = [0, t_f]$, for $t_f = \{1, 4, 8, 12, 16, 20\}$ and various $F(\text{CF})^n$ -relaxation strategies of $n = 0, 1, 2, 3, 4$.

The iteration counts shown in Table 6.9 do not change when using fine grid sizes $N = \{10000, 25000, 50000, 75000, 100000\}$, thus we show only results for $N = 50000$. We also see in Table 6.9 that in the $\epsilon = 1$ case, an increased number of CF-relaxation sweeps and an increased averaging window length are both individually capable of allowing two-level asymptotic MGRIT to converge over significantly longer time domains when using a 50 point coarse time grid, yielding convergence for $t_f = 12.0$. When increasing both the number of CF-sweeps and averaging window size, it is possible to achieve convergence for $t_f = 16.0$ with a 50 point coarse time grid. The number of CF-relaxation sweeps to achieve convergence for a time domain of $t_f = 16.0$ reaches a minimum of 2 at $\alpha = 12.0$, but for other values of α , more CF-sweeps are necessary to achieve convergence. This suggests that there is an optimal averaging window length $\eta > \Delta T$ for large time domains that would ordinarily diverge in asymptotic parareal where additional CF-relaxations are not an option.

Our experiments suggest the existence of a “coarse grid time domain size limit” in asymptotic MGRIT that prevents taking large coarse timesteps when $\epsilon = 1$, even when a problem with the same coarse timestep size converges for $\epsilon = 10^{-2}$. In Section 6.3 and Table 6.3, we saw that when $\epsilon = 1$, the introduction of a coarse grid with 4 or 2 points when $t_f = 5.0$ results in divergence with SAA-MGRIT and SAS-MGRIT. The same divergence occurs when a coarse grid with 32 points or fewer is used with $t_f = 10.0$. In contrast, we see in Table 6.4 the introduction of the same-sized coarse grids when $\epsilon = .01$ converges with SAA-MGRIT. We also see this in Table 6.9 where

increasing the time domain over a fixed coarse grid size $N_\Delta = 50$ eventually results in divergence when $\epsilon = 1$, although we see in Table 6.5 how a coarse grid of $N_\Delta = 16$ points and $\epsilon = 0.1$ works well with asymptotic MGRIT and much larger time domain sizes. In summary, each of these cases has a limit to how small N_Δ can be before MGRIT convergence stops.

The wall timings given in Table 6.10 further illustrate the findings in Section 6.4.1 that suggest the relationship between CF-sweeps and overall wall time depend primarily on the computational cost of the coarse grid operator Φ_Δ relative to the cost of relaxation. When $\alpha = 1$ and $\alpha = 4$, increasing the number of CF-sweeps increases the overall wall time, even when the iteration count is decreased by the extra relaxation. Since the number of points \bar{M} in the midpoint quadrature computation is small, the coarse grid operator Φ_Δ is cheap and the dominant cost of a Braid iteration in this case is the application of relaxation. However, for $\alpha = 12$ and above, \bar{M} is increased to the point where the dominant cost of a Braid iteration is the application of the coarse grid operator Φ_Δ . Here, reductions in iteration count as a result of the increased CF-sweeps on larger time domains translate to a reduction in wall time.

For instance, when $\alpha = 1$, although increasing the number of CF-sweeps from $n = 2$ to $n = 4$ reduces the iteration count required for convergence when $t_f = 12.0$ from 12 to 9, it also increases the wall time from 74.3 seconds to 90.0 seconds. The average wall time per iteration increases from 6.2 seconds when $n = 2$ to 10.0 seconds when $n = 4$; hence when $n = 4$, a Braid iteration is on average 38% more expensive than when $n = 2$. However, when $\alpha = 12$ and $t_f = 16$, 16 iterations are required to reach convergence with $n = 2$ and 11 iterations are required with $n = 4$. The corresponding wall times are 295.5 seconds (18.5 seconds per iteration) when $n = 2$ and 229.8 seconds (20.9 seconds per iteration) when $n = 4$, so a Braid iteration is merely 11% more expensive when $n = 4$.

We speculate that this MGRIT convergence benefit from larger α and n is due to

Chapter 6. Numerical Results

the eigenstructure of Φ . In [5], FCF-relaxation is shown to be effective in smoothing out error modes associated with eigenvalues of Φ that are $\ll 1$. Thus, we speculate that the troublesome error modes for MGRIT and longer time domains have an eigenvalue $\ll 1$. Enlarging α to 12 and 16 may also be having a positive effect by making the coarse grid Φ_Δ more accurate.

Chapter 6. Numerical Results

$\alpha = 1$	$t_f = 1.0$	4.0	8.0	12.0	16.0	20.0
$n = 0$	3	6	14	**	**	**
1	3	5	9	**	**	**
2	3	5	7	12	**	**
3	3	5	7	10	**	**
4	3	5	6	9	**	**

$\alpha = 4$	$t_f = 1.0$	4.0	8.0	12.0	16.0	20.0
$n = 0$	6	14	18	20	**	**
1	6	13	15	15	**	**
2	6	10	12	13	**	**
3	6	9	11	11	**	**
4	6	8	10	10	10	**

$\alpha = 12$	$t_f = 1.0$	4.0	8.0	12.0	16.0	20.0
$n = 0$	10	13	17	28	**	**
1	10	11	13	19	**	**
2	9	10	12	15	16	**
3	9	10	11	13	13	**
4	8	9	10	11	11	**

$\alpha = 16$	$t_f = 1.0$	4.0	8.0	12.0	16.0	20.0
$n = 0$	11	13	19	31	**	**
1	10	11	16	21	**	**
2	10	10	13	16	**	**
3	9	9	11	13	13	**
4	9	9	10	11	11	**

$\alpha = 20$	$t_f = 1.0$	4.0	8.0	12.0	16.0	20.0
$n = 0$	11	13	21	**	**	**
1	10	11	17	21	**	**
2	10	10	14	15	**	**
3	9	10	11	13	**	**
4	9	9	10	11	11	**

Table 6.9: Iteration counts for two-level asymptotic MGRIT solves of the RSWE. Here, the averaging window η and quadrature points are chosen using Equations (6.1) and (6.2), where α is given in the upper-left cell of each table and $\beta = 1$. The value $\epsilon = 1$ is used corresponding to no scale separation and the time grid size is fixed at $N_\Delta = 50$, $N = 50000$. Columns are ordered as the final time t_f varies from 1.0 to 20.0 as given by the top row. Rows are ordered by number of CF-relaxations applied in smoothing i.e., $F(CF)^n$ -relaxation where the left-most column gives n for each row. Runs that fail to converge are marked with “**”.

Chapter 6. Numerical Results

$\alpha = 1$	$t_f = 1.0$	4.0	8.0	12.0	16.0	20.0
$n = 0$	8.4	16.2	37.7	**	**	**
1	11.8	22.6	40.6	**	**	**
2	15.3	31.2	43.2	74.3	**	**
3	20.1	36.2	54.2	83.0	**	**
4	24.2	46.3	56.8	90.0	**	**

$\alpha = 4$	$t_f = 1.0$	4.0	8.0	12.0	16.0	20.0
$n = 0$	16.0	38.5	69.6	91.9	**	**
1	27.9	63.5	85.9	100.9	**	**
2	36.2	64.5	85.4	102.7	**	**
3	45.9	72.8	99.0	105.0	**	**
4	57.0	76.7	104.4	111.2	126.2	**

$\alpha = 12$	$t_f = 1.0$	4.0	8.0	12.0	16.0	20.0
$n = 0$	27.5	60.0	134.7	308.6	**	**
1	48.4	68.4	120.5	242.5	**	**
2	57.4	88.1	139.0	214.1	295.5	**
3	75.9	101.1	137.7	207.5	252.4	**
4	77.5	106.5	142.2	203.3	229.8	**

$\alpha = 16$	$t_f = 1.0$	4.0	8.0	12.0	16.0	20.0
$n = 0$	30.1	75.4	191.0	449.6	**	**
1	44.8	84.9	193.8	341.1	**	**
2	44.9	87.4	170.1	286.2	**	**
3	71.7	93.0	160.6	255.5	320.4	**
4	91.8	108.0	164.5	231.6	288.1	**

$\alpha = 20$	$t_f = 1.0$	4.0	8.0	12.0	16.0	20.0
$n = 0$	34.3	89.5	258.6	**	**	**
1	46.7	89.7	232.7	421.5	**	**
2	64.4	104.4	217.3	323.5	**	**
3	72.6	118.0	186.2	303.2	**	**
4	92.0	117.9	186.2	274.4	351.9	**

Table 6.10: MPI 16-core wall clock times for two-level asymptotic MGRIT solves of the RSWE. Here, the averaging window η and quadrature points are chosen using Equations (6.1) and (6.2), where α is given in the upper-left cell of each table and $\beta = 1$. The value $\epsilon = 1$ is used corresponding to no scale separation and the time grid size is fixed at $N_\Delta = 50$, $N = 50000$. Columns are ordered as the final time t_f varies from 1.0 to 20.0 as given by the top row. Rows are ordered by number of CF-relaxations applied in smoothing i.e., $F(CF)^n$ -relaxation where the left-most column gives n for each row. Runs that fail to converge are marked with “**”.

Chapter 7

Conclusions

In this thesis, we have developed and investigated variations of an asymptotic MGRIT approach for solving the RSWE parallel-in-time. These approaches are based on the the asymptotic parareal method, which uses the key component of an asymptotically time-averaged coarse grid timestepping scheme. The overall goal is to research improvements to asymptotic parareal by improving its overall convergence and its robustness to problem parameters, in particular ϵ and time-domain length.

In Section 6.3, we proposed the so-called SAS-MGRIT and SAA-MGRIT three-level techniques with the goal of introducing a nonintrusive and cheap third grid to the two-grid asymptotic MGRIT/parareal method. We note that for asymptotic parareal, a three-level scheme is novel. This strategy allows the expensive applications of the asymptotic coarse grid operator Φ_Δ on the second level (i.e., first coarse grid) to be parallelized. Thus, instead of solving a larger second level sequentially, asymptotic MGRIT instead solves a smaller third level (i.e., coarsest grid) sequentially. While SAS-MGRIT and SAA-MGRIT were successful in improving wall times for small time domains, we saw that SAS-MGRIT became divergent and SAA-MGRIT became inefficient (compared to two-level asymptotic MGRIT) for longer time domains. In

Chapter 7. Conclusions

response to this, we suggest that future research target an alternative time integration technique for the third level, which increases robustness for long time domains.

In Section 6.4, we research the use of $F(CF)^n$ relaxation inside of asymptotic MGRIT. We find that the use of additional CF -sweeps ($n > 0$), which is novel for asymptotic parareal-type algorithms, both increases robustness with respect to problem parameters and decreases runtime in some settings. For instance, we demonstrate that when $\epsilon = .01$, FCF-relaxation ($n = 1$) can halve the number of Braid iterations required to reach convergence. In addition, FCF-relaxation is able to reduce the wall time when compared to F-relaxation, with the runtime roughly proportional to the total iteration count. We demonstrate in Section 6.4.1 that this is the result of the coarse grid operator Φ_Δ dominating the cost of an individual Braid iteration. Thus, FCF-relaxation is able to reduce wall time by reducing the total number of solves on the coarse grid. However, the cost of the coarse grid operator Φ_Δ could be reduced by parallelizing the midpoint quadrature computation over \bar{M} in Algorithm 4, if spare compute nodes were available. In such a setting, the wall time benefits of FCF-relaxation are likely to be less pronounced.

In Section 6.7, further benefits of additional CF -sweeps ($n > 1$) in conjunction with larger time averaging window sizes η are demonstrated by showing greater robustness of the solver when $\epsilon = 1.0$. When η and the number of CF -sweeps are increased, we see that asymptotic MGRIT is able to quickly converge over relatively long time domains, compared to F-relaxation, i.e., asymptotic parareal. The benefit from increasing η for our test cases appears to peak when η is computed using the formulas (6.1) and (6.2) with $\alpha = 12.0, \beta = 1.0$ and $N_\Delta = 50$. The optimal n for a problem depends on the time domain size and the cost of an individual Braid iteration. For example, when $t_f = \{1.0, 4.0\}$ and $N_\Delta = 50$, it is clear that $n = 0$ is the best choice. For larger time domains, however, $n = 2$ can drastically reduce iteration counts over $n = 0$. In these cases $n = 3$ or larger does not offer as dramatic

Chapter 7. Conclusions

a reduction in iteration count.

Another avenue to reduce the cost of the expensive asymptotic coarse grid operator Φ_Δ , is to explore the reuse of stale data when computing Φ_Δ . In Section 6.5.1, we investigated techniques to reuse the slowly-varying coarse grid solution $\bar{\mathbf{u}}$ from previous iterations in an asymptotic MGRIT two-level solve. We observed that on iterations where $\bar{\mathbf{u}}$ was reused from a prior iteration, the convergence factor was near 1.0 in those iterations and ultimately, in order to reach a given residual halting tolerance, the same number of Φ_Δ evaluations was necessary whether or not $\bar{\mathbf{u}}$ was reused.

On the contrary, experiments performed in Section 6.5.2 show that it is possible in the $\epsilon = .01$ case to use a modified coarse-grid operator which reuses stale values of the expensive $\bar{\mathcal{N}}()$ computations from prior iterations without degrading MGRIT convergence. In particular, we saw that when $\bar{\mathcal{N}}()$ is computed on odd-numbered iterations and recycled in the following even-numbered iterations, the residual norm is marginally smaller than the residual norm for the non-reuse reference solution after 10 iterations. Reuse of $\bar{\mathcal{N}}()$ was not as efficient in the $\epsilon = 1$ case, where all considered reuse patterns led to an order of magnitude (or worse) penalty in convergence speed after 10 iterations. However, a speedup is still a possibility in both cases because of the amount of reduced computational work and is the subject of current investigation.

In Section 6.6, we investigated a new problem parameter setting based on Peddle’s work [31] when the Rossby deformation radius F is small. Here, three scales (instead of two) are present in the problem, because the linear operator \mathcal{L} can be split into separate timescales for a gravitational operator \mathcal{G} and rotational operator \mathcal{R} . In particular, when $F = 10^{-4}$, gravity waves are rapidly affected by rotation and the timescale of \mathcal{G} is smaller than the timescale of \mathcal{R} by two orders of magnitude. We show experimentally that when $\epsilon = 1.0$ and $F = 10^{-4}$, there are two optimal choices for η . Peddle conjectured that these two optimal η correspond to one ideal averaging window η for \mathcal{R} and a different one for \mathcal{G} . We consider two-level and three-level SAA-

Chapter 7. Conclusions

MGRIT techniques to handle these multiple averaging windows. Small improvements in MGRIT convergence were found for a two-level technique where the experimentally determined optimal averaging windows $\eta = \{.15, .32\}$ were alternated on even/odd iterations.

Chapter 8

Future Work

We believe that the results in Section 6.3, where three-level asymptotic MGRIT is explored (SAA- and SAS-MGRIT), indicate some promise for three-level (or more) solves with asymptotic MGRIT. In particular for small time-domains, the three-level solves showed an improvement over two-level, converging faster and in less wall clock time. However as discussed in Section in 6.3, more research is required to further develop the coarse level timestepping operator for longer time domains. There is some previous work that has been done with regards to taking large time steps in the shallow water setting which may be appropriate for consideration. In [41], Wingate shows that a semi-implicit method due to Dukowicz and Smith [6] is capable of recovering the Rossby wave amplitudes for large time steps when solving the linearized quasigeostrophic equations. LeVeque describes a generalization of Godunov’s method allowing arbitrarily large time steps in [23]. Further large time step schemes have been studied in a shallow water setting [12, 27, 42].

Another fruitful direction would be to parallelize the computation of $\overline{\mathcal{N}}$ by parallelizing over the quadrature points \overline{M} (see for-loop in Algorithm 4). This would be particularly useful in cases where asymptotic MGRIT benefits from longer

Chapter 8. Future Work

averaging windows η , which require more points \overline{M} to resolve. For instance in [31], it is shown that when $\epsilon = .01$, convergence improves as the size of the averaging window η increases. Additionally in Section 6.7, we demonstrate improved convergence and more efficient coarse grids (i.e., fewer coarse time points N_Δ) when $\epsilon = 1.0$ and larger averaging windows $\eta > \Delta T$ are chosen. Thus if given enough parallel resources, an increase in the averaging window size would not incur a sizeable penalty in compute time, because the for-loop in Algorithm 4 would be parallelized..

Regarding the three-scale problem from Section 6.6, further research is required to develop a wave-averaging technique suited to three-scales. Remember that here the Rossby radius of deformation F is small and the timescales of rotation and gravitation separate. Essentially, the fast linear operator \mathcal{L} can be written as a sum of a gravitational operator \mathcal{G} and a rotational operator \mathcal{R} , yielding two problem scales, whereas previously for $F = 1$, \mathcal{L} represented only one scale. Unfortunately, our attempts to use the optimal averaging window for both \mathcal{R} and \mathcal{G} inside of asymptotic MGRIT resulted only in rather limited improvements in convergence. If it is physically relevant, a reasonable future direction would be to consider the case where F is even smaller and the separate scales of \mathcal{R} and \mathcal{G} would become more pronounced. Here, the use of the optimal averaging window for both \mathcal{R} and \mathcal{G} could yield a larger improvement in convergence.

Lastly, we recommend further research on determining optimal η values and number of CF-sweeps for the case of long time domains. We demonstrated in Section 6.7 how the use of increased CF-relaxation sweeps and an increased time averaging window $\eta > \Delta T$ can be used to allow for small and efficient coarse grid sizes N_Δ with long time domains. Further research should target developing rules or a computational model to help in choosing η and the number of CF-sweeps.

References

- [1] *XBraid: Parallel multigrid in time*. <http://llnl.gov/casc/xbraid>.
- [2] A. BRANDT, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.
- [3] A. CHERTOCK, A. KURGANOV, AND G. PETROVA, *Fast explicit operator splitting method for convection–diffusion equations*, International Journal for Numerical Methods in Fluids, 59 (2009), pp. 309–332.
- [4] H. DE STERCK, S. FRIEDHOFF, A. J. HOWSE, AND S. P. MACLACHLAN, *Convergence analysis for parallel-in-time solution of hyperbolic systems*, arXiv preprint arXiv:1903.08928, (2019).
- [5] V. A. DOBREV, T. KOLEV, N. A. PETERSSON, AND J. B. SCHRODER, *Two-level convergence theory for multigrid reduction in time (mgrid)*, SIAM Journal on Scientific Computing, 39 (2017), pp. S501–S527.
- [6] J. K. DUKOWICZ AND R. D. SMITH, *Implicit free-surface method for the bryan-cox-semtner ocean model*, Journal of Geophysical Research: Oceans, 99 (1994), pp. 7991–8014.
- [7] P. EMBID AND A. MAJDA, *Averaging over fast gravity waves for geophysical flows with arbitrary potential vorticity*, Communications in Partial Differential Equations, 21 (1996), pp. 619–658.
- [8] B. ENGQUIST, H. HOLST, AND O. RUNBORG, *Multiscale methods for the wave equation*, PAMM, 7 (2007), pp. 1140903–1140904.
- [9] B. ENGQUIST AND Y.-H. TSAI, *Heterogeneous multiscale methods for stiff ordinary differential equations*, Mathematics of Computation, 74 (2005), pp. 1707–1743.

References

- [10] R. D. FALGOUT, S. FRIEDHOFF, T. V. KOLEV, S. P. MACLACHLAN, AND J. B. SCHRODER, *Parallel time integration with multigrid*, SIAM J. Sci. Comput., 36 (2014), pp. C635–C661.
- [11] 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. S298–S322.
- [12] L. FRACCAROLLO AND E. F. TORO, *Experimental and numerical assessment of the shallow water model for two-dimensional dam-break type problems*, Journal of Hydraulic Research, 33 (1995), pp. 843–864.
- [13] S. FRIEDHOFF, R. D. FALGOUT, T. KOLEV, S. MACLACHLAN, AND J. B. SCHRODER, *A multigrid-in-time algorithm for solving evolution equations in parallel*, tech. rep., Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2012.
- [14] M. J. GANDER, *Analysis of the parareal algorithm applied to hyperbolic problems using characteristics*, 2008.
- [15] M. J. GANDER, *50 years of time parallel time integration*, in Multiple Shooting and Time Domain Decomposition, T. Carraro, M. Geiger, S. Körkel, and R. Rannacher, eds., Springer, 2015, pp. 69–114.
- [16] M. J. GANDER AND S. VANDEWALLE, *Analysis of the parareal time-parallel time-integration method*, SIAM Journal on Scientific Computing, 29 (2007), pp. 556–578.
- [17] S. GÜNTHER, L. RUTHOTTO, J. B. SCHRODER, E. C. CYR, AND N. R. GAUGER, *Layer-parallel training of deep residual neural networks*, 2018.
- [18] T. HAUT AND B. WINGATE, *An asymptotic parallel-in-time method for highly oscillatory pdes*, 2013.
- [19] A. HESSENTHALER, D. NORDSLETTEN, O. RÖHRLE, J. B. SCHRODER, AND R. D. FALGOUT, *Convergence of the multigrid reduction in time algorithm for the linear elasticity equations*, Numerical Linear Algebra with Applications, 25 (2018), p. e2155.
- [20] A. J. HOWSE, H. D. STERCK, R. D. FALGOUT, S. MACLACHLAN, AND J. SCHRODER, *Parallel-in-time multigrid with adaptive spatial coarsening for the linear advection and inviscid burgers equations*, SIAM Journal on Scientific Computing, 41 (2019), pp. A538–A565.

References

- [21] O. A. KRZYSIK, H. D. STERCK, S. P. MACLACHLAN, AND S. FRIEDHOFF, *On selecting coarse-grid operators for parareal and mgrid applied to linear advection*, 2019.
- [22] S. KUMAR, *Fundamental limits to moore’s law*, 2015.
- [23] R. J. LEVEQUE, *A large time step generalization of godunov’s method for systems of conservation laws*, SIAM Journal on Numerical Analysis, 22 (1985), pp. 1051–1073.
- [24] J. L. LIONS, Y. MADAY, AND G. TURINICI, *Résolution d’EDP par un schéma en temps pararéel*, C.R.Acad Sci. Paris Sér. I Math, 332 (2001), pp. 661–668.
- [25] A. MAJDA, *Introduction to P.D.E.’s and waves for the atmosphere and ocean*, Courant Lecture Notes, New York University, Courant Institute of Mathematical Sciences and American Mathematical Society, 2002.
- [26] A. J. MAJDA AND P. EMBID, *Averaging over fast gravity waves for geophysical flows with unbalanced initial data*, Theoretical and Computational Fluid Dynamics, 11 (1998), pp. 155–169.
- [27] J. MURILLO, P. GARCÍA-NAVARRO, P. BRUFAU, AND J. BURGUETE, *Extension of an explicit finite volume method to large time steps (cfl₂1): application to shallow water flows*, International Journal for Numerical Methods in Fluids, 50 (2006), pp. 63–102.
- [28] A. S. NIELSEN, G. BRUNNER, AND J. S. HESTHAVEN, *Communication-aware adaptive parareal with application to a nonlinear hyperbolic system of partial differential equations*, Journal of Computational Physics, 371 (2018), pp. 483 – 505.
- [29] A. PASZKE, S. GROSS, F. MASSA, A. LERER, J. BRADBURY, G. CHANAN, T. KILLEEN, Z. LIN, N. GIMELSHEIN, L. ANTIGA, A. DESMAISON, A. KOPF, E. YANG, Z. DEVITO, M. RAISON, A. TEJANI, S. CHILAMKURTHY, B. STEINER, L. FANG, J. BAI, AND S. CHINTALA, *Pytorch: An imperative style, high-performance deep learning library*, in Advances in Neural Information Processing Systems 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché Buc, E. Fox, and R. Garnett, eds., Curran Associates, Inc., 2019, pp. 8024–8035.
- [30] A. PEDDLE, *Cyclops-lite*. <https://github.com/agpeddle/cyclops-lite>.
- [31] A. PEDDLE, *Components of nonlinear oscillation and optimal averaging for stiff pdes*, Jan 2018.

References

- [32] A. PEDDLE, T. HAUT, AND B. WINGATE, *Parareal convergence for oscillatory pdes with finite time-scale separation*, 2017.
- [33] M. RIES, U. TROTTEBERG, AND G. WINTER, *A note on mgr methods*, Linear Algebra and its Applications, 49 (1983), pp. 1 – 26.
- [34] D. RUPRECHT, *Wave propagation characteristics of parareal*, 2017.
- [35] M. SCHREIBER, A. PEDDLE, T. HAUT, AND B. WINGATE, *A decentralized parallelization-in-time approach with parareal*, 2015.
- [36] M. SCHREIBER, A. SCHMITT, P. PEIXOTO, AND M. SCHÄFER, *A numerical study of a semi-lagrangian parareal method applied to the viscous burgers equation*, Computing and Visualization in Science, 19 (2017).
- [37] J. B. SCHRODER AND B. W. ONG, *Applications of time parallelization*, Computing and Visualization in Science, Springer (submitted), (2019).
- [38] J. STEINER, D. RUPRECHT, R. SPECK, AND R. KRAUSE, *Convergence of parareal for the navier-stokes equations depending on the reynolds number*, 2013.
- [39] H. D. STERCK, R. D. FALGOUT, S. FRIEDHOFF, O. A. KRZYSIK, AND S. P. MACLACHLAN, *Optimizing mgrit and parareal coarse-grid operators for linear advection*, 2019.
- [40] J. WHITEHEAD, T. HAUT, AND B. WINGATE, *The effect of two distinct fast time scales in the rotating, stratified boussinesq equations: variations from quasi-geostrophy*, Theoretical and Computational Fluid Dynamics, 32 (2018).
- [41] B. A. WINGATE, *The maximum allowable time step for the shallow water α -model and its relation to time-implicit differencing*, Monthly Weather Review, 132 (2004), pp. 2719–2731.
- [42] R. XU, D. ZHONG, B. WU, X. FU, AND R. MIAO, *A large time step godunov scheme for free-surface shallow water equations*, Chinese Science Bulletin, 59 (2014), pp. 2534–2540.