Summer 7-15-2024

Parallel Multigrid in Time for Chaotic Dynamical Systems

David Alan Vargas
University of New Mexico

David Alan Vargas

Candidate

Mathematics and Statistics

Department

This dissertation is approved, and it is acceptable in quality and form for publication.

Approved by the Dissertation Committee:

Prof. Jacob B. Schroder, Chairperson
Prof. Jehanzeb Chaudhary
Prof. Amanda Bienz
Dr. Stefanie Günther

Parallel Multigrid in Time for Chaotic Dynamical Systems

by

David Alan Vargas

B.S., Applied Mathematics and Chemistry,
University of New Mexico, 2020

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Applied Mathematics

The University of New Mexico

Albuquerque, New Mexico

July, 2024

Dedication

For Domonique. Thanks for sticking around through all the chaos.

"Invention, it must be humbly admitted, does not consist in creating out of void, but out of chaos" – Mary Shelley

Acknowledgments

This work was funded by the US Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. I gratefully acknowledge my collaborators and co-authors at Lawrence Livermore National Laboratory, including Dr. Stefanie Günther and my grant sponsor Dr. Rob Falgout, whose support, mentorship, and experience were instrumental to this dissertation.

Above all I must acknowledge my advisor, collaborator, co-author, and friend, Professor Jacob B. Schroder, who got me back into research after a disastrous undergraduate internship caused me to nearly quit my bachelor's degree. Without his guidance I would not have discovered my love of numerical analysis and found a place in the wonderful multigrid community. Thank you, Jacob, for everything.

I am deeply indebted to my parents, for feeding my creativity and instilling in me a lifelong love of learning; surely I could never have accomplished this without their example. I must also thank my sister for being a bright spot in my life and a source of inspiration in the last few challenging years.

Words cannot express my gratitude for my wife, who helped me with the tedious algebra for the higher order θ method conditions found in Chapter 2, and whose unwavering support and friendship not only made this work possible, but made the last four years of graduate school the best of my life so far. I can't wait for our next adventure.

Parallel Multigrid in Time for Chaotic Dynamical Systems

by

David Alan Vargas

B.S., Applied Mathematics and Chemistry,
University of New Mexico, 2020

Ph.D., Mathematics, University of New Mexico, 2024

Abstract

Despite the fact that Parallel-in-Time (PinT) methods are predicted to become necessary to fully utilize next-generation exa- and zettascale machines, there are currently no known practical methods which scale well with the length of the time-domain for chaotic problems, due to exponential dependence of the condition number on the fastest chaotic timescale. I present modifications to the coarse-grid equations along with a novel rediscretization approach which together greatly improve convergence of the multigrid reduction in time (MGRIT) algorithm and allow the first known PinT speedup for a chaotic PDE. The novel Local Shadowing Relaxation (LSR) is presented as an alternative to classical FCF-relaxation for MGRIT and demonstrated to be a convergent, PinT smoother for chaotic PDE systems. Promising preliminary analytical results and numerical experiments with the Lorenz system indicate that LSR may solve the scaling problem for chaotic systems, potentially allowing space-time parallelization of turbulent computational fluid dynamics.

Contents

List of Figures

List of Figures

List of Tables

List of Algorithms

Glossary

👁	Denotes that a figure may be viewed as a parallel-view stereogram.
PinT	Parallel in time.
MGinT	Multigrid in time.
MGRIT	Multigrid Reduction in Time.
n_t	The number of time steps in a time grid having $n_t + 1$ time points.
n_s	The number of spatial degrees of freedom of a problem.
$\boldsymbol{u}$	A space-time block vector $[\boldsymbol{u}_0; \boldsymbol{u}_1; \ldots; \boldsymbol{u}_{n_t}] = [\boldsymbol{u}_0^T, \boldsymbol{u}_1^T, \ldots, \boldsymbol{u}_{n_t}^T]^T$.
A, A_*, A_c	All-at-once space-time system operators for the fine grid, Schur-complement coarse grid, and approximate coarse grid, respectively.
Φ	A time-stepping function propagating the state vector $\boldsymbol{u}_i$ to $\boldsymbol{u}_{i+1}$.
$\boldsymbol{\tau}$	The FAS multigrid τ correction; where for MGRIT, $\boldsymbol{\tau}_i(\boldsymbol{v}_{i-m}) = \Phi^m(\boldsymbol{v}_{i-m}) - \Phi_c(\boldsymbol{v}_{i-m})$.
Δ	The Δ correction, a block-subdiagonal matrix equal to the Jacobian of $\boldsymbol{\tau}$, $\Delta_i(\boldsymbol{v}_{i-m}) = (\partial_{\boldsymbol{u}_{i-m}}\Phi^m - \partial_{\boldsymbol{u}_{i-m}}\Phi_c)(\boldsymbol{v}_{i-m})$.
T_f	The final time of a simulation, equal to t_{n_t}.

Glossary

m	The MGRIT coarsening factor.
Ω^h	The fine time grid $\{t_i\}_{i=0}^{n_t}$.
Ω^{mh}	The coarse time grid $\{t_{im}\}_{i=0}^{n_t/m}$.
λ_j	The jth ordered Lyapunov exponent, where λ_1 is maximal.
$\boldsymbol{\psi}_i^{+,j}$, $\boldsymbol{\psi}_i^{-,j}$	The jth backward or forward Lyapunov vector at time point t_i.
F_i	The linear tangent propagator at time point t_i, equal to $\partial_{\boldsymbol{u}_i}\Phi(\boldsymbol{u}_i)$.
T_λ	The Lyapunov timescale, equal to $\ln(10)/\lambda_1$; an upper bound on the average time it takes for a perturbation in a chaotic system to grow by a factor of 10.
ε	Machine precision, an upper bound on relative floating point rounding error.
KS equation	The Kuramoto-Sivashinsky equation, $\boldsymbol{u}_t = -\boldsymbol{u}_{xx} - \boldsymbol{u}_{xxxx} - \boldsymbol{u}\boldsymbol{u}_x$, posed with periodic boundary conditions.
(E, SD, F)IRKk	(Explicit first stage, Singly diagonal, Fully) implicit Runge-Kutta method of order k.
Re	The Reynolds number from the theory of fluid dynamics.
τ_ν	The Kolmogorov timescale, a function of constants ν, the fluid viscosity, and ε, the energy dissipation rate per unit volume.
LSS	Least Squares Shadowing.
LSS	Local Shadowing Relaxation.
$\sigma_{i,j}$	The jth singular value of the Jacobian $\partial_{\boldsymbol{u}_i}\Phi^m$.

Chapter 1

Introduction

1.1 Motivating parallel in time (PinT)

For the past two decades, computer processor clock speeds have stagnated, while the number of logical cores in a single chip has grown exponentially to continue producing exponential performance gains. This point is demonstrated dramatically by Figure 1.1. Since the limitation on clock frequency is a physical one, it is reasonable to expect this trend to continue, with essentially all future performance improvements relying on increasingly parallel workloads. It is therefore only a matter of time before the time dimension must be parallelized to realize the performance improvements offered by exascale machines and beyond for the numerical simulation of partial differential equations (PDEs).

Parallel-in-space algorithms divide the spatial domain of a PDE problem into smaller parts, assign each part to its own processor, then find the solution in parallel on each of these subdomains. This does not scale indefinitely. Eventually, the spatial

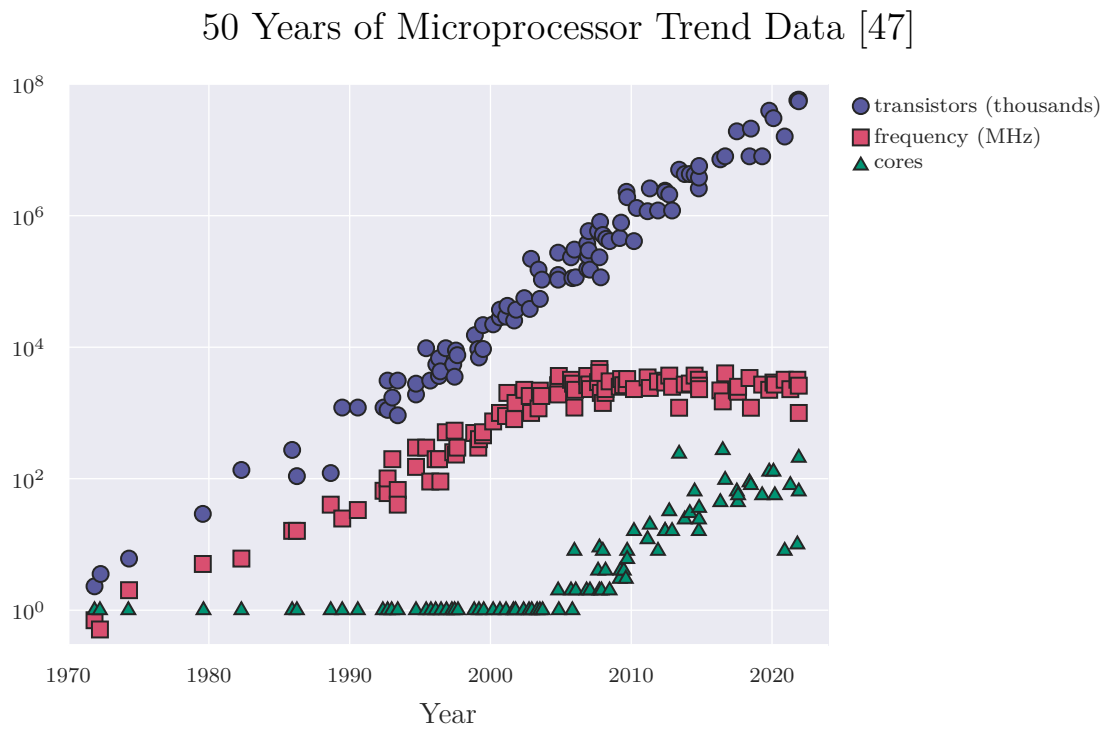


Figure 1.1: Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten. New plot and data collected for 2010-2021 by K. Rupp

domain becomes "saturated" such that the theoretical improvement provided by increased parallelism is completely offset by increased communication costs. This happens when the subdomains become small enough that the amount of neighbor-to-neighbor communication required, which is proportional to their surface area, overtakes the amount of work assigned to each processor, which is proportional to their volume. Of course, the obvious solution is to seek problems with larger spatial domains so that saturation occurs at ever higher concurrent core counts. However, the desire to run these extreme-scale simulations on long time domains and at high resolution results in a bottleneck. High resolution in space typically requires high resolution in time, and therefore increasingly many sequential time steps must be performed. Without parallel-in-time (PinT) methods, it is anticipated that no matter

what improvements come out of the field of time integration, the time it takes to run a simulation will remain proportional to the number of time steps required. Even if this isn't a serious concern now, it soon will be. Improvements to time integration or single chip performance will only delay the need for PinT, not prevent it [16].

There are many PinT algorithms, going as far back as 1964 [37]. Small-scale PinT is possible with direct methods like stage-parallel Runge-Kutta methods [38] and Revisionist Integral Deferred Correction (RIDC) [5]. Large-scale PinT, on which this work is focused, is most often achieved using iterative multigrid-in-time (MGinT) methods, with the most prominent examples being Parallel Full Approximation Scheme in Space and Time (PFASST) [14], Multigrid Reduction in Time (MGRIT) [15], and the most studied PinT method, Parareal [20]. A notable non-multigrid algorithm is the ParaDiag family of direct and iterative methods based on diagonalization of the all-at-once space-time system [23].

MGinT methods are especially promising for large scale PinT because they have been demonstrated to scale optimally for parabolic systems, such as the heat equation. That is to say that a well-behaved system discretized over n_t points in time and n_s points in space can be solved in $\mathcal{O}(\log(n_t))$ wall-clock time as long as the number of processors is scaled with the number of time points. Using space-time multigrid (STMG), the same simulation can be run in $\mathcal{O}(\log(n))$ time, where $n = n_s n_t$, given sufficient resources. At high enough n, this is effectively constant time complexity when compared to traditional time stepping, which is at best $\mathcal{O}(n_t \log(n_s))$ when using an optimal spatial solver such as Algebraic Multigrid (AMG) [2]. This remarkable result is due to the highly efficient structure of multigrid, which classically solves a problem in $\mathcal{O}(1)$ iterations, each doing $\mathcal{O}(n_t)$ work on $\mathcal{O}(\log(n_t))$ hierarchical grids, resulting in an overall complexity of $\mathcal{O}(n_t \log(n_t))$. Since the basic building block of multigrid is relaxation, which can be chosen to be embarrassingly parallel, given $\mathcal{O}(n_t)$ processors, the problem can be solved with overall logarithmic

complexity. Note that the time stepping problem, $\boldsymbol{u}_i = \Phi(\boldsymbol{u}_{i-1}) + \boldsymbol{g}_i$, is equivalent to the well-known *prefix sum* problem in the case that the operation $\Phi(\boldsymbol{u}) + \boldsymbol{g}$ is associative (e.g. if $\Phi(\boldsymbol{u}) = \boldsymbol{u}$, this is a cumulative sum). The prefix sum problem has a minimum circuit depth of $\log_2(n)$, and thus the minimum possible parallel time complexity is $\mathcal{O}(\log(n))$ [33]. This gives a strong indication that MGinT has optimal time complexity. In fact, when applied to the prefix sum problem, the MGRIT algorithm is equivalent to a binary tree algorithm which is known to have optimal work efficiency.

Despite these promises, and increasing interest over the past couple of decades, many are still skeptical that PinT will ever be practical. It is true that PinT faces an uphill battle. Whereas there is no single optimal choice of solver for the systems arising from elliptic problems in space, a parabolic initial-value problem is block lower-triangular in time, and thus forward substitution, also known as time stepping, is the most obvious and easily implemented solution method. Therefore, while multigrid is very attractive for spatial problems where other algorithms are extremely inefficient in comparison, PinT methods have to compete with time stepping, which is an optimal $\mathcal{O}(n_t)$ direct solve that is however sequential in time. Therefore, iterative PinT methods are at a disadvantage with respect to time stepping in efficiency, since even measuring the space-time residual often requires as much work as solving the entire system with time stepping. However, given an optimally scaling PinT algorithm, there will always be a crossover point after which PinT is faster than time stepping, because time stepping is $\mathcal{O}(n_t)$ and PinT is ideally $\mathcal{O}(\log(n_t))$. Because of this, and the increasing importance of parallelism, [16] argues that it is not a question of *if* PinT methods will be adopted, but *when*.

A more serious issue with PinT is the failure of many methods when applied naively to more physically complex problems, such as hyperbolic and chaotic systems, which has been documented in many published works [21, 39]. Multigrid only scales

optimally when the number of iterations required is independent of the size of the problem, which is often not the case for hyperbolic and chaotic systems when applied to the time dimension. For hyperbolic systems, information needs to be transmitted accurately (and in many cases exactly) along characteristics from the initial condition to the final time, which is a highly sequential process by nature. Early work failed to produce meaningful speedups for even linear advection [49, 28]. For chaotic systems, with which this work is primarily concerned, the condition number of the space-time system grows exponentially with the total length of the time domain, at a rate determined by the fastest chaotic timescale, such that naive PinT methods do not scale well for chaotic problems [53]. This has lead many to question whether PinT is possible for chaotic systems, a question which this work seeks to address.

1.2 Multigrid Reduction in Time (MGRIT)

MGRIT is an iterative multigrid method for solving discrete initial-value problems of the form

$$\begin{cases} \boldsymbol{u}_0 = \boldsymbol{g}_0, \\ \boldsymbol{u}_{i+1} = \Phi(\boldsymbol{u}_i) + \boldsymbol{g}_{i+1} \quad i = 0, 1, 2, \ldots, n-1, \end{cases} \tag{1.1}$$

where Φ is a nonlinear time-stepping operator [15]. Systems of this form typically arise from a time discretization of an ordinary differential equation (ODE) of the form $\boldsymbol{u}'(t) = \boldsymbol{f}(\boldsymbol{u}(t))$, in which case Φ is some time-stepping scheme such as Euler's method and $\boldsymbol{g}_i$ corresponds to constant forcing terms. The system is defined over a discrete time grid with $n_t + 1$ points, $\Omega^h = \{t_i\}_{i=0}^{n_t}$, and time-step size $h = t_{i+1} - t_i$. We will assume, without loss of generality, that h is constant. Let $\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_0; & \boldsymbol{u}_1; & \ldots; & \boldsymbol{u}_{n_t} \end{bmatrix}$ denote the state vector and let $\boldsymbol{g} = \begin{bmatrix} \boldsymbol{g}_0; & \boldsymbol{g}_1; & \ldots; & \boldsymbol{g}_{n_t} \end{bmatrix}$ be a constant forcing term which also encodes the initial condition $\boldsymbol{g}_0$. Then equation (1.1) may be written as

a block nonlinear operator equation,

$$A(\boldsymbol{u}) = \boldsymbol{g}, \text{ where } A(\boldsymbol{u}) = \begin{bmatrix} I & & & & \\ -\Phi & I & & & \\ & -\Phi & I & & \\ & & \ddots & \ddots & \\ & & & -\Phi & I \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_{n_t} \end{bmatrix}. \tag{1.2}$$

Typically, this system would be solved using forward substitution, which corresponds with traditional sequential time stepping. MGRIT instead applies Full Approximation Scheme (FAS) Multigrid Reduction (MGR) [1, 45] to the system (1.2), allowing it to be solved in parallel. To this end, (1.2) is approximated on a hierarchy of coarser time grids, e.g. $\Omega^{2h}, \Omega^{4h}, \Omega^{8h}, \ldots$, which provide error corrections to the finer grids, while the finer grids provide further corrections via local block Jacobi relaxation.

The multigrid method requires a coarsening scheme in time, inter-grid transfer operators, and a coarse-grid equation. Here, we define those for a two-level MGRIT method with fine grid, Ω^h, and coarse grid, Ω^{mh}, for coarsening factor m. To coarsen in time, label every mth time point in Ω^h a C-point and all other points an F-point, then Ω^{mh} is the set of size n_T containing the C-points in Ω^h (see Figure 1.2). A C-point, along with the following $m-1$ F-points to the right, is called a coarse interval. For grid transfer operations, MGRIT uses *injection*. For restriction, injection maps the values of $\boldsymbol{u}$ at the C-points in Ω^h to the corresponding points in Ω^{mh}. For interpolation, injection maps the points in Ω^{mh} to the corresponding C-points in Ω^h. The action of restriction by injection is given by the block matrix

$$R = \begin{bmatrix} I & & & & & & & \\ & 0 & \cdots & 0 & I & & & \\ & & & & & \ddots & & \\ & & & & & & 0 \cdots 0 & I \end{bmatrix}, \tag{1.3}$$

yielding a coarse-grid vector $\boldsymbol{u}_c = R\boldsymbol{u}_f$ on Ω^{mh}. Interpolation by injection is then

given by the action of R^T. Following interpolation from Ω^{mh} to Ω^h, i.e. $\boldsymbol{u}_f \leftarrow R^T \boldsymbol{u}_c$, MGRIT always relaxes the solution on Ω^h using F-relaxation (1.4), which evolves the state at each C-point to the following F-points in each coarse interval using Φ. F-relaxation is an important part of multigrid reduction since it sets the residual to zero at each F-point, meaning that the residual is exactly representable on the coarse grid. F-relaxation may also be viewed as part of the interpolation process, in that it updates each F-point based on the new C-point information injected from the coarse grid.

The two-level MGRIT method described thus far is equivalent to the popular Parareal algorithm [20]. However, apart from being a multi-level method, one of the key differences between MGRIT and Parareal is support for FCF-relaxation, where F-relaxation is followed by C-relaxation (1.5), propagating the solution from the right-most F-point in each interval to the following C-point, which is then followed by another F-relaxation [15]. FCF-relaxation has been shown in many cases to greatly improve convergence, especially in the multilevel setting [15, 52]. Importantly, since the intervals are disjoint, F-and FCF-relaxation can be done in parallel, with minimal communication between processors.

$$\boldsymbol{u}_{im+k} = \Phi(\boldsymbol{u}_{im+k-1}) + \boldsymbol{g}_{im+k} \quad \text{for } k = 1, \ldots, m-1 \text{ and each coarse-interval } i \tag{1.4}$$

$$\boldsymbol{u}_{im} = \Phi(\boldsymbol{u}_{im-1}) + \boldsymbol{g}_{im} \quad \text{for each coarse-interval } i \tag{1.5}$$

Coarsening in time induces a new system of equations posed on Ω^{mh}, where the

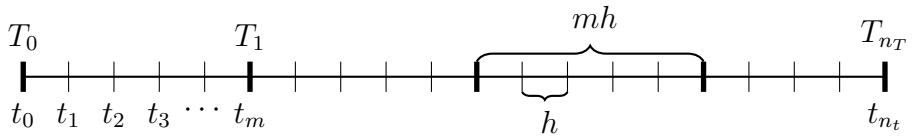


Figure 1.2: Coarsening of a uniformly spaced fine time grid with MGRIT C-point and F-point partitioning with coarsening factor m. The T_i are the C-points and form the coarse grid, while the small hash marks t_i are F-points. Together, the F- and C-points form the fine grid.

Schur-complement system is given by the space-time operator

$$A_* = \begin{bmatrix} I & & & & \\ -\Phi^m & I & & & \\ & -\Phi^m & I & & \\ & & \ddots & \ddots & \\ & & & -\Phi^m & I \end{bmatrix}. \tag{1.6}$$

Here $\Phi^m(\boldsymbol{u}_i)$ is understood to denote a fine-grid propagation of the solution across one coarse interval, from one C-point to the next, including the forcing term $\boldsymbol{g}$, e.g. $\Phi^2(\boldsymbol{v}_{i-2}) = \Phi(\Phi(\boldsymbol{v}_{i-2}) + \boldsymbol{g}_{i-1})$, $\Phi^3(\boldsymbol{v}_{i-3}) = \Phi(\Phi(\Phi(\boldsymbol{v}_{i-3}) + \boldsymbol{g}_{i-2}) + \boldsymbol{g}_{i-1})$, etc. Solving this ideal coarse-grid equation yields the exact solution for each C-point which would then yield the exact solution at every time point following F-relaxation, hence A_* is called the Schur-complement operator. However, solving this system requires just as much work as solving the original fine-grid problem, since the number of time points is reduced by a factor of m, but the cost of each time step is increased by the same factor. Instead, in practice one introduces the coarse-grid time-stepping operator Φ_c to cheaply approximate the action of Φ^m, and then A_* is replaced by

the approximate coarse-grid operator

$$A_c = \begin{bmatrix} I & & & & \\ -\Phi_c & I & & & \\ & -\Phi_c & I & & \\ & & \ddots & \ddots & \\ & & & -\Phi_c & I \end{bmatrix} \approx A_*. \tag{1.7}$$

In the case that Φ is derived from a continuous-time problem, then Φ_c is usually derived from some (possibly nonstandard) rediscretization of the continuous problem over the new coarse time-grid Ω^{mh}. Deriving coarse operators for general Φ is an open problem [39, 12] and motivates Chapter 2.

Because the Schur-complement coarse-grid equation gives the exact solution as the fine grid equation at the C-points, we may analyze MGRIT by ignoring the F-points and focusing entirely on A_*. To this end, we now drop the subscript $\boldsymbol{u}_c$ and use $\boldsymbol{u}_f$ explicitly to refer to fine-grid state vectors. From this perspective, two-level MGRIT using the approximation of A_* by A_c may be interpreted as a nonlinear splitting method. Let $\boldsymbol{\tau}(\boldsymbol{u}) = A_c(\boldsymbol{u}) - A_*(\boldsymbol{u})$, then $A_*(\boldsymbol{u}) = \boldsymbol{g}_c = A_c(\boldsymbol{u}) - \boldsymbol{\tau}(\boldsymbol{u})$, and one immediately gets the well-known τ-correction form of FAS multigrid [1, 2]:

$$A_c(\boldsymbol{v}^{k+1}) = \boldsymbol{g}_c + \boldsymbol{\tau}(\boldsymbol{v}^k), \tag{1.8}$$

where $\boldsymbol{v}^k$ denotes an approximate coarse solution on Ω^{mh} after k multigrid iterations, and for MGRIT, $\boldsymbol{\tau}_i = \Phi^m(\boldsymbol{v}^k_{i-1}) - \Phi_c(\boldsymbol{v}^k_{i-1})$. This splits the operator A_* into a part that is cheap to invert, A_c, and a part which can be computed efficiently in parallel, $\boldsymbol{\tau}$. One iteration of the two-level MGRIT scheme involves computing $\boldsymbol{\tau}(\boldsymbol{v}^k)$ at the C-points on Ω^h, injecting $\boldsymbol{g}$ and $\boldsymbol{\tau}$ to Ω^{mh}, solving (1.8) sequentially for $\boldsymbol{v}^{k+1}$, and then interpolating to Ω^h and applying F-relaxation. The vector $\boldsymbol{\tau}$ takes the form of a forcing term on the coarse-grid, being added to $\boldsymbol{g}_c$ on the right-hand side, and it steers the coarse grid equation toward the fine grid equation, as well as ensuring

that the exact fine-grid solution is a fixed point of the iteration, a property which is necessary for any iterative scheme. To see this last point, we plug the exact solution $\boldsymbol{u}_f$ satisfying $A_*(R\boldsymbol{u}_f) = \boldsymbol{g}_c$ into (1.8):

$$\begin{aligned} A_c(\boldsymbol{v}^{k+1}) &= \boldsymbol{g}_c + \boldsymbol{\tau}(R\boldsymbol{u}_f) \\ &= \boldsymbol{g}_c + A_c(R\boldsymbol{u}_f) - A_*(R\boldsymbol{u}_f) \\ &= A_c(R\boldsymbol{u}_f) \\ \implies \boldsymbol{v}^{k+1} &= R\boldsymbol{u}_f. \end{aligned}$$

Thus, any $\boldsymbol{u}$ satisfying the ideal coarse-grid equation is a fixed point of the MGRIT iteration, which is not true in general without $\boldsymbol{\tau}$ correction on the coarse-grid. The two-level MGRIT algorithm is detailed in Algorithm 1.1. A multi-level MGRIT algorithm then results from recursive application of the two-level scheme to solve the coarse-grid equation (1.8) with another MGRIT cycle. This recursion gives the V-cycle MGRIT$_{m_\ell}$ algorithm, with m_ℓ time-grid levels.

Algorithm 1.1 MGRIT$_2(\boldsymbol{v}_f, \boldsymbol{g}, m)$; MGRIT two-grid cycle

$\boldsymbol{v} \leftarrow R\boldsymbol{v}_f$, $\boldsymbol{g}_c \leftarrow R\boldsymbol{g}$

for each C-point, $i = 1, 2, 3, \ldots, n_T$ **do**

  $\boldsymbol{\tau}_i \leftarrow \Phi^m(\boldsymbol{v}_{i-1}) - \Phi_c(\boldsymbol{v}_{i-1})$

end for

for $i = 1, 2, 3, \ldots, n_T$ **do**

  $\boldsymbol{v}_i \leftarrow \Phi_c(\boldsymbol{v}_{i-1}) + \boldsymbol{\tau}_i + \boldsymbol{g}_{c,i}$

end for

$\boldsymbol{v}_f \leftarrow R^T\boldsymbol{v}$, followed by F(CF)-relaxation

1.3 MGRIT for chaotic problems

1.3.1 Chaos

Chaotic systems are globally stable, deterministic systems which demonstrate sensitive dependence on initial conditions and system parameters, and which have trajectories that never settle down to a steady state solution or a periodic orbit for almost every initial condition. Trajectories of chaotic systems are able to orbit forever within a finite region of space without ever visiting the same point because they approach a fractal limit set called a strange attractor. Trajectories beginning away from the attractor approach its surface exponentially fast, while nearby trajectories lying on the attractor's surface diverge away from each other exponentially fast.

The work [22] presents, to my knowledge, the first experiment involving PinT for a chaotic system, by solving the Lorenz system with Parareal over a moderately long time-scale, demonstrating the delay or stalling of convergence seen when naively applying MGRIT to chaotic problems. The work [14] introduces the PFASST algorithm, a PinT method with robust convergence for advection-diffusion problems, however, convergence degrades when applied to the chaotic Kuramoto-Sivashinsky (KS) equation. Parareal is applied to fully-developed plasma turbulence in [48, 44], and to the decay of a 3D Taylor-Green vortex configuration in [35], both reporting over 10× speedups. These works rely on spatial coarsening, where the coarse grid is solved with lower spatial accuracy, such that the coarse-grid dynamics are essentially non-chaotic due to the lack of small spatial scales. While promising, these works use an alternative convergence criterion based on the difference in the total energy of the system between iterations, and not the actual residual norm $\|\boldsymbol{g} - A(\boldsymbol{u})\|$. For this reason, the method more resembles a turbulence closure model, since the result is accurate with respect to some statistical measures, but fine space-time scales are not resolved. Thus, reporting speedup using these methods is problematic, since speedup

is measured relative to a DNS simulation which exactly resolves all scales. Recently, [25] takes this perspective and presents a PinT algorithm for turbulence based on an MGRIT framework, improving upon the Parareal algorithm by using spectral filtering in order to isolate space-time scales to their respective grids. Convergence in this work is based on a coarse-scale residual which ignores fine space-time scales that are still not resolved. In this way, fine-grid relaxation iterates the fine-scale dynamics to statistical equilibrium, while the FAS τ correction on the coarse grid acts like a correction from a turbulence closure. No parallel speedups are reported. All of these works rely on alternative convergence criteria and also rely on an energy cascade from large to small space-time scales which is present in their chosen model problems. In the presence of an inverse or bidirectional cascade, these approaches may break down.

To study MGRIT for chaotic systems, we will use the Lorenz system as a model problem. The Lorenz system is a three-dimensional system of ODEs which is widely studied as an archetypal example of a chaotic system, and is given by the system of ODEs

$$\begin{cases} x' &= \sigma(y - x) \\ y' &= x(\rho - z) - y \cdot \\ z' &= xy - \beta z \end{cases} \tag{1.9}$$

For the classical values of parameters $\sigma = 28$, $\rho = 10$, and $\beta = 8/3$, the Lorenz system is chaotic, with greatest *Lyapunov exponent* of $\lambda_1 \approx 0.9$ [51]. This can be understood to mean that two trajectories differing only infinitesimally in initial conditions will, almost surely, diverge exponentially from each other in time with average rate λ_1. Although trajectories diverge from each other locally, the Lorenz system is globally *Lyapunov stable*, meaning that trajectories are ultimately confined to a bounded trapping region in space, which contains a *strange attractor*, a fractal manifold which is the limit set of the Lorenz system. Generally, a system with n_s spatial

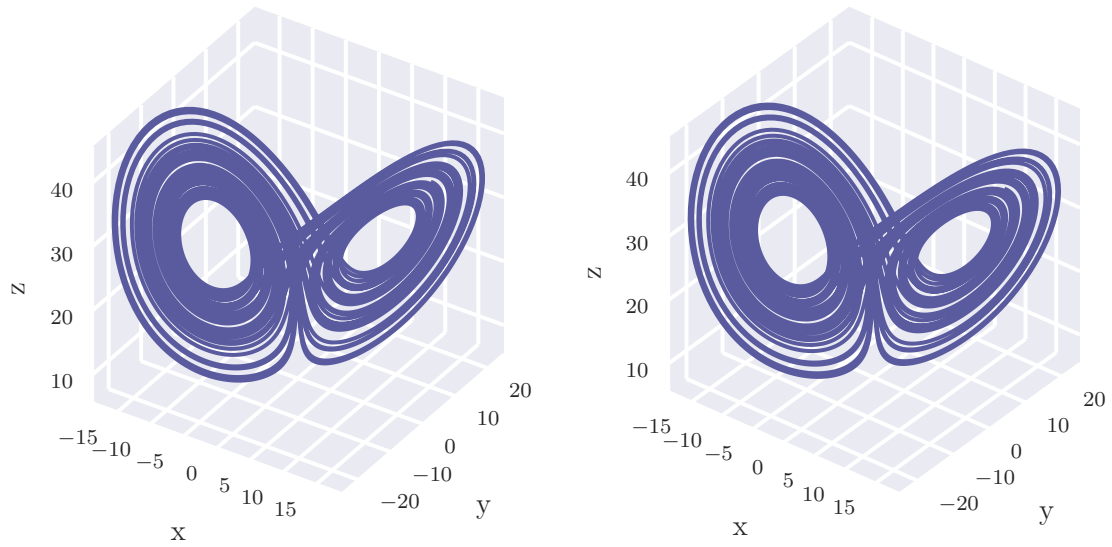


Figure 1.3: A trajectory of the Lorenz system, tracing out its famous butterfly-shaped strange attractor, rendered as a stereogram (In the remainder of this work, 👁▷ will demarcate a figure which can be viewed as a parallel-view stereogram).

dimensions has n_s Lyapunov exponents, which are characteristic of the qualitative behavior of the system, and every chaotic system has a greatest Lyapunov exponent which is greater than zero. The corresponding *Lyapunov vectors* are characteristic directions, $\boldsymbol{\psi}^k(t)$, $k = 1, 2, \ldots, n_s$, along which infinitesimal perturbations will grow exponentially with average rate λ_k, as illustrated in Figure 1.4.

As an example, the Lorenz system is three-dimensional and has three Lyapunov exponents: $\lambda_1 \approx 0.9$, which corresponds with perturbations lying tangent to the surface of the strange attractor, $\lambda_2 = 0$, which corresponds with perturbations tangent to the flow (resulting in a difference only in phase), and $\lambda_3 \approx -14$, which corresponds with perturbations away from the strange attractor. The Lyapunov vectors having negative Lyapunov exponent are tangent to the *stable manifold*, those having vanishing Lyapunov exponents are tangent to the *neutral manifold*, and those having positive Lyapunov exponents are tangent to the *unstable manifold*. The Lyapunov

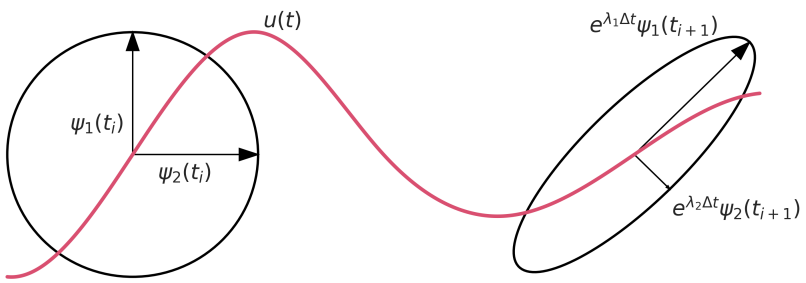


Figure 1.4: An illustration of the finite-time evolution of Lyapunov vectors.

vectors for a trajectory of the Lorenz system are shown in Figure 1.5, where we see that, roughly, the unstable Lyapunov vectors are tangent to the surface of the strange attractor, the neutral Lyapunov vectors are tangent to the flow, and the stable Lyapunov vectors are normal to the surface of the attractor, matching our physical intuition for these subspaces.

1.3.2 The Lyapunov spectrum

It is helpful to establish a distinction between *forward*, *backward*, and *covariant* Lyapunov vectors, $\boldsymbol{\psi}^{+,k}(t)$, $\boldsymbol{\psi}^{-,k}(t)$, and $\boldsymbol{\gamma}^{k}(t)$ respectively. What I have so far called "the Lyapunov vectors" are actually the covariant Lyapunov vectors, and the backward Lyapunov vectors are sometimes known as the Gram-Schmidt vectors, since, as we will see, they may be computed using a special QR iteration [31].

Let $\boldsymbol{u}(t_i)$ be a discrete trajectory of a nonlinear dynamical system defined on a fixed infinite time grid $\{t_i\}_{i=-\infty}^{\infty}$, having time propagator Φ, and which passes through some point $\boldsymbol{u}(t_0) = \boldsymbol{u}_0$. The Jacobian of the propagator Φ at each point $\boldsymbol{u}_i$

is the matrix $F_i = \partial_{\boldsymbol{u}_i}\Phi(\boldsymbol{u}_i)$, called the linear tangent propagator [31], as it describes the propagation of infinitesimal perturbations to $\boldsymbol{u}$ from time point i to time point $i+1$, i.e. $\Phi(\boldsymbol{u}_i + \boldsymbol{e}) - \boldsymbol{u}_{i+1} \approx F_i\boldsymbol{e}$. Thus, the propagation of a perturbation from time point t_i to point t_j is given by $W(t_i, t_j) = F_{j-1}\dots F_{i+1}F_i$. In the limit as $j \to \infty$, the time-average of the singular values of $W(t_i, t_j)$ are equal to $\exp(\lambda_i)$, where the λ_i are the Lyapunov exponents of the system. The Lyapunov exponents are independent of the times t_i, t_j, and are also the same for almost all $\boldsymbol{u}_0 \in \mathbb{R}^{n_s}$, and are thus considered constants of the system. The forward Lyapunov vectors at time t_i, $\psi^{+,k}(t_i)$, are given by the eigenvectors of $\left[W(t_i,t_j)^T W(t_i,t_j)\right]^{1/(2(t_j-t_i))}$ in the limit as $j \to \infty$, while the backward Lyapunov vectors at time t_j, $\psi^{-,k}(t_j)$, are given by the eigenvectors of $\left[W(t_i,t_j)W(t_i,t_j)^T\right]^{1/(2(t_j-t_i))}$ in the limit as $i \to -\infty$. Note that while the Lyapunov exponents are not time-dependent, the backward and forward Lyapunov vectors are. Because the forward and backward Lyapunov vectors are defined as eigenvectors of symmetric positive semi-definite matrices, they form an orthogonal basis as long as $W(t_i, t_j)$ remains invertible in the limits as $t_i \to -\infty$ and $t_j \to \infty$, which turns out to be equivalent to the requirement that the limit set of Φ is uniformly hyperbolic [42]. The Lyapunov exponents may not be distinct, in which case the forward and backward Lyapunov vectors are not unique, but any orthogonal set of vectors spanning the subspaces corresponding to the Lyapunov exponents with multiplicity greater than one will suffice, so this is not an issue in practice [31].

While the forward and backward Lyapunov vectors are orthogonal, they are not *covariant* with the dynamics. Consider the finite-time propagation of a perturbation $\boldsymbol{p}(t_i)$ from the point t_0 to the point t_1, where initially $\boldsymbol{p}(t_0)$ points entirely in the direction of one of the forward Lyapunov vectors, $\psi^{+,j}(t_0)$. Since it is known that the perturbation at time t_j will grow with asymptotic growth rate λ_j, it is therefore impossible for $\boldsymbol{p}(t_1)$ to have any component pointing in the direction $\boldsymbol{\psi}^{+,k}(t_1)$ for a larger Lyapunov exponent ($k < j$), since that would indicate that $\boldsymbol{p}(t_i)$ would then grow with asymptotically larger rate λ_k. However, if $\boldsymbol{p}(t_1)$ has a nonzero component

in the direction $\boldsymbol{\psi}^{+,k}(t_1)$ for a smaller Lyapunov exponent ($k \geq j$), it is clear that this will not affect the asymptotic growth of $\boldsymbol{p}$, since $\lambda_k \leq \lambda_j$. In other words, forward in time, the forward Lyapunov vectors are mapped to arbitrary vectors which are orthogonal to the set of Lyapunov vectors with larger Lyapunov exponent. Similarly, backward in time, the backward Lyapunov vectors are mapped to vectors which are orthogonal to the set of Lyapunov vectors with smaller Lyapunov exponent. Let $\Psi^+(t_i)$ be the matrix with columns equal to the forward Lyapunov vectors, and $\Psi^-(t_i)$ be the same for the backward Lyapunov vectors. Given the argument above, we derive the following relationships for the finite-time propagation of the Lyapunov vectors:

$$F_i \Psi^+(t_i) = \Psi^+(t_{i+1}) L_{i+1}, \tag{1.10}$$

$$F_i^{-1} \Psi^-(t_{i+1}) = \Psi^-(t_i) R_{i+1}^{-1}, \tag{1.11}$$

for some lower triangular matrix L_{i+1} and upper triangular matrix R_{i+1}, whose inverses are guaranteed to exist since the diagonals of each matrix will be non-zero by construction. Further, recall that the forward and backward Lyapunov vectors are orthogonal, so if we also require that the Lyapunov vectors are normalized such that $\|\boldsymbol{\psi}^k\| = 1$, the matrices $\Psi^+(t_i)$ and $\Psi^-(t_i)$ will be orthonormal and uniquely determined by the recurrence relationships (1.10) and (1.11), and by the QR and QL factorizations, respectively [31].

The covariant Lyapunov vectors $\boldsymbol{\gamma}^k(t_i)$ are defined such that

$$F_i \Gamma(t_i) = \Gamma(t_{i+1}) C_{i+1},$$

where C_{i+1} is a diagonal matrix. We see that in contrast to the forward and backward Lyapunov vectors, covariant Lyapunov vectors are mapped to covariant Lyapunov vectors in finite time. However, $\Gamma(t_i)$ is not in general orthogonal. The $\boldsymbol{\gamma}^k$ defined this way exist whenever the forward and backward Lyapunov vectors exist.

Rewriting Equation (1.11) as

$$F_i \Psi^-(t_i) = \Psi^-(t_{i+1}) R_{i+1} \tag{1.12}$$

gives a convenient way to computationally estimate the first k backward Lyapunov vectors of a system. Given a fixed trajectory, $\boldsymbol{u}$, we may initialize $\Psi^-(t_0) \in \mathbb{R}^{n_s \times k}$ with some initial guess, e.g. the first k columns of the identity matrix, then Equation (1.11) gives a time stepping relation that gives $\Psi^-(t_i)$ in terms of $\Psi^-(t_{i-1})$ as the forward propagation via F_{i-1} followed by orthonormalization via the QR factorization of the result. Indeed, starting from a non-singular initial guess to $\Psi^-(t_0)$, the columns of $\Psi^-(t_i)$ do converge to the first k backward Lyapunov vectors as $i \to \infty$ in a way analogous to the convergence of the power iteration for computation of eigenvectors [31]. Further, the long-time average of the diagonals of the upper triangular matrix R_i will converge to $e^{h\lambda_j}$, where λ_j are the Lyapunov exponents. Similarly, the first k forward Lyapunov vectors may be approximated via back-propagation with the adjoint linear tangent propagator F_i^T followed by QR decomposition of the result.

Given estimates of the first k backward and forward Lyapunov vectors, the covariant Lyapunov vectors may be estimated by forming $P(t) = \Psi^+(t)^T \Psi^-(t)$, computing the LU decomposition (without pivoting) $A^+(t)A^-(t) = P(t)$, and finally finding

$$\Gamma(t) = \Psi^+(t)A^+(t) = \Psi^-(t)(A^-(t))^{-1}, \tag{1.13}$$

assuming $A^-(t)$ is invertible [31].

1.3.3 Why even bother simulating chaotic systems?

Traditionally, the halting criterion for MGRIT is based on the space-time norm of the residual. This ignores the forward error, which measures the difference between the MGRIT and sequential time-stepping solutions. For a chaotic system, the forward

error grows exponentially fast along the unstable manifold, and will quickly saturate so that the error is of the same order of magnitude as the solution itself. This begs the question, "why bother simulating chaotic systems at all, if there can be no confidence in the given solution after a short time?" While this is a rather philosophical question which is fairly beyond the scope of this work, there is a resolution to this quandary, which is to consider a *backward* error analysis appropriate for an ill-conditioned initial-value problem. The shadowing lemma, Lemma 2, provides such a backward error analysis for numerical trajectories of systems with hyperbolic attracting sets, which chaotic systems such as the Lorenz system are widely thought to behave like [6, 42, 26, 41]. In such cases, even when the forward error is large, as long as an approximate trajectory has a *residual*, here given by $\boldsymbol{r}_i = \boldsymbol{g}_i + \Phi(\boldsymbol{u}_{i-1}) - \boldsymbol{u}_i$, which is small and uniformly bounded, there exists an exact solution which is uniformly close to the approximate trajectory but has a slightly perturbed initial condition and time grid– that is, the backward error is small.

The shadowing lemma will be discussed in more detail in Chapter 4, but it essentially provides evidence that in systems with chaotic attractors, these structures are stable, in the sense that small perturbations to the system result in small perturbations to the attractor itself, even if the same small perturbation results in large differences between individual trajectories having the same initial condition. The residual convergence criterion depends on this backward error argument, where, as long as a PinT method has reduced the global residual below a small uniform tolerance, the shadowing lemma holds and the obtained solution has backward error which is small and uniformly bounded. The shadowing lemma can be used to make a similar backward error argument when comparing the continuous, exact system and the exact solution of the discrete, numerical simulation.

Anecdotally, the shadowing lemma also says something quite philosophical about Human existence. Assuming our lives are chaotic (which I doubt many would argue

with), that you are at all times attempting to live according to some moral standard, and measuring the magnitude of your mistakes not by their consequences but by their instantaneous deviation from this standard, you can make constant small mistakes and still be assured that, in an alternate universe, there is some shadow of yourself who never made a mistake, yet has lived a life very similar to your own. Perhaps then the cure for perfectionism and personal shame is the acceptance of chaos. In the same way, numerical analysts are comforted by the fact that despite only being able to approximate the solutions to chaotic systems, making constant rounding and other numerical errors, their simulations do indeed represent reasonable approximations to the systems they care about.

1.3.4 MGRIT and chaotic systems: The stalling problem

PinT simulations of chaotic systems such as Lorenz are difficult because of two main problems. The first is that the sensitivity of chaotic systems makes them ill-conditioned. The other is that coarsening in time can cause serious global qualitative changes between fine and coarse grids for the discretized system because of changes in the Lyapunov spectrum. While chaotic systems are best known for their sensitivity to perturbations in the initial conditions, they are equally sensitive to changes to the system parameters. Coarsening in time can be considered a parametric perturbation, and thus the solution on the coarse-grid will diverge exponentially from that of the fine-grid. As long as there is *any* error on the fine grid, the τ correction will not be accurate enough to fix this problem. In fact, once the solution on the coarse grid has diverged sufficiently far from the solution on the fine-grid, the τ correction will become useless, since the τ correction is only valid in a small neighborhood around the point in space at which it was computed. Because of this, after a certain point, the τ correction becomes *harmful* to the quality of the coarse grid correction. Therefore, the challenge is to form a better coarse-grid equation that is both locally

precise and that also captures the global qualitative behavior of the system, even for very coarse time-grids.

Due to the existence of the unstable manifold, any numerical error committed in solving a chaotic initial-value problem with any method will grow exponentially in time and eventually result in error which is of the same order of magnitude as the solution itself. We can characterize this with the condition number of the initial-value problem (1.1). First, we define *Lyapunov time*, $T_\lambda = \frac{\ln(10)}{\lambda_1}$, which acts as an upper bound on the expected time for a perturbation to a trajectory to grow by a factor of 10 [51], where λ_1 is the greatest LE of the system. Hence, T_λ is roughly the time it takes for our numerical simulation of the system to lose one digit of accuracy on average. The Lyapunov time can be used to compute an estimate for the condition number, $\kappa = \mathcal{O}(10^{T_f/T_\lambda})$, where $T_f = n_t h$ is the length of the time-domain. Lyapunov time serves as a normalized timescale over which all chaotic systems have greatest Lyapunov exponent equal to unity. For example, one Lyapunov time for a weather simulation might be a few days, while one Lyapunov time for a simulation of planetary motion might be millions of years.

Note that other sources define the Lyapunov time alternatively, such as $1/\lambda_1$ as given in [51]. Another possible definition is $T_d = \ln(2)/\lambda_1$, often called the doubling time in weather modelling [30], which one could say is the time it takes for entropy in the system to increase by one bit. Incidentally, after $16T_\lambda$, $16\ln(10)/\ln(2) = 53.1$ bits of accuracy have been lost, which, rounded down, is the number of bits used to store the mantissa in standard double precision floating point.

For the Lorenz system, we choose a final time of $T_f = 16T_\lambda$, giving the condition number $\kappa = \mathcal{O}(10^{16})$, and exemplify MGRIT performance in Figure 1.6. This figure demonstrates that the residual grows exponentially in time, with average rate λ_1, and thus the solution after 30 iterations does not satisfy our backward error argument, since its residual is large at the later times and MGRIT iterations have stagnated. It

is clear that for chaotic problems, MGRIT alone does not converge to an acceptable solution within a reasonable number of iterations.

Because the residual is approximately an exponential function, the residual at the beginning of the time domain needs to be made exponentially smaller than at the end to converge to a given uniform tolerance. In exact arithmetic, it can be shown that this is possible, but in practice we are limited by floating-point precision. Rounding errors committed in the calculation of $\boldsymbol{\tau}$ on the fine grid are being exponentially magnified on the coarse grid, preventing the solution from being further improved. This implies that, in order to eliminate this stalling behavior without infinite precision, $\boldsymbol{\tau}$ needs to approach zero as the solution converges, implying that $\Phi_c \to \Phi^m$.

We can also use this intuition to estimate the maximum length time domain for which we can expect robust convergence of MGRIT. Given machine precision ε, and a final time T_f, if the subtraction used to calculate $\boldsymbol{\tau}$ introduces $\mathcal{O}(\varepsilon)$ rounding error, and assuming no rounding error in *all other calculations*, then we can expect to be able to reduce the residual at best to $\mathcal{O}(\varepsilon)$ near the beginning of the time domain, and therefore the minimum tol which can be reached at the end of the time domain is $\mathcal{O}(10^{T_f/T_\lambda}\varepsilon)$. Therefore, the "time limit," or maximum time domain size over which MGRIT can robustly converge to tolerance tol is then approximately $\log_{10}(\text{tol}/\varepsilon)T_\lambda$. As an example, assuming we want tol $= \sqrt{\varepsilon}$, we get approximate time limits of $4T_\lambda$ for single precision, $8T_\lambda$ for double precision, and $16T_\lambda$ for quad precision. Unfortunately, we need to double the precision to double the time limit, so this is not a scalable approach.

What's worse, this exponential dependence of the residual at later times on the residual at earlier times implies that not only do we need to double the precision to double the final time, we also need to double the number of MGRIT iterations, since we assume that the residual near the beginning of the time domain is reduced

to $\mathcal{O}(\varepsilon)$, and MGRIT typically requires $\mathcal{O}(-\log_{10}(\text{tol}))$ iterations to converge for well-behaved parabolic problems. Based on this, for fixed h, solving a chaotic system PinT with MGRIT has $\mathcal{O}(\log(n))$ time complexity up until the time limit, and with increasing precision this becomes $\mathcal{O}(n\log(n))$. This means that there is unlikely to be a crossover point after which naive MGRIT is faster than time stepping for many chaotic systems.

The following two chapters seek to remedy this stalling problem by improving the approximation $\Phi_c \approx \Phi^m$. Chapter 2 attempts to improve overall MGRIT convergence for challenging nonlinear problems by constructing higher order time discretizations on coarse grids, while Chapter 3 adds a linear correction term to the coarse grid which has the effect of cancelling the first order error terms that grow exponentially fast on the coarse grid. Chapter 4 attempts to eliminate the ill-conditioning problem by relaxing the initial condition and replacing the forward solve on the coarse grid with a stable PinT relaxation scheme.

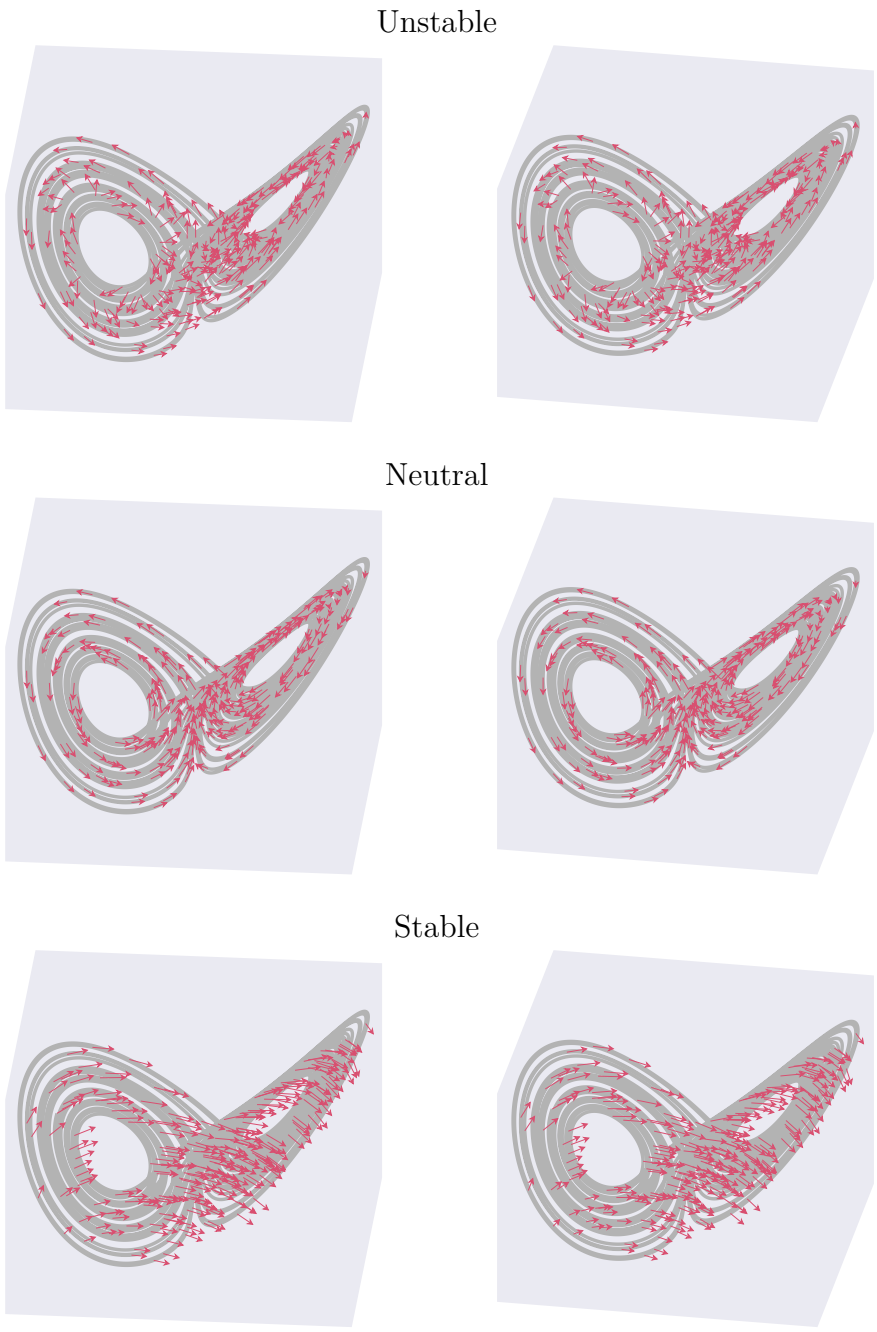


Figure 1.5: Lyapunov vectors of the Lorenz system. ⮞

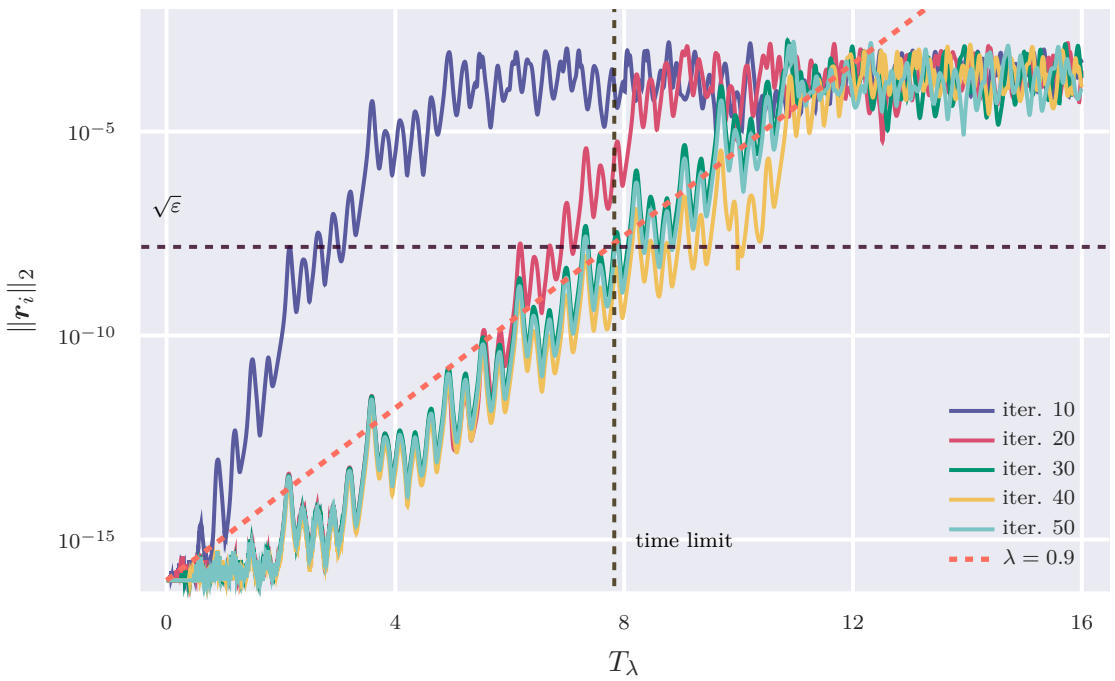


Figure 1.6: Residual over the time-domain (in units of Lyapunov time) for 30 two-level MGRIT iterations on the Lorenz system with $T_f = 16T_\lambda$ and coarsening factor $m = 4$. Convergence stalls after 24 iterations, due to the exponentially growing residual which is not damped by further iterations. The Lorenz system is discretized in time using forward Euler's method.

Chapter 2

θ Methods: Improved MGRIT coarse-grid operators

The material for this chapter is primarily adapted from [54].

It is well known that the convergence rate of MGRIT depends both on the physics of the problem and the properties of the time-stepping operators on the fine and coarse levels [52, 19]. However, the selection of an appropriate coarse operator is an open research problem, since the convergence of MGRIT is often very sensitive to how well the coarse operator approximates m steps of the fine operator. The most common choice of coarse operator is a re-discretization of the fine operator, e.g. if a single-step method is used on the fine grid with step size h, then the same method can be used on the coarse grid with step size mh. This presents an obvious problem for explicit methods; since the fine step size h may already be close to the stability limit of the method, the same method with step size mh may be unstable, making the entire MGRIT iteration unstable. Even stable implicit methods do not guarantee MGRIT stability. An example from [52] shows that when using the Crank-Nicolson

method on the fine and coarse grids, MGRIT has a finite stability region, resembling that of an explicit method, despite the fact that the Crank-Nicolson method is A-stable.

For more difficult problems, such as chaotic or advection dominated systems, when MGRIT is observed to diverge or converge slowly, it can be argued that this is entirely due to the inaccuracy of the approximation $\Phi_c \approx \Phi^m$. Using Φ^m on the coarse grid will result in convergence of MGRIT in a single iteration, so we may hypothesize that there is *some* approximation of Φ^m which is cheaper to evaluate than Φ^m and which results in fast MGRIT convergence, assuming MGRIT convergence is a smooth function of the difference $\Phi^m - \Phi_c$. One natural approach to better approximating Φ^m is to use a higher order method on the coarse grid relative to the fine grid when re-discretizing, although this alone will not necessarily improve the matching between Φ^m and Φ_c, so this should be done carefully, in a way that leverages the approximating power of a higher order method to cancel higher order terms in the Taylor series expansion of $\Phi^m - \Phi_c$.

The idea of enforcing a higher order matching between fine and coarse operators is not new to multigrid. For instance, the work [58] proposes an improved coarse grid operator for spatial multigrid, consisting of a linear combination of different coarse operators $L_{pq} = \alpha L_p + (1-\alpha)L_q$, where L_p and L_q are discretizations of orders p and q, respectively, and α is carefully chosen such that the leading truncation error term of L_{pq} is equal to that of the fine-grid operator. Similarly, [13] proposes a modified coarse-grid discretization for solving hyperbolic problems with MGRIT, based on matching the leading order truncation error terms in space between Φ^m and Φ_c with semi-Lagrangian discretizations. In this chapter, I consider the novel application of this approach to the time-dimension with a general framework for Runge-Kutta methods, resulting in the θ methods.

θ methods are a generalization of work presented by myself and coauthors in [53],

which will be reviewed in Section 2.1. Section 2.2 presents general order conditions which, when satisfied, allow for arbitrarily high order approximation of Φ^m. The general order conditions for fixed time-step size are presented, followed by a method for computing the order conditions for variable time-step size. A general, non-intrusive θ method based on Richardson extrapolation is also presented. Then, in Section 2.5 I apply the linear two-grid convergence theory of [52] to demonstrate the improved convergence provided by these methods for advection diffusion problems. Section 2.5 goes on to provide numerical confirmation of the theoretical convergence bounds and improved convergence for the linear advection-diffusion equation and demonstrates significantly improved parallel performance of MGRIT for a 2D linear advection-diffusion problem in the advection dominated regime. The 2D advection-diffusion problem is solved utilizing time-grid adaptivity provided by the XBraid [57] interface to ARKODE [43], part of the SUNDIALS software package [24, 27], in which I implemented the automatic generation of coarse grid θ methods as well as several critical performance optimizations for implicit Runge-Kutta methods utilizing iterative spatial solvers. Later, in Chapter 3, θ methods are used to greatly improve MGRIT convergence for the chaotic Lorenz system and Kuramoto-Sivashinsky equation.

2.1 Motivation for chaotic problems

Much of this section is adapted from [53].

As we have seen in Section 1.3, one difficulty in solving chaotic systems with MGRIT is that coarsening in time can cause dramatic qualitative changes to the global behavior of the system. Typically, when an explicit method is used on the fine grid, an implicit method is used on very coarse-grids for stability. However, switching from an explicit method to an implicit one causes serious changes to the

behavior of the discretized chaotic system. For example, when using backward Euler to solve the Lorenz equations, the measured greatest Lyapunov exponent, estimated by the average rate of divergence of two nearby trajectories, decreases with increasing time-step size, h, meaning that for large h, a chaotic system can become artificially stabilized, and even non-chaotic. Conversely, using forward Euler, the Lyapunov exponent increases with increasing h, and the system appears *more* chaotic on coarse-grids [6]. Thus, a time-stepping scheme that preserves the qualitative behavior of the system on coarse-grids, is one which lies somewhere between the binary of forward and backward Euler. Figure 2.1 demonstrates this dependence for different time-stepping schemes applied to the Lorenz system, including for the second-order θ method derived later in this section.

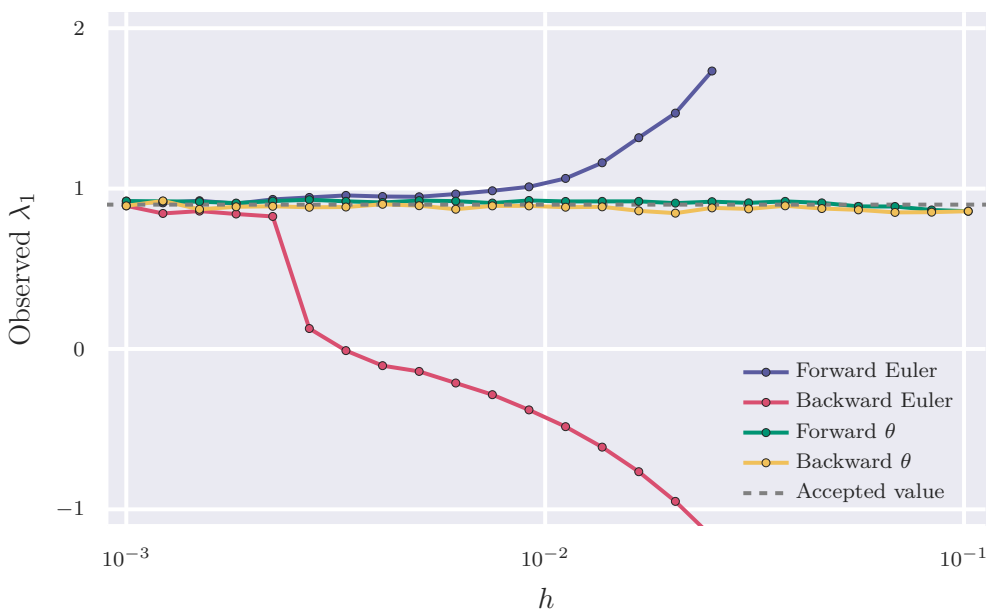


Figure 2.1: Plot of the observed greatest Lyapunov exponent λ_1 for different time-step sizes h, comparing forward Euler, backward Euler, as well as the θESDIRK2 method. For forward and backward Euler, coarsening in time changes the Lyapunov spectrum, while the θ method, using the "forward" and "backward" values of θ, better preserves the Lyapunov exponents.

As shown by this example, a suitable fine-grid method does not necessarily make a suitable coarse-grid method. One option would be to use a higher order method on the coarse-grid, such as ESDIRK2 (Crank-Nicolson method), which would likely better preserve the Lyapunov spectrum. However, higher order methods are more expensive, and in general do not improve MGRIT convergence. Although higher order methods better approximate the *continuous* problem, this is not necessarily useful, since the job of the coarse-grid operator is to approximate the discrete fine-grid problem. In fact, a higher order Φ_c will only agree with Φ^m to the same order of accuracy as Φ agrees with the continuous problem, so in a naive implementation, the accuracy of the approximation $\Phi_c \approx \Phi^m$ is limited by the lowest order of accuracy among the pair Φ_c, Φ, meaning that it is not useful in this context to use a more expensive, higher order method on the coarse-grid. This is illustrated by Figure 2.2, where we see that a coarse solution given by ESDIRK2 yields a better approximation to the exact solution, but is just as accurate as Euler's method with respect to approximating a discrete fine solution given by Euler's method.

What we actually seek is a coarse-grid propagator which better approximates the discrete fine-grid solution than simple re-discretization. Here, we rely on the fact that the fine-grid time-step size, h, is small enough that the operators Φ_c and Φ^m admit convergent Taylor series in h and can therefore be classified by their formal orders of accuracy. In the case where either h or m is large enough that this does not hold, it is likely that for most problems the accuracy of high order methods becomes roughly the same as low order methods. Thus, low order methods should be considered over high order methods on very coarse grids in the MGRIT hierarchy due to concerns of stability and cost, although this is not explored further in this work.

The work [53] gives three principles for deriving coarse-grid methods: First, Φ_c, should have at least the same order of accuracy as the fine-grid, Φ, so that the

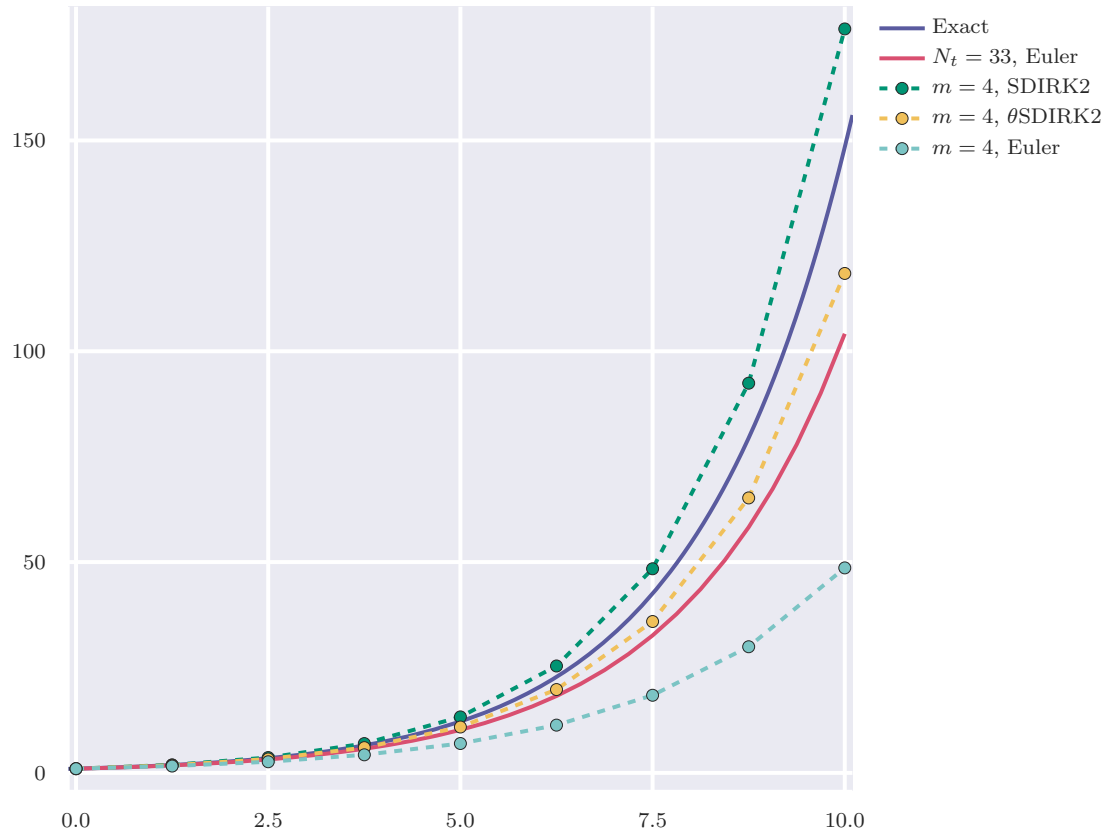


Figure 2.2: A comparison of different coarse solutions to an exact solution and a fine solution of the Dahlquist problem.

coarse-grid discretization is suitably consistent with the fine grid. Second, rather than choosing a single Φ_c, one should consider a k-parameter family of coarse-grid time-steppers, $\Phi_c(\boldsymbol{u}_i, \boldsymbol{\theta})$, whose parameters θ_j, $j = 1, 2, \ldots, k$ can be tuned depending on the physics of the problem and the coarsening factor m. Third, when possible, these parameters are tuned such that Φ_c approximates Φ^m to a higher order than Φ approximates the continuous equation.

More precisely, consider a p-th order fine-grid time stepping operator Φ with m-step stability function $\phi_m(z)$, representing m consecutive fine-grid steps, along with a k-parameter family of time-stepping operators Φ_θ of at least order p, with stability

function $\phi_\theta(mz)$, representing a single large coarse-grid step of size mh. If the system of equations

$$\phi_m(0) = \phi_\theta(0) \tag{2.1a}$$
$$\phi'_m(0) = \phi'_\theta(0) \tag{2.1b}$$
$$\vdots$$
$$\phi_m^{(p+1)}(0) = \phi_\theta^{(p+1)}(0) \tag{2.1c}$$
$$\vdots$$
$$\phi_m^{(p+k)}(0) = \phi_\theta^{(p+k)}(0) \tag{2.1d}$$

has a solution for $\{\theta_j\}_{j=1}^k$, then these parameters yield a method whose stability function ϕ_θ approximates the m-step stability function ϕ_m to order $p+k$, [53]. Note that higher order agreement between the stability functions is not sufficient for higher order agreement between the full operators, i.e. $\Phi_\theta(\boldsymbol{u}) - \Phi^m(\boldsymbol{u}) = \mathcal{O}((mh)^{p+k+1})$, for non-scalar, nonlinear systems beyond 2nd order. Later, in Section 2.2, sufficient conditions are presented for approximation of the full operator.

The work [53] exemplifies θ methods by deriving a coarse-grid propagator Φ_θ to better approximate m steps of forward or backward Euler's method. The ODE $u_t = f(u)$ can be solved to first order by any member of the family of implicit first order single step methods given by

$$\boldsymbol{u}_{i+1} = \boldsymbol{u}_i + h[\theta f(\boldsymbol{u}_i) + (1-\theta) f(\boldsymbol{u}_{i+1})], \tag{2.2}$$

parameterized by $\theta \in [0,1]$ which interpolates between forward ($\theta = 1$) and backward ($\theta = 0$) Euler's method. Note that when $\theta = 1/2$, this θ method becomes equivalent to the second-order Crank-Nicolson method, emphasizing the ability of the method to approximate to second order.

Since (2.2) is first-order for *any* value of θ, the extra degree of freedom in the parameter θ can be used to better approximate the fine-grid *discretization* by choosing

θ in such a way that m fine-grid steps of forward (or backward) Euler are approximated to *second* order in mh. For example, applying forward Euler's method and the θ method to the scalar Dahlquist problem, $u' = \lambda u$ for complex constant λ, let $z = h\lambda$, where h is the time-step size. Then, the stability functions for m steps of forward Euler with time-step h and for the θ method with time-step mh are given by

$$\phi_m(z) = \phi(z)^m = (1+z)^m \quad \text{and} \quad \phi_\theta(z) = \frac{1+\theta m z}{1-(1-\theta)mz}, \tag{2.3}$$

respectively. In order for ϕ_θ to approximate ϕ_m to second order in z, the following three equations must be satisfied:

$$\phi_m(0) = \phi_\theta(0), \quad \phi_m'(0) = \phi_\theta'(0), \quad \text{and} \quad \phi_m''(0) = \phi_\theta''(0). \tag{2.4}$$

While the first and second equality are already satisfied since forward Euler's method and the θ method are both first order in z, the third equality yields

$$m(m-1) = 2(1-\theta_m)m^2 \tag{2.5a}$$

$$1-\theta_m = \frac{m-1}{2m} \tag{2.5b}$$

$$\theta_m = \frac{m+1}{2m}, \tag{2.5c}$$

and a similar calculation yields $\theta_m = (m-1)/(2m)$ for backward Euler's method. Interestingly, in the limit as $m \to \infty$, $\theta_m \to 1/2$, and this θ method approaches the 2nd order Crank-Nicolson method.

Figure 2.3 demonstrates that the θ method indeed appears to have second-order convergence relative to a fixed fine grid solution computed with Euler's method. Notice that, as expected, there is a range of h values for which Euler's method is a better approximation to the fine grid than ESDIRK2, and that θESDIRK2 is a much better approximation to the fine grid than ESDIRK2 for a wide range of h values, while having the same computational cost.

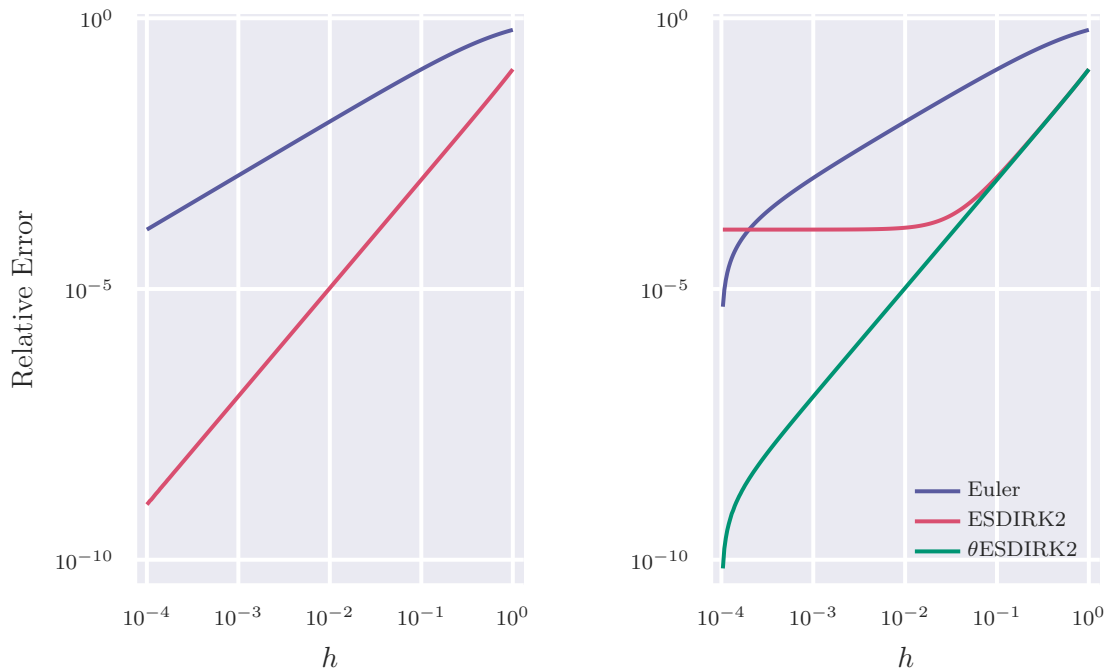


Figure 2.3: Error convergence curves for forward Euler's method, ESDIRK2, and θESDIRK2. Left: Relative error is measured with respect to the exact solution. Right: Relative error is measured with respect to a forward Euler solution with $h_{\text{fine}} = 10^{-4}$, and $\theta = (m+1)/2m$, where $m = h/h_{\text{fine}}$.

Since these values of θ give a method which approximates the fine grid to 2nd order in the scalar case, the natural question is whether this method also better represents the system on coarse grids for nonlinear, multivariable problems. This is indeed the case, as depicted in Figure 2.1, where the θ method is compared to forward and backward Euler applied to the Lorenz system. The θ method accurately preserves the greatest Lyapunov exponent even on coarse time-grids. Thus, for this example, we can say that the θ method addresses, at least in part, the fundamental difficulties with PinT and chaotic problems, preserving the global dynamics on coarse time grids and improving local accuracy. We will see in Section 2.2 that the derivation of this second-order θ method is equivalent to solving the θ method order conditions for second order using first order simplifying assumptions.

In Chapter 3, the MGRIT performance of this second-order θ method is demonstrated numerically on the Lorenz system, and a third order θ method is constructed using the techniques presented in this section which improves MGRIT convergence and parallel performance relative to re-discretization for the solution of the Kuramoto-Sivashinsky equation, a stiff, nonlinear, chaotic PDE.

2.2 Order conditions

For autonomous systems, having the general form

$$\begin{cases} \frac{d}{dt}\boldsymbol{u}(t) = f(\boldsymbol{u}(t)), \\ \boldsymbol{u}(0) = \boldsymbol{u}_0, \end{cases} \tag{2.6}$$

the exact solution at time h, $\boldsymbol{u}(h)$, and the approximate solution at time h, given by Runge-Kutta method Φ, each admit expansions as B-series. Each p-th order term in the B-series is associated with the set of all rooted trees T_p having p nodes. This is because the additive terms found in higher derivatives of $\boldsymbol{u}(t)$ have mathematical structure related to these trees. A rooted tree is a graph, with no simple cycles, having a single node designated as its root. The exact solution to the system of equations (2.6) and any Runge-Kutta method applied to the same system can each be expanded as:

$$\boldsymbol{u}(h) = \boldsymbol{u}_0 + \sum_{j=1}^{p} h^j \sum_{t \in T_j} \frac{1}{\sigma(t)t!} F(t)(\boldsymbol{u}_0) + \mathcal{O}(h^{p+1}), \tag{2.7}$$

$$\Phi(\boldsymbol{u}_0) = \boldsymbol{u}_0 + \sum_{j=1}^{p} h^j \sum_{t \in T_j} \frac{\psi(t)}{\sigma(t)} F(t)(\boldsymbol{u}_0) + \mathcal{O}(h^{p+1}), \tag{2.8}$$

given suitable scalar functions σ, ψ and $\cdot!$, and elementary differential F defined on rooted trees [3]. Evidently, the only difference between the two series are the scalar

multipliers $1/t!$ and $\psi(t)$, hence the formal order conditions for Φ to attain order p are

$$\psi(t) = \frac{1}{t!}, \quad t \in T_j, \quad j = 1, 2, \ldots, p \tag{2.9}$$

where $\psi(t)$ is called the elementary weight function of tree t and depends on the coefficients of the Runge-Kutta method.

Let Φ be a Runge-Kutta method of order p, having s stages, defined by the Butcher table:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \ldots & a_{1s} \\ c_2 & a_{21} & a_{22} & \ldots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \ldots & a_{ss} \\ \hline & b_1 & b_2 & \ldots & b_s \end{array} \equiv \begin{array}{c|c} \boldsymbol{c} & A \\ \hline & \boldsymbol{b}^T \end{array} \tag{2.10}$$

Let $C := \text{diag}(\boldsymbol{c})$, and let $\mathbf{1} = [1, 1, \ldots, 1]^T$ be of length s. Applying the first order simplifying assumptions that $\boldsymbol{b}^T\mathbf{1} = 1$ and $A\mathbf{1} = \boldsymbol{c}$, the order conditions, up to third order, are given in the third column of Table 2.1. Note that the left-hand sides of these equations are the elementary weights $\psi(t)$, and the right-hand sides are equal to $1/t!$, as in (2.9).

We first consider the uniform time stepping case, where both the time-step size and the Butcher table defining Φ is constant across all fine-grid time steps, followed by the more general, variable time stepping case in which h and Φ are allowed to vary across time steps.

2.2.1 Uniform time stepping

In order to derive a θ method, we expand Φ^m as a B-series, then derive the necessary order conditions that Φ_θ must satisfy so that $\Phi^m - \Phi_\theta = \mathcal{O}(h^{q+1})$, where $q > p$. The

Table 2.1: Order conditions for Runge-Kutta and θ Runge-Kutta methods, up to third order.

order, j	t	$\psi(t) = 1/t!$	$\psi_\theta(t) = \psi_m(t)/m^j$
2	⁝	$\boldsymbol{b}^T\boldsymbol{c} = \frac{1}{2}$	$\boldsymbol{b}_\theta^T\boldsymbol{c}_\theta = \frac{1}{2} + \frac{(\boldsymbol{b}^T\boldsymbol{c}-\frac{1}{2})}{m}$
3	⁘	$\boldsymbol{b}^T C\boldsymbol{c} = \frac{1}{3}$	$\boldsymbol{b}_\theta^T C_\theta\boldsymbol{c}_\theta = \frac{1}{3} + \frac{(\boldsymbol{b}^T C\boldsymbol{c}-\frac{1}{3})+(m-1)(\boldsymbol{b}^T\boldsymbol{c}-\frac{1}{2})}{m^2}$
3	⁝	$\boldsymbol{b}^T A\boldsymbol{c} = \frac{1}{6}$	$\boldsymbol{b}_\theta^T A_\theta\boldsymbol{c}_\theta = \frac{1}{6} + \frac{(\boldsymbol{b}^T A\boldsymbol{c}-\frac{1}{6})+(m-1)(\boldsymbol{b}^T\boldsymbol{c}-\frac{1}{2})}{m^2}$

operator Φ^m can be written as an extended, or "block" Butcher table:

$$\begin{array}{c|cccc} \boldsymbol{c} & A & & & \\ \boldsymbol{c}+\boldsymbol{1} & B^T & A & & \\ \vdots & \vdots & \vdots & \ddots & \\ \boldsymbol{c}+(m-1)\boldsymbol{1} & B^T & B^T & \dots & A \\ \hline & \boldsymbol{b}^T & \boldsymbol{b}^T & \dots & \boldsymbol{b}^T \end{array} \equiv \begin{array}{c|c} \hat{\boldsymbol{c}} & \hat{A} \\ \hline & \hat{\boldsymbol{b}}^T \end{array}$$

where $B = \{\boldsymbol{b}, \boldsymbol{b}, \dots, \boldsymbol{b}\}$. Now the elementary weights of Φ^m and Φ_θ can be computed according to the definition in [3], and the order conditions for the θ method can be derived. If $\psi_m(t)$ is the elementary weight of Φ^m for rooted tree t, and $\psi_\theta(t)$ is the elementary weight of Φ_θ for t, then the θ method order conditions are given by

$$\psi_\theta(t) = \frac{\psi_m(t)}{m^j}, \quad t \in T_j, \quad j = 1, 2, \dots, q, \tag{2.11}$$

where the factor of m^j comes from substituting mh for h in the B-series for Φ_θ. For example, the second-order elementary weight for Φ^m is computed by

$$\begin{aligned} \hat{\boldsymbol{b}}^T\hat{\boldsymbol{c}} &= \boldsymbol{b}^T\boldsymbol{c} + \boldsymbol{b}^T(\boldsymbol{c}+\boldsymbol{1}) + \cdots + \boldsymbol{b}^T(\boldsymbol{c}+(m-1)\boldsymbol{1}) = m\boldsymbol{b}^T\boldsymbol{c} + \sum_{k=1}^{m-1} k\boldsymbol{b}^T\boldsymbol{1} \\ &= m\boldsymbol{b}^T\boldsymbol{c} + \frac{m(m-1)}{2} = \frac{m^2}{2} + m\left(\boldsymbol{b}^T\boldsymbol{c} - \frac{1}{2}\right), \end{aligned}$$

and all the other higher order weights can similarly be found. The θ method order conditions, up to third order, are listed in the fourth column of Table 2.1. Note that the right-hand sides only differ from the Runge-Kutta order conditions by an

$\mathcal{O}(1/m)$ term, meaning that in the limit as $m \to \infty$, these order conditions reduce to the classical ones. This explains why the second-order θ method derived in [53] and in Section 2.1 approaches the Crank-Nicolson method as $m \to \infty$. Further, note that the parenthetical terms on the right-hand sides are determined by the order of Φ, e.g. if Φ is second-order, then the $(\boldsymbol{b}^T\boldsymbol{c} - 1/2)$ term vanishes, and the second-order θ condition for Φ_θ reduces to the usual second-order condition. In other words, if Φ is already order p, then Φ_θ must satisfy the classical Runge-Kutta order conditions up to order p, and the θ method order conditions up to order $q > p$ so that $\Phi_\theta - \Phi^m = \mathcal{O}(h^q)$.

2.2.2 Variable time stepping

When MGRIT is equipped with time adaptivity, the fine-grid will not have a uniform time-step size, h, and thus the coarse grids will not in general have a uniform step size either. Further, when using the θ-method coarse grid, the Runge-Kutta method defining Φ will also vary across time steps, since each step on a coarse grid corresponds to a coarse interval on the fine grid which will not have a uniform step size.

In the case that h and Φ are allowed to vary across time steps, the extended Butcher table for Φ^m is more general, and thus the order conditions do not admit simple closed forms. Consider a single coarse interval $k = 1, 2, \ldots, m$, having Runge-Kutta methods Φ_k with time-step sizes h_k. Let $\eta_k = \frac{h_k}{\sum_{s=1}^{m} h_s}$ be the normalized step size, then m-steps with Φ_k having Butcher tables given by $\boldsymbol{b}_k^T$, $\boldsymbol{c}_k$, A_k can be written

as the extended Butcher table

$$\begin{array}{c|ccccc}
\eta_1 \boldsymbol{c}_1 & \eta_1 A_1 & & & & \\
\eta_2 \boldsymbol{c}_2 + \eta_1 \mathbf{1} & \eta_1 B_1^T & \eta_2 A_2 & & & \\
\vdots & \vdots & \vdots & \ddots & & \\
\eta_{m-1} \boldsymbol{c}_{m-1} + \left(\sum_{s=1}^{m-2} \eta_s\right) \mathbf{1} & \eta_1 B_1^T & \eta_2 B_2^T & \dots & \eta_{m-1} A_{m-1} & \\
\eta_m \boldsymbol{c}_m + \left(\sum_{s=1}^{m-1} \eta_s\right) \mathbf{1} & \eta_1 B_1^T & \eta_2 B_2^T & \dots & \eta_{m-1} B_{m-1} & \eta_m A_m \\
\hline
 & \eta_1 \boldsymbol{b}_1^T & \eta_2 \boldsymbol{b}_2^T & \dots & \eta_{m-1} \boldsymbol{b}_{m-1}^T & \eta_m \boldsymbol{b}_m^T
\end{array} \equiv \begin{array}{c|c} \bar{\boldsymbol{c}} & \bar{A} \\ \hline & \bar{\boldsymbol{b}}^T \end{array}.$$

Then, the order conditions may be derived exactly as before. However, when using MGRIT, the time domain is distributed across parallel processors, and thus the processor that owns a particular time step may not have access to all the information needed to compute these order conditions without a significant amount of extra communication. Thus, I propose a method to compute the order conditions for each coarse interval while using the same number of messages as MGRIT traditionally does for F-relaxation followed by a residual calculation.

The above general extended Butcher table can be rewritten as a two-step extended Butcher table

$$\begin{array}{c|cc}
\bar{\boldsymbol{c}}_{m-1} & \bar{A}_{m-1} & \\
\eta_m \boldsymbol{c}_m + \bar{\eta}_{m-1} \mathbf{1} & \bar{B}_{m-1}^T & \eta_m A_m \\
\hline
 & \bar{\boldsymbol{b}}_{m-1}^T & \eta_m \boldsymbol{b}_m^T
\end{array}$$

where $\bar{\eta}_{m-1} = \sum_{s=1}^{m-1} \eta_s$. This splits the extended Butcher table into two parts, the first part comprised of $m-1$ steps, and the second part comprised of a single step. The elementary weights for Φ^m can then be computed in terms of the elementary weights for Φ^{m-1}, and the elementary weights for Φ_m. Given tree t, let $\Psi^{k-1}(t)$ denote the elementary weight for the multiple steps Φ^{k-1}, and let $\psi_k(t)$ denote the elementary weight for the single step Φ_k for $k = 1, 2, \dots, m-1$, where $\bar{\eta}_0 = 0$ and

$\Psi^0(t) = 0$. As an example, the second-order elementary weight for Φ^m is given by

$$\begin{aligned}\Psi^m\left(\begin{smallmatrix}\bullet\\ \bullet\end{smallmatrix}\right) &= \bar{\boldsymbol{b}}^T \bar{\boldsymbol{c}} \\ &= \bar{\boldsymbol{b}}_{m-1}^T \bar{\boldsymbol{c}}_{m-1} + \eta_m^2 \boldsymbol{b}_m^T \boldsymbol{c}_m + \bar{\eta}_{m-1} \boldsymbol{b}_m^T \mathbf{1} \\ &= \Psi^{m-1}\left(\begin{smallmatrix}\bullet\\ \bullet\end{smallmatrix}\right) + \eta_m^2 \psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\end{smallmatrix}\right) + \eta_m \bar{\eta}_{m-1}.\end{aligned}$$

Thus, the elementary weights for Φ^m can be computed recursively, starting with the elementary weights for Φ_1, and the order conditions for Φ_θ can be derived in the same way as before, where the pth order condition for each tree $t \in T_p$ is given by $\psi_\theta(t) = \Psi^m(t)$. Note that here there is no scaling factor since we already scaled each fine-grid time step by η_k/h_k so that the Butcher tables of Φ^m and Φ_θ each span a unit timescale.

Since only scalar elementary weights need to be communicated, and the computation of these elementary weights has the same communication pattern as the MGRIT F-relaxation and residual calculation, i.e. the calculation of $\Psi^k(t)$ depends only on $\Psi^{k-1}(t)$ (computed during the previous time step) and other local information, the order conditions can be computed with the same number of messages as MGRIT traditionally uses during F-relaxation by sending Ψ^{k-1} in the same message as that required to send the state vector $\boldsymbol{u}^{k-1}$. In this way, the order conditions may be derived and solved at the time of the first F-relaxation and residual calculation on each level, with essentially no additional communication. This will need to be done on the first MGRIT iteration, as well as any time the fine-grid is refined in time, since the order conditions will change. In practice, the order conditions may be solved algebraically, leaving the right-hand side as a free variable which is determined by the exact value of $\Psi^m(t)$, or they may be solved numerically using e.g. Newton's method.

The recursive formulae for the elementary weights up to fourth order are given

here:

$$\Psi^m\left(\begin{smallmatrix}\bullet\\ \bullet\end{smallmatrix}\right) = \Psi^{m-1}\left(\begin{smallmatrix}\bullet\\ \bullet\end{smallmatrix}\right) + \eta_m^2\psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\end{smallmatrix}\right) + \eta_m\bar{\eta}_{m-1}$$

$$\Psi^m\left(\begin{smallmatrix}\bullet\\ \bullet\bullet\end{smallmatrix}\right) = \Psi^{m-1}\left(\begin{smallmatrix}\bullet\\ \bullet\bullet\end{smallmatrix}\right) + \eta_m^3\psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\bullet\end{smallmatrix}\right) + 2\bar{\eta}_{m-1}\eta_m^2\psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\end{smallmatrix}\right) + \bar{\eta}_{m-1}^2\eta_m$$

$$\Psi^m\left(\begin{smallmatrix}\bullet\\ \bullet\\ \bullet\end{smallmatrix}\right) = \Psi^{m-1}\left(\begin{smallmatrix}\bullet\\ \bullet\\ \bullet\end{smallmatrix}\right) + \eta_m\Psi^{m-1}\left(\begin{smallmatrix}\bullet\\ \bullet\end{smallmatrix}\right) + \eta_m^3\psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\\ \bullet\end{smallmatrix}\right) + \bar{\eta}_{m-1}\eta_m^2\psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\end{smallmatrix}\right)$$

$$\Psi^m\left(\begin{smallmatrix}\bullet\\ \bullet\bullet\bullet\end{smallmatrix}\right) = \Psi^{m-1}\left(\begin{smallmatrix}\bullet\\ \bullet\bullet\bullet\end{smallmatrix}\right) + \eta_m^4\psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\bullet\bullet\end{smallmatrix}\right) + 3\bar{\eta}_{m-1}\eta_m^3\psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\bullet\end{smallmatrix}\right) + 3\bar{\eta}_{m-1}^2\eta_m^2\psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\end{smallmatrix}\right) + \bar{\eta}_{m-1}^3\eta_m$$

$$\Psi^m\left(\begin{smallmatrix}\bullet\\ \bullet\\ \bullet\bullet\end{smallmatrix}\right) = \Psi^{m-1}\left(\begin{smallmatrix}\bullet\\ \bullet\\ \bullet\bullet\end{smallmatrix}\right) + \eta_m^2\psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\end{smallmatrix}\right)\Psi^{m-1}\left(\begin{smallmatrix}\bullet\\ \bullet\end{smallmatrix}\right) + \eta_m^4\psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\\ \bullet\bullet\end{smallmatrix}\right) + \bar{\eta}_{m-1}\eta_m^3\psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\bullet\end{smallmatrix}\right)$$
$$+ \bar{\eta}_{m-1}\eta_m\Psi^{m-1}\left(\begin{smallmatrix}\bullet\\ \bullet\end{smallmatrix}\right) + \bar{\eta}_{m-1}\eta_m^3\psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\\ \bullet\end{smallmatrix}\right) + \bar{\eta}_{m-1}^2\eta_m^2\psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\end{smallmatrix}\right)$$

$$\Psi^m\left(\begin{smallmatrix}\bullet\\ \bullet\\ \bullet\bullet\end{smallmatrix}\right) = \Psi^{m-1}\left(\begin{smallmatrix}\bullet\\ \bullet\\ \bullet\bullet\end{smallmatrix}\right) + \eta_m\Psi^{m-1}\left(\begin{smallmatrix}\bullet\\ \bullet\bullet\end{smallmatrix}\right) + \eta_m^4\psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\\ \bullet\bullet\end{smallmatrix}\right) + 2\bar{\eta}_{m-1}\eta_m^3\psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\\ \bullet\end{smallmatrix}\right) + \bar{\eta}_{m-1}^2\eta_m^2\psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\end{smallmatrix}\right)$$

$$\Psi^m\left(\begin{smallmatrix}\bullet\\ \bullet\\ \bullet\\ \bullet\end{smallmatrix}\right) = \Psi^{m-1}\left(\begin{smallmatrix}\bullet\\ \bullet\\ \bullet\\ \bullet\end{smallmatrix}\right) + \eta_m\Psi^{m-1}\left(\begin{smallmatrix}\bullet\\ \bullet\\ \bullet\end{smallmatrix}\right) + \eta_m^2\psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\end{smallmatrix}\right)\Psi^{m-1}\left(\begin{smallmatrix}\bullet\\ \bullet\end{smallmatrix}\right) + \eta_m^4\psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\\ \bullet\\ \bullet\end{smallmatrix}\right) + \bar{\eta}_{m-1}\eta_m^3\psi_m\left(\begin{smallmatrix}\bullet\\ \bullet\\ \bullet\end{smallmatrix}\right)$$

2.2.3 Example θ methods

To illustrate the derivation of θ methods using the given order conditions, let Φ and Φ_θ be given by the Butcher tables

$$\Phi : \begin{array}{c|c} \alpha & \alpha \\ \hline & 1 \end{array} \text{ and } \Phi_{\theta\text{ESDIRK2}} : \begin{array}{c|cc} 0 & & \\ 1 & \theta & 1-\theta \\ \hline & \theta & 1-\theta \end{array}.$$

Solving the second-order condition from Table 2.1 for θ yields

$$\theta = \frac{m+1-2\alpha}{2m}.$$

When $\alpha = 1$, in which case Φ is equivalent to backward Euler's method, $\theta = (m-1)/(2m)$, which agrees with the optimal value of θ derived in [53]. This method will be referred to as θESDIRK2, where the ESDIRK2 (Crank-Nicolson) method is given when $\theta = 1/2$.

The θFIRK3 method used in Section 3.5.2 for the Kuramoto-Sivashinsky equation is a linear combination of the second order Lobatto IIIA, IIIB, IIIC, and IIIC* methods, where $\theta_A + \theta_B + \theta_C + \theta_{C^*} = 1$.

$$\theta\text{FIRK3}: \quad \begin{array}{c|cc} \frac{\theta_B}{2} & \frac{\theta_B}{2}+\frac{\theta_C}{2} & -\frac{\theta_C}{2} \\ 1 & 1-\frac{\theta_A}{2}-\frac{\theta_B}{2}-\frac{\theta_C}{2} & \frac{\theta_A}{2}+\frac{\theta_C}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}. \tag{2.12}$$

Other θ methods used in Section 2.5 and implemented in the XBraid-ARKODE interface are

$$\theta\text{SDIRK2}: \quad \begin{array}{c|cc} 1-\theta_1 & 1-\theta_1 & 0 \\ \theta_1 & 2\theta_1-1 & 1-\theta_1 \\ \hline & \theta_2 & 1-\theta_2 \end{array}, \tag{2.13}$$

$$\theta\text{SDIRK3}: \quad \begin{array}{c|ccc} \theta_1 & \theta_1 & 0 & 0 \\ \theta_2 & \theta_2-\theta_1 & \theta_1 & 0 \\ 1 & \theta_3 & 1-\theta_3-\theta_1 & \theta_1 \\ \hline & \theta_3 & 1-\theta_3-\theta_1 & \theta_1 \end{array}, \tag{2.14}$$

and

$$\theta\text{SDIRK4}: \quad \begin{array}{c|cccccc} \theta_1 & \theta_1 & 0 & 0 & 0 & 0 \\ 0.757 & 0.757-\theta_1 & \theta_1 & 0 & 0 & 0 \\ 0.572 & 0.572-\theta_1-\theta_2 & \theta_2 & \theta_1 & 0 & 0 \\ 0.234 & 0.234-\theta_1-\theta_3-\theta_4 & \theta_3 & \theta_4 & \theta_1 & 0 \\ 1 & 1-\theta_1-\theta_5-\theta_6-\theta_7 & \theta_5 & \theta_6 & \theta_7 & \theta_1 \\ \hline & 1-\theta_1-\theta_5-\theta_6-\theta_7 & \theta_5 & \theta_6 & \theta_7 & \theta_1 \end{array}. \tag{2.15}$$

The second-order SDIRK2 method is given by $\Phi_{\theta\text{SDIRK2}}$ with $\theta_1 = \sqrt{2}/2$ and $\theta_2 = 1/2$. The $\boldsymbol{c}$ coefficients for θSDIRK4 (namely 0.757, 0.572, and 0.234) were hand-tuned for performance on advection-dominated problems, but still provide a significant convergence improvement for more diffusive problems as well.

2.3 θ extrapolation

A θ method can be constructed for any time-stepping method, not just Runge-Kutta methods, by using a generalization of Richardson's extrapolation. Richardson's extrapolation is a method for improving the accuracy of a numerical method by combining multiple approximations of the same problem, each with a different step size. Given Φ_h of order p, we can similarly approximate Φ_h^m using two steps of $\Phi_{mh/2}$ with time-step size $\frac{mh}{2}$ and one step of Φ_{mh} with time-step size mh.

$$\theta\Phi_{mh/2}^2(\boldsymbol{u}) + (1-\theta)\Phi_{mh}(\boldsymbol{u}) = \Phi_h^m(\boldsymbol{u}) + \mathcal{O}(h^{p+1}), \tag{2.16}$$

where

$$\theta = \frac{2^p(m^p-1)}{m^p(2^p-1)}. \tag{2.17}$$

Note that in the limit as $m \to \infty$, $\theta \to 2^p/(2^p-1)$ which yields classical Richardson extrapolation.

Since this method only relies on the order of accuracy of Φ_h, it can be applied to any time-stepping method that admits a Taylor series, and which can be re-discretized in time by scaling the time-step size, not just Runge-Kutta methods. However, it does require significantly more work per coarse-grid time step, since it requires three time steps of the fine-grid method per coarse-grid time step, tripling the amount of work done on the coarse grid relative to re-discretization. Thus, it is not as efficient as a θ method can be for Runge-Kutta methods, but it is more general.

In classical linear multigrid, the *operator complexity*, defined as the total number of nonzero entries in all system matrices on every level divided by the number of nonzero entries in the fine-grid system matrix, is a measure of the total storage required by a multigrid method, but since the majority of the cost of a multigrid V-cycle is in relaxation on each level, and the cost of relaxation is typically proportional

to the number of nonzero entries in the system operator at that level, the cost of a V-cycle is proportional to the operator complexity of that V-cycle [2]. For MGRIT, the system matrices aren't stored explicitly on each level, but the majority of the cost of a V-cycle is indeed in F-relaxation, the cost of which is proportional to the number of time points on a given level times the work required for a single time step. Thus, we may analogously define the operator complexity for MGRIT as the total cost of F-relaxation on every level divided by the cost of F-relaxation on the fine grid.

Figure 2.4 demonstrates that MGRIT with θ extrapolation converges as fast, if not faster, than with a comparable Runge-Kutta θ method, and nearly twice as fast as with naive re-discretization, even when iterations are scaled by operator complexity. This provides evidence that theta methods are likely worth implementing in many cases, despite increasing the cost of coarse grids.

2.4 MGRIT convergence bounds

Let the error propagators, E^F and E^{FCF}, give the action of one MGRIT iteration on the error of the current solution guess using F- and FCF-relaxation, respectively. If k is the iteration count, and $\boldsymbol{e}^k$ is the difference between the current solution approximation and exact solution to (1.1), then for F- and FCF-relaxation we have:

$$\boldsymbol{e}^{k+1} = E^F - \boldsymbol{e}^k,$$
$$\boldsymbol{e}^{k+1} = E^{FCF}\boldsymbol{e}^k.$$

Here, we assume that Φ and Φ_c are Runge-Kutta methods applied to the linear system of equations:

$$\begin{cases} \frac{d}{dt}\boldsymbol{v}(t) = G\boldsymbol{v}(t) + \boldsymbol{f}(t),\ 0 \leq t \leq T, \\ \boldsymbol{v}(0) = \boldsymbol{g}_0 \end{cases} \tag{2.18}$$

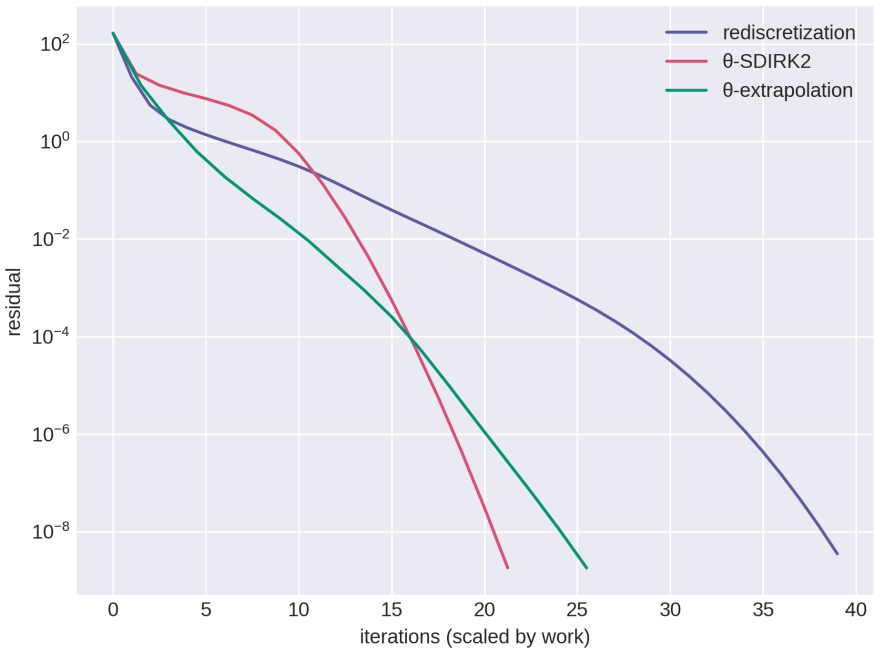


Figure 2.4: Multilevel MGRIT residual convergence for 1D linear advection comparing naive rediscretization with θ method coarse grids. The fine grid is discretized using backward Euler's method and the MGRIT coarsening factor $m = 4$. The iteration counts are scaled by the operator complexity of the V-cycle, as a proxy for the increase in work caused by the more expensive coarse-grid methods.

where the linear operator G is diagonalizable, with eigenvalues γ_ω. Further, we assume that Φ and Φ_c are stable in the sense that $\|\Phi\| < 1$ and $\|\Phi_c\| < 1$. In this case, given time-step size h, the operators Φ and Φ_c will be rational functions of hG and mhG, respectively, and will thus be diagonalized by the same basis as G. Let λ_ω and μ_ω denote the eigenvalues of Φ and Φ_c, respectively, then the two level MGRIT error propagators may be bound by

$$\|E^F - \boldsymbol{e}\|_2 \leq \max_\omega \left\{ |\lambda_\omega^m - \mu_\omega| \frac{(1 - |\mu_\omega|^N)}{(1 - |\mu_\omega|)} \right\} \|\boldsymbol{e}\|_2 \tag{2.19}$$

$$\|E^{FCF}\boldsymbol{e}\|_2 \leq \max_\omega \left\{ |\lambda_\omega^m - \mu_\omega| \frac{(1 - |\mu_\omega|^N)}{(1 - |\mu_\omega|)} |\lambda_\omega|^m \right\} \|\boldsymbol{e}\|_2 \tag{2.20}$$

[52]. Note that in the limit as $N \to \infty$, the numerator term $(1 - |\mu_\omega|^N) \to 1$, by the assumption $|\mu_\omega| < 1$. Thus, this term will be assumed to be 1 for the remainder of this paper, so that the convergence bounds will be independent of the grid size N. In this limit, the $|\lambda_\omega^m - \mu_\omega|/(1 - |\mu_\omega|)$ term dominates. The numerator of which depends on how well the eigenvalues of Φ^m are approximated by the eigenvalues of Φ_c. Also, note that FCF-relaxation has the effect of multiplying the convergence rate by a constant factor of $|\lambda_\omega|^m$, meaning that the convergence rate will be reduced by using FCF-relaxation in all cases where the $|\lambda_\omega|$ are smaller than unity.

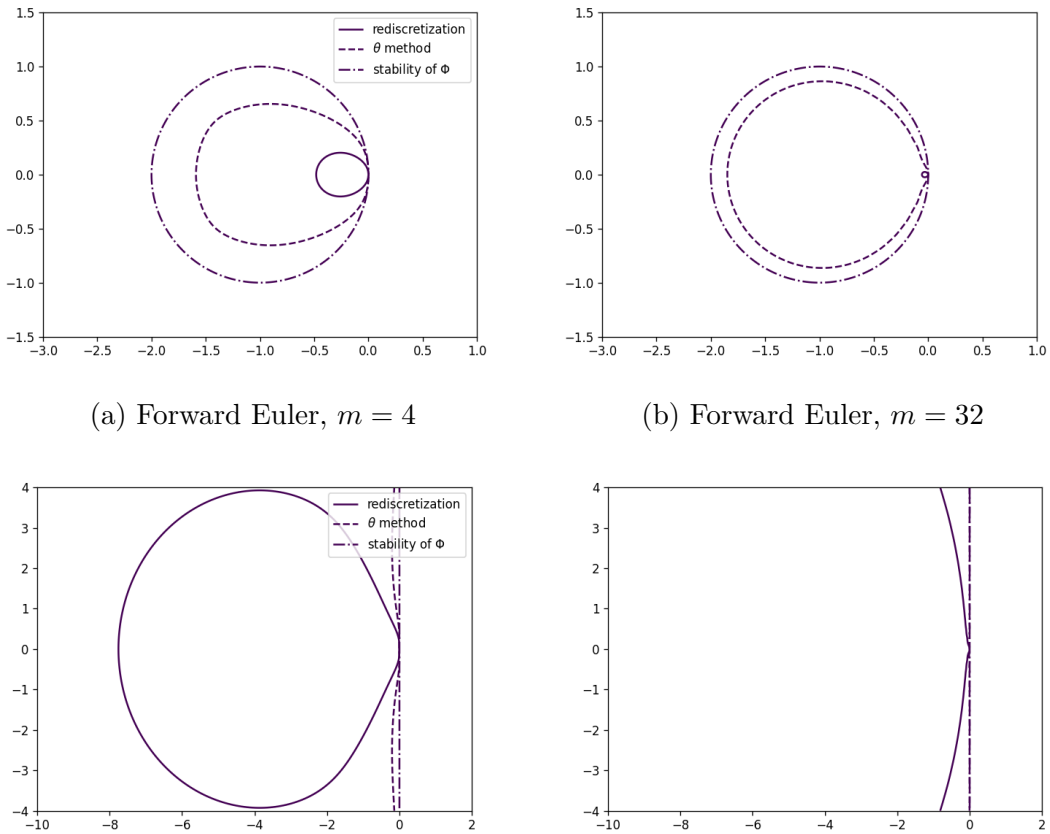
2.4.1 MGRIT stability

For a particular spatial eigenvalue γ, λ and μ are fixed, so, letting $z = h\gamma$,

$$\rho^F(z) = \frac{|\lambda^m(z) - \mu(mz)|}{(1 - |\mu(mz)|)}, \text{ and } \rho^{FCF}(z) = \frac{|\lambda^m(z) - \mu(mz)|}{(1 - |\mu(mz)|)}|\lambda(z)|^m, \tag{2.21}$$

act as bounds on the action of the error propagator on the particular eigenvector of G, and the functions $\lambda(z)$ and $\mu(z)$ are the stability functions of the Runge-Kutta methods Φ and Φ_c, respectively. If we plot the region in the complex plane where $\rho^F(z), \rho^{FCF}(z) \leq 1$, we can visualize the stability region for the MGRIT iteration, analogous to the stability region of a time-stepping method. Figure 2.5 plots the stability regions for MGRIT when using forward Euler's method and the Crank-Nicolson method. When forward Euler is used with re-discretization, (2.5a, 2.5b), the MGRIT stability is limited by the coarsening factor, m. When Crank-Nicolson is used with re-discretization, (2.5c, 2.5d), the stability of MGRIT resembles that of an explicit method, having a finite, bounded stability region for any m, despite the A-stability of the Crank-Nicolson method. While the instability of MGRIT when using forward Euler is to be expected (since the coarse grid equation becomes more unstable as the time-step size increases), the limited stability when using Crank-Nicolson is more surprising. According to [52], A-stability is not sufficient for MGRIT stability,

and at least L-stability is necessary. In the stiff limit, as $\gamma \to \infty$, [19] demonstrates that if the fine-grid method is A-stable, then it is sufficient that the coarse grid be L-stable to ensure MGRIT stability, although MGRIT may converge arbitrarily slowly in this case, since the convergence rate is bounded above by one. Indeed, MGRIT is only certain to converge quickly in the stiff limit if both the fine- and coarse-grid methods are L-stable.



(a) Forward Euler, $m = 4$

(b) Forward Euler, $m = 32$

(c) Crank-Nicolson, $m = 4$

(d) Crank-Nicolson, $m = 32$

Figure 2.5: The stability regions for MGRIT and the fine-grid operator using forward Euler and the Crank-Nicolson method. The contours plot the stability boundaries $\rho^{FCF}(\gamma) = 1$ (solid), $\rho_\theta^{FCF}(\gamma) = 1$ (dashed) and $|\Phi(\gamma)| = 1$ (dash-dot).

The proposed θ methods can be designed to improve MGRIT stability. As an

example, when forward Euler's method is used with the θESDIRK2 method on the coarse grid, the stability region for MGRIT actually increases in size with m, meaning that MGRIT stability is limited by the stability of Φ, not of the coarse grid Φ_c. When the Crank-Nicolson method is used with the θSDIRK3 method, the stability region expands to include the entire negative real axis. Thus, we say that the appropriate choice of coarse grid operator can stabilize what would be an unstable MGRIT iteration when using naive re-discretization.

Because hyperbolic problems are notoriously difficult to solve with PinT methods, with MGRIT typically diverging when G has any purely imaginary eigenvalues, we are encouraged to explore whether, given a fine-grid discretization of a hyperbolic problem, there exists some coarse-grid θ method which results in a convergent MGRIT iteration. Notice that in every plot in Figure 2.5, the imaginary axis is not included in the MGRIT stability region, this indicates that if the spatial eigenvalues γ_ω lie on the imaginary axis, we should not expect MGRIT to converge, and MGRIT will converge very slowly for eigenvalues near the boundary of the stability region, in particular near the imaginary axis. This illustrates the main problem with solving hyperbolic problems using MGRIT, since stability of the fine-grid method for imaginary eigenvalues does not imply MGRIT stability. Let us now examine this more carefully, under the assumptions of [52].

Lemma 1. *The MGRIT stability region, defined as* $P = \{z \in \mathbb{C} : \rho^F(z) \leq 1\}$ *is a subset of the fine-grid stability region* $L = \{z \in \mathbb{C} : |\lambda(z)| \leq 1\}$.

Proof. Close inspection of the proof given in [52] of the bounds in Equation (2.21) reveals that the assumption that $\|\Phi\| < 1$ is not necessary, and thus these bounds

are valid even if $|\lambda(z)| > 1$. For any $z \notin L$, $|\lambda(z)| > 1$, and

$$\rho^F(z) = \frac{|\lambda^m - \mu|}{(1 - |\mu|)} \tag{2.22}$$

$$\geq \frac{||\lambda^m| - |\mu||}{1 - |\mu|} \tag{2.23}$$

$$> \frac{1 - |\mu|}{1 - |\mu|} = 1, \tag{2.24}$$

$$\implies z \notin P \tag{2.25}$$

since by assumption $|\lambda^m| > |\lambda| > 1 > |\mu|$. Thus, $P \subseteq L$. ☠

A corollary of Lemma 1 is that it is increasingly difficult to stabilize MGRIT near the imaginary axis for higher order fine-grid methods for which the fine-grid stability boundary approaches the imaginary axis in a neighborhood of the origin, even for L-stable methods. Further, MGRIT can never be made to converge with a symplectic Runge-Kutta method on the fine grid with any Runge-Kutta method coarse grid, because symplectic methods by design have the imaginary axis itself as their stability boundary, meaning that the MGRIT stability region at best shares this boundary, and thus MGRIT cannot be convergent for any purely imaginary eigenvalue. Note that this argument only holds under the assumptions of [52], namely that the fine- and coarse-grid methods are simultaneously diagonalizable, which is true for the Runge-Kutta θ methods when the linear operator G is diagonalizable and also for the θ extrapolation method whenever Φ is diagonalizable. However, for more exotic discretizations, such as the semi-Lagrangian methods presented in [13], it is possible for MGRIT to be made convergent for purely hyperbolic problems, although the kind of theory that can express this rigorously for general discretizations remains an open problem.

Although the presented θ methods cannot fix the divergence of MGRIT for purely hyperbolic problems, it is often the case that θ methods improve MGRIT convergence for spatial eigenvalues near the imaginary axis and result in an MGRIT stability

region which lies closer to the imaginary axis, and therefore we should expect these methods to be very useful for parabolic problems which are advection dominated, as demonstrated in Section 2.5.

2.4.2 Considerations for multilevel MGRIT

For MGRIT with more than two levels and variable time-stepping, the θ method for each level can be derived with m^ℓ in place of m, where ℓ is the level in the MGRIT hierarchy, counting up from the fine grid which is level 0. This ensures that Φ_θ on each level approximates the fine-grid time-stepping operator to the same higher order of accuracy, implying that the method on each level approximates the method on the previous level to the same order of accuracy. This argument does not hold for variable time stepping, however, so the θ methods on each coarse interval need to be derived so that they approximate the m steps on the fine grid that span that interval. This can be done in practice during the first MGRIT down-cycle, or, if the first down-cycle is skipped, a mock down-cycle can be performed to compute the θ methods prior to the first up-cycle.

Multigrid algorithms, including classical multigrid, Parareal, and MGRIT, generally rely on the coarse grid being much cheaper to solve than the fine grid, so the θ method, which increases the cost of time stepping on the coarse grid relative to naive re-discretization, may appear counter to this goal. Indeed, for Parareal and two-level MGRIT, it is very important that the coarse grid be orders of magnitude cheaper than the fine-grid, or else one iteration of the algorithm will be expensive with little meaningful speedup. However, the MGRIT algorithm relies on paralleliz-ing the work on the expensive finer grids while only using sequential time stepping to solve the coarsest grid, which will only consist of one or two time steps. Thus, each processor only has one or a handful of sequential time steps to execute on each level. As we will see shortly, the extra cost incurred by θ methods is typically offset

by the improvement to MGRIT convergence they can provide, and this cost may be further mitigated by using a larger coarsening factor, which is often possible because of the improved convergence rate and stability due to the θ method, as demonstrated in the numerical results for multi-level MGRIT presented in Section 2.5.

It is also reasonable to ask whether, when using the multilevel algorithm, one should apply the θ method strategy recursively, using increasingly high order methods on coarser grids. However, it is my opinion that this is unlikely to be necessary. This is because the multilevel convergence rate is bounded from below by the two-level convergence rate given the fine grid and first coarse grid. For two levels, the first coarse grid is solved exactly, while for three levels, this exact coarse-grid solve is replaced by an inexact solve, i.e. a recursive application of MGRIT, meaning the three-level algorithm can at most converge as fast as the two-level one. For this reason, increasing the order of accuracy of the second coarse grid beyond that of the first coarse grid is not likely to improve the convergence rate very much, unless a serious degradation of convergence is observed when going from two levels to three.

2.5 Theoretical and numerical results

In order to illustrate the potential of the proposed θ methods for accelerating MGRIT convergence and improving stability, this section presents theoretical convergence bounds and observed asymptotic convergence rates for MGRIT applied to the one-dimensional linear advection-diffusion equation, along with space-time parallel strong scaling results for the two-dimensional linear advection-diffusion equation that demonstrate the superiority of θ methods as coarse grid discretizations. Then, a parallel study is presented for the space-time parallel solution of the 2D linear advection-diffusion equation in the advection dominated regime, where adaptive MGRIT is shown to significantly outperform adaptive time stepping and θ methods

are demonstrated to have a clear advantage over naive re-discretization in terms of parallel efficiency.

2.5.1 1D linear advection-diffusion

The one-dimensional linear advection-diffusion equation is given by

$$\boldsymbol{u}_t = \nu \boldsymbol{u}_{xx} - \alpha \boldsymbol{u}_x. \tag{2.26}$$

The parameters α, $\nu \in \mathbb{R}$ are used to tune how advective and diffusive the problem is, respectively. The problem is posed along with periodic boundary conditions for $x \in [0, 1)$ and $t \in [0, 1.25]$, with initial condition given by the half sin wave $\boldsymbol{u}(x, 0) = \sin(\pi x)$. This is then discretized on a uniform grid using centered second-order and upwind first-order finite differences for the second and first derivatives, respectively. The problem size considered here is $n_x \times n_t = 2^7 \times 2^{10}$, with time-step size h_t and spatial grid spacing h_x, and thus the ratio $h_t/h_x = 8$.

For the diffusive case, ($\nu = 1$, $\alpha = 0$) Table 2.2 records the observed MGRIT convergence rate along with the theoretical bound $\max\{\rho^{FCF}\}$ for a given combination of fine and coarse operators Φ, Φ_c. The theoretical bound given is the approximate maximum convergence rate for spatial eigenvalues on the negative real axis. When Φ_c is a θ method, the 'speedup' column records $\log(\max\{\rho_\theta^{FCF}\})/\log(\max\{\rho^{FCF}\})$, which roughly indicates the expected speedup due to the improvement in iteration count required for the method to converge to a fixed tolerance. An entry of 'div' indicates that the MGRIT iteration became numerically unstable within 32 iterations, '∞' indicates that the function ρ^{FCF} is unbounded, and '*' indicates that the speedup for a particular method is not available, due to numerical instability or a convergence rate greater than 1, which indicates divergence. Especially noteworthy are the around 2× speedups provided by the θESDIRK2, θESDIRK2, and θ extrapolation methods when used with the SDIRK1 fine grid. Also of note is the MGRIT

Table 2.2: Diffusion, ($\nu = 1$, $\alpha = 0$)

		$m = 4$, FCF			$m = 32$, FCF		
Φ	Φ_c	obs.	ρ^{FCF}	speedup	obs.	ρ^{FCF}	speedup
SDIRK1	SDIRK1	0.077	0.081		0.059	0.108	
	θESDIRK2	0.017	0.020	1.56	0.020	0.021	1.72
	θSDIRK2	0.004	0.006	2.04	0.008	0.011	2.04
	θ extrap.		0.005	2.11		0.012	1.99
SDIRK2	SDIRK2	0.007	0.010		0.009	0.011	
	θSDIRK3	0.003	0.005	1.15	0.002	0.005	1.17
	θ extrap.		0.0007	1.58		0.0009	1.56
ESDIRK2	ESDIRK2	div	∞		div	∞	
	θSDIRK3	0.670	1	*	0.038	1	*
	θ extrap.		∞	*		∞	*

convergence bound for ESDIRK2 which is unbounded with naive re-discretization, however, the iteration becomes stable when using the θSDIRK3 coarse grid, and although convergence is relatively slow for $m = 4$, convergence is very fast for $m = 32$.

For the advective case, ($\nu = 0$, $\alpha = 1$) Table 2.3 records the observed MGRIT convergence rate along with the theoretical bound $\max\{\rho^{FCF}\}$ for the same methods as before in Table 2.2. Here, the theoretical bound is the approximate maximum convergence rate for spatial eigenvalues on the imaginary axis. Of note here is the combination of SDIRK1 with the θSDIRK2 and θ extrapolation coarse grids, with theoretical speedups of 5.96× and 5.46× respectively, relative to naive re-discretization. This is a good result for a purely hyperbolic problem at a coarsening factor of $m = 4$. For the second-order fine-grid methods tested, no coarse-grid method yields a theoretical convergence bound less than one. However, although the theoretical bounds for the SDIRK2 and ESDIRK2 methods indicate divergence, interestingly, MGRIT converges quickly for SDIRK2 when $m = 4$ with SDIRK2 and

Table 2.3: Advection, ($\nu = 0$, $\alpha = 1$)

		$m = 4$, FCF			$m = 32$, FCF		
Φ	Φ_c	obs.	ρ^{FCF}	speedup	obs.	ρ^{FCF}	speedup
SDIRK1	SDIRK1	0.689	0.750		1.23	1.28	
	θESDIRK2	0.168	0.767	0.922	4.73	21.6	*
	θSDIRK2	0.085	0.180	5.96	2.60	3.03	*
	θ extrap.		0.208	5.46		1.2	*
SDIRK2	SDIRK2	0.100	∞		2.82	∞	
	θSDIRK3	0.060	2.22	*	1.95	2.54	*
	θ extrap.		∞	*		∞	*
ESDIRK2	ESDIRK2	div	∞		div	∞	
	θSDIRK3	div	2.17	*	div	2.54	*
	θ extrap.		∞	*		∞	*

θSDIRK3 coarse grids.

2.5.2 Space-time parallel 2D linear advection diffusion with adaptive time stepping

Although the previous section demonstrates that the MGRIT convergence rate is improved through the use of coarse grid θ methods, it remains to be seen if this improved convergence offsets the additional cost of using a more expensive coarse-grid method in an actual parallel study. To test this, consider the 2D linear advection diffusion equation with Dirichlet boundary conditions

$$u_t = \kappa(u_{xx} + u_{yy}) + \alpha_x u_x + \alpha_y u_y, \tag{2.27}$$

$$u_t(t, 0, y) = u_t(t, 1, y) = u_t(t, x, 0) = u_t(t, x, 1) = 0, \tag{2.28}$$

and initial condition

$$u(0, x, y) = \sin^2(\pi x) \sin^2(\pi y). \tag{2.29}$$

For the following experiments, the constants $\kappa = 10^{-3}$, $\alpha_x = \alpha_y = 1$, are fixed so that the problem is advection dominated. The Dirichlet outflow boundary makes for a numerically challenging simulation as the initial sine hump collides with the boundary and produces large spatial gradients. This problem is discretized in space on a uniform spatial grid with 512×512 points with second-order centered finite differencing. The temporal discretization is an SDIRK2 method with first order embedded error estimator provided by the ARKODE module of the SUNDIALS software package [43, 27, 24].

The implicit SDIRK stages are solved using GMRES preconditioned with a single V-cycle of PFMG, a parallel spatial multigrid algorithm provided by the hypre software package [29] with two pre- and post-iterations of symmetric red-black Gauss-Seidel relaxation. In the sequential time stepping case, the PID time-adaptivity controller from ARKODE is used along with error estimates from the embedded error estimator. The MGRIT implementation used is from XBraid [57], with coarse grid adaptive SDIRKθ methods implemented by myself on top of the ARKODE-XBraid interface. The θSDIRK methods are implemented in the XBraid-ARKODE interface in a very general way, such that if a user already has a SUNDIALS application code that utilizes an ARKODE SDIRK method for adaptive time stepping, they can enable the XBraid wrapper in just a few lines of code. Then, with the XBraid wrapper enabled, the user has access to θ methods up to fourth order with just two lines of code:

```
ARKBraid_SetTheta(app, true);
ARKBraid_SetCoarseOrder(app, fine_ord+1);
```

For this experiment, the MGRIT coarse-grid method is varied in order to compare

naive re-discretization (SDIRK2) with the θSDIRK3 and θSDIRK4 methods given in (2.14) and (2.15). Temporal adaptivity is achieved in MGRIT using the I controller from ARKODE along with error estimates given to generate refinement factors for XBraid's built in temporal refinement routine. Relative and absolute error tolerances of 10^{-7} were used for temporal refinement, an absolute tolerance of 5^{-9} was used for the GMRES solver, and an absolute residual tolerance of 10^{-7} was used for MGRIT convergence. An MGRIT coarsening factor of 4 was used beyond the first level for all experiments, while coarsening factors of 6, 8, and 12 were used on the first level for coarse-grid methods SDIRK2, θSDIRK3, and θSDIRK4 respectively. An initial study using a range of coarsening factors was used to select these as the optimal values for each method, based on the maximum parallel speedup given by each method and each coarsening factor.

Figure 2.6 demonstrates the improvement to scaling provided by MGRIT, and especially highlights the improvement in parallel speedup when using the third or fourth order θ methods on the coarse grid. This study was performed on Ruby, a parallel cluster at Lawrence Livermore National Laboratory.[1] We see here that MGRIT is especially useful once the spatial parallel algorithm has saturated, since after that point, further speedup is *only* possible by using PinT. In this regime, MGRIT with the θ method is always faster than parallel in space alone on the same number of processors for this problem, and the total speedup achieved using space-time parallelism is nearly an order of magnitude greater than the speedup with spatial parallel alone. Finally, note that MGRIT has not actually reached its theoretical saturation limit yet in this experiment, with only about %20 of the total available parallel work in the time dimension actually being performed in parallel, even for the run with 8192 processors. Unfortunately, scaling beyond this point stagnated

[1]Ruby is a peak 5.9 PFLOP cluster commissioned in 2020 from Supermicro utilizing an Intel Omni-Path Interconnect, with 1,480 batch nodes and two 28-core, 2.2 GHz Intel Xeon CLX-8276L processors with 192 GiB of memory per node.

unexpectedly due to what I suspect to be issues with the machine. Ideally, I would like to repeat these experiments on another machine to confirm this.

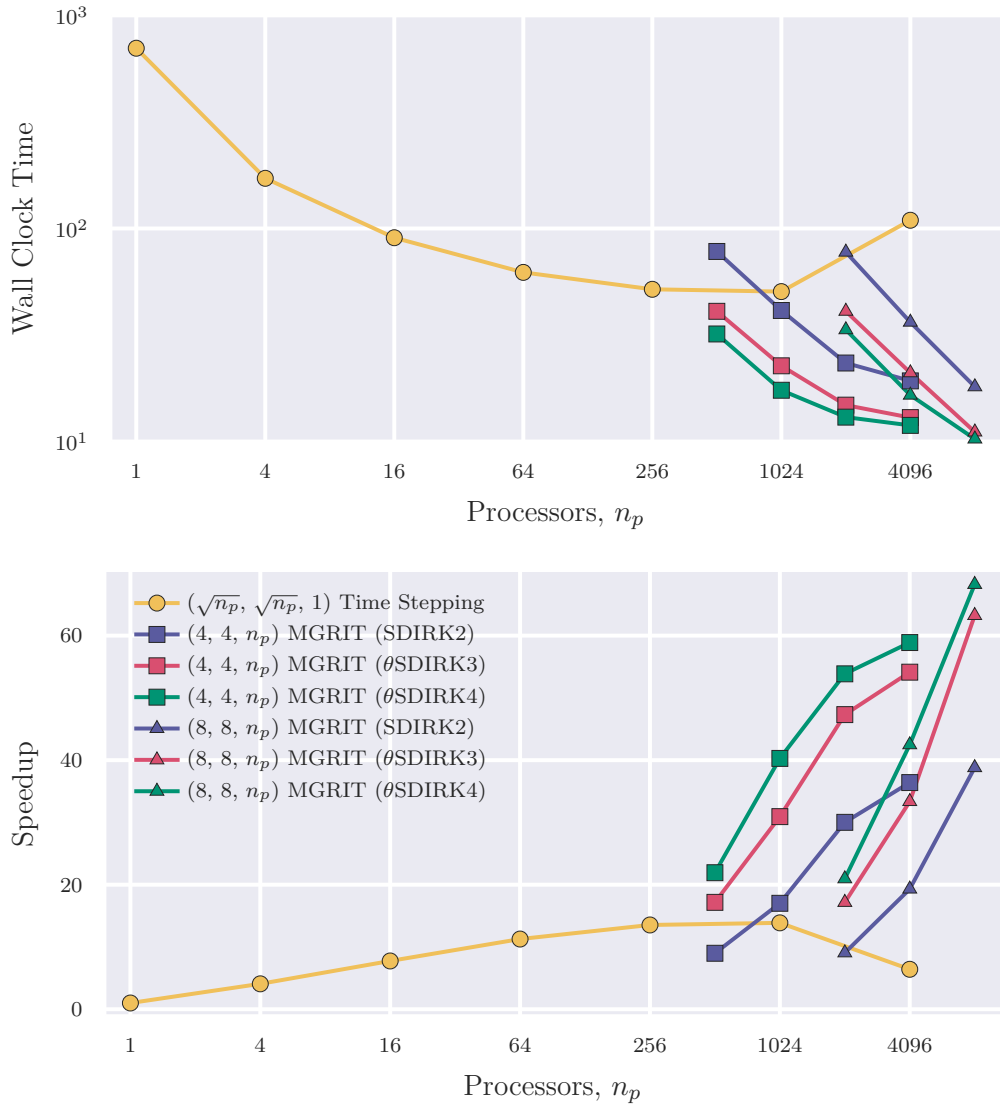


Figure 2.6: Results for a space-time strong scaling study for 2D linear advection diffusion equation. Legend entries indicate the number of processors allocated to each dimension, (p_x, p_y, p_t), as a function of the total number of available processors, n_p; the temporal solver, either time stepping or MGRIT; and the coarse-grid method used in MGRIT. The baseline for speedup is the time required for sequential time stepping on a single processor in space.

Table 2.4: Time grid sizes and error test failure rates using time stepping and MGRIT with different coarse-grid methods.

Method	m	time steps (n_t)	% error test fails
adaptive time stepping (PID)	-	8882	.03%
MGRIT (SDIRK2)	6	11650	3.7%
MGRIT (θSDIRK3)	8	7174	1.1%
MGRIT (θSDIRK4)	12	7352	2.4%

Since different controllers were used for time adaptivity in the sequential and parallel in time cases, it is interesting to compare the final sizes of the time grid after solving the problem with each method. As can be seen in Table 2.4. Note that even though for a fixed time grid MGRIT converges to the same solution as time stepping, the same is not strictly true with temporal adaptivity, even if the same controller were used. This is because refinement in XBraid is treated like adaptive mesh refinement (AMR), where an initial coarse grid is refined by subdividing time steps into sub-steps, whereas time adaptivity in time stepping employs feedback control to continuously control the error, only backtracking and subdividing a step when the error tolerance is violated.

For adaptive time stepping, it is considered expensive to recompute a time step after failing the error tolerance test, so an ideal controller would keep the error continuously below the tolerance without ever overshooting it. The PID controller uses the error estimates and step sizes from the last three time steps to continuously control the step size at the current time step, and generally it is quite efficient.

In contrast, for adaptive MGRIT, failing the error test at a given point in time is only expensive if it causes refinement after the residual has reached the halting tolerance, because this causes an extra MGRIT iteration to occur which could have been avoided if that refinement had taken place during an earlier iteration when the residual was much larger. Ideally, MGRIT would identify the optimal time grid during the first iteration and never need to refine again, as that would likely result in

the fastest convergence and prevent refinements which in XBraid involves reallocation of all coarse-grid data structures. However, since the solution has very low accuracy during the first couple of MGRIT iterations, early refinements can over-refine in some areas while under-refining others, resulting in a much larger than necessary time grid. Further, adaptivity in MGRIT is limited to an I controller which only uses the current step size and error estimate, because the PID controller requires the last three *refined* step sizes and error estimates to work, which are not available at a given point in time during a given MGRIT iteration. Note that for sequential time stepping, the I controller is typically less efficient than the PID controller.

Despite these difficulties, MGRIT has the potential to be more efficient than adaptive time stepping at temporal adaptivity. This is because adaptive MGRIT starts from a coarse time grid which is iteratively refined until the error test is satisfied at every time point. Since failing the error test at a given time point is not considered expensive, MGRIT can be fairly conservative when refining. Contrast this with adaptive sequential time stepping where repeating a step *is* considered expensive, so an initially small step size is chosen which is conservatively allowed to increase while still keeping the error estimate below the tolerance to avoid repeating steps. Therefore, all other things being equal, we should expect adaptive MGRIT to be less likely to over-refine than adaptive time stepping.

Table 2.4 demonstrates that, using the θ method coarse grids, MGRIT is able to find a more efficient grid than adaptive time stepping, as expected, although we don't observe this when comparing adaptive time stepping with MGRIT and naive re-discretization. Intuitively, this makes sense because on coarse time grids, the θ methods are roughly equivalent to their maximal order method (e.g. θSDIRK4 $\approx$ SDIRK4 for large m) and so the first iteration results in a much better initial guess for the whole time domain than when using the lower order re-discretization, and a better initial guess means the error estimates are more accurate and early refinements

are less likely to over-refine, resulting in a closer to optimal time grid. The difference in grid size between MGRIT with θSDIRK3 and θSDIRK4 coarse grids is small, and it is unknown what has caused this difference.

Overall, these results demonstrate the efficacy of MGRIT with θ method coarse grids, not only improving MGRIT convergence rates in nearly all cases, but more than making up for their increased cost in parallel speedup.

Chapter 3

$\triangle$ correction

The material for this chapter is adapted from [53].

As we have seen in Section 1.3, naive MGRIT fails to converge for the Lorenz system over long time domains due to a residual which grows exponentially in time. For systems where the underlying dynamics are chaotic, MGRIT is very sensitive to errors, no matter how small. This is because while the $\boldsymbol{\tau}$ correction makes the fine-grid solution a fixed point on the coarse grid, there may still be a significant mismatch between the Lyapunov spectra on the coarse and fine grids, making this fixed point unstable. This mismatch is why even small errors on the coarse grid are exponentially magnified along incorrect Lyapunov coordinate vectors, causing the observed exponentially increasing residual seen in Figure 1.6. The mismatch can be measured (and eventually corrected for) by considering the tangent linear propagator, F_i, along the trajectories of the fine and coarse operators. We will see that F_i on the coarse grid needs to be extremely accurate relative to the fine grid.

Since we know that perturbations only grow along the unstable manifold for trajectories of the Lorenz system, we should expect that the components of the error along the unstable manifold are the slowest to converge, while the other components

of the error converge more quickly. We further do not want the error in the unstable manifold, which may be large, to affect the residual. Thus, we need to understand how perturbations along these manifolds are propagated in finite time.

3.1 Lyapunov analysis of MGRIT convergence

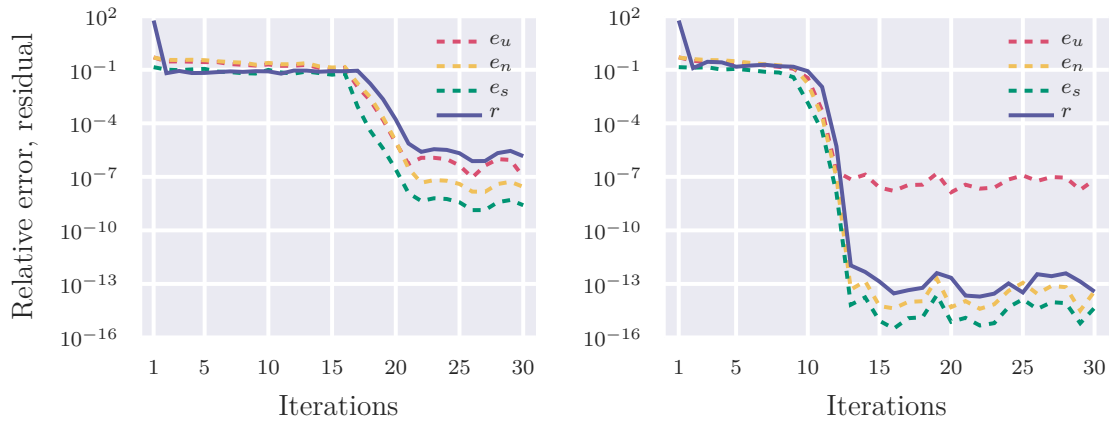
Recall the forward time stepping relation for the backward Lyapunov vectors,

$$F_i \Psi^-(t_i) = \Psi^-(t_{i+1}) R_{i+1}, \tag{1.12 revisited}$$

where R_{i+1} is upper triangular. This equation indicates that, forward in time, the *backward* Lyapunov vectors are mapped to vectors which are orthogonal to the set of Lyapunov vectors with lesser Lyapunov exponent. Thus, a perturbation along the unstable manifold will remain orthogonal to the neutral and stable manifolds for all time, while a perturbation along the stable manifold will almost surely have a nonzero projection on the neutral and unstable manifolds in finite time.

Returning to MGRIT, assuming there is a large component of error along the backward unstable manifold, we would expect that error to remain orthogonal to the other manifolds, and if there is a small component of error along the backward stable manifold, it should only contribute a small amount to the error in the less stable manifolds. Thus, if MGRIT is capable of damping errors along the stable manifold but not the unstable, we should be able to observe this by computing the components of error using the matrices $\Psi^-(t_i)^T$ as a time-dependent change of basis. However, in practice, since the coarse grid uses an approximate time-stepping operator, Φ_c, the coarse-grid equation will not have the same Lyapunov spectrum. This mismatch between the Lyapunov vectors on the fine and coarse grid causes some error along the unstable backward manifold to "leak" into the neutral and stable manifolds during the coarse grid correction, thus stalling convergence. Figure 3.1a demonstrates this

phenomenon, where we see that, after some initial convergence, the error in the unstable manifold contributes significantly to the error in the other manifolds as well, preventing the residual from decreasing further. While naive MGRIT is able to achieve a residual norm of 10^{-6} here, the level at which naive MGRIT stalls grows exponentially with the length of the time domain, and thus naive MGRIT will suffer dramatically for longer time domains as seen in the results section.



(a) MGRIT$_2$ (b) ΔMGRIT$_2$

Figure 3.1: The 2-norm of the residual (r) and the components of relative error along the unstable, neutral, and stable manifolds, (e_u, e_n, e_s, respectively) for 30 two-level MGRIT iterations on the Lorenz equation with $T_f = 8T_\lambda$ and coarsening factor $m = 2$. (3.1a) Convergence stalls after 22 iterations, and there is significant error even in the stable and neutral manifolds, which MGRIT should normally be expected to correct. (3.1b) The Δ correction allows MGRIT to correct errors in the stable and neutral manifolds, even in the presence of error in the unstable manifold, resulting in convergence of the residual.

3.2 Deriving Δ correction

The Δ correction remedies the mismatch in the Lyapunov spectrum between the fine and coarse grids by using *linearizations* of the fine operator, computed at the current

solution guess, to correct the coarse operator Φ_c.

Let Φ, Φ_c, and Φ^m be defined as above, and let $\boldsymbol{v} = \boldsymbol{u} - \boldsymbol{e}$ be an approximate solution on the fine grid. Now define

$$\Delta_i := \partial_{\boldsymbol{u}} \Phi^m(\boldsymbol{v}_{i-1}) - \partial_{\boldsymbol{u}} \Phi_c(\boldsymbol{v}_{i-1}), \tag{3.1}$$

where Δ_i is a matrix valued function of $\boldsymbol{v}_i$, which encodes the difference between the linearizations of the ideal and coarse operators. The matrix Δ_i will naturally have the same number of dimensions as the number of spatial dimensions of the system. Contrast this with the $\boldsymbol{\tau}$ correction, which encodes the difference between the *values* of these two operators applied to $\boldsymbol{v}_i$. We then use the computed Δ_i to form a correction to the time stepper on each coarse interval:

$$\Phi_{\Delta_i}(\boldsymbol{v}_{i-1}) := \Phi_c(\boldsymbol{v}_{i-1}) + \Delta_i \boldsymbol{v}_{i-1}, \tag{3.2}$$

which ensures that as $\boldsymbol{v}$ approaches the solution $\boldsymbol{u}$, i.e. near MGRIT convergence, $\partial_{\boldsymbol{u}_i} \Phi_\Delta$ approaches $\partial_{\boldsymbol{u}_i} \Phi^m$. Because $\partial_{\boldsymbol{u}_i} \Phi_\Delta$ is the linear tangent propagator on the coarse grid, which determines the Lyapunov spectrum, the Δ correction is able to correct the mismatch in the Lyapunov spectrum, even when Φ_c is a poor approximation to Φ^m.

Together with the $\boldsymbol{\tau}$ correction, which is computed at the same time, this gives the modified MGRIT algorithm 3.1, where the new additions are highlighted in red. A reduced memory variant of this algorithm is explored in Section 3.4 for problems where direct computation of the Δ_i matrices is intractable.

Note that the first loop does not update the values of $\boldsymbol{v}$ at each time point, and may thus be done in parallel, while the loop on the coarse grid must be solved sequentially. Remember that the multilevel method replaces the sequential solve on the coarse grid with a recursive call to the algorithm. In Figure 3.1b, we see the effect this Δ correction has on the convergence of MGRIT for the Lorenz system.

Algorithm 3.1 $\Delta\text{MGRIT}_2(\boldsymbol{v}_f, \boldsymbol{g}, m)$; MGRIT two-level cycle with Δ correction

$\boldsymbol{v} \leftarrow R\boldsymbol{v}_f$, $\boldsymbol{g}_c \leftarrow R\boldsymbol{g}$

for each C-point, $i = 1, 2, 3, \ldots, N_T$ **do**

$\Delta_i \leftarrow \partial_{\boldsymbol{u}_{i-1}} \Phi^m(\boldsymbol{v}_{i-1}) - \partial_{\boldsymbol{u}_{i-1}} \Phi_c(\boldsymbol{v}_{i-1})$

$\boldsymbol{\tau}_i \leftarrow \Phi^m(\boldsymbol{v}_{i-1}) - \Phi_{\Delta_i}(\boldsymbol{v}_{i-1})$

end for

for $i = 1, 2, 3, \ldots, N_T$ **do**

$\boldsymbol{v}_i \leftarrow \Phi_{\Delta_i}(\boldsymbol{v}_{i-1}) + \boldsymbol{\tau}_i + \boldsymbol{g}_{c,i}$

end for

$\boldsymbol{v}_f \leftarrow R^T\boldsymbol{v}$, followed by F(CF)-relaxation on $\boldsymbol{v}_f$ and Ψ_f

Without the Δ correction, Figure 3.1a shows that there are significant components of error along the neutral and stable manifolds, causing a stall in residual convergence, however, the Δ correction in Figure 3.1b completely eliminates this problem, allowing the iteration to converge in residual. Thus, we see that the Δ correction addresses the main difficulty with applying MGRIT to chaotic systems by correcting the mismatch in the Lyapunov spectrum between the fine and coarse grids and thereby eliminating the stalling problem.

3.3 Modified FAS coarse-grid equation and quadratic convergence

Since the Δ correction updates the coarse operator, this represents a modification to the FAS coarse-grid equation. Recall that in Section 1.2 we saw that the splitting $\boldsymbol{\tau}(\boldsymbol{v}) = A_c(\boldsymbol{v}) - A_*(\boldsymbol{v})$ resulted in the FAS coarse-grid Equation (1.8). Now, define the global linear correction on the coarse grid as $\Delta(\boldsymbol{v}) = \partial_{\boldsymbol{u}} A_*(\boldsymbol{v}) - \partial_{\boldsymbol{u}} A_c(\boldsymbol{v})$, and

the modified coarse-grid equation is

$$[A_c + \Delta(\boldsymbol{v}^k)](\boldsymbol{v}^{k+1}) = \boldsymbol{g}_c + \boldsymbol{\tau}(\boldsymbol{v}^k) + [\Delta(\boldsymbol{v}^k)]\boldsymbol{v}^k. \tag{3.3}$$

Equation (3.3) is equivalent to the Multilevel Nonlinear Method from [59] applied to the time dimension.

For the simplest choice of coarse-grid propagator where $\Phi_c \equiv 0$, we have that $A_c = I$, the identity operator, and (3.3) becomes

$$\begin{aligned}
&[I + \partial_{\boldsymbol{u}_i} A_*(\boldsymbol{v}^k) - I](\boldsymbol{v}^{k+1}) = \boldsymbol{g}_c + \boldsymbol{\tau}(\boldsymbol{v}^k) + [\partial_{\boldsymbol{u}_i} A_*(\boldsymbol{v}^k) - I](\boldsymbol{v}^k) \\
&\quad \implies [\partial_{\boldsymbol{u}_i} A_*(\boldsymbol{v}^k)]\boldsymbol{v}^{k+1} = \boldsymbol{g}_c + \boldsymbol{v}^k - A_*(\boldsymbol{v}^k) + [\partial_{\boldsymbol{u}_i} A_*(\boldsymbol{v}^k)]\boldsymbol{v}^k - \boldsymbol{v}^k \\
&\quad \implies [\partial_{\boldsymbol{u}_i} A_*(\boldsymbol{v}^k)]\boldsymbol{v}^{k+1} = [\partial_{\boldsymbol{u}_i} A_*(\boldsymbol{v}^k)]\boldsymbol{v}^k + \boldsymbol{g} - A_*(\boldsymbol{v}^k) \\
&\qquad\quad \implies \boldsymbol{v}^{k+1} = \boldsymbol{v}^k - [\partial_{\boldsymbol{u}_i} A_*(\boldsymbol{v}^k)]^{-1}(A_*(\boldsymbol{v}^k) - \boldsymbol{g}),
\end{aligned}$$

which is equivalent to an iteration of Newton's method applied to the residual equation $A_*(\boldsymbol{v}^k) - \boldsymbol{f}_c$. We should then expect MGRIT with this Δ correction to converge at least as well as Newton's method, i.e. with local quadratic convergence, as long as A_c approximates A_* better than the identity I. In Section 3.5, we provide numerical evidence for the quadratic convergence of MGRIT with Δ correction.

3.4 Low-rank Δ correction for PDEs

For the Lorenz system with dimension $n_s = 3$, the computation and storage requirements for the Δ correction are small relative to the improvement in convergence they provide. However, the computation and storage of $n_s \times n_s$ matrices will be prohibitive for the case of most PDEs, which can have n_s in the millions or larger. However, it is often the case that only a small, finite number of the Lyapunov exponents are positive, meaning that the unstable manifold of the discretized system has dimension much smaller than n_s. Moreover, there is evidence that the number

of positive Lyapunov exponents can be bounded in many cases. For example in [17], the authors measure the Lyapunov spectrum for a 2D simulation of a chaotic flow around an airfoil at high Reynolds number while refining in space. They observed that although the Lyapunov spectrum of the system was mesh dependent, beyond a certain resolution threshold, the total number of unstable modes was no more than 5, even as the number of degrees of freedom was increased to $\mathcal{O}(10^6)$. My own numerical computation of the Lyapunov spectrum for the Kuramoto-Sivashinsky equation on the spatial domain considered below reveals the unstable manifold to be roughly nine or ten dimensional.

Since the tangent dynamics in the stable manifold are equivalent to a parabolic system, we should expect MGRIT to converge well for these modes of error [52], even without Δ correction. In contrast, we have already seen that the unstable modes must be represented very accurately on the coarse grid, or else exponentially growing error will be mapped incorrectly to the stable manifold. This motivates a low-rank approximation to the Δ correction which targets only the low-dimensional unstable manifold. Given an approximate solution $\{\boldsymbol{u}_i\}_{i=0}^{n_T}$, let $\{\Psi_i\}_{i=0}^{n_T}$ be such that each Ψ_i is a rectangular, orthonormal matrix with columns equal to the first k backward Lyapunov vectors $\psi_i^{-,j}$ for $j = 0, 1, \ldots k-1$, then

$$\hat{\Delta}_i = \Delta_i \Psi_{i-1} \Psi_{i-1}^T \tag{3.4}$$

is a rank k approximation to Δ_i, which is exact for the first k backward Lyapunov vectors. Next, note that since Δ_i is a linearization of the function $\Phi^m - \Phi_c$, the columns of $\Delta_i \Psi_{i-1} = (\partial_{\boldsymbol{u}_i} \Phi^m - \partial_{\boldsymbol{u}_i} \Phi_c) \cdot \Psi_{i-1} = D_{\Psi_{i-1}} \Phi^m - D_{\Psi_{i-1}} \Phi_c$ are just directional derivatives along the first k Lyapunov vectors. Thus, we only need to compute these k directional derivatives and Ψ_{i-1} to form $\hat{\Delta}_i$, never forming the full matrix Δ_i. This requires storage of the factors $\Delta_i \Psi_{i-1}$ and Ψ_{i-1}, which are $n_s \times k$ matrices, meaning that as long as $k < n_s/2$, the low-rank approximation requires less storage and computational work than the full Δ correction.

However, implementation of this approach requires overcoming the cost of computing the first k Lyapunov vectors, which are considered expensive to obtain and are classically computed sequentially in time. The expense is largely due to the time propagation of Lyapunov vector matrices and linearizations of the time-stepping operator, as well as the QR factorizations needed for orthogonalization. Here, we outline an efficient way to estimate the Lyapunov vectors simultaneously while solving the original equation parallel-in-time. As we saw in Equation (1.12), given a trajectory $\{\boldsymbol{u}_i\}_{i=0}^{\infty}$ and an arbitrary initial orthogonal $n_s \times k$ matrix Ψ_0, $k \leq n_s$, the QR iteration

$$\Psi_{i+1} R_i = [\partial_{\boldsymbol{u}_i} \Phi(\boldsymbol{u}_i)] \Psi_i \quad i = 0, 1, \ldots \tag{3.5}$$

will result in convergence of the columns of Ψ_i to the first k backward Lyapunov vectors of the trajectory as $i \to \infty$. The QR decomposition orthonormalizes the columns of Ψ_i, preventing numerical instability, however in practice, Ψ_i may be normalized every m steps while still converging to the Lyapunov vectors.

Given a finite trajectory, this QR iteration will yield an estimate to the true Lyapunov vectors which is more accurate toward the end of the time domain. However, the sequential computation of these Lyapunov vector estimates can be more expensive than solving the state equation sequentially. Thus, [53] proposes to compute these estimates parallel in time using MGRIT. Since the Lyapunov vector computation takes the form of an initial value problem (IVP), we may apply MGRIT directly to (3.5), using $\partial_{\boldsymbol{u}_i} \Phi_c$ as the coarse-grid time-stepping operator. In this way, the MGRIT cycle will be simultaneously solving for the state vector $\boldsymbol{u}_f$ and the Lyapunov vectors Ψ. It is important to note that while the IVP for chaotic systems is very sensitive to initial conditions, this is *not* the case for the Lyapunov vectors. The first column of Ψ_i will almost surely converge to the first backward Lyapunov vector $\psi^{-,1}(t_i)$ as $i \to \infty$, independent of the initial matrix Ψ_0. This closely resembles the convergence of the linear power iteration to the first eigenvector of a

matrix. Therefore, although the state equation is chaotic, and thus ill-conditioned, the IVP for the backward Lyapunov vectors appears to be well-conditioned in the sense that the Lyapunov vectors at later times do not exhibit sensitive dependence on the Lyapunov vectors at earlier times, and thus we expect fast MGRIT convergence for the Lyapunov vectors without modification of the algorithm, and perhaps counter-intuitively, the most unstable modes will also be the most accurate, as they are the fastest to converge in the QR iteration algorithm.

In order for MGRIT to solve for the Lyapunov vectors Ψ_i, we also need to consider the MGRIT FAS coarse grid for the problem of finding Lyapunov vectors. Let $F^m = \partial_{\boldsymbol{u}_i}\Phi^m(\boldsymbol{u}_i)$ and $F_c = \partial_{\boldsymbol{u}_i}\Phi_c(\boldsymbol{u}_i)$, then an appropriate τ correction term on the coarse grid for the Lyapunov vectors is given by

$$\boldsymbol{\tau}_{i+1} = (F^m - F_c)\Psi_i.$$

However, if we are using the low-rank Δ correction, then this becomes

$$\begin{aligned}\boldsymbol{\tau}_{i+1} &= (F^m - (F_c + \Delta\Psi_i\Psi_i^T))\Psi_i \\ &= (F^m\Psi_i - F_c\Psi_i - \Delta\Psi_i) \\ &= (\Delta - \Delta)\Psi_i \\ &= 0.\end{aligned}$$

Therefore, as long as we are also using the low-rank Δ correction, the τ correction for the Lyapunov vectors vanishes. From another perspective, the Δ correction already acts as a τ correction for the Lyapunov vectors.

Regarding computational cost, if Ψ_i is only orthonormalized with the Gram-Schmidt method at C-points and $\Psi_{i+1} = [\partial_{\boldsymbol{u}_i}\Phi(\boldsymbol{u})]\Psi_i$ is used at F-points, then there is potential to save a great deal of work without much loss in accuracy of the computed Lyapunov vectors, since we anyway need the un-normalized product $(\partial_{\boldsymbol{u}_{i+m}}\Phi(\boldsymbol{u}_{i+m})\dots\partial_{\boldsymbol{u}_i}\Phi(\boldsymbol{u}_i))\Psi_i$ to form the Δ correction for the coarse grid.

We now have a low-rank Δ correction for the state equation, and we have an efficient algorithm for finding the needed Lyapunov vector estimates given a trajectory. Combining the two, we get the low-rank Δ correction Algorithm 3.2, where changes relative to original MGRIT (Algorithm 1.1) are again highlighted in red. In this algorithm, F-relaxation on the state vector is always followed by F-relaxation on the Lyapunov vectors, and likewise for C-relaxation. The Lyapunov vectors are also solved for sequentially on the coarsest grid alongside the state vector. The modified Gram-Schmidt algorithm is used to compute the QR factorization.

Algorithm 3.2 $\Delta_k\text{MGRIT}_2(\boldsymbol{v}_f, \Psi_f \boldsymbol{g}, m)$; MGRIT two-grid cycle with low-rank Δ correction and Lyapunov vector estimates

---

$\boldsymbol{v} \leftarrow R\boldsymbol{v}_f$, $\Psi \leftarrow R\Psi_f$, $\boldsymbol{g}_c \leftarrow R\boldsymbol{g}$

for each C-point, $i = 1, 2, 3, \ldots, N_T$ **do**

$\hat{\Delta}_i \leftarrow [\partial_{\boldsymbol{u}_{i-m}} \Phi^m(\boldsymbol{v}_{i-m}) - \partial_{\boldsymbol{u}_{i-1}} \Phi_c(\boldsymbol{v}_{i-1})]\Psi_{i-1}$

$\boldsymbol{\tau}_i \leftarrow \Phi^m(\boldsymbol{v}_{i-1}) - \Phi_{\hat{\Delta}_i}(\boldsymbol{v}_{i-1})$

end for

for $i = 1, 2, 3, \ldots, N_T$ **do**

$\boldsymbol{v}_i \leftarrow \Phi_{\hat{\Delta}_i}(\boldsymbol{v}_{i-1}) + \boldsymbol{\tau}_i + \boldsymbol{g}_i$

$\Psi_i \leftarrow \text{GramSchmidt}([\partial_{\boldsymbol{u}_{i-1}} \Phi_{\hat{\Delta}_i}(\boldsymbol{v}_{i-1})]\Psi_{i-1})$

end for

$\boldsymbol{v}_f \leftarrow R^T\boldsymbol{v}$, $\Psi_f \leftarrow R^T\Psi$, followed by F(CF)-relaxation

---

The parallel performance of MGRIT is highly dependent on the cost of solving the coarsest grid, and the sequential propagation of the Lyapunov vectors on the coarse grid adds significantly to this cost. However, experimentation has shown that this sequential solve can either be skipped or approximated with parallel FCF-relaxation, without much change to the effectiveness of the algorithm in many cases. In fact, for many of the experiments presented in Section 3.5.2, no coarse-grid propagation of Lyapunov vectors is performed, since it was determined experimentally that FCF-

relaxation on finer grids resolves the Lyapunov vector estimates sufficiently well for MGRIT convergence. In this case, the only extra work is performed in parallel on the fine grid (i.e., the Gram-Schmidt steps on the coarse grid are skipped). However, for long time domain sizes, the propagation of Lyapunov vectors on the coarsest grid is necessary in order to prevent the residual stalling, as discussed in the results section.

3.5 Numerical results

I implemented the low-rank Δ correction by modifying the XBraid software package [57], which is written in C and MPI. This was not a trivial task due to the complexity of communicating the extra information required to compute the Lyapunov vectors alongside the solution vector, as well as the additional API functions required to implement the algorithm in a general way. In order to use the low-rank Δ correction, the user only needs to define functions for computing a vector inner product with two state vectors $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$ and for computing a Jacobian-vector product $[\partial_{\boldsymbol{u}_i} \Phi]\boldsymbol{v}$. Then, a simple interface function allows the user to set the rank of Δ correction desired, and XBraid will estimate that many Lyapunov vectors and use them to form low-rank Δ corrections for the coarse grid.

In all experiments in this section, the stopping criterion used is an absolute tolerance on the global 2-norm of the residual, which is specified for each experiment.

3.5.1 Convergence for the Lorenz system

In the following experiments, we solve the discretized Lorenz system (1.9) using forward Euler on the fine grid. A coarsening factor of $m = 2$ and an absolute residual halting tolerance of 10^{-10} is used across all the studies. When the θESDIRK2 method is used on a coarse grid, the values of θ are dependent on the grid level ℓ and are

computed according to (2.5c) with $m = 2^\ell$, where ℓ refers to the grid level, starting at $\ell = 0$ for the fine grid. The implicit equation (2.2) is solved numerically using Newton's method. When the θ method is not used on the coarse grid, forward Euler is used, with coarsened time-step size $m^\ell h$. Only F-relaxation is considered in this section. First, we study convergence rates for various two-level algorithms on a small problem. Then we perform a refinement study in time and a time domain size scaling study. Finally, we explore the effect of adding more coarse levels for different problem sizes.

Two-level results

Figure 3.2 plots the convergence history of the modified two-level MGRIT algorithms, solving the Lorenz system with $T_f = 8T_\lambda$. This experiment demonstrates that MGRIT_2, even when using the θ method on coarse grids, stalls for long enough time domains, which is expected given the significant mismatch between the fine and coarse grid Lyapunov spectra. However, the Δ correction allows the method to converge quickly, with the combination of Δ correction and the θ method leading to the fastest convergence. My intuition is that the increased accuracy of the θ method in approximating the fine grid widens the basin of attraction for the quadratic convergence region of Δ-corrected MGRIT.

In order to study the effect of varying time-step size h on MGRIT_2 performance, Table 3.1 presents iteration counts required to reach a residual tolerance of 10^{-10} for the Lorenz system with fixed T_f and increasing numbers of time-points n_t. In all cases fewer iterations are needed for smaller h. However, compared to naive MGRIT_2, both the Δ correction and the θ method require roughly half as many iterations to converge, and when used together, they require a quarter as many. Further, we see that the θ method can correct for instabilities on the coarse grid which cause MGRIT to diverge (see first column).

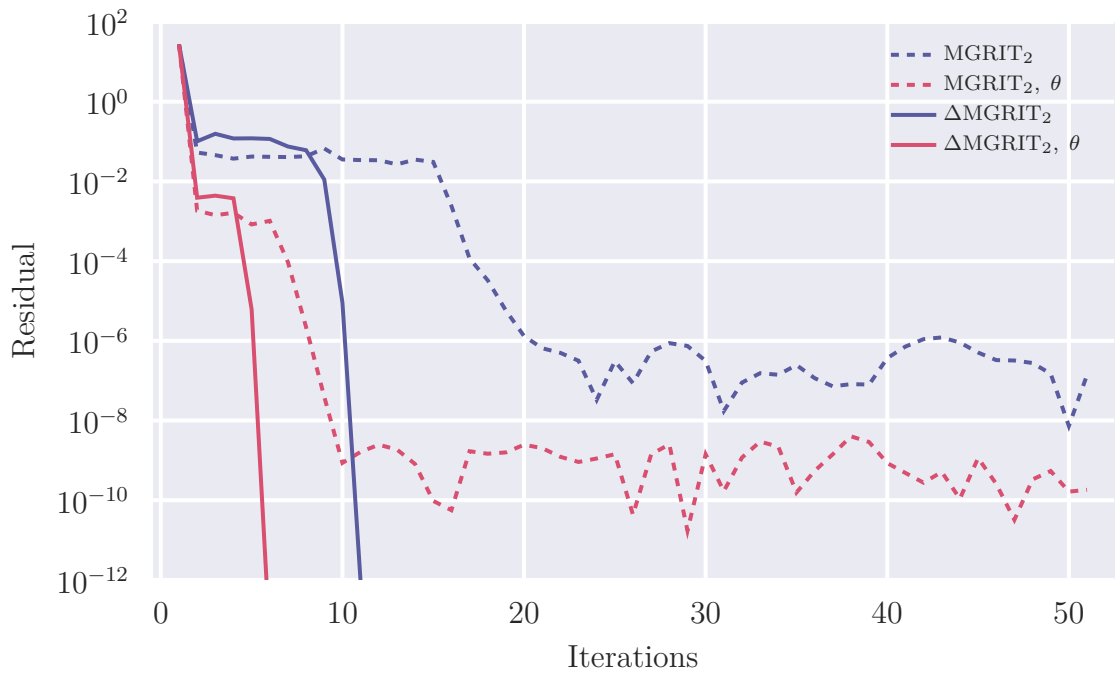


Figure 3.2: Two-level MGRIT residual convergence for the Lorenz system, comparing classical MGRIT with Δ correction, with and without the θ method coarse grid. The problem is posed on a time domain of $T_f = 8T_\lambda$ with 8192 time points. We see that standard MGRIT$_2$ stalls as expected, but that the Δ correction provides fast convergence. Using Δ correction and the θ method together leads to the fastest convergence.

Table 3.1: Two-level MGRIT iteration counts for the Lorenz system with fixed $T_f = 8T_\lambda$ and increasing number of time points, n_t. An entry of '*' indicates that the algorithm diverged for the given grid size.

	T_f, n_t				
Algorithm	4, 512	4, 1024	4, 2048	4, 4096	4, 8192
MGRIT$_2$	*	44	22	15	12
MGRIT$_2$, θ	19	13	9	7	6
ΔMGRIT$_2$,	*	11	8	6	6
ΔMGRIT$_2$, θ	8	6	5	4	4

Table 3.2: Two-level MGRIT iteration counts for the Lorenz system with fixed time-step size and increasing T_f (in units of Lyapunov time). An entry of '-' indicates that the algorithm did not converge within 100 iterations

	T_f, n_t					
Algorithm	2, 4096	4, 8192	6, 12288	8, 16384	10, 20480	12, 24576
MGRIT_2	10	13	17	64	-	-
MGRIT_2, θ	4	5	6	7	41	-
ΔMGRIT_2	5	6	7	8	9	94
ΔMGRIT_2, θ	3	4	4	5	5	48

Table 3.2 shows iteration counts for the two-level algorithm on the Lorenz system with increasing time domain size T_f and fixed time-step size h. For naive MGRIT_2, iteration counts increase linearly up until the critical time $T_f = 6T_\lambda$, after which naive MGRIT_2 stalls. In contrast, the Δ correction and θ method greatly improve convergence for all time domain sizes. Notably, the iteration counts for the Δ corrected algorithm are nearly flat (until the last column), even for long time domain sizes, with the combined Δ method and θ method providing the fastest convergence.

Multilevel results

While MGRIT_2 is not typically used in practice, it is used as a stepping stone toward understanding the multilevel algorithm. Recall that MGRIT_2 solves the coarse grid equation (1.8) using a sequential solve and that MGRIT_3 replaces this direct sequential solve with a recursive application of MGRIT_2 to inexactly solve the initial coarse grid. Thus, we expect that MGRIT_3 will converge no faster than MGRIT_2, and as we add more levels, this trend should continue. However, since the coarsest grid is solved sequentially, it is very important for parallel performance that the coarse grid be as small as possible, since the proportion of the algorithm which is not parallelizable limits the scaling of the method.

Table 3.3 depicts the effect of increasing the number of coarse grids on MGRIT convergence for various algorithmic configurations. For naive MGRIT (MGRIT$_k$), the effects of coarsening beyond two levels quickly leads to divergence and an unusable method. For the θ method (MGRIT$_k$, θ), convergence improves. However for the 7 level solver (where the coarsest grid size is small enough to be practical), the method diverges or takes too many iterations (43) to be practical. For Δ correction (MGRIT$_k$, Δ), convergence with 3 levels is impressively stable across all T_f, but for 5 and 7 levels divergence is observed. This is primarily due to the fact that the time-stepping scheme becomes unstable on such coarse grids. Finally, the combination of both approaches (MGRIT$_k$, Δ, θ) combines the robust convergence for long T_f with the improved coarse-grid stability and convergence of the θ method. The result is promising convergence at 7 levels for $T_f = 2$ and $T_f = 4$, since the coarsest grid here is small (64 and 128 time-points respectively) and the iteration counts are similar to previous cases demonstrating parallel speedup for linear parabolic problems [15]. Unfortunately, the small spatial size of this problem (3) makes a parallel performance study difficult, as computations would always be dominated by network latency. Thus, we next consider a larger problem.

3.5.2 Parallel scaling for the Kuramoto-Sivashinsky equation

In one spatial dimension, the Kuramoto-Sivashinsky (KS) equation is given by

$$\boldsymbol{u}_t = -\boldsymbol{u}_{xx} - \boldsymbol{u}_{xxxx} - \boldsymbol{u}\boldsymbol{u}_x, \tag{3.6}$$

and is posed with periodic boundary condition $\boldsymbol{u}(0,t) = \boldsymbol{u}(L,t)$, for some length L. This equation is widely studied as an archetypal example of a chaotic PDE and is considered one of the simplest such PDEs. It is also a useful surrogate for many fluid-dynamics applications, since it exhibits a wide range of complex dynamics including spatio-temporal chaos [32]. The KS equation combines the linear anti-diffusion and

Table 3.3: Multilevel MGRIT iteration counts for the Lorenz system with fixed time-step size and increasing T_f (in Lyapunov time), using varying numbers of grids. An entry of '-' indicates that the algorithm did not converge within 100 iterations, and an entry of '*' indicates that the algorithm diverged.

	T_f, n_t			
Algorithm	2, 4096	4, 8192	6, 12288	8, 16384
MGRIT_2	10	13	17	64
MGRIT_3	13	18	-	-
MGRIT_5	26	-	-	-
MGRIT_7	*	*	*	*
MGRIT_2, θ	4	5	6	7
MGRIT_3, θ	6	7	9	11
MGRIT_5, θ	10	13	19	63
MGRIT_7, θ	43	-	-	-
ΔMGRIT_2	5	6	7	8
ΔMGRIT_3	6	8	11	13
ΔMGRIT_5	*	*	*	*
ΔMGRIT_7	*	*	*	*
ΔMGRIT_2, θ	3	4	4	5
ΔMGRIT_3, θ	3	4	5	5
ΔMGRIT_5, θ	5	6	7	9
ΔMGRIT_7, θ	9	15	20	23

hyper-diffusion terms with a nonlinear advection term. Because of the structure of the linear terms, high-frequency modes are stiffly damped by the hyperdiffusive part, while low-frequency modes are *excited* by the anti-diffusive part. Although the linear part of this equation is apparently unstable, the nonlinear term stabilizes the equation as the advection causes an energy cascade from low- to high-frequency components. A typical trajectory for the KS equation over a chaotic timescale is shown in Figure 3.3. The maximal Lyapunov exponent for the KS equation depends on the length-scale L, which we fix at $L = 64$ for our experiments. With this parameter, we observe a maximal Lyapunov exponent of approximately 0.1, so one Lyapunov time T_λ for the KS equation with these parameters is around 23 real time

units. Being a stiff, nonlinear, and chaotic PDE, the KS equation is a challenging problem to solve with PinT methods.

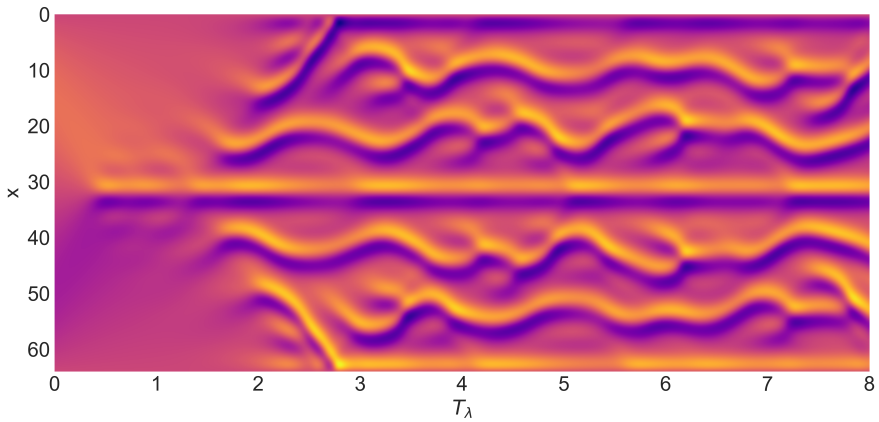


Figure 3.3: A numerical trajectory of the KS equation for 8 Lyapunov time with the initial condition $\boldsymbol{u}(x, 0) = \sin(2\pi x/L)$.

In the following experiments the KS equation is discretized using 4th order finite differencing in space and the fully implicit, two stage, 2nd order Lobatto IIIC method (FIRK2) in time, which is stiffly accurate and a fairly simple fully implicit Runge-Kutta method to implement. The nonlinear implicit equation is solved using Newton's method, while the linear part of the Newton iteration is solved using UMFPACK from SuiteSparse [11]. When using MGRIT with naive time coarsening and re-discretization, we use the same FIRK2 method with a larger time-step size.

For the θ method, since the two-stage Lobatto methods are actually a three parameter family of methods, also including Lobatto IIIA (Crank-Nicolson method), Lobatto IIIB, and Lobatto IIIC* (explicit trapezoid method) which all use function evaluations calculated at the end points of the time step, we can combine of all four of these methods into a θ method akin to the θESDIRK2 method from Chapter 2 which is a linear combination of the four methods. This is the θFIRK3 method given in (2.12).

Solving algebraic equations analogous to those in Section 2.1 allows us to find values of the three parameters, θ_A, θ_B, and θ_C, which approximate the m-step stability function for FIRK2 up to fourth order in mh. One degree of freedom is lost since we find the additional constraint that $\theta_A = \theta_B$. The resulting Runge-Kutta method is A-stable, and not stiffly accurate, and is thus a poor solver for the KS equation on coarse time grids which was verified experimentally. So, we instead may use one of the available degrees of freedom to enforce a stiff constraint on the stability function,

$$\lim_{z \to -\infty} \phi_\theta(z) = 0,$$

yielding a θ method of up to third order in mh which is also stiffly accurate. In all cases where the θ method is used for the KS equation, this is the method used. Note that this method does not satisfy the general order conditions from Section 2.2, instead, it uses the simpler approach of matching the Taylor series of the stability functions of Φ^m and Φ_θ as in Section 2.1. In the following experiments, we will consider the initial condition $\boldsymbol{u}(x, 0) = \sin(2\pi x/L)$, which is chosen since it is a smooth function which satisfies the periodic boundary condition and is easily generalized to any spatial grid resolution.

Multilevel results

FCF-relaxation is used in the following scaling studies, since it improves convergence for MGRIT by relaxing parabolic modes of error (in the stable manifold) more effectively than F-relaxation [52] and generally leads to faster time to solution in our experiments. Unless stated otherwise, a coarsening factor of $m = 4$ is used. However, in certain cases, I found that parallel efficiency could be improved by coarsening by a bigger factor (here $m = 16$) between the fine grid and the first coarse grid, while a coarsening factor of 4 was used for coarser grids, with little to no degradation in convergence, similar to the approach taken in [15]. Further, as demonstrated in Section 3.5.1, two-level convergence can be fast enough that the Δ correction is not

needed, however, the Δ correction greatly improves the convergence for the multilevel method. Thus, we may defer the Δ correction to a coarser grid, while the first coarse grid uses standard FAS MGRIT. This works well because the Δ correction is able to improve the accuracy of the coarsest grids where it is needed most, while significant work is saved by not computing Δ corrections on the finer grids where they are not as useful. In this context, the Δ correction serves as a correction to help the multilevel method approximate the convergence of the 2-level method.

In the following experiments, the Δ correction has been deferred to the second coarse grid whenever more than two levels are used. An absolute residual stopping tolerance of 10^{-8} is used to terminate the MGRIT iterations. Unless stated otherwise, the Lyapunov vector estimates are not propagated on the coarsest grid, as they are sufficiently resolved by FCF-relaxation on the finer grids in most cases, as discussed in Section 3.4. MGRIT is an iterative method, and thus requires an initial guess for the solution across the whole time domain. For the following experiments, the initial guess is produced by solving the coarsest grid with sequential time stepping, then interpolating the solution to the fine grid. Therefore, the quality of the initial guess is dependent on the accuracy of the coarsest grid.

First, we explore weak scaling results for the KS equation. Figure 3.4 plots the wall time to solution for increasing problem sizes, comparing sequential time stepping against naive MGRIT, the θ method, and the θ method with rank-9 Δ correction, which was found a posteriori to be the optimal choice. The KS equation is solved for 4 Lyapunov time (92.1 time-units), and the number of points in space are doubled as the points in time are quadrupled to maintain a fixed h/h_x^2 ratio where h_x is the mesh size in space. By fixing h/h_x^2, we maintain a reasonable balance between the error in space and time, since the spatial and temporal discretizations are fourth and second order, respectively. The problem size per processor remains fixed at 256×16 points in space and time. The coarsest grid size is chosen individually for each algorithm

to be as small as each algorithm stably supports for all problem sizes, i.e. remaining fixed at 128 time-points for naive MGRIT and MGRIT with the θ method, and 32 time-points for the θ method with rank-9 Δ correction. The ideal scaling plot is increasing because the spatial solver used in Newton's method is UMFPACK, which has no parallelism in the spatial dimension and scales here for this 1D problem like $\mathcal{O}(n_s)$. By choosing a suitable and stable coarsest grid size for naive MGRIT, naive MGRIT is able to scale well for a wide variety of problem sizes, albeit with a higher iteration count than for the other solvers. This is reflected in the longer run-times for naive MGRIT. We believe that all algorithms benefit from the fact that the size of the unstable manifold is a property of the continuous equation, rather than being mesh dependent. The θ method with rank-9 Δ correction achieves a max speedup of 21.5$\times$ over sequential time stepping for the largest problem size, (512×32768).

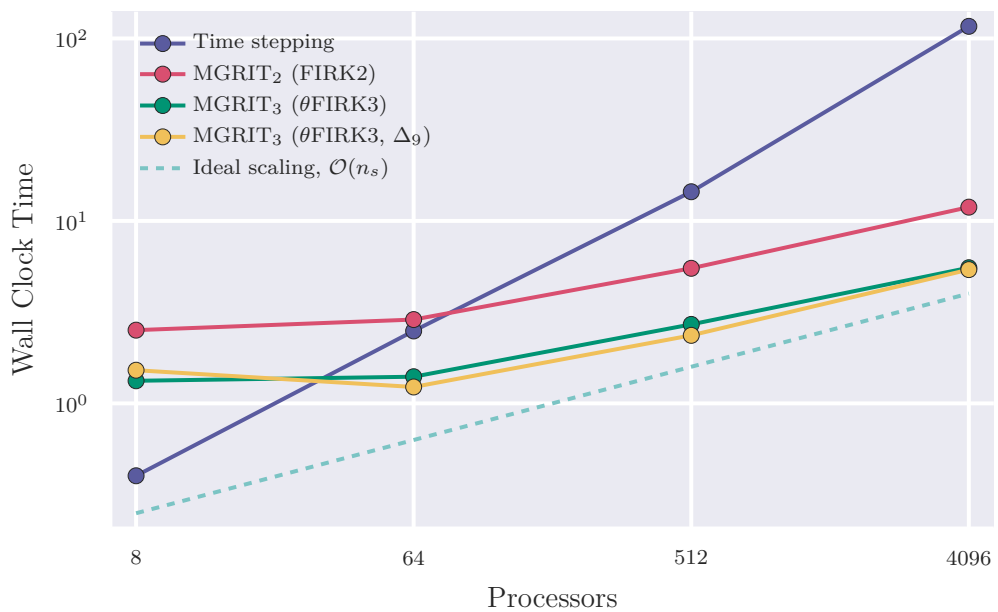


Figure 3.4: Weak scaling study for the KS equation with refinement in space and time, comparing naive rediscretization with θ methods and a rank-9 Δ correction. Here we see that the low-rank Δ-corrected MGRIT with a fixed rank weak scales just as well as classical MGRIT, despite the increase in n_s.

Next we explore strong scaling for the KS equation. Figure 3.5 plots the wall-clock time to solution for naive MGRIT, the θ method, and θ method with a rank-9 Δ correction, as compared to sequential time stepping. The KS equation is solved for a time domain of 4 Lyapunov time (92.1 time-units) with 256 points in space and 8192 points in time. This corresponds with the second-largest problem size from Figure 3.4. Naive MGRIT is limited to a three-level method, since the method becomes unstable for four levels, and thus the sequential cost of solving the coarsest grid leads to quick run-time stagnation. The θ method, however, provides a stabilized coarse grid, thus enabling a 4-level method, which scales better than naive MGRIT. The low-rank Δ correction reduces the number of iterations required for convergence with only a rank-9 correction, thus making the method slightly faster, and we observe a maximum speedup of 7.8×. However, increasing the rank of the Δ correction beyond 9 does not further improve convergence, and only increases the cost of each iteration. Thus, a Δ correction rank of 9 appears to be optimal for this problem, which is expected, as experiments measuring the full Lyapunov spectrum indicate that the unstable manifold is roughly 9-dimensional. The dependence of the maximum speedup on the rank of Δ correction is illustrated in Figure 3.6.

Last, we present a strong scaling study for an even longer time domain. Figure 3.7 demonstrates strong scaling for the KS equation over 8 Lyapunov time (184.2 time-units). As before, the KS equation is solved with 256 points in space, but the time-grid is increased to 16384 points in time, to match the fine-grid time-step size from the previous problem. On this timescale, naive MGRIT is unable to converge even with only two levels, and thus timing data is not available. With the θ method, MGRIT converges with two levels but begins to stall when a third level is added. However, a rank-9 Δ correction is able to remedy this, allowing rapid convergence for both the three and four level methods. For this experiment, accuracy for the Lyapunov vector estimates requires that the propagation of the Lyapunov vectors be turned on for the coarsest grid (i.e., the Gram-Schmidt step in Algorithm 3.2),

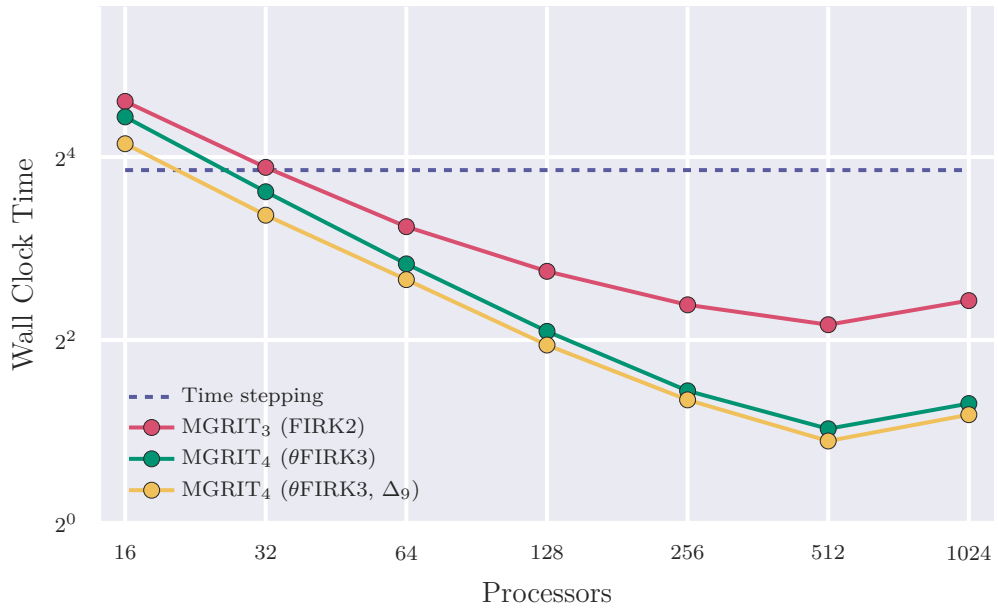


Figure 3.5: Results of a strong scaling study for the KS equation with $T_f = 4T_\lambda$ and $(n_x, n_t) = (256, 8192)$, demonstrating a maximum speedup of 7.8× over sequential time stepping.

otherwise convergence stalls. Here, Δ correction allows for a dramatic improvement to the scaling of the method, and we observe a maximum speedup of 6.1×. Using F-cycles, which can improve convergence at the cost of some parallel efficiency [15], the four-level method can be made even faster, achieving a maximum speedup of 9.6× for this problem.

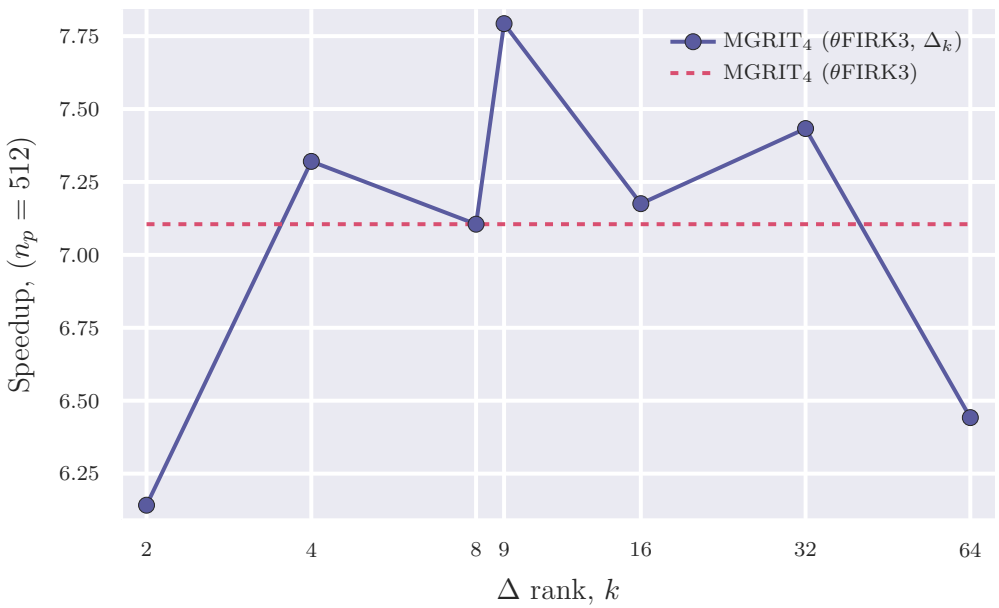


Figure 3.6: Speedup over sequential time stepping with 512 cores for the KS equation with varying ranks of low-rank Δ correction and the θ method coarse grid (solid line), as compared to MGRIT with the θ method coarse grid but without Δ correction (dashed line). The best speedup is achieved when the rank of $\hat{\Delta}$ is roughly equal to the dimension of the unstable manifold (measured experimentally to be 9 dimensional).

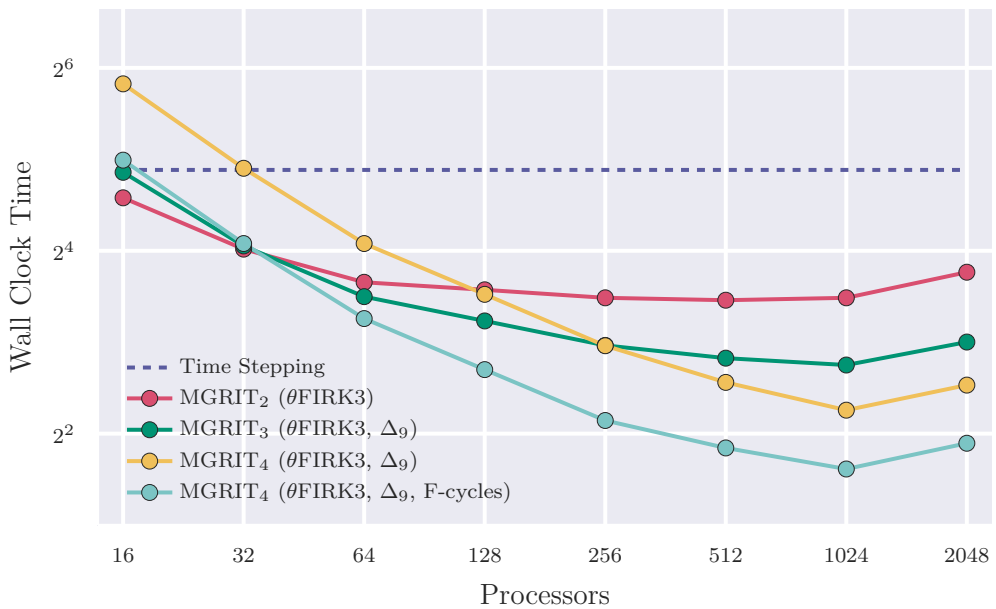


Figure 3.7: Results of a strong scaling study for the KS equation with $T_f = 8T_\lambda$ and $(n_x, n_t) = (256, 16384)$, demonstrating a maximum speedup of 9.6× when using a four-level F-cycle with the θ method and rank-9 Δ correction.

Chapter 4

Local Shadowing Relaxation: Toward optimal-scaling PinT for chaos

So far, we have seen that θ methods and Δ correction can significantly improve MGRIT convergence for chaotic systems. This has even permitted a significant speedup over time stepping for the Kuramoto-Sivashinsky (KS) equation by increasing the length of the time domain over which MGRIT can converge, thereby increasing the number of time-steps available to be simultaneously parallelized [53]. This is especially significant because it is to my knowledge the first such speedup for a chaotic system. Despite this success, the fundamental problem of PinT for chaotic systems has thus far not been addressed: the quality of the coarse-grid correction at later times depends exponentially on the residual at earlier times, and thus convergence depends at least linearly on the length of the physical time domain. In this chapter, I present a novel PinT relaxation scheme for chaotic systems which is stable and convergent even on very long time domains, with some evidence that this approach may be robust on arbitrarily long time domains.

MGRIT with Δ correction and θ methods seems to be robust on timescales smaller than roughly 8 Lyapunov time (T_λ) when using double precision arithmetic, with convergence breaking down on longer time domains. However, remember that this is a relative timescale.

For many "weakly" chaotic systems, i.e. systems for which T_λ is long relative to all other physical time-scales of interest, $8T_\lambda$ is likely long enough to enable a PinT speedup. Many systems of importance to climate and weather simulation as well as viscous fluid dynamics problems can be considered such weakly chaotic systems. For example, Lyapunov analysis of a simulation of the flow around an airfoil in 2D found a maximal Lyapunov exponent (LE) of 0.04 [17], which corresponds to a $T_\lambda \approx 60$ normalized to the characteristic timescale of the bulk flow. A common rule of thumb in atmospheric modeling, first posited by the Global Atmospheric Research Program, says that the errors accumulating in a simulation of atmospheric weather will double roughly every 5 days [30], which translates to a $T_\lambda \approx 17$ days. For this class of problems, it may already be possible to achieve a PinT speedup with MGRIT, using the tools already presented in this work, although proof of concept for this is still needed since many of these systems are advection dominated– a challenge independent of their chaotic nature.

In contrast, turbulent systems have fast chaotic timescales. For a turbulent simulation of the Navier-Stokes equations at high Reynolds number (Re), $8T_\lambda$ may only consist of a few time steps on the fine grid. Worse, T_λ could be sub-grid-scale if the fastest chaotic features aren't even resolved on the fine grid, as is often the case for turbulent simulations where direct numerical simulation (DNS) is not possible. There is strong evidence that the maximal LE, λ_1 of a turbulent flow, is roughly inversely proportional to the Kolmogorov timescale, meaning that

$$\lambda_1 \sim \frac{1}{\tau_\nu} = \mathrm{Re}^{1/2}, \tag{4.1}$$

and therefore T_λ may be *much* smaller than the reference timescale of the bulk flow

[46, 18].

Recall that PinT methods rely on the existence of a problem size crossover point, beyond which there are enough time steps to parallelize over such that a given PinT method is faster than sequential time stepping. Combined with the knowledge that current methods will eventually break down after the timescale increases beyond some multiple of $1/\lambda_1$, it becomes obvious that it doesn't matter how much existing PinT algorithms are improved to increase the timescale length on which convergence is robust, since there will always be a Reynolds number large enough so that there is no such crossover point, and sequential in time will always be faster for any time-domain length.

Of course, if one assumes that the fine grid is sufficiently fine to resolve the Kolmogorov timescale, that is $h = \mathcal{O}(\tau_\nu)$, then this problem seems a little less dire, since $\lambda_1\tau_\nu$ is roughly constant, and in fact the quantity $\lambda_1\tau_\nu \leq 1/\sqrt{3}$ even as $\mathrm{Re} \to \infty$ [18]. Accordingly, if the Kolmogorov timescale is always resolved on the fine grid, then the number of time steps in a given interval of length T_λ will be roughly constant as well. Assume that $h = 0.1\tau_\nu$, then using the value of $\lambda_1\tau_\nu \approx 0.14$ given by [18], we can estimate the number of time steps needed per T_λ as

$$\frac{T_\lambda}{h} = \frac{\ln(10)}{\lambda_1 h} = \frac{10\ln(10)}{\lambda_1\tau_\nu} \approx 165.$$

Therefore, assuming our current limit of $8T_\lambda$, this means there are around 1,320 time steps available to parallelize in a DNS of turbulence at high Re. For comparison, we only saw a maximum speedup of 10× for the KS equation at around 16,000 time steps in Chapter 3.

Even if it is eventually possible to achieve some PinT speedup for a turbulent system at high Re, perhaps by further increasing the time domain limit for MGRIT, or by reducing the overhead so that large speedups are possible on $\mathcal{O}(1,000)$ time steps, the only way to solve the problem on time domains longer than the given

limit is by so-called *windowing*. That is, dividing the whole time domain into time windows or sub-intervals and then solving each window in sequence with a PinT algorithm. Obviously, the overall speedup achieved with windowing will be equal to the speedup on a single window, so this approach cannot be said to scale optimally with the size of the problem, as the time complexity is still $\mathcal{O}(n_t)$. Put plainly, a PinT speedup for a turbulent system at a given Re means the same speedup at all other Re, and while this would be a significant achievement, it falls quite short of the true promise of optimally scaling multigrid in time, that is, the ability to perform DNS of the Navier-Stokes equations at arbitrarily high Re in nearly constant time, assuming computational resources are scaled commensurately.

Optimal scaling for a turbulent system demands an algorithm with timescale independent convergence, that is, robust convergence on arbitrarily long time domains. This is certainly a tall order due to the exponential dependence of the residual at later times on the residual at earlier times, but this is only true as long as the problem is posed as an initial value problem (IVP). We will now see how a reformulation of the problem using Least Squares Shadowing (LSS) completely sidesteps the ill-conditioning problem by relaxing the initial condition, in theory allowing for scalable PinT simulation of turbulence [55].

4.1 Shadowing

4.1.1 The shadowing lemma

As discussed in Chapter 1, chaotic systems are in a sense structurally stable, meaning that a small perturbation to the system results in small backward error. This stability is formalized by the *shadowing lemma*, Lemma 2, for which we need to establish a

few definitions. First, $\boldsymbol{u}^\varepsilon$ is called an *ε-pseudo-orbit* of the propagator $\Phi(\boldsymbol{u}, h)$ if

$$\|\Phi(\boldsymbol{u}_{i-1}^\varepsilon, h) - \boldsymbol{u}_i^\varepsilon\| < \varepsilon \;\; \forall i = 1, 2, \ldots, n_t,$$

where we make explicit that the second argument of Φ is the time-step size, h. Next, we say that the orbit $\boldsymbol{u}^s$ δ-shadows $\boldsymbol{u}^\varepsilon$ if there is a sequence of step sizes $\{\eta_i\}$ such that for all $i = 1, 2, \ldots, n_t$,

$$\|\boldsymbol{u}_i^s - \boldsymbol{u}_i^\varepsilon\| < \delta,$$

$$\boldsymbol{u}_i^s = \Phi(\boldsymbol{u}_{i-1}^s, \eta_i),$$

and

$$\left\| \frac{\eta_i}{h} - 1 \right\| < \delta.$$

Note that by this definition, $\boldsymbol{u}^s$ is an exact trajectory of Φ on a perturbed time grid defined by $\{\eta_i\}$. We are now ready to write the shadowing lemma.

Lemma 2 (Shadowing Lemma)**.** *Let Λ be a hyperbolic invariant set of Φ, then in some neighborhood U of Λ, for any $\delta > 0$ there exists $\varepsilon > 0$ such that every ε-pseudo-orbit $\boldsymbol{u}^\varepsilon \in U$ is δ-shadowed by some $\boldsymbol{u}^s \in \Lambda$ [42].*

The hyperbolic invariant set Λ is key to making the shadowing lemma work. In [42], the author defines Λ as a hyperbolic invariant of Φ if Λ is compact and if the tangent space at every point $\boldsymbol{p} \in \Lambda$ can be split into subspaces

$$E^u(\boldsymbol{p}) \oplus E^s(\boldsymbol{p}) \oplus E^n(\boldsymbol{p})$$

where E^u, E^s, and E^n are unstable, stable, and neutrally stable subspaces which are $\partial_{\boldsymbol{p}}\Phi$-invariant so that

$$[\partial_{\boldsymbol{p}}\Phi(\boldsymbol{p}, h)]E^{u|n|s}(\boldsymbol{p}) = E^{u|n|s}(\Phi(\boldsymbol{p}, h)),$$

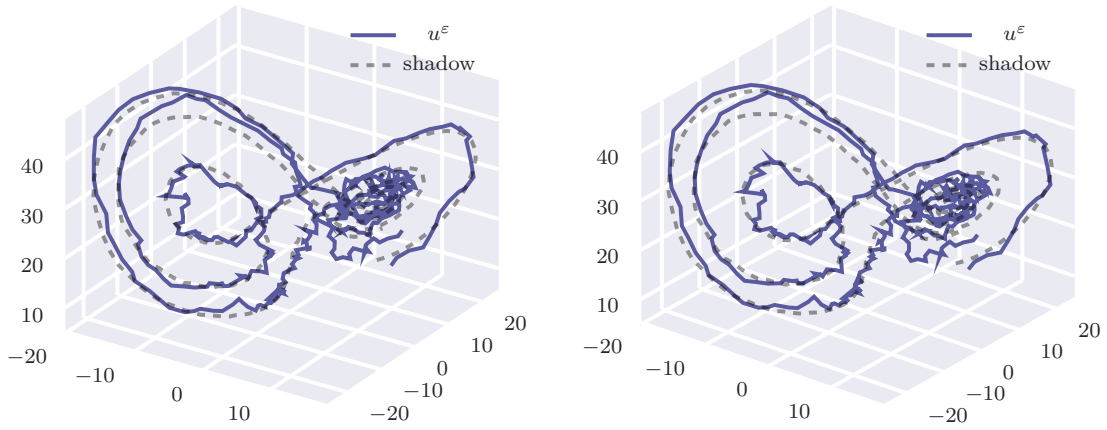


Figure 4.1: A noisy ε-orbit of the Lorenz system and its numerically computed δ-shadow. ➢

and where $E^n = \text{span}(\partial_h \Phi(\boldsymbol{p}))$. Then, intuitively, the shadow trajectory $\boldsymbol{u}^s$ is found by projecting the perturbations $\Phi(\boldsymbol{u}^\varepsilon_{i-1}) - \boldsymbol{u}^\varepsilon_i$ onto the subspaces E^u, E^s, and E^n, and integrating them forward in time along E^s, backward in time along E^u, and accounting for perturbations in E^n using a shifted time grid defined by $\{\eta_i\}$. In a sense, the shadow trajectory is the orthogonal projection of $\boldsymbol{u}^\varepsilon$ onto the space of exact orbits of Φ. This motivates the Least Squares Shadowing approach discussed in the following subsection.

It should be acknowledged that shadowing has recently become a topic of some amount of controversy, with a recent paper even going as far as the say that "we cannot trust the long-time averages generated by numerical simulations of chaotic processes, even when we account for model uncertainties and statistical noise due to finite-time averaging" [4]. The argument goes that while the shadowing lemma guarantees the existence of a shadow trajectory, there is no such guarantee that the trajectory will represent the statistics of the system in the same way that a randomly chosen initial condition will. The work [4] presents several examples where a seemingly innocent application of shadowing results in finding a shadow trajectory

which, despite being a true orbit of the system, has an initial condition starting from a set of measure zero with a demonstrably different statistical distribution. That is to say that shadowing should be used with care, since there are still several open questions as to when a shadow trajectory can be trusted as a statistical representative of the system. However, despite these concerns, I believe that shadowing is still the most promising potential solution to the PinT scaling problem for chaotic and turbulent systems, so it is still important to explore these methods.

4.1.2 Least Squares Shadowing (LSS)

Rather than computing the covariant Lyapunov vectors and explicitly integrating local perturbations forward and backward in time, which in practice is very expensive and not numerically stable with respect to errors in the computed Lyapunov vectors [56], in the Least Squares Shadowing (LSS) formulation, the shadow trajectory is approximated by the solution to the convex constrained minimization problem

$$\min_{\boldsymbol{v},\eta} \left\{ \frac{1}{2} \sum_{i=0}^{n_t} \|\boldsymbol{u}_i - \boldsymbol{v}_i\|^2 + \frac{1}{2} \sum_{i=1}^{n_t} \left| \frac{\eta_i}{h_i} - 1 \right|^2 \right\}, \tag{4.2}$$

$$\text{s.t. } \boldsymbol{v}_i = \Phi(\boldsymbol{v}_{i-1}, \eta_i), \quad i = 1, 2, \ldots, n_t, \tag{4.3}$$

where $\boldsymbol{u}$ takes the role of $\boldsymbol{u}^\varepsilon$, and thus it is assumed that $\boldsymbol{u}$ resembles a trajectory of Φ in some way, and $\boldsymbol{v}$ takes the role of the shadow trajectory. In the context of LSS, $\boldsymbol{u}$ is called the *reference* trajectory, and $\boldsymbol{v}$ is called the shadow trajectory. Importantly, Equation (4.3) does not specify an initial condition for $\boldsymbol{v}$, it only ensures that the distance $\|\boldsymbol{u}_0 - \boldsymbol{v}_0\|$ is small via minimization. For this reason, LSS is said to relax the initial condition. Because the sensitivity of chaotic systems on their initial conditions is what causes the ill-conditioning of the IVP, we might expect the relaxation of the initial condition to improve the conditioning of the problem, and that is indeed the case.

In [55], the authors show that the LSS problem is well conditioned, with condition number approximately proportional to the ratio between the maximum and minimum growth factors $e^{\lambda_1}/e^{\lambda_{n_s}}$, where λ_1 and λ_{n_s} are the greatest positive and negative Lyapunov exponents of the system. Most importantly, note that this condition number is only dependent on the average rates of growth in the system, and not on the length of the time domain. Thus, [55] proposes that if the LSS problem can be solved PinT, then it would likely result in a scalable solver for chaotic and turbulent systems. However, the proposed algorithm is considered too expensive for practical application, since solving the LSS problem for a system typically involves using Newton's method, and thus inversion of an $n_s n_t \times n_s n_t$ SPD block tridiagonal matrix, which, although amenable to space-time multigrid, is a very difficult and expensive problem to solve relative to time stepping, especially for a PDE, where n_s can be large. Further, LSS requires an initial guess, in the form of the reference trajectory $\boldsymbol{u}$, which already must be close to an exact orbit of Φ for Newton's method to converge, and there is no general way to generate this initial guess given in [55].

4.2 LSS and MGRIT

We saw in Section 1.3.4 that poor MGRIT convergence for chaotic systems is caused by exponential divergence of the coarse grid solution from the restricted fine grid solution, resulting in a failure of the τ correction at later times to properly correct the coarse grid equation. The ability of LSS to find an exact trajectory of a chaotic system that is uniformly close to an inexact trajectory seems to make it a good candidate to fix this problem on the coarse grid for MGRIT, if one is able to accept a solution with a nearby, albeit different, initial condition than the one originally provided to the solver.

Consider solving the LSS problem on the coarse grid, where the restricted fine grid

solution is used as the reference trajectory, and the coarse grid solution is the shadow trajectory which is an exact solution to the τ-corrected FAS coarse grid equation, Equation (1.8) with relaxed initial condition. The coarse grid solution would then remain uniformly close to the restricted fine grid solution on the whole time domain while still being an exact solution to the $\boldsymbol{\tau}$-corrected coarse grid equation with a slightly different initial condition. Thus, the residual could, in theory, be reduced uniformly everywhere in time by such a coarse grid correction. According to the shadowing lemma, the closeness of the shadow to its ε-orbit is implicitly related to the size of ε, that is, the uniform bound on the residual norm. Therefore, in this context, if the residual were reduced following a correction from a LSS solution on the coarse grid, then the coarse grid solution in the next iteration could be closer to the restricted fine grid solution than in the previous iteration, resulting in an even better coarse grid correction and lower residual everywhere. This line of reasoning seems to imply that such an iteration could be convergent, and convergence of the solution on the fine grid to any exact orbit of Φ would result in a zero coarse grid correction, since the τ correction ensures that the restricted fine grid solution is, roughly speaking, an exact orbit of $\Phi_c + \tau$. In other words, this process would be a fixed point iteration.

The idea to use LSS as a replacement for the FAS coarse grid equation was the original motivation for this dissertation work, and while it still has yet to be realized, it has inspired the recently developed Local Shadowing Relaxation presented in Section 4.3, which I believe represents a major step toward the realization of a scalable MGinT solver based on shadowing.

MGinT can also potentially provide a general way to produce an initial guess for a shadowing-based approach, using full multigrid (FMG). In FMG, instead of starting with relaxation on the fine grid, it is assumed that no initial guess exists for the solution value, so the cycle begins by solving the coarsest grid exactly, which

is interpolated to the next finest grid where V-cycling takes place until the solution on that level is solved to discretization accuracy, at which point the solution is interpolated to the next finest grid, and so on. FMG is a remarkable algorithm in that it can be said to solve the problem on the fine grid to discretization accuracy in a single iteration without the need for an initial guess. In a shadowing based MGinT solver, an FMG cycle could similarly provide an excellent intial guess to the fine grid by initializing on a very coarse grid and iteratively refining that guess through the two-level process described above. In contrast, FMG based on naive MGRIT does not work for chaotic systems, since the iterative refinement process breaks down– the initial guess provided to a fine grid after refinement comes from solving the previous coarse grid to discretization accuracy, but this initial guess will naturally be exponentially far from the solution obtained after solving the fine grid to discretization accuracy.

4.3 Local Shadowing Relaxation (LSR)

Due to the exponential growth of tangent vectors along a chaotic trajectory, FCF-relaxation, commonly used with MGRIT, is not convergent, and will in fact magnify a small residual at a given point in time to an exponentially larger residual at a later point in time in a few iterations. Figure 4.2 demonstrates this for the Lorenz system, where initially noisy but uniform errors are magnified in the direction of the unstable Lyapunov vectors. This behavior, coupled with the coarse grid correction, which is increasingly low-quality as the time-horizon expands, leads to very poor convergence of the MGRIT algorithm for chaotic problems. However, the LSS problem is significantly more expensive to solve than the time stepping problem, and existing methods for solving LSS are $\mathcal{O}(n_t)$ and are not PinT, so this LSS coarse grid is not likely to be practical in application. Thus, in this chapter, we will devise a PinT relaxation

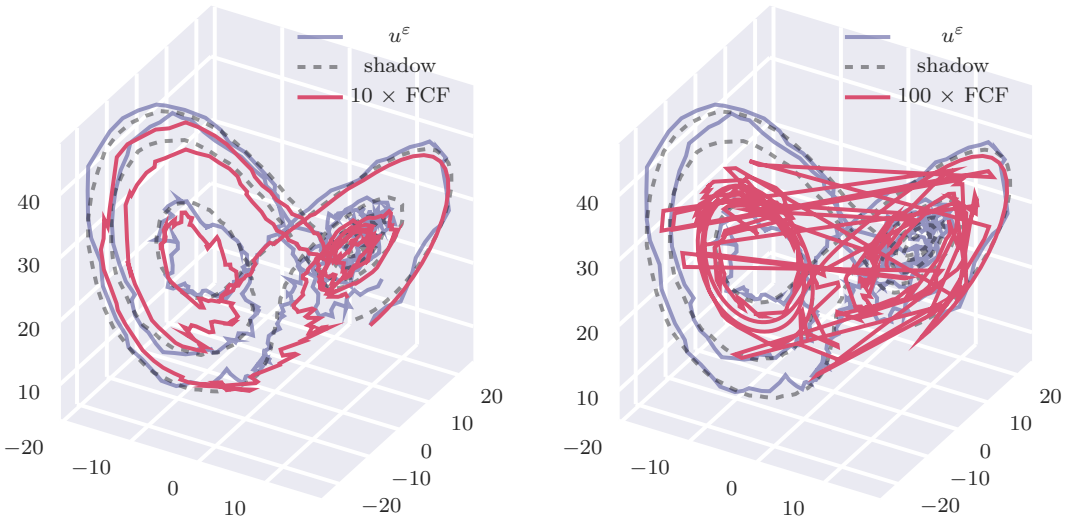


Figure 4.2: The effect of FCF-relaxation on initially uniform error in the Lorenz system.

method which we will see converges to the same shadow trajectory as LSS, and we will see how this relaxation performs in a MGinT setting.

Consider the LSS problem posed over a single interval containing $m+1$ time-points i, $i+1$, $\cdots$, $i+m$, with no time dilation, $\{\eta_i\}$, and only minimizing over the two endpoints, as in Equation (4.4). Notice that the constraint can be solved by substitution. This, along with the change of variables $\boldsymbol{v}_i = \boldsymbol{u}_i + \boldsymbol{p}_i$ resulting in Equation (4.5), makes it clear that this is equivalent to the least squares minimization of the residual $\boldsymbol{r}_{i+m} = \boldsymbol{u}_{i+m} - \Phi^m(\boldsymbol{u}_i + \boldsymbol{p}_i)$ with respect to a perturbation $\boldsymbol{p}_i$ along with a term regularizing the size of $\boldsymbol{p}_i$.

$$\min_{\boldsymbol{v}_i,\, \boldsymbol{v}_{i+m}} \frac{1}{2}\|\boldsymbol{v}_i - \boldsymbol{u}_i\|^2 + \frac{1}{2}\|\boldsymbol{v}_{i+m} - \boldsymbol{u}_{i+m}\|^2 \text{ s.t. } \boldsymbol{v}_{i+m} = \Phi^m(\boldsymbol{v}_i) \tag{4.4}$$

$$\equiv \min_{\boldsymbol{p}_i} \frac{1}{2}\|\boldsymbol{p}_i\|^2 + \frac{1}{2}\|\Phi^m(\boldsymbol{u}_i + \boldsymbol{p}_i) - \boldsymbol{u}_{i+m}\|^2. \tag{4.5}$$

While solving the fully coupled LSS problem is considered too expensive, solving the minimization problem in Equation (4.5) on sub-intervals of the time-grid is relatively trivial and highly parallel in time. Thus, our relaxation scheme finds a perturbation

$\boldsymbol{p}_i$ at each C-point t_i, that minimizes the residual at the following C-point, updates each initial point with $\boldsymbol{u}_i \rightarrow \boldsymbol{u}_i + \boldsymbol{p}_i$, and then uses sequential time stepping to update the following points in that interval. This procedure, given in Algorithm 4.1 for a single interval, is the atomic algorithm from which we will construct a stable relaxation scheme which, when deployed in a multigrid algorithm, approximates the ideal convergence of this hypothetical LSS coarse grid algorithm.

Algorithm 4.1 LSR($\boldsymbol{u}_i, \boldsymbol{u}_{i+m}$); Local Shadowing Relaxation update

LSR update:

Solve Equation 4.5 for $\boldsymbol{p}_i$

$\boldsymbol{u}_i \leftarrow \boldsymbol{u}_i + \boldsymbol{p}_i$

FC-relaxation:

for each $\boldsymbol{u}_j$, $j = i+1, i+2, \ldots, i+m$ **do**

$\boldsymbol{u}_j \leftarrow \Phi(\boldsymbol{u}_{j-1})$

end for

When posed on a coarse interval, where $\boldsymbol{u}_i$ and $\boldsymbol{u}_{i+m}$ are both C-points, Algorithm 4.1 resembles FC-relaxation, although the solution of (4.5) and subsequent perturbation of $\boldsymbol{u}_i$ allows information to travel backward in time before FC-relaxation takes place.

The first-order necessary conditions for the LSR update are given via the gradient of (4.5), which shows that, to a first-order approximation,

$$\boldsymbol{p}_i + \left[\frac{\partial \Phi^m}{\partial \boldsymbol{u}}(\boldsymbol{u}_i + \boldsymbol{p}_i)\right]^T (\Phi^m(\boldsymbol{u}_i + \boldsymbol{p}_i) - \boldsymbol{u}_{i+m}) = \boldsymbol{0}, \tag{4.6}$$

$$\boldsymbol{p}_i + \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}^T (\Phi^m(\boldsymbol{u}_i + \boldsymbol{p}_i) - \boldsymbol{u}_{i+m}) + \mathcal{O}\left(\|\boldsymbol{p}_i\|^2\right) = \boldsymbol{0},$$

$$\left(I + \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}^T \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}\right) \boldsymbol{p}_i + \mathcal{O}\left(\|\boldsymbol{p}_i\|^2\right) = -\frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}^T \boldsymbol{r}_{i+m}, \tag{4.7}$$

where

$$\boldsymbol{r}_{i+m} = \Phi^m(\boldsymbol{u}_i) - \boldsymbol{u}_{i+m} \tag{4.8}$$

is the residual at time point $i+m$. Ignoring second order terms, we can see that $\|\boldsymbol{p}_i\| \approx \mathcal{O}(\|\boldsymbol{r}_{i+1}\|)$ since the matrix implicitly defining $\boldsymbol{p}_i$ on the left-hand side of equation (4.7) is clearly positive definite, with eigenvalues greater than or equal to one in magnitude. This loosely argues that the perturbation $\boldsymbol{p}_i$ is small whenever the residual $\boldsymbol{r}_{i+m}$ is small, and vanishes in the limit as $\boldsymbol{r}_{i+m} \to 0$. It remains to prove this by showing that the solution $\boldsymbol{p}_i$ exists and that the norm of the Jacobian of $\boldsymbol{p}_i$ with respect to $\boldsymbol{r}_{i+m}$ is less than or equal to 1.

Lemma 3. *Assume Φ is twice continuously differentiable and $\partial_{\boldsymbol{u}_i}\Phi$ is full rank. In some neighborhood of a true trajectory u^*, equation* (4.6) *implicitly defines the LSR update $\boldsymbol{p}_i$ as a unique, continuously differentiable function of $(\boldsymbol{u}_i, \boldsymbol{u}_{i+m})$ such that $\|\boldsymbol{p}_i\| \lessapprox \|\boldsymbol{r}_{i+m}\|/2$*

Proof. Define $\boldsymbol{g}_i(\boldsymbol{u}, \boldsymbol{p}) : \mathbb{R}^{2n_x} \to \mathbb{R}^{n_x}$ such that

$$\boldsymbol{g}_i(\boldsymbol{u}, \boldsymbol{p}) = \boldsymbol{p}_i + \left[\frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}(\boldsymbol{u}_i + \boldsymbol{p}_i)\right]^T (\Phi^m(\boldsymbol{u}_i + \boldsymbol{p}_i) - \boldsymbol{u}_{i+m}). \tag{4.9}$$

The Jacobian of $\boldsymbol{g}_i$ is given by

$$\frac{\partial \boldsymbol{g}_i}{\partial \boldsymbol{p}_i} = I + \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}^T \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i} + \sum_k \frac{\partial^2 \Phi^m_k}{\partial \boldsymbol{u}_i \partial \boldsymbol{u}_i^T}(\boldsymbol{u}_{i+m,k} - \Phi^m_k(\boldsymbol{u}_i + \boldsymbol{p}_i)) \tag{4.10}$$

$$\frac{\partial \boldsymbol{g}_i}{\partial \boldsymbol{u}_i} = \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}^T \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i} + \sum_k \frac{\partial^2 \Phi^m_k}{\partial \boldsymbol{u}_i \partial \boldsymbol{u}_i^T}(\boldsymbol{u}_{i+m,k} - \Phi^m_k(\boldsymbol{u}_i + \boldsymbol{p}_i)) \tag{4.11}$$

$$\frac{\partial \boldsymbol{g}_i}{\partial \boldsymbol{u}_{i+m}} = -\frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}^T \tag{4.12}$$

where all first and second derivatives of Φ^m are understood to be evaluated at the point $\boldsymbol{u}_i + \boldsymbol{p}_i$. Clearly, $g(\boldsymbol{u}^*, 0) = 0$, and the Jacobian at this point

$$\frac{\partial \boldsymbol{g}_i}{\partial \boldsymbol{p}_i}(\boldsymbol{u}^*, 0) = I + \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}^T \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i} \tag{4.13}$$

is symmetric positive definite and thus invertible. Therefore, there exists some neighborhood about the point $(\boldsymbol{u}^*, 0)$ in which $\boldsymbol{p}_i$ is defined as a unique differentiable function of $\boldsymbol{u}$ by the implicit function theorem, and further we have in this neighborhood

that

$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{u}}(\boldsymbol{u}) = -\left[\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{p}}(\boldsymbol{u}, \boldsymbol{p}(\boldsymbol{u}))\right]^{-1} \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{p}(\boldsymbol{u})), \tag{4.14}$$

from which we get that

$$\frac{\partial \boldsymbol{p}_i}{\partial \boldsymbol{u}_i}(\boldsymbol{u}^*) = -\left(I + \frac{\partial \Phi^m{}^T}{\partial \boldsymbol{u}_i}\frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}\right)^{-1} \frac{\partial \Phi^m{}^T}{\partial \boldsymbol{u}_i}\frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}, \tag{4.15}$$

$$\frac{\partial \boldsymbol{p}_i}{\partial \boldsymbol{u}_{i+m}}(\boldsymbol{u}^*) = \left(I + \frac{\partial \Phi^m{}^T}{\partial \boldsymbol{u}_i}\frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}\right)^{-1} \frac{\partial \Phi^m{}^T}{\partial \boldsymbol{u}_i}. \tag{4.16}$$

Finally, we bound the norm of the Jacobian $\partial \boldsymbol{p}_i / \partial \boldsymbol{r}_{i+1}$ at the point $(\boldsymbol{u}^*, 0)$ by implicit differentiation of Equations (4.6) and (4.8):

$$\frac{\partial \boldsymbol{p}_i}{\partial \boldsymbol{r}_{i+m}} + \frac{\partial \Phi^m{}^T}{\partial \boldsymbol{u}_i}\left(\frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}\left(\frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{r}_{i+m}} + \frac{\partial \boldsymbol{p}_i}{\partial \boldsymbol{r}_{i+m}}\right) - \frac{\partial \boldsymbol{u}_{i+m}}{\partial \boldsymbol{r}_{i+m}}\right) \tag{4.17}$$

$$+ \frac{\partial^2 \Phi^m}{\partial \boldsymbol{u}_i^2}\frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{r}_{i+m}}(\Phi^m(\boldsymbol{u}_i + \boldsymbol{p}_i) - \boldsymbol{u}_{i+m}) \qquad = 0 \tag{4.18}$$

$$\tag{4.19}$$

$$\frac{\partial \boldsymbol{p}_i}{\partial \boldsymbol{r}_{i+m}} + \frac{\partial \Phi^m{}^T}{\partial \boldsymbol{u}_i}\frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}\frac{\partial \boldsymbol{p}_i}{\partial \boldsymbol{r}_{i+m}} = \frac{\partial \Phi^m{}^T}{\partial \boldsymbol{u}_i}\frac{\partial \boldsymbol{u}_{i+m}}{\partial \boldsymbol{r}_{i+m}} - \frac{\partial \Phi^m{}^T}{\partial \boldsymbol{u}_i}\frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}\frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{r}_{i+m}} \tag{4.20}$$

$$\left(I + \frac{\partial \Phi^m{}^T}{\partial \boldsymbol{u}_i}\frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}\right)\frac{\partial \boldsymbol{p}_i}{\partial \boldsymbol{r}_{i+m}} = \frac{\partial \Phi^m{}^T}{\partial \boldsymbol{u}_i}\left(\frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}\frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{r}_{i+m}} - I\right) - \frac{\partial \Phi^m{}^T}{\partial \boldsymbol{u}_i}\frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}\frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{r}_{i+m}} \tag{4.21}$$

$$\left(I + \frac{\partial \Phi^m{}^T}{\partial \boldsymbol{u}_i}\frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}\right)\frac{\partial \boldsymbol{p}_i}{\partial \boldsymbol{r}_{i+m}} = -\frac{\partial \Phi^m{}^T}{\partial \boldsymbol{u}_i} \tag{4.22}$$

$$\frac{\partial \boldsymbol{p}_i}{\partial \boldsymbol{r}_{i+m}} = -\left(I + \frac{\partial \Phi^m{}^T}{\partial \boldsymbol{u}_i}\frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}\right)^{-1} \frac{\partial \Phi^m{}^T}{\partial \boldsymbol{u}_i} \tag{4.23}$$

$$\left\|\frac{\partial \boldsymbol{p}_i}{\partial \boldsymbol{r}_{i+m}}\right\|_2 = \left\|\left(I + \frac{\partial \Phi^m{}^T}{\partial \boldsymbol{u}_i}\frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}\right)^{-1} \frac{\partial \Phi^m{}^T}{\partial \boldsymbol{u}_i}\right\|_2 \tag{4.24}$$

$$= \max_j \left\|\frac{\sigma_{i,j}}{1 + \sigma_{i,j}^2}\right\| \leq \frac{1}{2}, \tag{4.25}$$

where $\sigma_{i,j}$ is the jth singular value of $\partial \Phi^m / \partial \boldsymbol{u}_i$. ☠

Lemma 3 shows that $\boldsymbol{p}_i$ is well-defined, and further that in a neighborhood of an exact orbit $\boldsymbol{u}^*$, $\|\boldsymbol{p}_i\| \lessapprox \|\boldsymbol{r}_{i+1}\|/2$, indicating that the perturbation $\boldsymbol{p}_i$ is small when the residual $\boldsymbol{r}_{i+m}$ is small. Further, note that this procedure eliminates the residual at every point in the interval except for at t_i. Next, we find the Jacobian of Algorithm 4.1 and see that it does not excite modes of error in a neighborhood of $\boldsymbol{u}^*$, in contrast to FCF-relaxation.

Lemma 4. *Algorithm 4.1 is stable at any fixed point* $\boldsymbol{u}^*$.

Proof. Algorithm 4.1 is given in the equations

$$\boldsymbol{u}_i^{k+1} = \boldsymbol{u}_i^k + \boldsymbol{p}_i, \tag{4.26}$$

$$\boldsymbol{u}_{i+m}^{k+1} = \Phi(\boldsymbol{u}_i^k + \boldsymbol{p}_i), \tag{4.27}$$

with Jacobian

$$\frac{\partial \boldsymbol{u}^{k+1}}{\partial \boldsymbol{u}^k} = \begin{bmatrix} \left(I + \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}^T \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}\right)^{-1} & \left(I + \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}^T \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}\right)^{-1} \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}^T \\ \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i} \left(I + \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}^T \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}\right)^{-1} & \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i} \left(I + \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}^T \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}\right)^{-1} \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i}^T \end{bmatrix}, \tag{4.28}$$

which follows directly from Equations (4.15) and (4.16). Note that this Jacobian is symmetric positive semi-definite. Next, let $U\Sigma V^T$ be the singular value decomposition of $\partial \Phi^m / \partial \boldsymbol{u}_i$, and we see that

$$J_{\text{LSR}} = \frac{\partial \boldsymbol{u}^{k+1}}{\partial \boldsymbol{u}^k} = \begin{bmatrix} V & \\ & U \end{bmatrix} \begin{bmatrix} (I + \Sigma^2)^{-1} & \Sigma\,(I + \Sigma^2)^{-1} \\ \Sigma\,(I + \Sigma)^{-1} & \Sigma^2\,(I + \Sigma^2)^{-1} \end{bmatrix} \begin{bmatrix} V & \\ & U \end{bmatrix}^T, \tag{4.29}$$

which has eigenvalues $\lambda_1 = 0$ and $\lambda_2 = 1$, each with multiplicity n_s. For any vector

$\boldsymbol{v}_i \in \mathbb{R}^{n_s}$, let $\boldsymbol{c} = V^T \boldsymbol{v}_i$, then

$$
\begin{aligned}
&\begin{bmatrix} V & \\ & U \end{bmatrix} \begin{bmatrix} (I+\Sigma^2)^{-1} & \Sigma\,(I+\Sigma^2)^{-1} \\ \Sigma\,(I+\Sigma)^{-1} & \Sigma^2\,(I+\Sigma^2)^{-1} \end{bmatrix} \begin{bmatrix} V & \\ & U \end{bmatrix}^T \begin{bmatrix} V\boldsymbol{c} \\ [\partial_{\boldsymbol{u}_i}\Phi]V\boldsymbol{c} \end{bmatrix} = \\
&= \begin{bmatrix} V & \\ & U \end{bmatrix} \begin{bmatrix} (I+\Sigma^2)^{-1} & \Sigma\,(I+\Sigma^2)^{-1} \\ \Sigma\,(I+\Sigma)^{-1} & \Sigma^2\,(I+\Sigma^2)^{-1} \end{bmatrix} \begin{bmatrix} V & \\ & U \end{bmatrix}^T \begin{bmatrix} V\boldsymbol{c} \\ U\Sigma\boldsymbol{c} \end{bmatrix} \\
&= \begin{bmatrix} V & \\ & U \end{bmatrix} \begin{bmatrix} (I+\Sigma^2)^{-1} & \Sigma\,(I+\Sigma^2)^{-1} \\ \Sigma\,(I+\Sigma)^{-1} & \Sigma^2\,(I+\Sigma^2)^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{c} \\ \Sigma\boldsymbol{c} \end{bmatrix} \\
&= \begin{bmatrix} V & \\ & U \end{bmatrix} \begin{bmatrix} (I+\Sigma^2)^{-1}\,(\boldsymbol{c}+\Sigma^2\boldsymbol{c}) \\ \Sigma\,(I+\Sigma^2)^{-1}\,(\boldsymbol{c}+\Sigma^2\boldsymbol{c}) \end{bmatrix} = \begin{bmatrix} V\boldsymbol{c} \\ U\Sigma\boldsymbol{c} \end{bmatrix} = \begin{bmatrix} V\boldsymbol{c} \\ U\Sigma V^T V\boldsymbol{c} \end{bmatrix} \\
&= \begin{bmatrix} V\boldsymbol{c} \\ [\partial_{\boldsymbol{u}_i}\Phi]V\boldsymbol{c} \end{bmatrix},
\end{aligned}
$$

and we see that $(\boldsymbol{v}_i; [\partial_{\boldsymbol{u}_i}\Phi]\boldsymbol{v}_i)$ is an eigenvector of J_{LSR} corresponding to eigenvalue $\lambda_2 = 1$. Similarly, the orthogonal vector $([\partial_{\boldsymbol{u}_i}\Phi]\boldsymbol{v}_i; -\boldsymbol{v}_i)$ is an eigenvector corresponding to eigenvalue $\lambda_1 = 0$. ☠

Notice that the eigenvectors of (4.28) with unit eigenvalue correspond with perturbations to the initial point $\boldsymbol{u}_i$, and the eigenvectors with zero eigenvalue are orthogonal to these. One might call the former *physical* errors and the latter *unphysical* errors, since a physical error results in a trajectory which is indistinguishable from an exact solution locally in time. That is, physical errors do not affect the residual locally in time, while unphysical errors do, and thus the physical errors make up the null-space of the local residual operator while the unphysical errors make up the null-space of the LSR relaxation operator. This makes sense, because the shadow trajectory is the orbit nearest the reference trajectory in the space of true orbits, that is, orbits which only differ from each other in initial condition, so translating that to tangent space, the difference between the reference trajectory and its shadow

is orthogonal to the space of unphysical error. We see that LSR is an orthogonal projection of the local error onto the subspace of physical error.

Contrast this result with FC-relaxation, which has a Jacobian of

$$J_{FC} = \begin{bmatrix} I & 0 \\ \frac{\partial \Phi^m}{\partial \boldsymbol{u}_i} & 0 \end{bmatrix},$$

with singular values $\sigma_1 = 0$ and $\sigma_{2,j} = \sqrt{1 + \sigma_{i,j}^2} \geq 1$, where $\sigma_{i,j}$ is the jth singular value of $\partial_{\boldsymbol{u}_i}\Phi$. The right singular vectors of J_{FC} are the columns of the identity matrix, and the left singular vectors are the eigenvectors of J_{LSR}. Therefore, the symmetrization $\sqrt{J_{\text{FC}} J_{\text{FC}}^T}$ gives the action of FC-relaxation on the eigenspace of J_{LSR}, which shows that FC-relaxation and LSR both eliminate unphysical errors, but FC-relaxation magnifies physical errors, while LSR does not.

4.3.1 Initial attempts at stable LSR

Despite these promising results for a single interval, LSR is not stable when applied naively to the whole time grid. To illustrate this, let us make two initial attempts to use Algorithm 4.1 as a PinT relaxation scheme. First, notice that while FC-relaxation only updates $\boldsymbol{u}_{i+m}$, LSR provides an update for both endpoints $\boldsymbol{u}_i$ and $\boldsymbol{u}_{i+m}$, so there is not a single obvious way to use the overlapping updates one gets from performing Algorithm 4.1 on each C-interval in parallel. Three options are

$$\boldsymbol{u}_i \leftarrow \text{LSR}^+{}_i(\boldsymbol{u}) := \Phi^m(\boldsymbol{u}_{i-m} + \boldsymbol{p}_{i-m}) \tag{4.30}$$

$$\boldsymbol{u}_i \leftarrow \text{LSR}^-{}_i(\boldsymbol{u}) := \boldsymbol{u}_i + \boldsymbol{p}_i \tag{4.31}$$

$$\boldsymbol{u}_i \leftarrow \text{LSR}_i^{1/2}(\boldsymbol{u}) := \boldsymbol{u}_i + \frac{1}{2}\boldsymbol{p}_i + \frac{1}{2}\left(\Phi^m(\boldsymbol{u}_{i-m} + \boldsymbol{p}_{i-m}) - \boldsymbol{u}_i\right) \tag{4.32}$$

Where LSR^- uses the update from the $(\boldsymbol{u}_i; \boldsymbol{u}_{i+m})$ interval, LSR^+ uses the update from the $(\boldsymbol{u}_{i-m}; \boldsymbol{u}_i)$ interval, and $\text{LSR}^{1/2}$ uses an average between the two updates.

LSR^+ seems reasonably symmetric, since there is first a flow of information backward in time from $\boldsymbol{u}_i \to \boldsymbol{u}_{i-m}$ via $\boldsymbol{p}_{i-m}$, followed by a flow of information forward in time from $\boldsymbol{u}_{i-m} \to \boldsymbol{u}_i$ via FC-relaxation with Φ^m, so we select LSR^+ as our first candidate. Let the superscript k denote the current relaxation iteration, then $\boldsymbol{u}^{k+1} = \text{LSR}^+(\boldsymbol{u}^k)$. We seek a bound for $\|\partial \boldsymbol{u}^{k+1}/\partial \boldsymbol{u}^k\|$ at the fixed point $\boldsymbol{u}^*$ where $\|\cdot\| = \|\cdot\|_{2,\infty}$ is the 2-norm in space and ∞-norm in time. First, we bound the blocks in each row,

$$\left\| \frac{\partial \boldsymbol{u}_i^{k+1}}{\partial \boldsymbol{u}_{i-m}^k} \right\|_2 = \left\| \frac{\partial \Phi^m}{\partial \boldsymbol{u}_{i-m}^k} \left(I + \frac{\partial \boldsymbol{p}_{i-m}}{\partial \boldsymbol{u}_{i-m}^k} \right) \right\|_2 \tag{4.33}$$

$$= \left\| \frac{\partial \Phi^m}{\partial \boldsymbol{u}_{i-m}^k} \left(I + \frac{\partial \Phi^m}{\partial \boldsymbol{u}_{i-m}^k}^T \frac{\partial \Phi^m}{\partial \boldsymbol{u}_{i-m}^k} \right)^{-1} \right\|_2 \tag{4.34}$$

$$= \max_j \left\{ \frac{\sigma_{i-m,j}}{1+\sigma_{i-m,j}^2} \right\}, \tag{4.35}$$

where the second to last equality is a direct result of Equation (4.15) and $\sigma_{i-m,j}$ is the jth singular value of $\partial \Phi^m / \partial \boldsymbol{u}_{i-m}$, and

$$\left\| \frac{\partial \boldsymbol{u}_i^{k+1}}{\partial \boldsymbol{u}_i^k} \right\|_2 = \left\| \frac{\partial \Phi^m}{\partial \boldsymbol{u}_{i-m}^k} \left(\frac{\partial \boldsymbol{p}_{i-m}}{\partial \boldsymbol{u}_i^k} \right) \right\|_2 \tag{4.36}$$

$$= \left\| \frac{\partial \Phi^m}{\partial \boldsymbol{u}_{i-m}^k} \left(I + \frac{\partial \Phi^m}{\partial \boldsymbol{u}_{i-m}^k}^T \frac{\partial \Phi^m}{\partial \boldsymbol{u}_{i-m}^k} \right)^{-1} \frac{\partial \Phi^m}{\partial \boldsymbol{u}_{i-m}^k}^T \right\|_2 \tag{4.37}$$

$$\leq \max_j \left\{ \frac{\sigma_{i-m,j}^2}{1+\sigma_{i-m,j}^2} \right\}. \tag{4.38}$$

Finally, we see that

$$\left\| \frac{\partial \boldsymbol{u}^{k+1}}{\partial \boldsymbol{u}^k} \right\|_{2,\infty} = \max_i \left\{ \left\| \frac{\partial \boldsymbol{u}_i^{k+1}}{\partial \boldsymbol{u}_{i-m}^k} \right\|_2 + \left\| \frac{\partial \boldsymbol{u}_i^{k+1}}{\partial \boldsymbol{u}_i^k} \right\|_2 \right\} \tag{4.39}$$

$$= \max_{i,j} \left\{ \frac{\sigma_{i-m,j} + \sigma_{i-m,j}^2}{1+\sigma_{i-m,j}^2} \right\} \leq \frac{1+\sqrt{2}}{2}. \tag{4.40}$$

Unfortunately, this upper bound on the Jacobian of our tentative LSR^+ is greater than one, and it is attained when $\sigma_{i-m,j} = \sqrt{2}+1 \approx 2.41$ which means that this iteration excites error for locally *unstable* modes.

Placing the blame for the failure of our first attempt on the forward propagation with Φ, which perhaps is allowing errors to grow along the unstable manifold forward

in time, we might decide to forgo this step and try just computing $\boldsymbol{p}_i$ and adding it to each $\boldsymbol{u}_i$ as in LSR$^-$. The propagation from $i \to i+m$ is only responsible for a single factor of $\sigma_{i,j}$ in our bound, so the new bound becomes

$$\left\| \frac{\partial \boldsymbol{u}^{k+1}}{\partial \boldsymbol{u}^k} \right\|_{2,\infty} = \max_{i,j} \left\{ \frac{1+\sigma_{i,j}}{1+\sigma_{i,j}^2} \right\} \leq \frac{1+\sqrt{2}}{2}, \tag{4.41}$$

which is attained when $\sigma_{i,j} = \sqrt{2}-1 \approx 0.41$. The bound has not changed! However, now the iteration is exciting error for certain locally *stable* modes, backward in time.

Although these initial attempts are not stable, they are tantalizingly close. Notice that the upper bound in Equations (4.40) and (4.41), is $(1+\sqrt{2})/2 \approx 1.21$, and that this bound is completely independent on the physics of the problem or the stability of Φ. Contrast this with FCF-relaxation, which has Jacobian norm equal to $\max_{i,j} \sigma_{i,j}$ that is unbounded. In both cases we have explored so far, the Jacobian of the iteration is block triangular, and this seems to introduce a slight bias either forward or backward in time. The convergence bounds for LSR$^+$ and LSR$^-$ are plotted in Figure 4.3 for a given singular value σ, under the assumption that σ is constant in time and that the singular vectors satisfy $V_{i+m} = U_i^T$. Notice the skew-symmetry in the convergence bounds for LSR$^+$ and LSR$^-$, where, apparently, $\|\text{LSR}^-{}_\Phi\| = \|\text{LSR}^+{}_{\Phi^{-1}}\|$, i.e. the two methods are equivalent up to the direction of the arrow of time.

The third method listed above, LSR$^{1/2}$ which averages the two updates from LSR$^+$ and LSR$^-$, appears to be stable for all modes, as seen in Figure 4.3, although the convergence rate is bounded from below by $1/2$, which is not ideal. Further, dropping the assumption that σ is constant in time, but keeping the orthogonality relationship, $V_{i+m} = U_i^T$, we see that

$$\|J_{\text{LSR}^{1/2}}\|_{2,\infty} = \frac{1}{2} \max_{i,j} \left\{ \frac{1+\sigma_{i,j}}{1+\sigma_{i,j}^2} + \frac{\sigma_{i-m,j}+\sigma_{i-m,j}^2}{1+\sigma_{i-m,j}^2} \right\} \leq \frac{1+\sqrt{2}}{2},$$

where the maximum is attained when $\sigma_{i-m,j} = 1+\sqrt{2}$ and $\sigma_{i,j} = \sqrt{2}-1$. Thus, if a given mode is expanded by $\partial_{\boldsymbol{u}_{i-m}} \Phi^m$ and contracted by $\partial_{\boldsymbol{u}_i} \Phi^m$, then LSR$^{1/2}$ will be

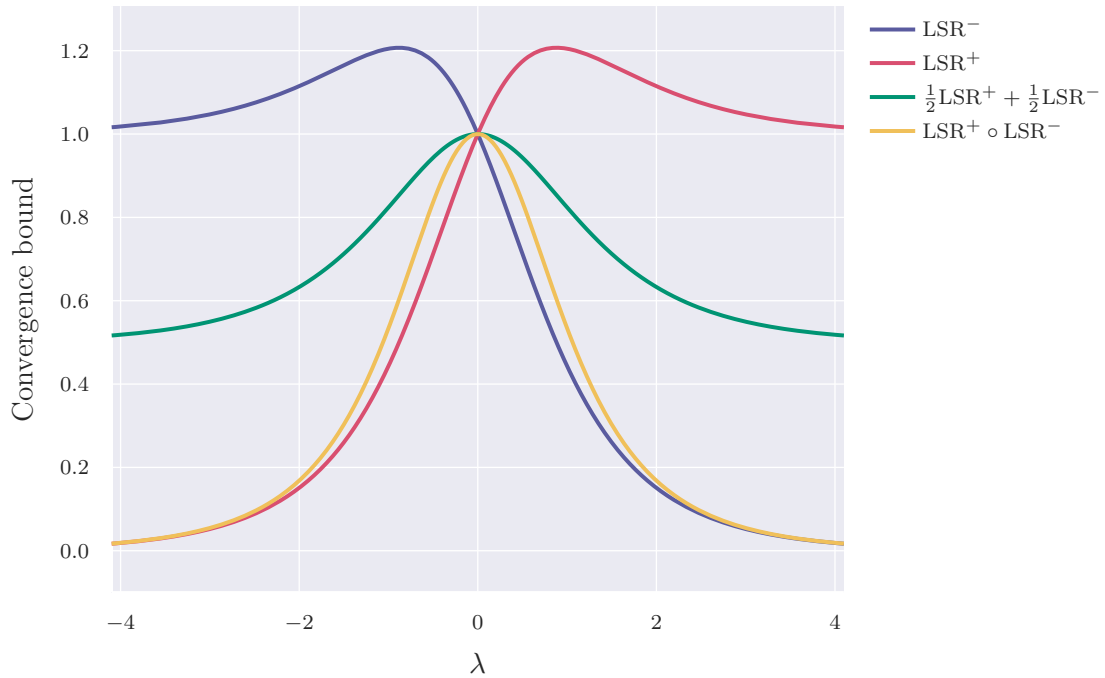


Figure 4.3: Plot of convergence bounds for LSR^+, LSR^-, the average $\mathrm{LSR}^+/2 + \mathrm{LSR}^-/2$ and the product $\mathrm{LSR}^+ \circ \mathrm{LSR}^-$, as a function of $\lambda := \ln(\sigma)$, where σ is a singular value of $\partial\Phi/\partial \boldsymbol{u}_i$, assuming σ is constant in time.

unstable. This is in fact very likely to happen, especially for more neutrally stable modes, since the local Lyapunov exponents $\ln(\sigma_{i,j})/mh$ fluctuate around the true Lyapunov exponents, and they may change sign without changing their long time averages.

Figure 4.3 also suggests that the composition $\mathrm{LSR}^+ \circ \mathrm{LSR}^-$, which has the effect of multiplying the two bounds together, would result in a method which is stable for all σ, and which has a convergence rate that approaches zero as $\sigma \to 0$ or $\sigma \to \infty$, which is what we are looking for– a stable, convergent relaxation method. However, this method suffers from the same problem as $\mathrm{LSR}^{1/2}$, where time dependence of σ results in the same bound of $(1 + \sqrt{2})/2$ at the same point.

4.4 Red-black LSR

Since our Jacobi-like iterations have failed to produce a stable relaxation method, we will now attempt to construct a red-black Gauss-Seidel type iteration, which is almost as parallel, and allows us to apply Algorithm 4.1 to disjoint, non-overlapping intervals. Let the even numbered C-points $\{0, 2m, \ldots, n_t - 2m, n_t\}$ be colored black, and the odd numbered C-points $\{m, 3m, \ldots, n_t - 3m, n_t - m\}$ be colored red, then let a red interval be an interval starting on a red point and ending on the following black point and define a black interval similarly. LSR_{red} and LSR_{blk} will refer to the application of Algorithm 4.1 to all red or black intervals respectively, with any point not contained in either a red or black interval being left unchanged.

At the fixed point $\boldsymbol{u}^*$, the Jacobian of both LSR_{red} and LSR_{blk} will be block diagonal, with

$$J_{\text{LSR}_{\text{red}}} = \begin{bmatrix} M_0 & M_0 F_0^T & & & \\ F_0 M_0 & F_0 M_0 F_0^T & & & \\ & & \ddots & & \\ & & & M_{n_t} & M_{n_t} F_{n_t}^T \\ & & & F_{n_t} M_{n_t} & F_{n_t} M_{n_t} F_{n_t}^T \end{bmatrix}, \tag{4.42}$$

and

$$J_{\text{LSR}_{\text{blk}}} = \begin{bmatrix} I & & & & & \\ & M_1 & M_1 F_1^T & & & \\ & F_1 M_1 & F_1 M_1 F_1^T & & & \\ & & & \ddots & & \\ & & & M_{n_t-1} & M_{n_t-1} F_{n_t-1}^T & \\ & & & F_{n_t-1} M_{n_t-1} & F_{n_t-1} M_{n_t-1} F_{n_t-1}^T & \\ & & & & & I \end{bmatrix}, \tag{4.43}$$

where $F_i = \partial_{\boldsymbol{u}_i}\Phi$, and $M_i = \left(I + F_i^T F_i\right)^{-1}$. The 2×2 blocks are given by Equation (4.28) and thus both $J_{\mathrm{LSR}_{\mathrm{red}}}$ and $J_{\mathrm{LSR}_{\mathrm{blk}}}$ are symmetric positive semi-definite and bounded by 1 in norm. This immediately implies that any composition of $\mathrm{LSR}_{\mathrm{red}}$ and $\mathrm{LSR}_{\mathrm{blk}}$ will also have a fixed-point Jacobian which is bounded by 1 by submultiplicativity of the spectral operator norm, e.g.

$$\left\| \frac{\partial \mathrm{LSR}_{\mathrm{red}} \circ \mathrm{LSR}_{\mathrm{blk}}}{\partial \boldsymbol{u}} \right\|_2 \leq \left\| \frac{\partial \mathrm{LSR}_{\mathrm{red}}}{\partial \boldsymbol{u}} \right\|_2 \left\| \frac{\partial \mathrm{LSR}_{\mathrm{blk}}}{\partial \boldsymbol{u}} \right\|_2 = 1,$$

although this particular case will not result in a symmetric positive semi-definite Jacobian. The composition $\mathrm{LSR}_{\mathrm{blk}} \circ \mathrm{LSR}_{\mathrm{red}} \circ \mathrm{LSR}_{\mathrm{blk}}$ will have a fixed-point Jacobian which is symmetric and positive semi-definite, since we have that $(J_{\mathrm{blk}} J_{\mathrm{red}} J_{\mathrm{blk}})^T = J_{\mathrm{blk}}^T J_{\mathrm{red}}^T J_{\mathrm{blk}}^T = J_{\mathrm{blk}} J_{\mathrm{red}} J_{\mathrm{blk}}$.

4.4.1 Convergence of LSR to a shadow trajectory

The important question remains, if LSR converges, then to what does it converge? Intuition tells us that it should converge to the LSS shadow trajectory, but this is not so obvious. At no point do we construct the global LSS minimization problem, and what's more, LSR overwrites the original reference trajectory or initial guess after one iteration, and so the LSS objective function cannot even be evaluated after a single iteration. We will now see that $\mathrm{LSR}_{\mathrm{red}} \circ \mathrm{LSR}_{\mathrm{blk}}$ in particular can be derived from the KKT system of the full, global-in-time LSS problem (without time dilation), and that this red-black LSR in some sense "remembers" the original reference trajectory, implying that when LSR converges, it converges to the LSS shadow trajectory.

Theorem 5. *The red-black LSR iteration* $\mathrm{LSR}_{\mathrm{blk}} \circ \mathrm{LSR}_{\mathrm{red}}$ *is equivalent to red-black Gauss-Seidel iteration on the KKT system of the global-in-time LSS problem without time dilation.*

Proof. Without loss of generality, let $m = 1$, since any other m can be recovered by

substituting $\Phi \leftarrow \Phi^m$. The LSS problem without time dilation reads

$$\min_{\boldsymbol{v}} \left\{ \frac{1}{2} \sum_{i=0}^{n_t} \|\boldsymbol{v}_i - \boldsymbol{u}_i\|^2 \right\} \text{ s.t. } \boldsymbol{v}_i = \Phi(\boldsymbol{v}_{i-1}). \tag{4.44}$$

Substituting $\boldsymbol{p}_i = \boldsymbol{v}_i - \boldsymbol{u}_i$ and $\boldsymbol{p}_{2i+1} = \Phi(\boldsymbol{u}_{2i} + \boldsymbol{p}_{2i}) - \boldsymbol{u}_{2i+1}$, Equation (4.44) becomes

$$\min_{\boldsymbol{p}_{2i}} \left\{ \frac{1}{2} \sum_{i=0}^{n_t/2} \|\boldsymbol{p}_{2i}\|^2 + \|\Phi(\boldsymbol{u}_{2i} + \boldsymbol{p}_{2i}) - \boldsymbol{u}_{2i+1}\|^2 \right\} \text{ s.t. } \boldsymbol{v}_{2i} = \Phi^2(\boldsymbol{v}_{2i-2}), \tag{4.45}$$

which could be called the Schur-complement minimization problem, because Equation (4.45) has the same solution as Equation (4.44) but is minimizing over half as many variables. Further, note that without the constraint, Equation (4.45) is identical to the disjoint minimization problems solved in LSR_{red}.

Now, using the method of Lagrange multipliers we get the KKT system for Equation (4.44):

$$\boldsymbol{p}_i + \partial_i \Phi^T \boldsymbol{\lambda}_{i+1} - \boldsymbol{\lambda}_i = 0, \tag{4.46}$$

$$\boldsymbol{p}_{i+1} = \Phi(\boldsymbol{u}_i + \boldsymbol{p}_i) - \boldsymbol{u}_{i+1}. \tag{4.47}$$

Letting $\boldsymbol{\lambda}^0 = \boldsymbol{0}$, where the superscript indicates the relaxation iteration, write the red-black Gauss-Seidel iteration as

$$\boldsymbol{p}_{2i-1}^{r,k} + \partial_{2i-1} \Phi^T \boldsymbol{\lambda}_{2i}^{r,k} = \boldsymbol{\lambda}_{2i-1}^{b,k-1} \tag{4.48}$$

$$\boldsymbol{p}_{2i}^{r,k} - \boldsymbol{\lambda}_{2i}^{r,k} = -\partial_{2i} \Phi^T \boldsymbol{\lambda}_{2i+1}^{b,k-1} \tag{4.49}$$

$$\boldsymbol{p}_{2i}^{r,k} = \Phi(\boldsymbol{u}_{2i-1} + \boldsymbol{p}_{2i-1}^{r,k}) - \boldsymbol{u}_{2i} \tag{4.50}$$

$$\boldsymbol{p}_{2i}^{b,k} + \partial_{2i} \Phi^T \boldsymbol{\lambda}_{2i+1}^{b,k} = \boldsymbol{\lambda}_{2i}^{r,k} \tag{4.51}$$

$$\boldsymbol{p}_{2i+1}^{b,k} - \boldsymbol{\lambda}_{2i+1}^{b,k} = -\partial_{2i+1} \Phi^T \boldsymbol{\lambda}_{2i+2}^{r,k} \tag{4.52}$$

$$\boldsymbol{p}_{2i+1}^{b,k} = \Phi(\boldsymbol{u}_{2i} + \boldsymbol{p}_{2i}^{b,k}) - \boldsymbol{u}_{2i+1} \tag{4.53}$$

where unknowns are on the left-hand side and known variables are on the right. Now, subtracting (4.49) from (4.51) and (4.48) from (4.52) we get

$$\boldsymbol{p}_{2i}^{b,k} - \boldsymbol{p}_{2i}^{b,k} + \partial_{2i}\Phi^T \left[\boldsymbol{\lambda}_{2i+1}^{b,k} - \boldsymbol{\lambda}_{2i+1}^{b,k-1}\right] = 0, \tag{4.54}$$

$$\boldsymbol{p}_{2i+1}^{b,k} - \boldsymbol{p}_{2i+1}^{r,k} - \left[\boldsymbol{\lambda}_{2i+1}^{b,k} - \boldsymbol{\lambda}_{2i+1}^{b,k-1}\right] = 0, \tag{4.55}$$

$$\boldsymbol{p}_{2i+1}^{b,k} - \boldsymbol{p}_{2i+1}^{r,k} = \Phi(\boldsymbol{u}_{2i} + \boldsymbol{p}_{2i}^{r,k} + (\boldsymbol{p}_{2i}^{b,k} - \boldsymbol{p}_{2i}^{r,k})) \tag{4.56}$$

$$- \boldsymbol{u}_{2i+1} - \boldsymbol{p}_{2i+1}^{r,k}. \tag{4.57}$$

Finally, define new variables $\tilde{\boldsymbol{p}^k} = \boldsymbol{p}^{b,k} - \boldsymbol{p}^{r,k}$ and $\tilde{\boldsymbol{\lambda}}^k = \boldsymbol{\lambda}^{b,k} - \boldsymbol{\lambda}^{b,k-1}$ and $\boldsymbol{u}^{r,k} = \boldsymbol{u} + \boldsymbol{p}^{r,k}$, and we get

$$\tilde{\boldsymbol{p}}_{2i}^{b,k} + \partial_{2i}\Phi^T \tilde{\boldsymbol{\lambda}}_{2i+1}^{b,k} = 0, \tag{4.58}$$

$$\tilde{\boldsymbol{p}}_{2i+1}^{b,k} - \tilde{\boldsymbol{\lambda}}_{2i+1}^{b,k} = 0, \tag{4.59}$$

$$\tilde{\boldsymbol{p}}_{2i+1}^{b,k} = \Phi(\boldsymbol{u}_{2i}^{r,k} + \tilde{\boldsymbol{p}}_{2i}^{b,k}) - \boldsymbol{u}_{2i+1}^{r,k}, \tag{4.60}$$

from which we can eliminate $\tilde{\boldsymbol{\lambda}}^{b,k}$ to get, finally,

$$\tilde{\boldsymbol{p}}_{2i}^{b,k} + \partial_{2i}\Phi^T \left[\Phi(\boldsymbol{u}_{2i}^{r,k} + \tilde{\boldsymbol{p}}_{2i}^{b,k}) - \boldsymbol{u}_{2i+1}^{r,k}\right] = 0, \tag{4.61}$$

$$\boldsymbol{u}_{2i+1}^{b,k} = \Phi(\boldsymbol{u}_{2i}^{r,k} + \tilde{\boldsymbol{p}}_{2i}^{b,k}), \tag{4.62}$$

which we recognize as being the update given by $\mathrm{LSR}_{\mathrm{blk}}$. The same argument also applies to a red update following a black update, with the temporal index shifted by one, such that

$$\tilde{\boldsymbol{p}}_{2i-1}^{r,k} + \partial_{2i-1}\Phi^T \left[\Phi(\boldsymbol{u}_{2i-1}^{b,k-1} + \tilde{\boldsymbol{p}}_{2i-1}^{r,k}) - \boldsymbol{u}_{2i}^{b,k-1}\right] = 0, \tag{4.63}$$

$$\boldsymbol{u}_{2i}^{r,k} = \Phi(\boldsymbol{u}_{2i}^{b,k-1} + \tilde{\boldsymbol{p}}_{2i}^{r,k}), \tag{4.64}$$

and along with the initial iterates $\boldsymbol{u}^{b,0} = \boldsymbol{u}$ and $\boldsymbol{\lambda}^{b,0} = 0$, we get by induction on k that this Gauss-Seidel iteration on the KKT system of the LSS minimization problem with no time dilation is equivalent to the $\mathrm{LSR}_{\mathrm{red}} \circ \mathrm{LSR}_{\mathrm{blk}}$ iteration. ☠

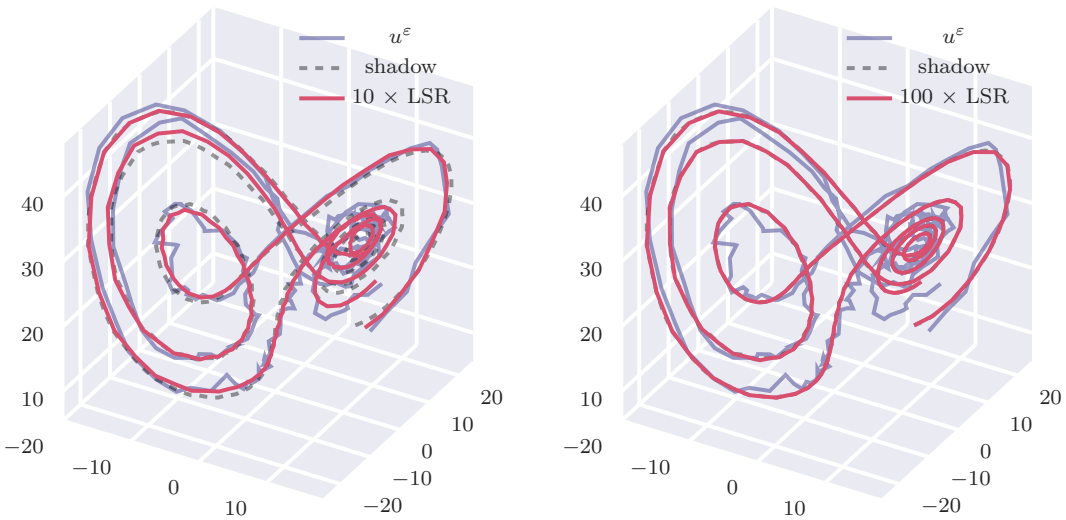


Figure 4.4: The effect of red-black LSR on initially uniform error in the Lorenz system.

Figure 4.4 demonstrates the convergence of red-black LSR to the shadow trajectory of the Lorenz system. Contrasting this result with Figure 4.2, red-black LSR appears to not only converge to the shadow trajectory, it seems to damp highly oscillatory errors in time much faster than smooth errors in time. This indicates that LSR might be a *smoother*, that is, a relaxation which damps high frequency modes of error fast enough that it effectively contracts the error to a lower dimensional subspace consisting of low-frequency modes. This is an excellent property for a relaxation scheme to have in a multigrid setting, since low-frequency modes are precisely what the coarse grid is geometrically good at representing, so we should now confirm that red-black LSR is in fact a smoother.

While I have not yet worked out a formal proof that red-black LSR is a smoother, it is fairly straightforward to see under our simplifying assumptions from earlier. Assume the singular values $\sigma_{i,j}$ of $\partial_i \Phi$ are constant in time and that the singular vectors satisfy the time-dependent orthonormality condition $V_{i+m}^T U_i = I$. Note that we do not have to worry about this assumption here because we already know that

the red-black LSR iteration is stable. Under these assumptions, after multiplication of $J_{\mathrm{LSR}_{\mathrm{red}}} J_{\mathrm{LSR}_{\mathrm{blk}}} J_{\mathrm{LSR}_{\mathrm{red}}}$ and expansion of $\partial_{\boldsymbol{u}_i} \Phi = U \Sigma V^T$, we can see that the fixed-point Jacobian of the symmetric red-black-red iteration is congruent to

$$\mathrm{diag}_j \left(\frac{\sigma_j^2}{\left(1+\sigma_j^2\right)^2} \begin{bmatrix} 2+\sigma_j^{-2} & 1 & & & \\ 1 & 2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 2 & 1 \\ & & & 1 & 2+\sigma_j^2 \end{bmatrix} \otimes \begin{bmatrix} \frac{1}{1+\sigma_j^2} & \frac{\sigma_j}{1+\sigma_j^2} \\ \frac{\sigma_j}{1+\sigma_j^2} & \frac{\sigma_j^2}{1+\sigma_j^2} \end{bmatrix} \right), \quad (4.65)$$

The tridiagonal part on the left of the Kronecker product governs the damping of error vectors in time, and it has eigenvectors $\boldsymbol{e}_\omega$ and eigenvalues ν_ω given by

$$\begin{aligned} \boldsymbol{e}_{0,j,i} &= \sigma_j^{2i}, \\ \nu_{0,j} &= 1, \\ \boldsymbol{e}_{\omega,j,i} &= \left\{ \cos(i\omega) + \frac{\cos(\omega) - \sigma_j^{-2}}{\sin(\omega)} \sin(i\omega) \right\}, \\ \nu_{\omega,j} &= \frac{\sigma_j^2}{\left(1+\sigma_j^2\right)^2} (2 + 2\cos(\omega)), \qquad\qquad \omega = \frac{\pi}{n_t+1}, \frac{2\pi}{n_t+1}, \ldots, \frac{n_t \pi}{n_t+1}, \end{aligned}$$

where ω is the wavenumber. The first few of these eigenvectors are plotted in Figure 4.5 for an unstable, neutral, and stable singular value σ_j. A particular set of eigenvectors of this matrix are the vectors with $\omega = 0$, $\boldsymbol{e}_{0,j} = [1, \sigma_j^2, \sigma_j^2 \ldots, \sigma_j^m]^T$, with eigenvalue 1 of multiplicity n_s. These eigenvectors are the global-in-time version of the local "physical errors" that we saw before, and so it should not be surprising that they are not damped by LSR, since they do not affect the residual. For $\omega > 0$, the eigenvectors are shifted cosine functions with eigenvalues $\nu_\omega < 1$ that vanish as $\sigma_j \to 0$, $\sigma \to \infty$, and $\omega \to \pi$. These vectors represent the global-in-time "unphysical" errors which affect the residual, and we observe that these errors are damped quickly when they are highly oscillatory or when they correspond to very stable or very unstable manifolds. Thus, we have shown that the red-black LSR is a smoother under

these assumptions. These results are very encouraging because red-black LSR seems to damp errors that are particularly difficult to represent accurately on the coarse grid, namely those which correspond to fast transients and which are oscillatory in time.

4.5 Multilevel algorithm and preliminary results

Despite these promising analytical results for LSR, it remains to be seen if the algorithm can be used in a practical setting. It is clear from the eigenvector analysis of red-black LSR (which for the rest of this chapter will just be called LSR) in the previous section that LSR, while stable, is very slow to converge for smooth errors in near-neutral manifolds, for which the convergence rate can be arbitrarily close to 1. However, we have seen evidence that red-black LSR is a smoother, and thus it is particularly well suited for use in multigrid. Following relaxation on the fine-grid with LSR, the remaining error will be smooth, and thus will damp slowly on the fine grid. However, following restriction, the smooth fine grid error will appear more oscillatory on the coarse grid, and will thus be damped more quickly, leaving only smooth error which is able to be accurately represented on an even coarser grid, and so on until the coarsest grid, which is either solved or treated similarly with relaxation. In the case where relaxation is used on every level, including the coarsest grid, we expect to see faster convergence as more grid levels are added. This is because, in contrast to MGRIT, we are not replacing an exact coarse grid solve with an inexact multilevel solve, we are replacing a slow coarse grid relaxation sweep with a more effective multilevel relaxation.

We will now see whether LSR can be used in a V-cycle to improve on the convergence of fine-grid only LSR for the Lorenz system. For the following experiment, the red-black LSR iteration developed in the previous section is used on every level

of an MGRIT V-cycle algorithm in place of FCF-relaxation, where LSR is always followed by F-relaxation in order to compute the FAS τ-correction for the coarse grid. The sequential coarse grid solve is replaced by a single iteration of LSR, and a single iteration of LSR is used on every level for pre- and post-smoothing. The initial guess used for the trajectory is a time stepping solution of the Lorenz system which has been perturbed by significant Gaussian noise. LSR is implemented using NonlinearSolve.jl [40] to solve the nonlinear LSR optimization problem given in Equation (4.5), with multithreading for limited parallelism in time. The Lorenz system is discretized using forward Euler's method, and the same method is used for rediscretization on all coarse grids.

Figure 4.6 shows that, at least initially, adding more coarse grid levels does improve convergence of LSR when used as a smoother in MGRIT. Up to 7 levels, adding 2 more levels to the hierarchy improves overall convergence by roughly one digit of accuracy over 20 iterations. However, convergence degrades when moving to 7 levels, and the iteration begins diverging with 8 levels. Despite these less than perfect results, Figure 4.6 demonstrates more robust convergence with respect to the time domain length than before, with the algorithm converging nearly identically on time domains of $16T_\lambda$ and $32T_\lambda$. Recall that this is four times longer than the time limit of $8T_\lambda$ for robust convergence of classical MGRIT for a chaotic system, so even though the target residual tolerance of $\sqrt{\varepsilon} \approx 10^{-8}$ wasn't reached in this experiment, this is still a very promising result, since it indicates that if these convergence issues can be solved on very coarse grids, LSR is likely to be an effective and scalable parallel solver for chaotic systems.

While I am not sure what exactly is causing the divergence on 7 and 8 levels, there are a few possible culprits. The first is stability of forward Euler's method. The legend in Figure 4.6 records the time-step size h_c on the coarsest grid for a given number of levels, where h_c is colored red for time steps beyond the stability

limit of forward Euler's method. Remarkably, the fastest solver setup tested, with 6 levels, had a coarse grid time-step size which was 4 times this stability limit, meaning that if sequential time stepping were attempted on the coarse grid rather than LSR, the solution would blow up in a single iteration. Even when the algorithm begins to diverge at 8 levels, the rate of divergence is slow, and doesn't seem to indicate numerical instability. Indeed, according to the theory we've developed in the previous sections, LSR should be stable regardless of how unstable Φ_c is. However, it is unknown what effect numerical instability has on the hyperbolic structure of the system. It may be the case that on some coarse grid, there is no longer a splitting of the tangent space into stable and unstable manifolds, which could cause a breakdown of the shadowing lemma. Simply performing this experiment again with an unconditionally stable method should be able to easily rule this out.

Another possibility is that the trajectory becomes so coarse on the coarse grid that it no longer qualitatively resembles the solution on the fine grid, and therefore the shadow trajectory to which LSR is converging differs enough on the fine and coarse grids to cause divergence. Even on a single level, changing the points in time over which the LSS problem is posed changes the solution. There is currently no mechanism in place for correcting this difference, but, the multigrid optimization (MGOPT) framework [36] provides a way to correct the objective function in such a way that ensures the coarse grid optimization has the same solution as the fine grid.

Finally, it may be necessary to add time dilation to the LSR formulation. The shadowing lemma for flows given in Lemma 2 requires time dilation in the form of perturbed time-step sizes η_i in order to deal with the neutral manifold, since otherwise, hyperbolicity is violated and a shadow trajectory may or may not exist. This seems to be a likely explaination for the slowdown of convergence after the initial iterations, but perhaps not for the subsequent divergence. Recall that LSR converges slowest for neutrally stable modes, since they are not transient either

forward or backward in time, and thus they must be resolved on all time scales, from the beginning of the time domain to the end, accurately, unlike the unstable and stable modes. While a formulation of LSR with time dilation exists, the algorithm is not as stable as the one presented here without time dilation, so more work needs to be done to address this.

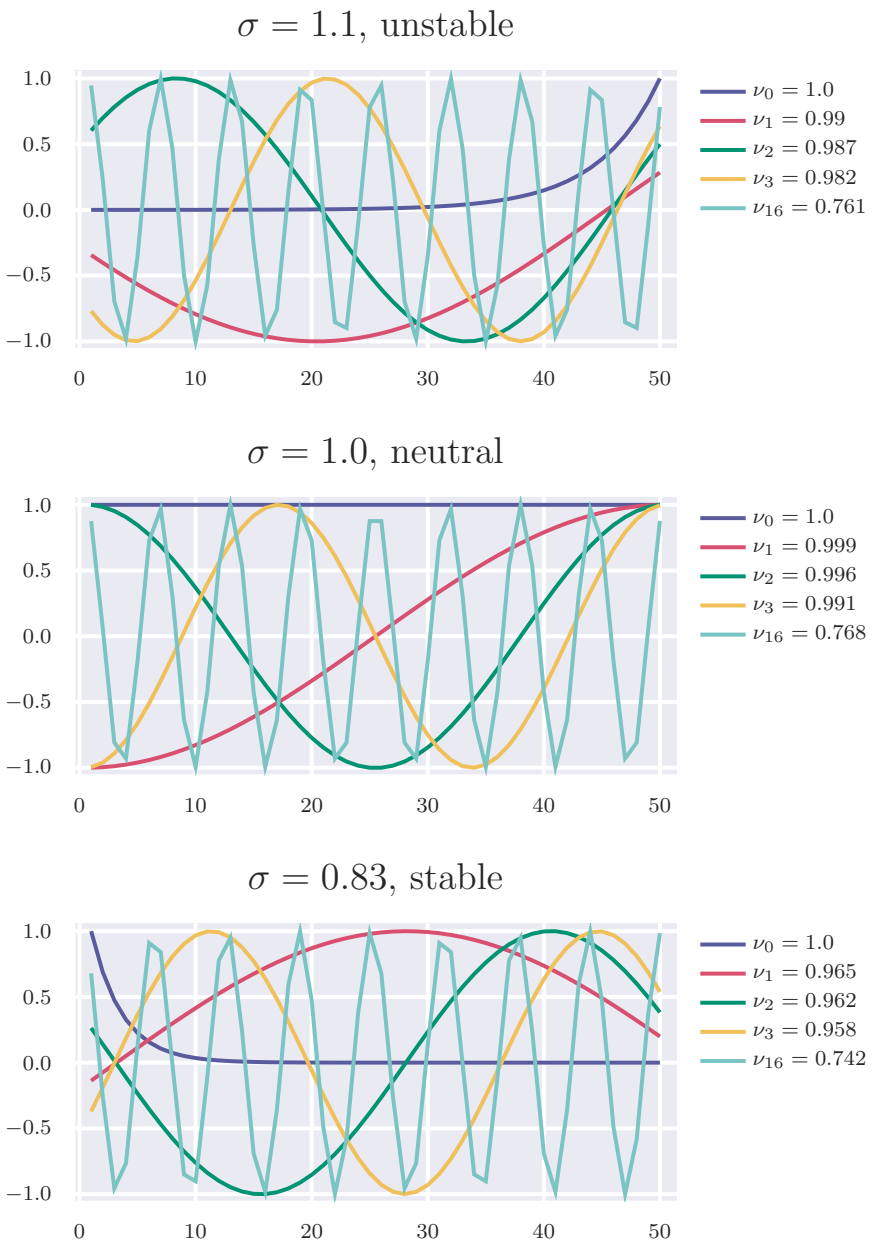


Figure 4.5: The first few eigenvectors and eigenvalues of the red-black LSR fixed-point Jacobian for three different fixed singular values σ.

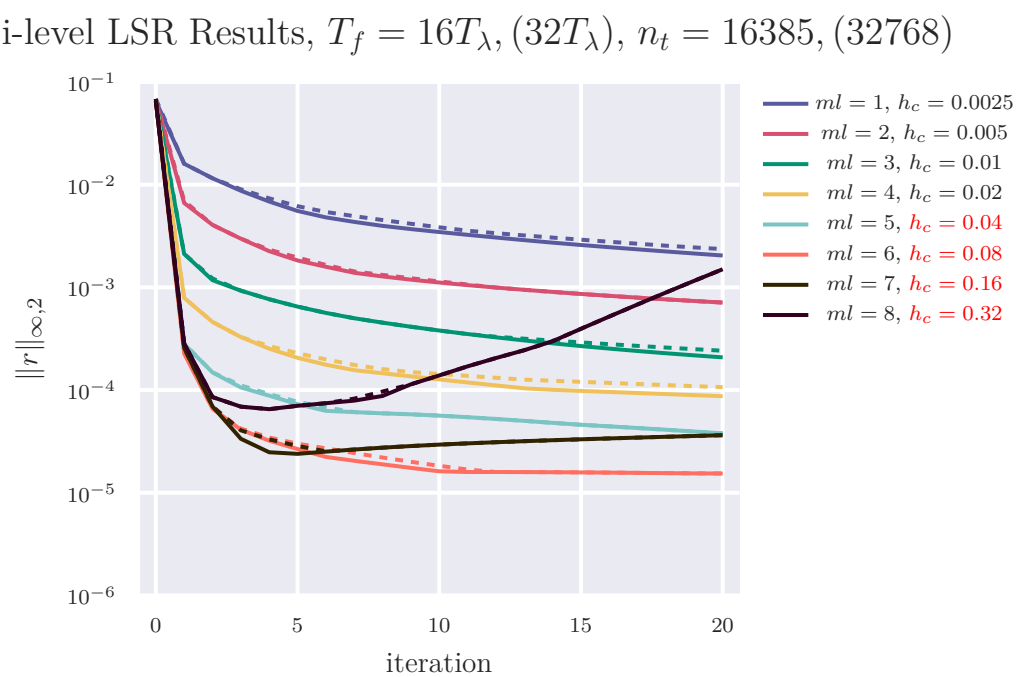


Figure 4.6: MGRIT residual convergence histories for the Lorenz system with increasing number of levels using LSR smoothing and coarse grid 'solve'. Solid line: $T_f = 16T_\lambda$. Dashed line: $T_f = 32T_\lambda$.

Chapter 5

Conclusion

5.1 Summary

As part of this dissertation work, I have developed θ methods, a new approach for deriving stable and accurate coarse grid operators [54], as well as the Δ correction, a modification to the FAS coarse grid equation that targets chaotic systems directly in a general way [53]. Finally, I have developed Local Shadowing Relaxation (LSR) which exhibits robust convergence over very long timescales, well past the theoretical limit for classical MGRIT.

The θ methods make it possible to approximate the fine-grid propagator to arbitrarily-high order on the coarse grid, even when using temporal adaptivity, greatly improving MGRIT convergence in most cases, while also allowing larger coarsening factors which can improve parallel efficiency of the algorithm. I present a sizeable speedup for the space-time parallel solution of a 2D advection diffusion problem in the advection-dominated regime on an adaptive time grid, with the θ methods

nearly doubling the speedup achieved. For Δ correction, the modified coarse grid equation

$$[A_c + \Delta(\boldsymbol{u}^{k-1})]\boldsymbol{u}^k = \boldsymbol{g}_c + \boldsymbol{\tau}(u^{k-1}) + \Delta(\boldsymbol{u}^{k-1})\boldsymbol{u}^{k-1}, \tag{5.1}$$

adds Δ as a linear correction to the coarse grid system operator, in contrast with classical FAS τ-correction which only adds a constant correction to the right-hand-side. This ensures that as the solution converges, the Jacobian of $A_c + \Delta$ converges to the Jacobian of the Schur-complement operator A_*, correcting the Lyapunov spectrum on the coarse grid as the solution converges. I have then developed a low-rank approximation of the full Δ correction which specifically targets and corrects unstable modes of the system while greatly reducing the cost associated with Δ correction. These algorithmic improvements allows MGRIT to reach its theoretical convergence limit of 8 Lyapunov time (T_λ) for the Lorenz system and results in a 10$\times$ PinT speedup for the Kuramoto-Sivashinsky equation, the first such meaningful PinT speedup for a chaotic PDE to my knowledge. Despite this success, MGRIT still breaks down on longer time-domains where convergence stalls.

The work [55] first proposed the use of Least Squares Shadowing (LSS) as a scalable PinT solution method that theoretically overcomes the ill-conditioning of chaotic systems by relaxation the initial condition. Given an approximate trajectory of a chaotic system, LSS attempts to find its *shadow*, a nearby trajectory, with a perturbed initial condition, which is an exact solution to the system. Since the (LSS) problem is well conditioned, independent of the length of the simulation, it stands to reason that a successful PinT algorithm for the solution of the LSS problem could constitute a scalable solver for chaotic systems. Inspired by this idea, and motivated by the extremely fast chaotic timescales that characterize turbulent flows, I show that if posed over a small time-interval, only minimizing over the two endpoints, LSS is equivalent to a minimization of the residual at the terminal point, with respect to a small perturbation at the initial point. The solution of this minimization problem

on many disjoint sub-intervals is trivial, and forms the basis of a PinT relaxation method, which I call Local Shadowing Relaxation (LSR). Following a theoretical treatment of LSR demonstrating its stability, I show that, when it converges, LSR converges to the same shadow trajectory as LSS. Following this, I provide evidence that LSR is a smoother and present very encouraging multilevel results indicating that LSR indeed has robust convergence, independent of the total length of the timescale of the simulation.

5.2 Outlook and Future Work

Ultimately, PinT promise to enable massive scaling and speedups, the kind of scaling that will become increasingly necessary as computing performance improvements in the near future will come from greater parallellism, not improved single core performance. Despite this, many challenges remain for PinT, namely for application to chaotic and hyperbolic systems, although I see these problems as entirely solvable.

Since many systems of interest are weakly chaotic, $8T_\lambda$ is already a fairly significant time domain for systems arising from weather and climate modeling and other non-turbulent CFD applications. It may be the case that existing technology is already sufficient to provide a PinT speedup from some of these methods. Future work should include applying the Δ-correction algorithm to a parabolic chaotic problem targeting climate modeling, such as the viscous rotating shallow water equations, with the goal of running a PinT climate simulation. Although current PinT technology cannot yet handle the strongly advection dominated case, there is possibly enough diffusion in current climate models for MGRIT to converge sufficiently quickly, based on conversations with the E3SM group at Sandia National Laboratories [34]. I also plan to continue developing the SUNDIALS-XBraid interface code to improve θ methods as a black-box, non-intrusive way for users to apply PinT

to their applications, especially for improving convergence for advection dominated problems.

The Δ correction is also attractive in an STMG setting, especially for highly advective problems, where STMG has had more success than MGRIT. The work [50], points out that a time-dependent advection problem in d dimensions appears as a steady state advection problem in $d+1$ dimensions. Thus, the authors use approximate ideal restriction algebraic multigrid (AIR-AMG), designed for steady state advection, to solve 3rd order space-time discretizations of linear advection, with excellent scaling. Following this, [10, 9] introduce space-time block preconditioners based on [7, 8] for the all-at-once solution of incompressible CFD and MHD systems, using an outer Newton's method to handle nonlinearity and solving the linear velocity space-time block with AIR-AMG. Targeting turbulent CFD and MHD, future work will be to develop all-at-once STMG solvers for turbulent flows based on space-time block preconditioning.

While STMG methods based on Newton's method and AIR-AMG work well for non-turbulent advective problems, they will not work for turbulent systems. This is because the time-evolution of the linearization of a chaotic system is unstable by definition, thus Newton's method is unstable on any appreciable timescale. This mirrors the inability of MGRIT to converge past a certain time limit for a chaotic system, but worse, because with Newton's method there is no nonlinear mechanism to stabilize the Newton correction, so it may grow exponentially fast without bound. Of course, this is even worse for a turbulent system which has very fast chaotic timescales. However, Δ-correction converges quadratically with a non-linear coarse grid equation based on FAS and exhibits robust convergence for chaotic problems, so a generalization of Δ correction to the STMG setting is called for.

LSR could be very useful for turbulent systems, as it would resolve the small space-time scales that have previously been ignored as in [25] entirely through PinT

relaxation. This is especially promising coupled with Full Multigrid (FMG), since a coarse initial guess for the whole trajectory could be produced with a single FMG cycle. Ultimately, if LSR is successful, it would represent be a monumental development in PinT technology that could be directly applicable to many important application areas.

References

[1] Achi Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333 – 390, 1977.

[2] Williams L. Briggs, Van Emden Henson, and Steve F. McCormick. *A Multigrid Tutorial*. SIAM, 2000.

[3] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations.* John Wiley & Sons, Incorporated, 2016.

[4] Nisha Chandramoorthy and Qiqi Wang. On the probability of finding nonphysical solutions through shadowing. *Journal of Computational Physics*, 440:110389, 2021.

[5] Andrew J. Christlieb, Colin B. Macdonald, and Benjamin W. Ong. Parallel high-order integrators. *SIAM Journal on Scientific Computing*, 32(2):818–835, 2010.

[6] R. M. Corless. What good are numerical simulations of chaotic dynamical systems?. In *COMPUTERS AND MATHEMATICS WITH APPLICATIONS*, volume 28, page 107, 1994.

[7] Eric C. Cyr, John N. Shadid, and Raymond S. Tuminaro. Stabilization and scalable block preconditioning for the Navier–Stokes equations. *Journal of Computational Physics*, 231(2):345–363, 2012.

[8] Eric C. Cyr, John N Shadid, Raymond S. Tuminaro, Roger P. Pawlowski, and Luis Chacón. A new approximate block factorization preconditioner for two-dimensional incompressible (reduced) resistive MHD. *SIAM Journal on Scientific Computing*, 35(3):B701–B730, 2013.

[9] Federico Danieli, Ben S. Southworth, and Jacob B. Schroder. Space-time block preconditioning for incompressible resistive magnetohydrodynamics, 2023.

[10] Federico Danieli, Ben S. Southworth, and Andrew J. Wathen. Space-time block preconditioning for incompressible flow. *SIAM Journal on Scientific Computing*, 44(1):A337–A363, 2022.

[11] Timothy A. Davis. Algorithm 832: Umfpack v4.3—an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Softw.*, 30(2):196–199, jun 2004.

[12] Hans De Sterck, Robert D. Falgout, Stephanie Friedhoff, Oliver A. Krzysik, and Scott P. MacLachlan. Optimizing multigrid reduction-in-time and parareal coarse-grid operators for linear advection. *Numerical Linear Algebra with Applications*, 28(4):e2367, 2021.

[13] Hans De Sterck, Robert D. Falgout, and Oliver A. Krzysik. Fast multigrid reduction-in-time for advection via modified semi-Lagrangian coarse-grid operators. *SIAM Journal on Scientific Computing*, 45(4):A1890–A1916, 2023.

[14] Matthew Emmett and Michael Minion. Toward an efficient parallel in time method for partial differential equations. *Communications in Applied Mathematics and Computational Science*, 7(1):105–132, 2012.

[15] R. D. Falgout, S. Friedhoff, Tz. V. Kolev, S. P. MacLachlan, and J. B. Schroder. Parallel time integration with multigrid. *SIAM J. Sci. Comput.*, 36(6):C635–C661, 2014. LLNL-JRNL-645325.

[16] R. D. Falgout and J. B. Schroder. Parallel time integration – an approaching paradigm shift for scientific computing. Technical report, Lawrence Livermore National Laboratory, 2023. LLNL-TR-851068.

[17] Pablo Fernandez and Qiqi Wang. Lyapunov spectrum of scale-resolving turbulent simulations. application to chaotic adjoints. In *23rd AIAA Computational Fluid Dynamics Conference*, Denver, Colorado, 2017.

[18] Itzhak Fouxon, Joshua Feinberg, Petri Käpylä, and Michael Mond. Reynolds number dependence of Lyapunov exponents of turbulence and fluid particles. *Physical Review E*, 103(3):033110, 2021.

[19] Stephanie Friedhoff and Ben S. Southworth. On "optimal" h-independent convergence of parareal and multigrid-reduction-in-time using Runge-Kutta time integration. *Numerical Linear Algebra with Applications*, 28(3):e2301, 2021.

[20] M. J. Gander and S. Vandewalle. Analysis of the parareal time-parallel time-integration method. *SIAM J. Sci. Comput.*, 29(2):556–578, 2007.

[21] Martin J. Gander. *50 Years of Time Parallel Time Integration.*, volume 9 of *Contributions in Mathematical and Computational Sciences. 9.* Springer International Publishing, 2015.

[22] Martin J. Gander and Ernst Hairer. Nonlinear convergence analysis for the parareal algorithm. In Ulrich Langer, Marco Discacciati, David E. Keyes, Olof B. Widlund, and Walter Zulehner, editors, *Domain Decomposition Methods in Science and Engineering XVII*, pages 45–56, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[23] Martin J Gander, Jun Liu, Shu-Lin Wu, Xiaoqiang Yue, and Tao Zhou. Paradiag: Parallel-in-time algorithms based on the diagonalization technique. *arXiv preprint arXiv:2005.09158*, 2020.

[24] David J Gardner, Daniel R Reynolds, Carol S Woodward, and Cody J Balos. Enabling new flexibility in the SUNDIALS suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, 48(3):1–24, 2022.

[25] Stephen M. Guzik, Joshua Christopher, Sean Walters, Xinfeng Gao, Jacob B. Schroder, and Robert D. Falgout. On the use of a multigrid-reduction-in-time algorithm for multiscale convergence of turbulence simulations. *Computers & Fluids*, 261:105910, 2023.

[26] Stephan M. Hammel, James A. Yorke, and Celso Grebogi. Numerical orbits of chaotic processes represent true orbits. *Bulletin (New Series) of the American Mathematical Society*, 19(2):465 – 469, 1988.

[27] Alan C Hindmarsh, Peter N Brown, Keith E Grant, Steven L Lee, Radu Serban, Dan E Shumaker, and Carol S Woodward. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):363–396, 2005.

[28] Alexander J. Howse, Hans De Sterck, Robert D. Falgout, Scott MacLachlan, and Jacob Schroder. Parallel-in-time multigrid with adaptive spatial coarsening for the linear advection and inviscid burgers equations. *SIAM Journal on Scientific Computing*, 41(1):A538–A565, 2019.

[29] *hypre*: High performance preconditioners. `https://llnl.gov/casc/hypre`, `https://github.com/hypre-space/hypre`.

[30] Venkataramanaiah Krishnamurthy. Predictability of weather and climate. *Earth and Space Science*, 6(7):1043–1056, 2019.

[31] P. V. Kuptsov and U. Parlitz. Theory and computation of covariant Lyapunov vectors. *JOURNAL OF NONLINEAR SCIENCE*, 22(5):727 – 762, 2012.

[32] Yoshiki Kuramoto. Diffusion-induced chaos in reaction systems. *Progress of Theoretical Physics Supplement*, 64:346–367, 1978.

[33] Richard E Ladner and Michael J Fischer. Parallel prefix computation. *Journal of the ACM (JACM)*, 27(4):831–838, 1980.

[34] L. Ruby Leung, David C. Bader, Mark A. Taylor, and Renata B. McCoy. An introduction to the E3SM special collection: Goals, science drivers, development, and analysis. *Journal of Advances in Modeling Earth Systems*, 12(11):e2019MS001821, 2020. e2019MS001821 2019MS001821.

[35] Thibaut Lunet, Julien Bodart, Serge Gratton, and Xavier Vasseur. Time-parallel simulation of the decay of homogeneous turbulence using parareal with spatial coarsening. *Computing and Visualization in Science*, 19(1-2):31 – 44, 2018.

[36] Stephen G Nash. A multigrid approach to discretized optimization problems. *Optimization Methods and Software*, 14(1-2):99–116, 2000.

[37] J. Nievergelt. Parallel methods for integrating ordinary differential equations. *Commun. ACM*, 7(12):731–733, dec 1964.

[38] SP Nørsett. On the theory of parallel Runge—Kutta methods. *IMA Journal of numerical Analysis*, 10(4):463–488, 1990.

[39] Benjamin W. Ong and Jacob B. Schroder. Applications of time parallelization. *Computing and Visualization in Science*, 23:1,10, 2020.

[40] Avik Pal, Flemming Holtorf, Axel Larsson, Torkel Loman, Frank Schaefer, Qingyu Qu, Alan Edelman, Chris Rackauckas, et al. Nonlinearsolve.jl: High-performance and robust solvers for systems of nonlinear equations in julia. *arXiv preprint arXiv:2403.16341*, 2024.

[41] Kenneth James Palmer. *Shadowing in dynamical systems: theory and applications*, volume 501. Springer Science & Business Media, 2000.

[42] Sergei Yu. Pilyugin. *Numerical applications of shadowing*, pages 219–257. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[43] Daniel R Reynolds, David J Gardner, Carol S Woodward, and Rujeko Chinomona. ARKODE: A flexible IVP solver infrastructure for one-step methods. *ACM Transactions on Mathematical Software*, 49(2):1–26, 2023.

[44] J.M. Reynolds-Barredo, D.E. Newman, R. Sanchez, D. Samaddar, L.A. Berry, and W.R. Elwasif. Mechanisms for the convergence of time-parallelized, parareal turbulent plasma simulations. *Journal of Computational Physics*, 231(23):7851–7867, 2012.

[45] Manfred Ries, Ulrich Trottenberg, and Gerd Winter. A note on mgr methods. *Linear Algebra and its Applications*, 49:1–26, 1983.

[46] David Ruelle. Microscopic fluctuations and turbulence. *Physics Letters A*, 72(2):81–82, 1979.

[47] Karl Rupp. 50 years of microprocessor trend data. Data Set, 02 2022.

[48] D. Samaddar, D.E. Newman, and R. Sánchez. Parallelization in time of numerical simulations of fully-developed plasma turbulence using the parareal algorithm. *Journal of Computational Physics*, 229(18):6558–6573, 2010.

[49] J.B. Schroder. On the use of artificial dissipation for hyperbolic problems and multigrid reduction in time (MGRIT). Technical report, Lawrence Livermore National Lab, 2018. LLNL-TR-750825.

[50] A. A. Sivas, B. S. Southworth, and S. Rhebergen. AIR algebraic multigrid for a space-time hybridizable discontinuous Galerkin discretization of advection(-diffusion). *SIAM Journal on Scientific Computing*, 43(5):A3393–A3416, 2021.

[51] Steven H. Strogatz. *Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry, and Engineering*. studies in nonlinearity. Addison-Wesley, 1994.

[52] N. A. Petersson V. Dobrev, Tz. Kolev and J. B. Schroder. Two-level convergence theory for multigrid reduction in time (MGRIT). *SIAM J. Sci. Comput.*, 39(5):S501–S527, 2017.

[53] D. A. Vargas, R. D. Falgout, S. Günther, and J. B. Schroder. Multigrid reduction in time for chaotic dynamical systems. *SIAM Journal on Scientific Computing*, 45(4):A2019–A2042, 2023.

[54] D. A. Vargas, R. D. Falgout, S. Günther, and J. B. Schroder. Improved coarse-grid methods for multigrid reduction in time with Runge-Kutta θ methods. *SIAM Journal on Scientific Computing*, 2024.

[55] Qiqi Wang, Steven A. Gomez, Patrick J. Blonigan, Alastair L. Gregory, and Elizabeth Y. Qian. Towards scalable parallel-in-time turbulent flow simulations. *Physics of Fluids*, 25(11):110818, 2013.

[56] Qiqi Wang, Rui Hu, and Patrick Blonigan. Least squares shadowing sensitivity analysis of chaotic limit cycle oscillations. *Journal of Computational Physics*, 267:210–224, 2014.

[57] XBraid: Parallel multigrid in time v3.0. `http://llnl.gov/casc/xbraid`.

[58] Irad Yavneh. Coarse-grid correction for nonelliptic and singular perturbation problems. *SIAM JOURNAL ON SCIENTIFIC COMPUTING*, 19(5):1682 – 1699, 1998.

[59] Irad Yavneh and Gregory Dardyk. A multilevel nonlinear method. *SIAM Journal on Scientific Computing*, 28(1):24 – 46, 2006.