Seeking Space Aliens and the Strong Approximation Property: A (disjoint) Study in Dust Plumes on Planetary Satellites and Nonsymmetric Algebraic Multigrid

by

Benjamin Scott Southworth

B.A. Mathematics, Dartmouth College, 2013

M.S. Applied Mathematics, University of Colorado, 2015

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Applied Mathematics

2017

This thesis entitled:
Seeking Space Aliens and the Strong Approximation Property: A (disjoint) Study in Dust Plumes on Planetary Satellites and Nonsymmetric Algebraic Multigrid
written by Benjamin Scott Southworth
has been approved for the Department of Applied Mathematics

Professor Thomas Manteuffel

Professor Sascha Kempf

Professor Steve McCormick

Dr. Jacob Schroder

Dr. John Ruge

Professor Stephen Becker

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Southworth, Benjamin Scott (Ph.D., Applied Mathematics)

Seeking Space Aliens and the Strong Approximation Property: A (disjoint) Study in Dust Plumes on Planetary Satellites and Nonsymmetric Algebraic Multigrid

Thesis directed by Professor Thomas Manteuffel and Professor Sascha Kempf

PART I: One of the most fascinating questions to humans has long been whether life exists outside of our planet. To our knowledge, water is a fundamental building block of life, which makes liquid water on other bodies in the universe a topic of great interest. In fact, there are large bodies of water right here in our solar system, underneath the icy crust of moons around Saturn and Jupiter. The NASA-ESA Cassini Mission spent two decades studying the Saturnian system. One of the many exciting discoveries was a "plume" on the south pole of Enceladus, emitting hundreds of kg/s of water vapor and frozen water-ice particles from Enceladus' subsurface ocean. It has since been determined that Enceladus likely has a global liquid water ocean separating its rocky core from icy surface, with conditions that are relatively favorable to support life. The plume is of particular interest because it gives direct access to ocean particles *from space*, by flying through the plume. Recently, evidence has been found for similar geological activity occurring on Jupiter's moon Europa, long considered one of the most likely candidate bodies to support life in our solar system. Here, a model for plume-particle dynamics is developed based on studies of the Enceladus plume and data from the Cassini Cosmic Dust Analyzer. A C++, OpenMP/MPI parallel software package is then built to run large scale simulations of dust plumes on planetary satellites. In the case of Enceladus, data from simulations and the Cassini mission provide insight into the structure of emissions on the surface, the total mass production of the plume, and the distribution of particles being emitted. Each of these are fundamental to understanding the plume and, for Europa and Enceladus, simulation data provide important results for the planning of future missions to these icy moons. In particular, this work has contributed to the Europa Clipper mission and proposed Enceladus Life Finder.

PART II: Solving large, sparse linear systems arises often in the modeling of biological and physical phenomenon, data analysis through graphs and networks, and other scientific applications. This work focuses primarily on linear systems resulting from the discretization of partial differential equations (PDEs). Because solving linear systems is the bottleneck of many large simulation codes, there is a rich field of research in developing "fast" solvers, with the ultimate goal being a method that solves an $n \times n$ linear system in $O(n)$ operations. One of the most effective classes of solvers is algebraic multigrid (AMG), which is a multilevel iterative method based on projecting the problem into progressively smaller spaces, and scales like $O(n)$ or $O(n \log n)$ for certain classes of problems. The field of AMG is well-developed for symmetric positive definite matrices, and is typically most effective on linear systems resulting from the discretization of scalar elliptic PDEs, such as the heat equation. Systems of PDEs can add additional difficulties, but the underlying linear algebraic theory is consistent and, in many cases, an elliptic system of PDEs can be handled well by AMG with appropriate modifications of the solver. Solving general, nonsymmetric linear systems remains the wild west of AMG (and other fast solvers), lacking significant results in convergence theory as well as robust methods. Here, we develop new theoretical motivation and practical variations of AMG to solve nonsymmetric linear systems, often resulting from the discretization of hyperbolic PDEs. In particular, multilevel convergence of AMG for nonsymmetric systems is proven for the first time. A new nonsymmetric AMG solver is also developed based on an approximate ideal restriction, referred to as AIR, which is able to solve advection-dominated, hyperbolic-type problems that are outside the scope of existing AMG solvers and other fast iterative methods. AIR demonstrates scalable convergence on unstructured meshes, in multiple dimensions, and with high-order finite elements, expanding the applicability of AMG to a new class of problems.

Dedication

To all of my teachers and mentors along the way.

“If a mistake is found, the reader should substitute what was originally intended.”

Acknowledgements

I want to extend my gratitude to all of the teachers and professors I have had that encouraged and supported my interest in mathematics, through grade school, college, and graduate school. At Dartmouth, I discovered how stimulating research can be while working with Professors Dan Rockmore, Alex Barnett, Dorothy Wallace, and Brenden Epps, and learned that there can be a future in just studying mathematics (as opposed to moving to the dark side of, for example, engineering). While working towards my PhD, I have been fortunate enough to continue working with great scientists and great people, which has made the entire experience very enjoyable. My advisors, Tom and Sascha, are both excellent academic mentors, as well as people who I admire and consider role models. Through working with them, I have also also been able to meet and collaborate with many of their colleagues. In no particular order, I have appreciated working with Jacob Schroder, Rob Falgout, and friends, at Lawrence Livermore National Lab; officially retired and Professor Emeritus, Steve McCormick, permanent post-doc and AMG-whisperer, John Ruge; our German friend, Steffen Münzenmaier; Joe Spitale, of the Planetary Sciences Institute; Jürgen Schmidt, at the University of Oulu; and many others, indirectly. I would also like to thank Stephen Becker for being on my committee even though we didn't get the chance to work together, and Mihály Horányi for general guidance as I started my graduate career. Finally, I would like to thank my wife, Laura Spector, for her continued support while she works towards a PhD in genetics, and my family for their enthusiasm about my research, even if they don't understand it.

Contents

Tables

Table

Figures

Figure

PART I: Modeling dust plumes on planetary satellites

Chapter 1

Model and software development

1.1 Dust plumes

1.1.1 What are plumes and why do we care?

Human's have long been asking the question: "are we alone?" Is Earth unique in its supporting of life and biological activity? To our knowledge, liquid water is one of the fundamental building blocks of life, so often when we search for life outside of Earth, we search for liquid water on other bodies in the universe. As it turns out, we don't have to look very far: Enceladus and Europa, large, icy moons around Saturn and Jupiter, respectively, harbor large bodies of liquid water underneath their icy crust. Europa is actually expected to have more liquid water than we have here on Earth, and the subsurface ocean on Enceladus is believed to be a global ocean, separating Enceladus' rocky core from its icy surface [174]. Such moons are of great interest in the search for biological activity outside of Earth, in particular because they are right here in our solar system, so we can send spacecraft to study the environments and search for indications of life. The NASA-ESA Cassini Mission spent 20 years studying the Saturnian system, including Enceladus, making numerous exciting discoveries. In the early 2020s, the NASA Clipper mission will head to Europa for a detailed reconnaissance of conditions there, and the proposed Enceladus Life Finder will (hopefully) return to Enceladus in search of life.

So what is a dust plume? A dust plume can be any cryovolcanic activity emitting "dust particles." Here, we are particularly interested in plumes that consist of water vapor and frozen water-ice particles, originating in subsurface oceans on the icy moons of Enceladus and Europa. As early as the 1980s, there was speculation of cryovocanic activity on Enceladus [132], but it was not until 2005 that the NASA-ESA Cassini mission made the exciting discovery of a plume of particles erupting from the south polar terrain of Enceladus [51, 66, 134, 165, 167, 181]. Interestingly, the plume is located on Enceladus' south pole, but anomalies were actually detected that led to its discovery from a north polar flyby. Over the course of the Cassini mission, multiple spacecraft flybys of Enceladus and traversals through the plume allowed Cassini in-situ instruments to collect important data on the plume.

Much research has been devoted to understanding the Enceladus plume and its driving mechanism. There is convincing evidence that the plume is by far the strongest source of E-ring particles [for example, 77, 164] and also the dominant source of the resurfacing of Enceladus [for example, 89, 91].Arguably, the most exciting result is that Enceladus plume emissions are believed to be connected to the moon's global, subsurface ocean [79, 83, 137, 138, 151, 174], emitting water vapor and frozen water-ice particles, some of which are large ice particles that originate at the ocean's boiling surface [138]. Recent Hubble Space Telescope observations [144] gave evidence for similar, possibly intermittent, active water-vapor plumes on Europa. An alternative explanation for the observations was provided in Shemansky et al. [156]); however, follow up observations in Sparks et al. [166] provide further evidence of episodic plume activity on Europa. Episodic plume activity on Europa was also suggested previously to create surface features seen in Galileo images [95, 140]. Europa is widely considered one of the likeliest candidate bodies in our solar system able to support biological activity [155], so potential plume activity there is of great interest.

The perspective that plumes on Enceladus and, potentially, Europa are fed by a subsurface ocean offers an opportunity to study the oceans' composition and investigate their habitability. Due to the association of liquid water with life, biological activity would most likely be found in an ocean beneath these moons' icy crusts. The plumes offer a unique opportunity to directly sample subsurface material *in situ* from space. This work develops a model for plume-particle dynamics after particles have been emitted, and a parallel, C++, software package to run large-scale plume simulations. The purpose here is two-fold. First, data on the Enceladus plume gathered during the Cassini mission is used to constrain our model and resolve our understanding of the physical mechanism of the plume. Second, simulation data and software is publicly available for future projects, and has been used in the planning and development of dust detector instruments for the NASA Europa Clipper Mission and proposed Enceladus Life Finder.

1.1.2 Models of plume dynamics

When the Enceladus plume was discovered by Cassini in 2005, it was not known that Enceladus had a liquid water ocean, nor that the plume originated there. Due to the excitement about cryovalcanic activity on Enceladus, a number of models were proposed to explain the plume's physical mechanism. In particular, there were three primary theories developed to explain the origin of plumes on Enceladus, termed *cold faithful*, *frigid faithful*, and *deep source*. Over the course of the Cassini mission, data was gathered from multiple instruments that, altogether, indicates two of the models are inconsistent with observations, and suggests that the deep-source plume model is an accurate description of subsurface plume dynamics.

The *cold faithful* model [87, 134] is based on boiling explosions of water in shallow ice pockets, ~ 10 m beneath the surface, suddenly exposed to vacuum. This process, however, would be self-limiting, and thermodynamic arguments [30] imply that the resulting gas velocities would be fairly low, way below the high gas velocities seen on Enceladus [68] and Europa [144]. Furthermore, such a mechanism is inconsistent with the stratification of plume particles by size and chemical species observed at different altitudes above the surface of Enceladus Postberg et al. [138]. The *frigid faithful* model [93] assumes that the venting on Enceladus is driven by the decomposition of CO_2, CH_4, and N_2 clathrates located underneath an H_2O-CO_2 ice shell. The model favors temperatures around 140K. At such low temperatures, the gas in the vents is very dilute, and transporting ice grains in the flow at the high observed rates is difficult in any non-straight vent geometry [30, 31, 151]. A plume driven by the decomposition of clathrates would also result in high concentrations of clathrates, which is inconsistent with data on the chemical composition of plume particles on Enceladus [68, 138]. The *deep source* model [151] is based on fractures in the icy crust extending down to a large, liquid water reservoir. A vacuum effect leads to back-pressurized vents that emit a gas-particle flow, evaporating at depth from a reservoir at temperatures close to or at the triple point of water. In this case the ice particles, in part, may condense from the supersaturated vapor stream in the vents [31, 31, 151]. Another population, exhibiting a salinity of about 1%, are frozen droplets [138], forming above a liquid reservoir at depth, freezing in the vents. These frozen droplets are deemed "large" particles in the context of a plume, and particularly interesting as they are direct ocean particles emitted at high velocities that can be collected by spacecraft.

Here, we build on the deep-source plume model of subsurface particle dynamics to model particle dynamics *after* ejection. The deep-source model and data from the Cassini mission are used to motivate an initial particle state upon ejection, and particles are then integrated until they either establish a stable orbit about the central planet, or re-collide with the surface of the moon. Details on the model, sampling of the initial state vector, and software package are provided in Section 1.2.

1.2 Software package

Particle simulations are performed using an MPI/OpenMP hybrid parallel software package developed in C++11. Simulations of a plume source output binary files with data on plume-particle residence times in a discretized grid about the satellite and the particle states upon collision with the surface. Data is post-processed and visualized in IDL; HDF5 files of the post-processed surface-collision and plume-density data are available. The C++11 software package is an object-oriented C++ code, with two main classes, *Solver* and *Jet*, a namespace, *genFunctions*, and driver scripts to run simulations. This section introduces the software packages developed, detailing the underlying model and its numerical implementation.

1.2.1 Coordinate systems and transformations

Particle trajectories are integrated in a planet-centered quasi-inertial coordinate system, assuming ejection at 00:00, January 1, 2014 with respect to the J2000 epoch; denote this time t_0 and the coordinate frame $\mathcal{F}_1$. Spice [121] provides the position and velocity of the moon, right ascension, and declination, with respect to the parent body at time t_0 in the J2000 coordinate system (denoted $\mathbf{r}^M_{J2000}$, $\mathbf{v}^M_{J2000}$, RA^M_{J2000}, and δ^M_{J2000}, respectively). The pole of the moon has direction vector in J2000

$$\mathbf{p}^M_{J2000} = \Big(\cos(\delta^M_{J2000})\cos(\text{RA}^M_{J2000}), \cos(\delta^M_{J2000})\sin(\text{RA}^M_{J2000}), \sin(\delta^M_{J2000})\Big).$$

TransformSystem(...) then transforms $\mathbf{r}^M_{J2000}$, $\mathbf{v}^M_{J2000}$, and $\mathbf{p}^M_{J2000}$ to $\mathcal{F}_1$, where the xy-plane of the $\mathcal{F}_1$ corresponds to the moon's orbital plane and the moon's initial position is given by $(x, 0, 0)$. Given the moon's position and velocity in J2000, respectively, the angular momentum is given by $L := \mathbf{r}^M_{J2000} \times \mathbf{v}^M_{J2000}$. To initialize the moon orbiting in the xy-plane, a basis vector in z is taken to be the normalized angular momentum vector, and a basis vector in x is built orthogonal to angular momentum and $\mathbf{e}_z$ in J2000. An orthonormal y-basis vector is constructed to complete the basis :

$$\mathbf{b}^{(1)}_z := \frac{L}{\|L\|}, \quad \mathbf{b}^{(1)}_x := \frac{L \times \mathbf{e}_z}{\|L \times \mathbf{e}_z\|}, \quad \mathbf{b}^{(1)}_y := \mathbf{b}^{(1)}_z \times \mathbf{b}^{(1)}_x.$$

We then rotate this coordinate system to initialize the moon at position $(x, 0, 0)$ using a rotation matrix in the xy-plane. Assuming basis vectors are stored as row vectors, vectors in J2000 are transformed to $\mathcal{F}_1$ by applying the transformation, for example,

$$\mathbf{r}^M_{\mathcal{F}_1} \hookleftarrow \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{b}^{(1)}_x \\ \mathbf{b}^{(1)}_y \\ \mathbf{b}^{(1)}_z \end{pmatrix} \mathbf{r}^M_{J2000},$$

where θ is the angle of rotation (easily computable, complicated to write in terms of $\mathbf{r}^M_{J2000}$). Moving forward, vectors without subscripts correspond to $\mathcal{F}_1$ unless otherwise specified.

Although particles are integrated in a planet-centered quasi inertial frame, plume source location, plume density profiles and surface collision data all use a moon-centered quasi-interial frame. A coordinate transformation is constructed in *SetChangeBasis(...)* or *SetInitCond(...)* of the Jet class (see Section 1.2.3) that maps from the planet-centered frame to a moon-centered frame with the $+y$-axis pointing the direction of rotation and $-x$-axis pointing towards Saturn; denote this coordinate frame $\mathcal{F}_2$. The z-axis will be the same in $\mathcal{F}_2$ as in $\mathcal{F}_1$. A y-basis vector is then constructed orthogonal to the plane containing the z-axis and the position of the moon, corresponding to the direction of the moon's orbit (because $\mathbf{b}^{(2)}_z$ is the direction of angular momentum), and an x-basis vector is constructed by orthonormalization to complete the set:

$$\mathbf{b}^{(2)}_z = (0, 0, 1), \quad \mathbf{b}^{(2)}_y = \frac{\mathbf{b}^{(2)}_z \times \mathbf{r}^M}{\|\mathbf{b}^{(2)}_z \times \mathbf{r}^M\|}, \quad \mathbf{b}^{(2)}_x = \frac{\mathbf{b}^{(2)}_y \times \mathbf{b}^{(2)}_z}{\|\mathbf{b}^{(2)}_y \times \mathbf{b}^{(2)}_z\|}.$$

Particle position $\mathbf{r}$ in $\mathcal{F}_1$ is then transformed to $\mathcal{F}_2$ by taking the difference of $\mathbf{r}$ and $\mathbf{r}_M$ and applying the transformation:

$$\mathbf{r}_{(\mathcal{F}_2)} \hookleftarrow \begin{pmatrix} \mathbf{b}^{(2)}_x \\ \mathbf{b}^{(2)}_y \\ \mathbf{b}^{(2)}_z \end{pmatrix} \left(\mathbf{r} - \mathbf{r}^M\right).$$

Due to orthonormal basis vectors, transferring from $\mathcal{F}_2$ back to $\mathcal{F}_1$ can be seen by multiplying by the change-of-basis adjoint (i.e., inverse) and adding the position of Enceladus.

Above, $\mathcal{F}_2$ is in Euclidean coordinates; however, locations on the moon for plume sources as well as the surface deposition profiles considered later are given in planetographic coordinates. Here we assume a spherical moon, as this is sufficiently accurate for simulation purposes and simpler from an implementation perspective. Given longitude and latitude for some plume source, (λ, ϕ), respectively, a unit-norm direction vector for the source location in Euclidean coordinates $(\mathcal{F}_2)$ is given as $\mathbf{s} = (\cos(\lambda)\cos(\phi), \sin(\lambda)\cos(\phi), \sin(\phi))^T$. Note that here, it is important that the moon's prime meridian lies in the xz-plane of $\mathcal{F}_2$ and the moon orbits in the xy-plane so that longitude and latitude are well defined as simple translations of spherical coordinates. Let R_M be the radius of the moon; the plume location and, thus, the initial particle position in $\mathcal{F}_1$ is then given by

$$\mathbf{r} = \mathbf{r}^M + R_M \begin{pmatrix} \mathbf{b}_x^{(2)} \\ \mathbf{b}_y^{(2)} \\ \mathbf{b}_z^{(2)} \end{pmatrix}^T \begin{pmatrix} \cos(\lambda)\cos(\phi) \\ \sin(\lambda)\cos(\phi) \\ \sin(\phi) \end{pmatrix}.$$

Finally, each particle has an initial velocity, consisting of three parts: the velocity of the moon, the rotational velocity of the moon at the source location, and the initial launch velocity from the plume. Accounting for moon's velocity is straightforward. For the rotational velocity, according to Kepler's third law, the moon's orbital period about the planet is given by $\tau = 2\pi\sqrt{\frac{a^3}{\mu_P}}$, where a is the moon's semi-major axis and μ_P the planet's standard gravitational parameter. As previously, let $\mathbf{s}(\lambda, \phi)$ be a unit direction vector for a plume location with longitude λ and latitude ϕ. Because the moon is orbiting, as well as rotating, in the xy-plane, the tangential direction of a launched particle is $\mathbf{t} = \mathbf{e}_z \times \mathbf{s}$, and the tangential velocity imparted on a particle given by

$$\mathbf{v}_\tau := \cos(\phi) R^M \omega \mathbf{t},$$

where

$$\omega := \frac{2\pi}{\tau} = \sqrt{\frac{\mu_P}{a^3}}$$

is the angular speed of the moon's rotation at latitude ϕ. Note that the speed depends on the latitude – for example, at the south pole, $\cos(\phi) = 0$ and, thus, $\mathbf{v}_\tau = \mathbf{0}$ because there is no rotational velocity.

Last, we account for the initial particle velocity from the plume; the initial speed is discussed in Section 1.2.3 and here we account for the direction of ejection. Proposed sources of plume emissions are typically directed at some angle from the surface, not necessarily orthogonal [135, 168]. If the jet azimuthal angle is zero, the jet direction in $\mathcal{F}_1$ is equivalent to the source location, $\mathbf{d} = \mathbf{s}$. Otherwise, define $\mathbf{b}_z^{(3)} = \mathbf{s}$, that is, define the z-axis in the direction of the jet. Then, define a y-basis vector to be orthogonal to $\mathbf{s}$ and the angular momentum, $(0, 0, 1)$); this ensures that the definition of longitude in $\mathcal{F}_1$ is preserved with respect to deriving jet direction in this new frame, say $\mathcal{F}_3$. A basis for $\mathcal{F}_3$ is then

$$\mathbf{b}_z^{(3)} = \mathbf{s}, \qquad \mathbf{b}_y^{(3)} = \frac{\mathbf{b}_z^{(3)} \times (0,0,1)}{\|\mathbf{b}_z^{(3)} \times (0,0,1)\|}, \qquad \mathbf{b}_x^{(3)} = \frac{\mathbf{b}_y^{(3)} \times \mathbf{b}_z^{(3)}}{\|\mathbf{b}_y^{(3)} \times \mathbf{b}_z^{(3)}\|},$$

and the direction of a jet with azimuthal angle θ and polar angle φ given by

$$\mathbf{d} = \begin{pmatrix} \mathbf{b}_x^{(3)} \\ \mathbf{b}_y^{(3)} \\ \mathbf{b}_z^{(3)} \end{pmatrix}^T \begin{pmatrix} \cos(\theta)\sin(\varphi) \\ \sin(\theta)\sin(\varphi) \\ \cos(\varphi) \end{pmatrix}. \tag{1.1}$$

Finally, we construct an orthonormal frame with the $+z$-direction corresponding to the jet's direction, $\mathbf{d}$, and $+x$ in the moon's apex-direction. The process is analogous to the transformations above and the direction of ejection for a particle launched at a given opening angle and azimuthal angle computed as in Equation (1.1) (albeit, with different basis vectors).

1.2.2 Integrating and tracking trajectories

The *genFunctions* namespace contains parameters and utilities used in multiple functions and classes in the larger code (for example, radius of the central body, orbital period of moon, etc., or vector dot and cross products) and also provides a simple interface to switch between simulation type:

```
#include "genFunctions.h"
using namespace genFunctions;

int main(int argc, char *argv[])
{
        SetEnceladus();          // Set system parameters for Enceladus/Saturn
        SetEuropa();             // Set system parameters for Europa/Jupiter
        ...
}
```

Setting the appropriate system is the first step in integrating particles. Currently, the Enceladus/Saturn system and Europa/Jupiter system are available; however, if plumes are to be studied elsewhere, it is easy to generalize the software by adding appropriate parameters and a switch to the namespace.

The Solver class is then responsible for the integration and tracking of individual plume particles, primarily through the method *ParticleSim(...)*. An object representation for the solver is used in order to maintain and modify member variables for a single particle simulation that change across multiple simulations run in parallel (in particular, parameters relevant to particle charging). Instantiating an object requires setting a number of parameters for the solver as follows:

Listing 1.1: Solver setup

```
#include ”solver.h”
...
        Solver systemSolver;    // Create solver.

        // Set planet's pole, initial particle size and charge, and charging model.
        systemSolver.SetPole(double pole_x, double pole_y, double pole_z);
        systemSolver.SetSize(double r);
        systemSolver.SetCharge(double pot);
        systemSolver.SetCharging();      // Or SetNoCharging() or SetConstCharge().

        // Set Bfield model: 1 = Connerney (1993), 2 = Simon (2014) and initialize plasma
        systemSolver.SetBfield(int bFieldModel);
        systemSolver.SetPlasma(moonPos[0], moonPos[1], moonPos[2]);

        // Dimensions of discrete grid in F_2 to track particle trajectories
        systemSolver.CreateDensityGrid(xmin,xmax,ymin,ymax,zmin,zmax,gridSize);
```

Once a Solver object has been created, the user must specify if and how to use particle charging when integrating a particle trajectory. If charge is not used when integrating particle trajectories (*SetNoCharging(...)*), the choice of B-field, particle size, and initial particle potential are not relevant, and particle trajectories depend only on three-body gravitational effects. Constant charging (*SetConstCharge(...)*) refers to using a fixed charge-to-mass ratio; that is, a particle's charge does not change over its trajectory, but the effects of the magnetic and electric fields on the particle are considered for a fixed charge; particle trajectories then depend on particle mass. Full charging (*SetCharging(...)*) updates the particle's charge each time step [76]. This is the most accurate model considered, but also significantly more expensive computationally than considering no charge or even a constant charge. In the case of Europa and Jupiter, we do not account for particle charging because Europa's gravitational force dominates particle dynamics on the time scales in which we are interested, that is, for motion in the vicinity of Europa. Electromagnetic and radiation forces are important perturbations only for the long-term particle dynamics of micron or smaller sized grains in the Jovian system.[1]

[1] Integrating particle charge in the Jupiter/Europa system also offers additional complications over Saturn/Enceladus. In Saturn, the magnetic field rigidly co-rotates with Saturn. Jupiter's fields are more complicated and do not co-rotate with Jupiter in the same sense as Saturn, making their inclusion in the integration more difficult and expensive.

"Large" particles (here, on the order of several μm in radius or larger) are sufficiently heavy and charge sufficiently fast upon ejection that the charging equations have almost no effect on a particle's trajectory; for this reason, the Solver class automatically uses no charging equations for particles larger than three μm. The Z3-Voyager magnetic field model for Saturn is a planet-wide field [38] in which particles achieve an equilibrium charge at some point after ejection. To avoid the unnecessary computation of updating the charge of a particle that has reached equilibrium, if the Z3 magnetic field model is used, a particle is checked to have reached equilibrium at each time step; if it has, then the particle charge is fixed and the Solver object set to integrate over a fixed charge-to-mass ratio. A local magnetic field about the Enceladus plume is also implemented [157]; here, particles do not necessarily achieve equilibrium within vicinity of Enceladus, so entire particle trajectories are integrated using the full charging equations. One way to see the effects of particle charging is to consider the three-body particle escape speed on Enceladus for particles between 0.2 μm and 2 μm, shown in Figure 1.1. The 0.2-μm particle has a distinct escape speed profile compared with larger particles, but from 0.8 μm and larger, escape speeds are nearly identical. In our plume simulations, we are interested in particles in the μm-range and larger, because the smallest CDA threshold is about 1.6μm and these particles (i) dominate total mass production and (ii) are more likely to contain organic material. It is demonstrated in Chapter 3 that particle charging has a negligible effect on aggregate plume dynamics for the particle sizes we are interested in, (that is, the resulting plume density profile and spacecraft impact rates resulting from millions of particle simulations is almost identical with and without particle charging). Nevertheless, for Enceladus simulations, particle charging is included for completeness and it should be noted that charging does have a progressively larger effect on smaller particles (as seen in Figure 1.1).

Next, a three-dimensional discretized grid in $\mathcal{F}_2$ is constructed to track particle number densities about the moon (*systemSolver.CreateDensityGrid(...)*). Discretization of $\mathcal{F}_2$ is centered at the moon with grid-spacing $\Delta x = \Delta y = \Delta z$ km, resulting in a cell volume of $V = \Delta x^3$ km^3. Here, we choose $\Delta x = 2.5$ km for Enceladus and $\Delta x = 1$ km for Europa (because the Europa plumes are smaller and require finer resolution). Using the discrete grid, we define and track a "residence-time" as the length of time a particle spends in each grid cell. This time is proportional to the particle's contribution to aggregate plume number density in a given cell. The maximum time step for integrating a particle trajectory is then chosen based

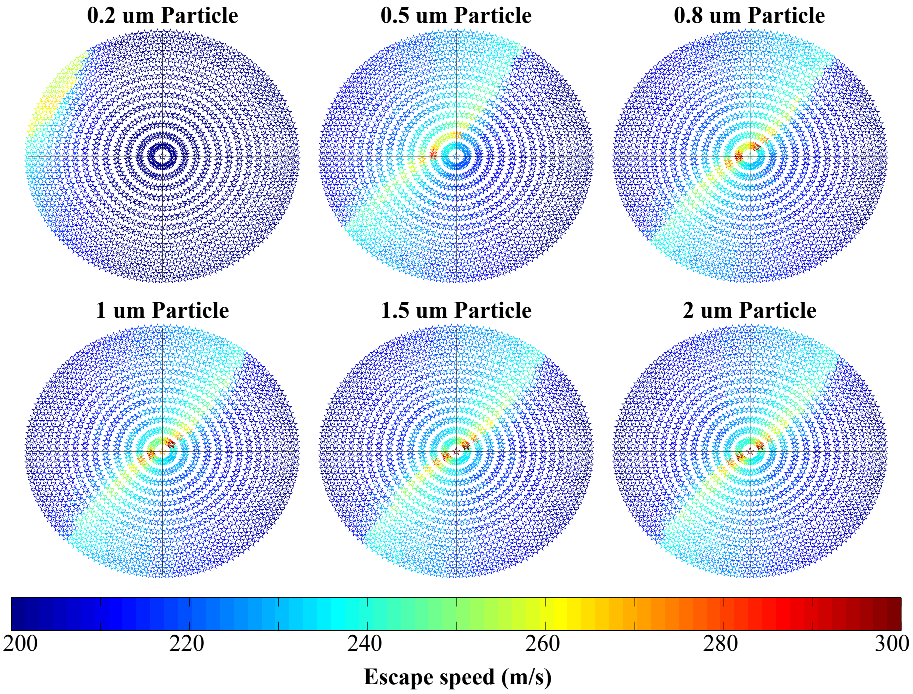


Figure 1.1: Three-body escape speed for particles launched from the south polar region of Enceladus, latitudes between −90° and −60°, including charging based on the Z3-Voyager Saturn magnetic field. If no particle charging is considered, the escape speed of all particles resembles that of the 2-μm particle.

on cell size and initial particle velocity to avoid integrating a particle through a grid cell without accounting for its residence time. Assuming that a particle will attain its maximum velocity at time $t = 0$, we choose

$$dt \leq \frac{\Delta x}{2v_0}. \tag{1.2}$$

Thus a particle traveling through a cell in the direction of x, y or z will be captured in at least two time steps; other trajectories may result in more or less detections, but we can safely expect to capture the majority of particles that enter a cell. Residence times are normalized by total grid volume V to get a one-particle normalized residence time $t_{n,m}(i,j,k)$ for the (i,j,k) cell with regards to the nth particle of size r (note that the normalization is passed in to *ParticleSim(...)* through the parameter *weight* and is not explicitly accounted for in the Solver class).

Once a Solver object has been constructed, particle trajectories can be integrated. The equation of motion for a particle in $\mathcal{F}_1$ is given by

$$\begin{aligned}\ddot{\mathbf{r}} = -\frac{\mu_P}{|\mathbf{r}|^5}\Big\{\Big[|\mathbf{r}|^2 - \frac{3}{2}J_2R_P^2(5\sin^2\delta - 1)\Big]\mathbf{r} + 3J_2R_P^2\mathbf{e}_z r_z\Big\} - \mu_M\frac{\mathbf{r}-\mathbf{r}^M}{|\mathbf{r}-\mathbf{r}^M|^3} + \\ \frac{Q_d}{m_d}\Big(\mathbf{E}^c(\mathbf{r}) + \mathbf{r}'\times\mathbf{B}^P(\mathbf{r})\Big)\end{aligned} \tag{1.3}$$

where $\mathbf{r} = (r_x, r_y, r_z)$ is the particle's position vector. The first term in Equation (1.3) accounts for the planet's gravitational field, where R_P km is the planet's equatorial radius; μ_P km^{3}s^{-2} the gravitational constant, G, times the planet's mass, M_P kg; J_2 the planet's second moment, which accounts for deviations from point mass gravity due to the oblateness of planet; and δ the particle's declination. The second term in Equation (1.3) accounts for the moon's gravity, where $\mu_P = GM_P$ km^{3}s^{-2} and $\mathbf{r}^M$ is the moon's position. Note that we assume the moon to be spherical, as the oblateness is typically not significant enough to affect particle dynamics. Finally, the interaction of a dust particle with the electromagnetic field is accounted for in the third term. Here, Q_d denotes the particle's charge, m_d the particle's mass, $\mathbf{B}^P$ the planet's magnetic field, and $\mathbf{E}^P$ the planet's electric field. Recall that particle charging is only considered for Enceladus, where the electromagnetic field of Saturn rigidly co-rotates with the planet. In this case, $\mathbf{E}^c := -(\Omega^P \times \mathbf{r}) \times \mathbf{B}^P$, where Ω_P is the angular rate of the co-rotating magnetic field.

For integration, we transform Equation (1.3) to a system of first-order, single-variable ordinary differential equations (ODEs), where $\mathbf{r}' = \mathbf{v}$ is the particle's velocity and x, y, and z subscripts denote vectors restricted to that variable (for example, $r_x = \mathbf{e}_x \cdot \mathbf{r}$):

$$\begin{aligned}
r'_x &= v_x, \\
v'_x &= -\frac{\mu_J}{|\mathbf{r}|^5}\Big\{\Big[|\mathbf{r}|^2 - \frac{3}{2}J_2R_J^2(5\sin^2\delta - 1)\Big]r_x - -\mu_E\frac{r_x - r_x^M}{|\mathbf{r}-\mathbf{r}^M|^3} - \\
&\qquad (v_y - \Omega_z^P r_x + \Omega_x^P r_z)B_z^P - (v_z - \Omega_x^P r_y + \Omega_y^P r_x)B_y^P, \\
r'_y &= v_y, \\
v'_y &= -\frac{\mu_J}{|\mathbf{r}|^5}\Big\{\Big[|\mathbf{r}|^2 - \frac{3}{2}J_2R_J^2(5\sin^2\delta - 1)\Big]r_y - -\mu_E\frac{r_y - r_y^M}{|\mathbf{r}-\mathbf{r}^M|^3} - \\
&\qquad (v_z - \Omega_x^P r_y + \Omega_y^P r_x)B_x^P - (v_x - \Omega_y^P r_z + \Omega_z^P r_y)B_z^P, \\
r'_z &= v_z, \\
v'_z &= -\frac{\mu_J}{|\mathbf{r}|^5}\Big\{\Big[|\mathbf{r}|^2 - \frac{3}{2}J_2R_J^2(5\sin^2\delta - 3)\Big]r_x - -\mu_E\frac{r_z - r_z^M}{|\mathbf{r}-\mathbf{r}^M|^3} + \\
&\qquad (v_x - \Omega_y^P r_z + \Omega_z^P r_y)B_y^P - (v_y - \Omega_z^P r_x + \Omega_x^P r_z)B_x^P.
\end{aligned}$$

Particle charge, Q_d, is evolved as a function of the particle position and velocity in the electromagnetic field, depending on how particle charging is accounted for in the current simulation (no charging, constant charge, or full charging with prescribed electromagnetic field). In fact, particle trajectories are relatively smooth functions outside of the initial particle charging upon ejection, which happens quite suddenly and can have significant effects on particle dynamics. Such behavior is typically referred to as as a “stiff” ODE, where much smaller time steps are required to capture dynamics at one point in space-time compared with others. For this reason, particles are integrated using a modified midpoint method with adaptive time steps, also referred to as a Bulirsch-Stoer integrator [139], to ensure that the effects of charging early in a particle’s trajectory are accurately captured. When integrating particles, a residence-time profile is constructed based on the grid defined in setup (Listing 1.1) and stored in a reference variable passed into the integration routine. Reference variables are also used to return whether or not a particle collided and, if so, its state vector upon collision.

1.2.3 Parallel simulation of jets

A full plume simulation is broken down by simulating emissions at specified source locations or “jets” [135, 168]. On Enceladus, the plume consists of many jets. Developing separate density profiles for each jet is important in studying temporal variability of emissions, where certain jets may be active at different times. A similar approach is taken for polar angle of particle ejection and particle size, simulating particles and storing results for a set of discrete values. This allows us to post-process simulation data with different angular and size distributions. Due to limited data available on plumes, the ability to modify and study the effect of different distributions is important to help resolve physical models.

Of course, we can only break down simulation data into so many individual pieces, so a fixed size-dependent speed distribution is used for each simulation. On Enceladus, we primarily use discrete speed distribution weights determined in Schmidt et al. [151] through Monte Carlo simulations of the subsurface venting process. Weights are available for 768 discrete particle sizes between 0.01μm and 1000μm and 393 velocities between 12m/s and 1km/s. These weights are stored in a .csv file and imported into the code at runtime. A set of discrete particle radii are then chosen and particles simulated for all velocity weights corresponding to the given particle sizes. The size-dependent speed distribution corresponding to the weights and Monte Carlo simulations takes on the approximate form [151][2]

$$p(v|r) = \left(1 + \frac{r}{r_c}\right)\frac{r}{r_c}\frac{v}{v_{gas}^2}\left(1 - \frac{v}{v_{gas}}\right)^{\frac{r}{r_c}-1}, \tag{1.4}$$

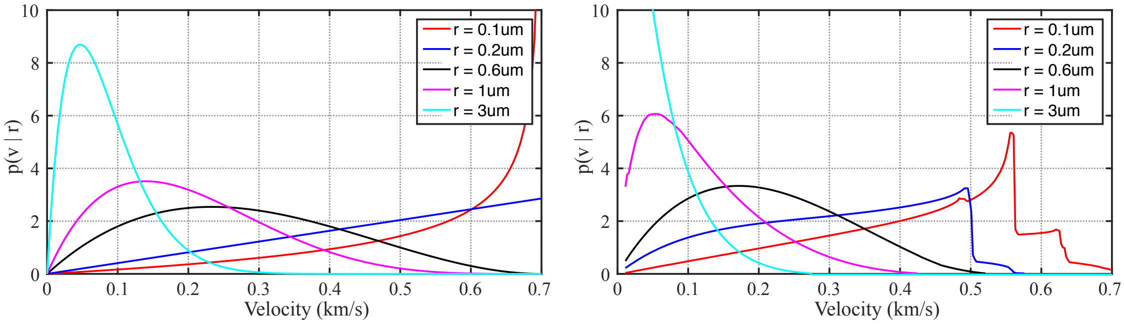
normalized such that

$$\int_0^{v_{gas}} p(v|r)dv = 1\,. \tag{1.5}$$

Initial particle velocities can also be determined in the software by random sampling from (1.4) for a given particle size. To do so, a user must use the executable flags, e.g., *-montecarlo -vgas 0.7 -RC 0.2*, for parameters v_{gas} in km/s and r_c in μm.

Here, v_{gas} is the gas velocity at the outlet. For our model, v_{gas} provides an upper bound for the particle speed as plume particles are accelerated by the gas and therefore cannot be launched at faster speeds. The critical radius r_c separates two regimes of acceleration of grains by the gas in the vents. Repeated re-acceleration of grains entrained in the gas flow will be necessary in any non-straight vent geometry. The critical radius r_c depends on parameters of the gas (velocity, density, temperature), and grain bulk density. It also depends on a length scale that corresponds to the typical depth of a particle's final collision with a vent wall prior to ejection. This depth is roughly proportional to the vent diameter, which is much smaller than the length of a vent, and thus the thickness of the moon's icy crust does not play a role in ejection velocities. Particles with radius $r < r_c$ will be re-accelerated very efficiently by gas friction to velocities approaching gas velocity. For larger particles, $r > r_c$, inertial forces become significant relative to gas friction and such grains move on average in the gas flow with speeds much smaller than v_{gas}. Note that this model fully accounts for latent heat effects due to condensation of grains [151]. Gao et al. [62] recently suggested that icy particles in Enceladus' plume contain a considerable number of aggregates, which are less dense than pure water ice. Preliminary analysis suggests that the resulting velocity distribution for aggregates will be similar in shape and well approximated by Equation (1.4). The choice of v_{gas} and r_c is discussed later. Figure 1.2 shows sample speed distributions for the weights computed in Schmidt et al. [151] compared with the analytical model with parameters $r_c = 0.2\mu$m and $v_{gas} = 700$m/s.

[2] Equation 1.4 includes a correction of $1/v_{gas}$ that was omitted in Schmidt et al. [151]. That correction also appeared without comment in Southworth et al. [161].



(a) Analytical distribution with $r_c = 0.2\mu$m and $v_{gas} = 700$ m/s.

(b) Weights from Monte Carlo simulations of venting process [151].

Figure 1.2: Sample particle speed distributions for various particle sizes. For a particle radius $r < r_c$, 0.1 μm in this case, probabilities increase sharply as $v \to v_{gas}$. For $r > r_c$, the most probable ejection speed is $v = v_{gas}\frac{r_c}{r}$. From this we notice two things: (i) larger particles are launched at lower speeds than smaller particles, and (ii) the critical radius of subsurface plume dynamics has a strong effect on the ejection speed distribution. Results from Schmidt et al. [151] (b) indicate small particles being launched slower than expected by an analytic model (a), and large particles being launched faster on average, although rarely achieving gas velocities. This is indicative of r_c not necessarily being an independent, fixed parameter. Nevertheless, results between (a) and (b) are overall comparable, and provide reference parameter values for the weights computed in Schmidt et al. [151].

Running the parallel simulation executable corresponds to simulating one jet, indicated by an ID number that corresponds to the location and tilt. Jet IDs and the corresponding jet details are given in */Data/EncJetSources.csv* or */Data/EurJetSources.csv*, and which jet to simulate is indicated with the executable flag, for example, *-jetid 5*, for the fifth jet. There are three sets of Enceladus sources: the original eight sources identified in Spitale and Porco [168] (and one modified jet direction used in Kempf et al. [91]), with IDs $-9, ..., -1$; the 98 sources proposed in Porco et al. [135], with IDs $1, ..., 98$; and a discrete set of orthogonal jets spaced evenly along the Tiger Stripes, simulating a curtain-type emission [170], with IDs $201, ..., 424$. Sources on Europa are given by the two location identified in Roth et al. [144] and two jets located at the closest approach of planned flybys of the Clipper mission (see Chapter 4).

Each jet is simulated for a set of particle sizes specified in a the driver file and a set of opening angles for the plume. Particle sizes are specified by index, where the index/size relation corresponds to those used in the speed distribution weights, and are available in /Data/Size_m.csv. Opening angles are discretized between 0°, that is, in the direction of the jet, and some maximum opening angle, $\theta_{\max}$. For Enceladus, the maximum opening angle is set to 15°, which is consistent with opening angles measured in images in Spitale et al. [170] as well as the widths of plume gas estimated in Hansen et al. [67]. Particles are then simulated for opening angles $\{0°, 0.5°, ..., 14.5°, 15°\}$. Little data is available on Europa, so θ_{max} is chosen to be 45°, which includes a wide range of ejection angles; substructure within a plume, such as that found at Enceladus, can be studied in detail when more data on Europa plumes are available. Plumes on Europa are then simulated for opening angles $\{0°, 1°, ..., 44°, 45°\}$. For a given opening angle, the jet is assumed to be azimuthally uniform, that is, particles are equally likely to be ejected at any azimuthal angle between 0° and 360°. To make a relatively uniform density for the simulated steady-state plume, the number of particles simulated for a given particle size and opening angle is chosen such that one-particle has the same representation in phase-space for each opening angle. For example, one particle is simulated for the orthogonal opening angle, seven for 0.5°, etc.. The software allows for azimuthal angles can be randomly distributed or uniformly distributed.

When an MPI process is free, the master process assigns it to simulate a given opening angle and particle size. Each opening angle and particle size consists of several hundred to several thousand particle simulations, broken down over azimuth angle and particle velocity, and each particle simulation tracks the particle's residence time in a discretized grid about the moon and the particle state upon collision. Two variables (azimuth and velocity) results in a nested for-loop and OpenMP shared-memory parallelism is applied to the outer loop to accelerate computation. In this case, the outer loop is over initial particle velocity, so each OpenMP thread is independently simulating all azimuthal angles for a given initial velocity. A C++ vector of vectors is used to track collisions and an unordered map to track residence-time profiles *within each thread*. After all azimuthal angles have been simulated on a given thread, an OpenMP critical segment is used to merge the results into global collision and residence-time profiles without interference between threads. The global residence time profile is normalized to represent the expected time that *one* ejected particle spends in a given cell, referred to as the *one-particle residence time*. The list of collisions consists of particle state vectors (particle location and velocity in $\mathcal{F}_2$) upon impact and a weight representing the probability of this particle being ejected. Once all velocities have been simulated for the given particle size and opening angle, two binary files are exported for the collision list and residence time profile.[3]

[3] Exporting a residence time profile for a fixed opening angle and particle size as opposed to, e.g., all opening angles for a given particle size, serves two purposes. First, we are free to post-process data with different size distributions and polar-angle

Data is post-processed in IDL to create binary collision and residence-time profiles for each simulated particle size. An IDL package is then used to adjust the size distribution and mass production rate and visualize data. In merging initial binary files corresponding to a single opening angle, we assume ejected particles follow a $\cos^2(\theta)$ distribution over polar angle, normalized between 0° and the maximum angle, θ_{max}. The $\cos^2$ distribution simply means that particles are most likely launched orthogonally to the surface and decreasingly likely to be launched at larger angles. This is consistent with data from CDA, showing a smooth onset of particle impacts when approaching the plume, a maximum impact rate directly over emissions, and a smooth decay when flying away from the plume. For example, using a uniform polar-angle distribution produces plume dynamics inconsistent with CDA data. Collision data is merged into collision profiles for each simulated particle size, which is discussed in Chapter 2. Residence-time profiles are weighted with the $\cos^2$ distribution to reflect the density of particles of size r_i that results from a source emitting one such particle per unit time. Converting this density into impact rates for spacecraft flybys is discussed in Chapter 3.

Particle sizes between $0.6 - 10$ μm are simulated for each source location, leading to $10^5 - 6 \cdot 10^5$ simulations per particle size (larger particles take on a smaller range of initial velocities and do not require as many simulations) and $10^6 - 10^7$ particle simulations per source (depending on the number of sizes considered for a given source). More than 200 source locations are simulated altogether, from Porco et al. [135], Spitale and Porco [168], and Spitale et al. [170].

1.2.4 Notes on usability

In a hybrid distributed- and shared-memory master/worker parallel structure, the master process distributing work should consist of one processor and all other processes should consist of multiple processors (albeit limited by the number of processors per node on a given machine). This avoids allocating multiple processors to the master process that remain idle (because the master process only needs one processor), which happens by default if one were to naively specify, for example, two processes per node or twelve

distributions. However, we only consider one polar-angle distribution moving forward, as it is well-motivated by Cassini data. The main purpose of exporting each opening angle is to limit the size in memory at runtime that the residence-time profile occupies. In particular, when multiple opening angles are simulated without exporting the data and starting a new residence-time data structure, the global residence-time profile can extend well beyond the cache in memory. Insertion then becomes several orders of magnitude slower and the code's performance can suffer significantly.

processors per MPI process. Although conceptually simple, distributing processes in this manner is somewhat complicated in practice. For example, on the Summit super computer at CU Boulder, each node contains two sockets and 24 processors (12 processors / socket). In general, suppose we want two processes per node (one per socket). If we allocate two processes per node in a batch script, the compiler will typically do this automatically. However, on one node we want three processes (the master and two workers) and on all other nodes two processes. To do this, we use a shell script to create a *hostfile*, which is passed to *mpirun* when the executable is called.

Listing 1.2: get_hostfile.sh

```
#!/bin/bash
scontrol show hostnames >> hostname_${SLURM_JOB_ID}.txt
echo $(head -n 1 hostname_${SLURM_JOB_ID}.txt) slots=3 >> hostfile_${SLURM_JOB_ID}.txt
L=$SLURM_JOB_NUM_NODES
for i in $(seq 2 $L); do
    echo $(sed -n ${i}p hostname_${SLURM_JOB_ID}.txt) slots=2 >>hostfile_${SLURM_JOB_ID}.txt
done
```

Running *get_hostfile.sh* creates a .txt file named *hostfile_ID.txt* (where ID denotes the numerical SLURM job ID) with the following format:

```
shas0321 slots=3
shas0322 slots=2
⋮
shas0329 slots=2
```

When passed to *mpirun*, this specifies that there are “slots” for three processes on node shas0321 and slots for two processes on all other nodes. An example batch script to allocate 20 nodes on Summit for 24 hours, simulating Enceladus Jets 1–98 is given in Listing 1.3 (recall sizes to simulate are specified in the driver file).

Listing 1.3: batch_script.txt

```
#!/bin/bash
#SBATCH --nodes 20
#SBATCH -t 24:00:00
#SBATCH --partition shas

# Make hostfile using names of nodes in $PBS_NODEFILE
sh get_hostfile.sh

# Number of nodes, numNodes. Three processes are started on first node (one
# master, two workers), and two worker process on remaining numP0 nodes.
numNodes=$SLURM_JOB_NUM_NODES
numP0=$(expr $numNodes + $numNodes - 2)

for JETID in {1..98}; do
```

```
    mpirun --hostfile hostfile_${SLURM_JOB_ID}.txt -np 3 -x OMP_NUM_THREADS=11  \
                ./EncPar -jetid $JETID >> ./jet${JETID}.txt :                   \
                -np $numP0 -x OMP_NUM_THREADS=12                                \
                ./EncPar -jetid $JETID >> ./jet${JETID}.txt
done
```

Here, the first three MPI processes are allocated on the first node through the hostfile, each of which is allowed eleven shared-memory threads. The remaining MPI processes are allocated on a two-per-node basis by the hostfile, with twelve shared-memory threads per process. Note that there is still one idle processor because each node has 24 processors; on the first node, two MPI processes use eleven processors each and the master process uses only one. However, when using several hundred processors per simulation, the parallel inefficiency of one idle processors is not significant, while addressing this from an implementation perspective is significantly more difficult. Each process outputs its status during the simulation to a .txt file to ensure that the software is acting as expected.

The software package is relatively self-contained and the only external package required is the GNU Scientific Library, which is used for an integration procedure in computing the background plasma of the magnetic field in *Solver.cpp*. Compiling an executable and submitting a parallel job requires GSL, an MPI compiler compatible with C++11, and an OpenMP package. The parallel executable is compiled with the *parallel* makefile, resulting in two executables: *EncPar* and *EurPar*. On Summit, the process of compiling and submitting the batch script in Listing 1.3 proceeds as follows:

Listing 1.4: Compiling and submitting simulation

```
module load intel/16.0.3
module load gsl
module load openmpi
make -f parallel
sbatch batch_script.txt
```

Chapter 2

Surface deposition and the angle of emissions

2.1 Background

Despite significant research on the Enceladus plume, there remain open questions about the plume, some of which surface deposits may help answer. As the dominant source of E-ring particles [77, 164], the Enceladus plume has probably been active for at least as long as the existence of the E ring. Studying surface deposits provides a lower limit for ice particle ejection from the current location of emission and, thus, also a lower estimate of plume lifetime. Surface deposits also indicate from where ice jets are erupting. As long as we have been aware of the Enceladus plume, observed emissions have been close to the four linear structures (commonly referred to as the Tiger Stripes) at the south pole terrain. However, due to tidal stresses opening and closing fractures [71, 81] and fracture fill-in from ejected particles, there may be other areas on Enceladus that have experienced fractures and plume activity previously. Understanding the surface deposition pattern arising from emissions along the Tiger Stripes indicates if a surface pattern can be attributed to Tiger Stripes or if its source must be located elsewhere. Finally, reproducing surface patterns provides validation to models for plume particle dynamics [91, 151, 161, 162] and insight into the structure of plume emissions at the interface between subsurface vents and particle ejection [135, 168, 170].

The purpose of this chapter is two-fold. First, we provide simulated data for surface deposition resulting from the three primary proposals for plume emission structure: the eight jets identified in Spitale and Porco [168], an updated set of approximately 100 sources identified in Porco et al. [135], and a contrasting "curtain-like" plume proposed in Spitale et al. [170]. Multiple particle sizes from $0.6 - 10$ μm are simulated for each source location, and data is generated on the impact rate in particles/sec and mass deposition in mm/year across the surface of Enceladus. Initial simulated maps of surface deposition from the Enceladus plume published in [91] have received interest from the larger research community [for example, 47, 119, 153]. Here, we provide detailed methods and a more complete set of maps and data with respect to source location and particle size. Using the newly generated surface data for a curtain-style plume [170] and the ~ 100 discrete jets proposed in Porco et al. [135], we then provide new insight into the zenith angle of plume emissions, that is, the "tilt" of the jets. Specifically, comparing simulated surface deposition patterns with the surface pattern seen in IR/UV images [150] indicates that most highly tilted jets (zenith angle $\gg 15°$) identified in Porco et al. [135] either (i) are not active long enough to contribute visible areas of resurfacing, or (ii) are so-called "phantom jets," that is, the jet in question does not exist and was (falsely) identified due to an inadvertent spacecraft viewing angle relative to emissions that create the appearance of a jet [170]. In either case, over a long timeframe, most emissions must be directed approximately orthogonal to the surface.

A background on the plume model and simulations is given in Chapter 1, along with a description of the data. Details on computing impact rate and surface deposition can be found in Section 2.2. Images of surface deposition as a function of time are given in Section 2.3 and all associated data cubes, stored in HDF5 format [173], can be obtained from `http://impact.colorado.edu/southworth_data`. Section 2.4 introduces the jets vs. curtains controversy and provides evidence that, regardless whether emissions originate from discrete jets or in a continuous curtain-style emission, the zenith angle of emissions is largely close to orthogonal to the surface. Implications and other open questions that surface deposition may provide insight towards are discussed in Section 2.5.

2.2 Particle flux and surface deposition

In considering resurfacing of Enceladus from plume particles, we are interested in two rates: the particle collision rate in particles/sec and the depth of particle deposition in mm/year. First, define $p_{size}(r) = Cr^{-\alpha}$ as the power-law particle size distribution, where C is chosen such that $\int_{r_{min}}^{r_{max}} p_{size}(r)\mathrm{d}r = 1$, r_{min} is the minimum particle radius, r_{max} is the maximum particle radius, and $\alpha > 0$ the size-distribution slope.[1]

[1] Note that for a well-defined size distribution and average plume particle mass, we must choose some minimum and maximum particle radius, $r_{\min} > 0$ and $r_{\max} < \infty$. The minimum size particle is largely based on the mechanical origin of

Assuming spherical particles, the average volume of a plume particle is given by

$$V_{av} = \int_{r_{min}}^{r_{max}} \frac{4}{3}\pi r^3 p_{size}(r)\mathrm{d}r = \begin{cases} \frac{4\pi(\alpha-1)(r_{min}^{4-\alpha}-r_{max}^{4-\alpha})}{3(\alpha-4)(r_{min}^{1-\alpha}-r_{max}^{1-\alpha})} & \alpha \neq 4 \\ \\ \frac{4\pi(\alpha-1)(\log(r_{max})-\log(r_{min}))}{3(r_{min}^{1-\alpha}-r_{max}^{1-\alpha})} & \alpha = 4 \end{cases}, \tag{2.1}$$

and average mass $M_{av} = \rho V_{av}$, for particle density ρ. Note that $r_{min} \neq 0$ and $r_{max} \neq \infty$ must be fixed for V_{av} to be well-defined. Now consider the particle impact rate, $r_{imp}(\lambda, \phi)$, as a function of surface location, for longitude λ and latitude ϕ. Let M^+ denote the plume mass production in kg/sec and $N^+ = \frac{M^+}{M_{av}}$ the expected plume production rate in particles/sec. Impact rate can then be written as $r_{imp}(\lambda, \phi) = N^+ \hat{r}_{imp}(\lambda, \phi)$, where $\hat{r}_{imp}(\lambda, \phi)$ is the normalized contribution of a single plume particle to the impact rate at location (λ, ϕ). Impact rate can be obtained by integrating the normalized impact rate over the size distribution as a function of particle radius r:

$$r_{imp}(\lambda, \phi) = \frac{M^+}{\rho V_{av}} \int_{r_{min}}^{r_{max}} \hat{r}_{imp}(\lambda, \phi, r) p_{size}(r)\ \mathrm{d}r. \tag{2.2}$$

Now, given $r_{imp}(\lambda, \phi)$ expressed in geographical coordinates, suppose we want the expected deposition height of plume particles covering some area $S(\lambda_0 \leq \lambda \leq \lambda_1, \phi_0 \leq \phi \leq \phi_1)$ per second. The expected total volume of particles per second is given by the product of the average volume of a plume particle *impacting in* S, $V_{av,S}$ (generally not equal to V_{av}), with the expected number of particle impacts in area S per second, $n_S = \iint_S r_{imp}\ \mathrm{d}S$. To compute the average volume of impacting particles at a given location (λ, ϕ), we define the normalized size distribution of particles impacting at (λ, ϕ) as $p_{imp}(\lambda, \phi, r) := \frac{\hat{r}_{imp}(\lambda,\phi,r)p_{size}(r)}{\int_{r_{min}}^{r_{max}} \hat{r}_{imp}(\lambda,\phi,r)p_{size}(r)\ \mathrm{d}r}$. Then, averaging over S,

$$V_{av,S} = \frac{\int_{\lambda_0}^{\lambda_1} \int_{\phi_0}^{\phi_1} \int_{r_{min}}^{r_{max}} \frac{4}{3}\pi r^3 p_{imp}(\lambda, \phi, r)\cos(\phi)\ \mathrm{d}r\mathrm{d}\lambda\mathrm{d}\phi}{\int_{\lambda_0}^{\lambda_1} \int_{\phi_0}^{\phi_1} \cos(\phi)\mathrm{d}\lambda\mathrm{d}\phi} \tag{2.3}$$

Let $R_E = 249.1$ km be Enceladus' radius. Volume of particles per second in S is then given by:

$$\begin{aligned} V_S &= V_{av,S} n_S \\ &= V_{av,S} \cdot \int_{\lambda_0}^{\lambda_1} \int_{\phi_0}^{\phi_1} R_E^2 r_{imp}(\lambda, \phi)\cos(\phi)\ \mathrm{d}\lambda\mathrm{d}\phi. \end{aligned} \tag{2.4}$$

the particle, of which we are interested in frozen ice grains from the subsurface ocean. A separate population of nano grains likely result from supersonic bursts through Laval nozzles in the fractures, and corresponding to a different size distribution. A maximum size is necessary to bound the average mass of particles, but also makes sense physically because ejecta particle size is at least limited by the channel width of fractures from which particles are emitted (and likely much smaller).

The depth or height of surface deposition if we assume perfect compaction of particles is given by h such that the volume of particles (Equation 2.4) is equal to the volume of S integrated to height h:

$$\begin{aligned}\int_{R_E}^{R_E+h}\int_{\lambda_0}^{\lambda_1}\int_{\phi_0}^{\phi_1} R^2\cos(\phi)\ \mathrm{d}\lambda\mathrm{d}\phi\mathrm{d}R &= h\left(R_E^2 + R_E h + \frac{h^2}{3}\right)\int_{\lambda_0}^{\lambda_1}\int_{\phi_0}^{\phi_1}\cos(\phi)\ \mathrm{d}\lambda\mathrm{d}\phi \\ &= h\left[R_E^2\int_{\lambda_0}^{\lambda_1}\int_{\phi_0}^{\phi_1}\cos(\phi)\ \mathrm{d}\lambda\mathrm{d}\phi + O(R_E h + h^2)\right]. \qquad (2.5)\end{aligned}$$

Here, h is expected on the order of mm or $\approx 10^{-7}R_E$, which justifies dropping terms $O(R_E h + h^2)$, and we find that

$$h \approx \frac{M^+}{\rho}\cdot\frac{V_{av,S}}{V_{av}}\cdot\frac{\int_{\lambda_0}^{\lambda_1}\int_{\phi_0}^{\phi_1}\int_{r_{min}}^{r_{max}}\hat{I}(\lambda,\phi,r)p_{size}(r)\cos(\phi)\ \mathrm{d}r\mathrm{d}\lambda\mathrm{d}\phi}{\int_{\lambda_0}^{\lambda_1}\int_{\phi_0}^{\phi_1}\cos(\phi)\ \mathrm{d}\lambda\mathrm{d}\phi}. \qquad (2.6)$$

Note that dropping terms $O(R_E h + h^2)$ is equivalent to estimating the total volume as the surface area times height. To estimate h over the moon's surface, we consider a mesh on the moon's surface of 1° longitude × 1° latitude cells and approximate Equation 2.6 for each cell. Cells are sufficiently small that we assume $\hat{r}_{imp}$ to be constant over each cell, denoted $\hat{r}_{imp(\lambda,\phi)}$, with units $1/\mathrm{m}^2$. Average volume of impacting particles (Equation 2.3) reduces to

$$V_{av,S} = \frac{\int_{r_{min}}^{r_{max}}\frac{4}{3}\pi r^3\hat{r}_{imp(\lambda,\phi)}(r)p_{size}(r)\ \mathrm{d}r}{\int_{r_{min}}^{r_{max}}\hat{r}_{imp(\lambda,\phi)}(r)p_{size}(r)\ \mathrm{d}r},$$

and we can then separate integrals in Equation 2.6 to get:

$$\begin{aligned}h(\lambda,\phi) &\approx \frac{M^+}{\rho}\cdot\frac{V_{av,S}}{V_{av}}\cdot\frac{\int_{r_{min}}^{r_{max}}\hat{r}_{imp(\lambda,\phi)}(r)p_{size}(r)\mathrm{d}r\cdot\int_{\lambda_0}^{\lambda_1}\int_{\phi_0}^{\phi_1}\cos(\phi)\ \mathrm{d}\lambda\mathrm{d}\phi}{\rho\int_{\lambda_0}^{\lambda_1}\int_{\phi_0}^{\phi_1}\cos(\phi)\ \mathrm{d}\lambda\mathrm{d}\phi} \\ &= \frac{M^+}{\rho V_{av}}\int_{r_{min}}^{r_{max}}\frac{4}{3}\pi r^3\hat{r}_{imp(\lambda,\phi)}(r)p_{size}(r)\ \mathrm{d}r. \qquad (2.7)\end{aligned}$$

Notice that the (approximate) total volume (Equation 2.7) takes a similar form to the impact rate (Equation 2.2), but now we are integrating over particle volume, $\frac{4}{3}\pi r^3\mathrm{d}r$. Each can be approximated using some quadrature method with sample particle sizes $\{r_0,...,r_k\}$ and data $\{\hat{r}_{imp(\lambda,\phi)}(r_i)\}_{i=0}^{k}$:

$$r_{imp}(\lambda,\phi) \approx \frac{M^+}{\rho V_{av}}\sum_i \hat{r}_{imp(\lambda,\phi)}(r_i)p_{size}(r_i)w_i, \qquad (2.8)$$

$$h(\lambda,\phi) \approx \frac{4\pi M^+}{3\rho V_{av}}\sum_i r_i^3\hat{r}_{imp(\lambda,\phi)}(r_i)p_{size}(r_i)w_i, \qquad (2.9)$$

for quadrature weights $\{w_i\}$. We use a simple trapezoid method to approximate (2.8) and (2.9). Although more accurate methods could be used for quadrature as well as higher resolution (non-constant) estimates

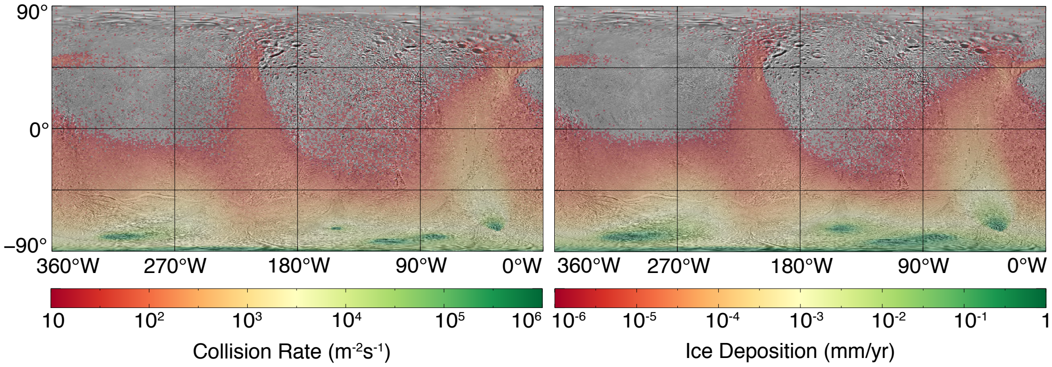


Figure 2.1: Cumulative plume particle deposition on Enceladus' surface in mm/year for the eight sources proposed in Spitale and Porco [168], particle sizes $0.6 - 15$ μm, assuming a mass production rate of $M^+ = 20$ kg/s, and slope of the power-law size distribution $\alpha = 3$.

of $\hat{r}_{imp}$, the underlying physical model is not sufficiently resolved to warrant such accuracy. The function $\hat{r}_{imp(\lambda,\phi)}(r_i)$ is exactly what we build from simulation data and store in 360×180 arrays corresponding to 1° longitude × 1° latitude areas on the moon's surface.

2.3 Surface maps

This section provides maps of surface deposition for various particle sizes and model parameters. It is important to note that our simulated deposition depth assumes perfect compaction of particles and particle density equal to that of water ice (about 916 kg/m^3). Evidence for "fluffy" (less dense) particles [62] and the fact that particles are unlikely to pack perfectly on the surface suggests that deposition may be greater than presented here; nevertheless, here we present a safe lower bound on deposition. Figures 2.1, 2.2, and 2.3 are based on the original eight sources identified in Spitale and Porco [168], and Figure 2.4 is based on a continuous curtain emission over each individual fracture. Maps for jet sources proposed in Porco et al. [135] and a full curtain scenario as proposed in Spitale et al. [170] are given in Section 2.4.

Figure 2.1 maps the particle impact rate per m^2 per sec and surface deposition in mm/year for particles $0.6 - 15$ μm, and Figure 2.2 breaks the total deposition down into particle size ranges starting at $0.5 - 2.0$ μm, and increasing up to $6 - 15$ μm.[2] For each of these figures, we choose $r_{min} = 0.6$ μm and $r_{max} = 15$ μm, a mass production rate of $M^+ = 20$ kg/s, and a size distribution slope of $\alpha = 3$. Moving forward, all

[2] The last bin contains a large range of sizes because there are very few particles of that size, and they do not travel far from the fractures

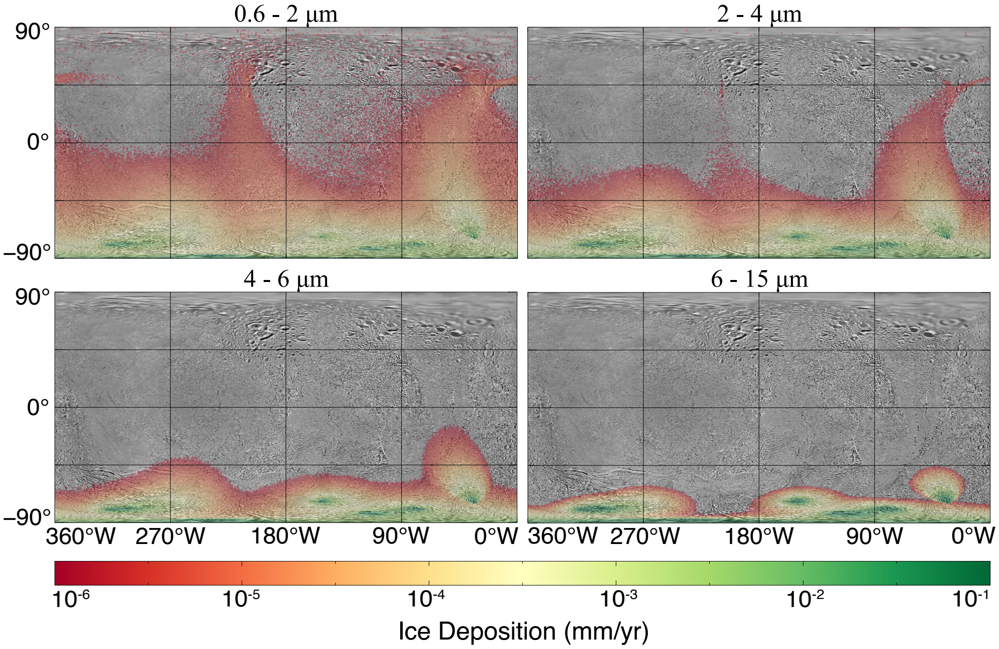


Figure 2.2: Plume particle deposition rates on Enceladus' surface in mm/year for the eight sources proposed in Spitale and Porco [168] and for particle sizes $0.6 - 2$ μm, $2 - 4$ μm, $4 - 6$ μm, and $6 - 15$ μm. Mass production rate is $M^+ = 20$ kg/s and the slope of the power-law size distribution is $\alpha = 3$.

result will use these parameters unless stated otherwise.

The mass production and size distribution slope are directly motivated through CDA data on low-altitude flybys [162], which provide the most direct measurements to date of large particles in the Enceladus plume. Results are also relatively consistent with estimates of mass poduction in Meier et al. [113], Porco et al. [136]. Note that in fixing r_{min} and r_{max}, M^+ corresponds to the mass production of particles *in this size range*. A maximum particle size must be chosen so that the average mass of a particle is well-defined; here we choose r_{max} to be sufficiently large that it includes all particles encountered by Cassini (see Chapter 3) and that plume particles of that size or larger are very unlikely. We only consider micron-size particles (formed through condensation in gas flow) because the assumed power-law size distribution does not necessarily propagate back to nano-grains. Nano-grains can also be formed through supersonic nucleation bursts at the narrowest channel points, which can occur at various depths, leading to a bi-modal or multi-modal size distribution. In any case, it is easily verified that the total ejected mass and the total redeposition mass are dominated by large particles, so extending r_{min} to nano-grains also does not have a significant effect

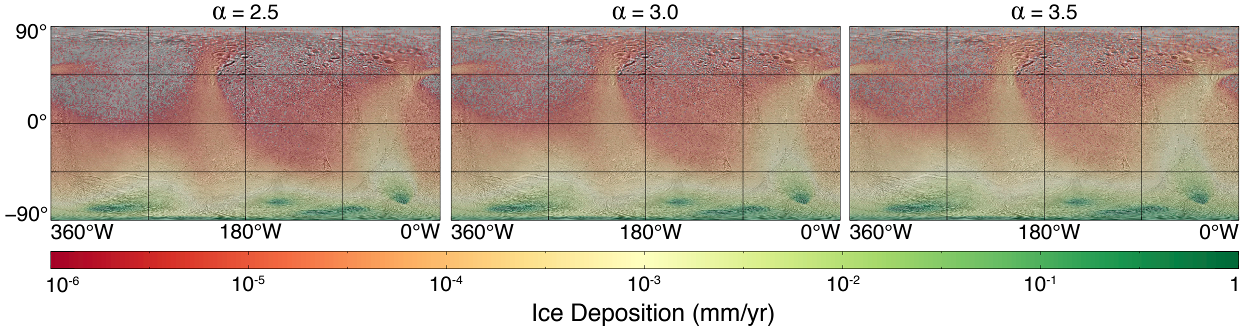


Figure 2.3: Cumulative plume particle deposition on Enceladus' surface in mm/year for the eight sources proposed in Spitale and Porco [168], particle sizes $0.6 - 15$ μm, assuming a mass production rate of $M^{+} = 20$ kg/s, and varying the slope of the power-law size distribution, α.

on results. Increasing $r_{\max}$ may lead to increased total mass production (in order to reproduce CDA data; see Chapter 3)), but the underlying structure would be consistent with that shown here. Although there is evidence that plume strength varies over time [71, 122, 162], in considering mass deposition on the surface, we need only consider an average mass production.

In Equation 2.9, we show that mass deposition depends linearly on the mass production rate and, thus, the structure of surface deposition is consistent across changes in M^{+}. Similarly, Figure 2.3 shows that the structure of the surface deposition pattern is not strongly affected by changes in size distribution slope, α, either. A steeper size distribution (larger α; see $\alpha = 3.5$, Figure 2.3) results in more small particles ejected, which tend to have higher ejection velocities [70, 151] and travel larger distances from the source, leading to an increase in deposition away from the pole. Conversely, a flatter size distribution slope (smaller α; see $\alpha = 2.5$, Figure 2.3) results in more large particles ejected, which have slower initial velocities and impact the moon closer to the source location. Then, there are increased deposition rates close to the source, and decreased deposition rates far from the source. In any case, the structure of plume resurfacing seen in Figures 2.1, 2.2, and 2.3 is consistent on some scale, regardless of α and M^{+}.

So far we have only considered the eight sources published in Spitale and Porco [168]. Figure 2.4 shows the deposition from a curtain-style emission (see Section 2.4.2) isolated to the four main Tiger Stripes of the Enceladus plume. Due to three-body effects and the angle of ejection, emissions from the outermost fractures, Alexandria and Damascus, are most likely to reach the north polar region and would likely

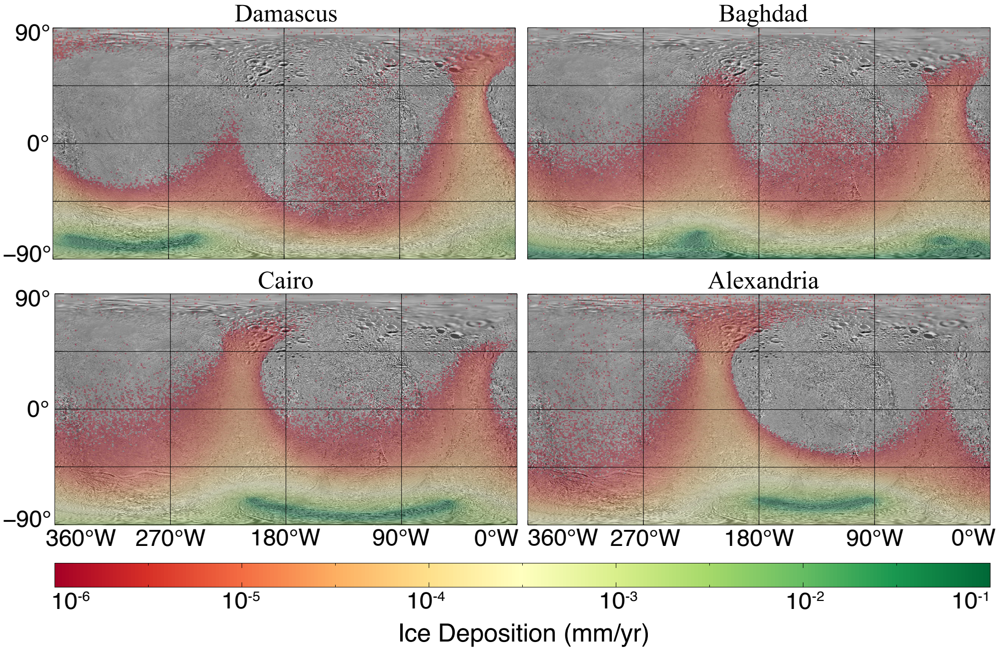


Figure 2.4: Plume particle deposition rates on Enceladus' surface in mm/year for particle sizes $0.6 - 15$ μm and all particles emitted in a curtain-style plume [170] from a single fracture. Mass production rate is $M^{+} = 20$ kg/s and the slope of the power-law size distribution is $\alpha = 3$. The fractures can be seen in the dark green region of each image where deposition rate is highest.

dominate resurfacing there. Here we have assumed emissions are directed orthogonal to the surface; note that highly-tilted emissions from Baghdad or Cairo may also be likely to reach the north pole, but in Section 2.4 we discuss that such emissions are generally not active for long periods of time. Further images of deposition from a curtain-style plume [170] and the jets proposed in Porco et al. [135] can be found in Section 2.4.

2.4 Jets vs. curtains and tilting of emissions

2.4.1 Competing theories of emission structure

The structure and location of plume emissions on Enceladus' surface was first studied by triangulating images of observed jetting activity from different angles, and projecting the result back to an approximate source location [168]. Spitale and Porco [168] identified eight distinct jets that are largely consistent with thermal emission signatures measured by the Cassini Composite Infrared Spectrometer (CIRS) [167]; however, the resolution of data was relatively coarse and the accuracy of proposed source locations no better than 10 to 20 km. This led to a set of follow-up observations at lower altitudes to better resolve the emission structure of the plume. Porco et al. [135] analyzed six years of imaging data from the Cassini Imaging Science Subsystem (ISS) using a triangulation-based approach, resulting in the identification of approximately 100 discrete "jets." Results from Porco et al. [135] are consistent with temperatures measured across the south polar terrain by CIRS [78] as well as localized hot spots identified in Cassini Visible and Infrared Mapping Spectrometer (VIMS) observations [64].

More recently, Spitale et al. [170] noticed that plume activity actually appears as a relatively continuous glow in ISS images, as opposed to discrete jet-like features as proposed in Porco et al. [135], and that finer structure within the plume is difficult to reliably identify over successive ISS images. This motivated a different analysis applied to many of the same data sets used in Porco et al. [135], where a continuous "curtain" emission is simulated over fractures and compared with images of the plume to identify active regions. One result that came out of this study is that so-called "phantom-jets" may appear in an image with the Porco et al. [135] approach, where multiple continuous curtain emissions overlap from the viewing angle and (falsely) appear as discrete jetting activity. Although many jets identified in Porco et al. [135] are undoubtedly real and have shown to be consistent with other data [72, 78], it is likely that some of the jets identified in Porco et al. [135] are not real jets.

The approaches and results of Porco et al. [135] and Spitale et al. [170] are very different and have stimulated an in-depth review of the interpretation of ISS images and plume emission structure. Here, we simulate the jets proposed in Porco et al. [135] as well as an approximate curtain, consistent with Spitale et al. [170]. Although results do not evidence one approach to necessarily be more accurate than the other, a comparison with surface color maps [150] does provide constraints on the angle or "tilt" of emissions relative to the surface.

2.4.2 Surface deposition and highly-tilted emissions

In Schenk et al. [150], near-global, high-resolution color maps of Enceladus were constructed using data from the Cassini ISS in three colors, UV, Green, and near-IR. Looking at the IR/UV ratio provides a color contrast, where a "reddish" area on the surface appears bright, and a "blueish" area, potentially corresponding to unaltered water ice, appears darker [150]. It is generally believed that Enceladus' unique color pattern, differing from other Saturnian satellites, is a result of surface re-deposition due to plume activity [73, 91, 150], which agrees with comparisons of surface maps and simulated deposition patterns. However, it should be noted that a detailed motivation and study of this topic is ongoing (Schenk et al., in preparation). Here, we assume that the surface pattern seen on Enceladus and shown in Figure 2.5 does indeed result from plume deposition and use this as a basis for expected surface deposition patterns in simulated plumes.

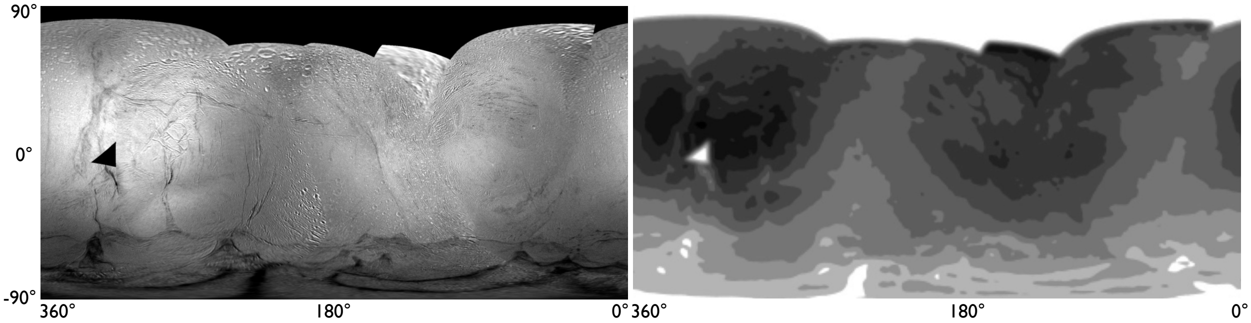


Figure 2.5: IR/UV color maps of Enceladus, as provided in Schenk et al. [150], along with a log of the IR/UV ratio to compare with simulated deposition rates in log-scale. Darker areas in the IR/UV ratio correspond to surface area with reflectivity similar to that of unaltered water-ice, which is believed due to a resurfacing effect of Enceladus by plume particles from a subsurface ocean.

The Enceladus plume model and structure can be constrained by ensuring that data collected by spacecraft are reproducible. Here, we use IR/UV images (Figure 2.5) as a reference to compare simulated surface deposition profiles. Figure 2.6 shows simulated global surface deposition profiles of plume particles size $0.6 - 15$ μm, for a curtain-style plume [170] and discrete jet sources [135]. Each simulated plume leads to a surface deposition pattern that is largely consistent with the IR/UV ratio seen in surface images of Enceladus (Figure 2.5). However, the discrete jets proposed in Porco et al. [135] do not reproduce the finer structure of surface deposition, while the curtain-style plume does. In particular, jets from Porco et al. [135] lead to surface deposition features not seen in surface images.

In looking at the simulated curtain- and jet-plumes, the fundamental difference between the two is the direction that jets are pointing. Like the jets, the curtain is also "discrete" and not simulated as a truly

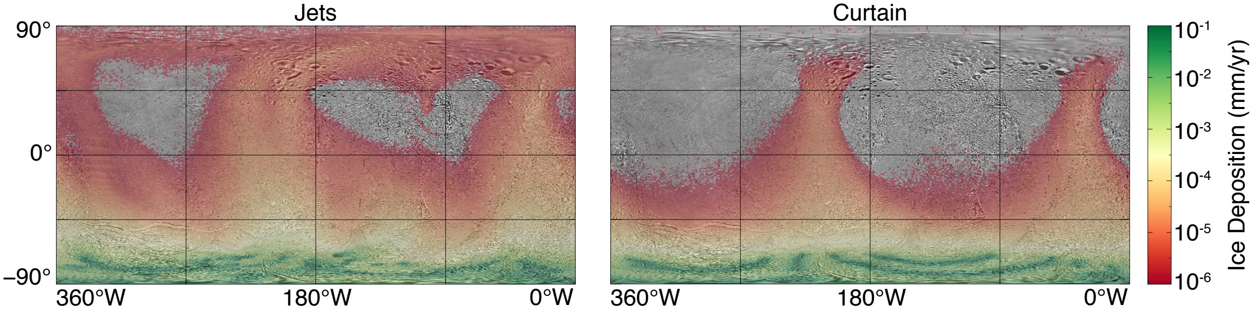


Figure 2.6: Surface deposition rates in mm/year for jets from Porco et al. [135] on the left, and a curtain [170] on the right. Particle size ranges considered are $0.6 - 15$ μm.

continuous curtain, and both models have emissions primarily aligned on the Tiger Stripes. However, all sources simulated for the curtain are directed orthogonal to the surface, while each of the jets proposed in Porco et al. [135] have a given zenith and azimuthal angle. A number of the proposed zenith angles are as large as 30–62° (measured from orthogonal to the surface). Such strongly tilted jets lead to very distinct deposition patterns on the surface, which do not always agree with surface images. As an example, Jets 23 and 95 from Porco et al. [135] originate in close proximity to each other, but Jet 23 has a near-orthogonal zenith angle of 3°, while Jet 95 has a large zenith of 42°. Figure 2.7 compares the deposition pattern for Jets 23 and 95.

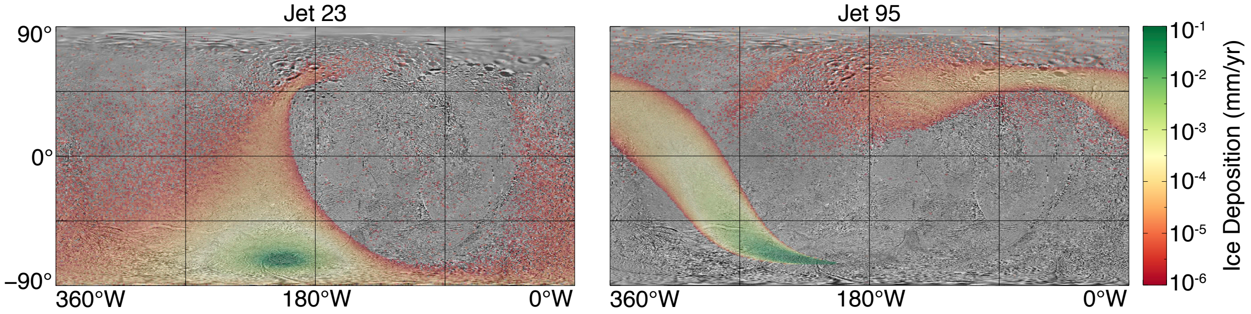


Figure 2.7: Surface deposition rates in mm/year for jets 23 and 95 from Porco et al. [135], with particle sizes $0.6 - 15$ μm, and all mass production allocated to the single jet. The contribution in mm/year with respect to all 98 jets proposed in Porco et al. [135] is approximately two orders of magnitude smaller.

Surface deposition from Jet 23 is consistent with surface images, but Jet 95 has a long, narrow deposition pattern, aligned effectively the opposite direction as the pattern seen in surface images. Although this is partially due to the proposed azimuthal angle as well, in fact, most highly tilted jets lead to deposition patterns that are at odds with surface images. A natural conclusion from this is that, for the most part, highly tilted jets are not contributing to surface deposition. Figure 2.8 shows the jet-plume surface deposition for three scenarios: all jets in Porco et al. [135], jets with zenith angle less than 30°, and jets with zenith angle less than 20°. We can see that by simply removing highly tilted jets, the surface deposition pattern is now consistent with that of the simulated curtain and, more importantly, surface images.

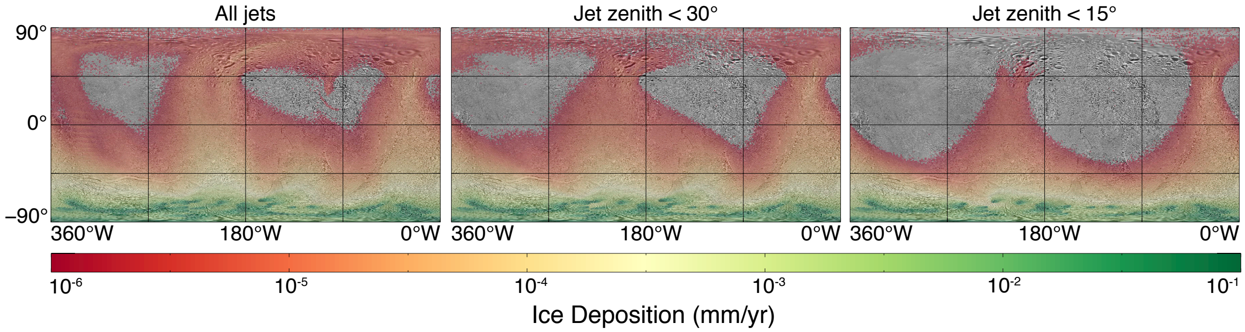


Figure 2.8: Surface deposition rates in mm/year for jets from Porco et al. [135] and particle sizes $0.6 - 10$ μm. The leftmost figure corresponds to the 98 principle jets proposed in Porco et al. [135], the center figure to the 87 jets with zenith angle less than 30°, and the rightmost figure to the 70 jets with zenith angles less than 20°.

There are several possible explanations as to why highly-tilted emissions are not contributing to surface deposition. First, recall Spitale et al. [170] showed that the approach used in Porco et al. [135] can lead to so-called phantom jets, where overlapping continuous emissions appear to be a jet based on viewing plume emissions from a certain angle. This is one possible explanation for the discrepancy between deposition patterns for the simulated jet-plume and surface images: many of the highly tilted jets, which are the source of surface patterns that differ from surface images, simply do not exist and were identified due to images of overlapping continuous emissions. Less than 30% of jets proposed in Porco et al. [135] have a zenith angle greater than or equal to 20°, and only about 10% have a zenith angle greater than or equal to 30°. A few of these high-tilt jets lead to surface deposition patterns that are consistent with surface images (for example, Jet 5), and we can reproduce surface images well using approximately 80% of jets proposed in Porco et al. [135], leaving about 20% of which may be phantoms.

Although surface deposition of tilted jets is largely inconsistent with ISS surface images, the distinct angle of such jets, differing notably from most plume emissions, does decrease the probability of these being phantom jets [170]. A second explanation as to why highly-tilted jets do not contribute to surface deposition is that they are not active long enough (at least for a fixed direction of emission) to create visible surface patterns. Roughly, particles covering the surface should be visible in images when their depth is greater than the reflectivity wavelength (on the order of nanometers). All figures shown use a minimum value for the deposition profile of 1 nm/year. Looking at Figures 2.7 and 2.8, one can faintly notice the deposition pattern of Jet 95 in the aggregate deposition pattern (center two rows, leftmost column), contributing on the order of 1 nm per year in particle deposition. Thus, for certain deposition contributions from Jet 95 (or other highly tilted jets), particularly areas that do not overlap with the deposition of other jets, to be visible in surface images, Jet 95 would have to be continuously active for around one year or longer.

Although the plume itself has been active for much longer than one year, there is evidence for temporal variability of plume emissions in several forms. There has long been speculation and confirmation of tidal stresses along Enceladus' orbit modulating the emissions [67, 71, 81, 82, 122], evidence from CDA of entire fractures turning on and off [162], and, recently, evidence of long-term change in emission rates between 2005–2015 [86]. Recent images also show prominent jets in the plume suddenly turning on or off in successive images, while the curtain-emissions remain relatively constant [169]. It is thus plausible that highly-tilted jet sources do not stay active for significant periods of time, either turning on and off or changing direction sufficiently often that their deposition signature cannot be seen in ISS surface images. In the case of temporal variability, an interesting question is whether highly-tilted jets are more susceptible to variability and short lifespans compared with jets near orthogonal to the surface, or if all or most jets experience short lifespans. However, as of now we can only comment on the highly-tilted jets due to their unique surface signature.

Finally, the ISS images showing jet sources turning on and off over a short period of time [169] also provide evidence that the discrete jet sources identified in Porco et al. [135] have significantly higher velocities than the continuous curtain emissions proposed in Spitale et al. [170]. Depending on the gas velocity of the jets, it is possible that a large percentage of particles from discrete jet sources escape Enceladus' gravity and do not land back on the surface compared with curtain emissions. This would also help explain why the deposition patterns from highly-tilted jets in Porco et al. [135] are not apparent in surface images. A more detailed analysis of the differing velocity of jets and curtains can be found in Spitale and Southworth [169].

2.5 Moving forward

Open questions remain on the Enceladus plume, in particular how long plumes have been active on Enceladus as well as the lifetime of the current plume. Some of the more heavily cratered regions on Enceladus' surface lie in the middle of some of the heaviest resurfacing area (Figure 2.5) and vice-versa for smooth regions that are not being resurfaced by the current plume configuration. Were such smooth areas previously resurfaced? Why is the cratered terrain so distinct in images, despite consistent resurfacing by the current plume? Each of these are topics of future research. Full colormaps of the north pole will also be available soon, which will provide greater insight into historical plume activity. Initial data shows deposition patterns on the north pole that are inconsistent with south polar emissions, prompting the question as to whether there was once plume activity on the north pole.

Chapter 3

Plume density and spacecraft impact rates

3.1 CDA and simulation data

Cassini has recently performed two flybys at low altitudes: *E7* in late 2009, which had a closest approach of approximately 100 km; and *E21* in 2015, which dove all the way to 50 km above the surface at closest approach. These flybys provide invaluable insight into plume dynamics as the only direct source of data available on larger plume particles (large being on the order of μm). Here, we perform the most detailed modeling of large particles in the Enceladus plume to date and use Cassini's low altitude flybys to study and constrain the plume model. This section details methods used to derive spacecraft impact rates and estimate the size distribution at different altitudes, based on simulated data. Section 3.3 will then introduce major findings that have resulted from Cassini's low-altitude flybys as well as new questions that have arisen based on the results. Referencing spacecraft data, parameter studies are performed in Section 3.2 to indicate which model parameters may explain discrepancies in results and which are less significant moving forward. The two flybys considered here are the 7th and 21st Cassini flybys of Enceladus, denoted E7 and E21, respectively. E7 took place on day 306 of 2009, with a closest approach of about 100 km, and E21 took place on day 301 of 2015, with a closest approach of about 50 km. Trajectories and the observed particle number density for each flyby are shown in Figure 3.1. Note that data shows Alexandria to emit significantly less particles than the other fractures.

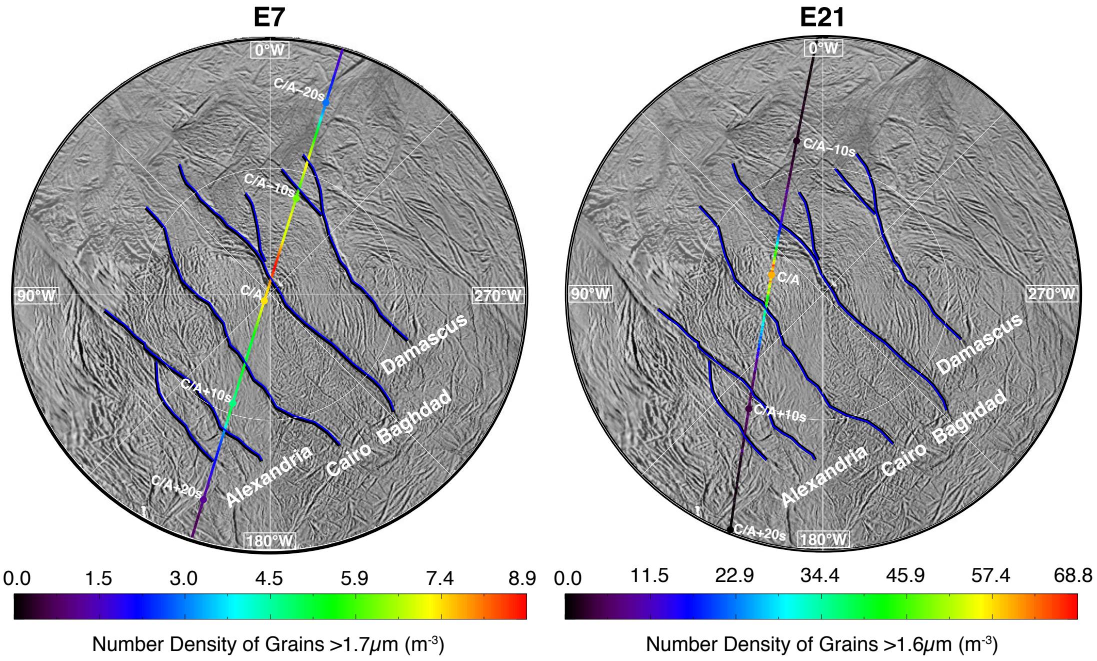


Figure 3.1: Spacecraft trajectories for E7 and E21 plotted over a south polar map of Enceladus. Heat maps along the trajectory indicate the observed particle number density during each flyby. Closest spacecraft approach is denoted by "C/A," and western longitude coordinates are plotted around the region for reference.

3.1.1 Integrating impact rates

Suppose the plume has a steady-state particle number density $n(x, y, z)$ particles/m^3 in an Enceladus-centered inertial frame and a spacecraft with sensitive area S_A m^2 traverses the plume on trajectory $\mathbf{sc}(t)$. We assume the spacecraft primarily encounters particles with slow velocities relative to the spacecraft, because the probability of an encounter for two small objects moving different directions at several km/sec is very small. In this case, the relative speed of particles to the spacecraft is given by $\mathbf{v}_{\text{imp}}(t) \approx \mathbf{v}_{\text{sc}}(t)$, where $\mathbf{v}_{\text{sc}}(t)$ is the spacecraft velocity and is effectively constant over the timescales considered here. Then, the impact rate over $\mathbf{sc}(t)$ is given by

$$r(t) = |\mathbf{v}_{\text{sc}}|\widehat{S_A} n(\mathbf{sc}(t)),$$

where $\widehat{S_A}(t)$ is approximately constant, given by the sensitive area of the dust detector facing forward along $\mathbf{sc}(t)$.

Recall from Section 2.2 that plume production rate is given by $N^+ = \frac{M^+}{M_{av}}$ particles/sec, where M^+

is the mass production rate in kg/sec and M_{av} the average mass of a plume particle, given by integrating the mass of a uniform-density sphere, $m = \rho\frac{4\pi r^3}{3}$, over the particle size distribution (2.1). As discussed in Chapter 2, a minimum particle size must be chosen so that the power law is well defined and a maximum particle size chosen so that the average volume is finite. A minimum particle size is chosen in the μm-range, specifically $r_{\text{min}} = 0.6\ \mu$m, to avoid potential multi-modal distributions resulting from different potential ejection mechanisms of nano-grains (Section 2.2). Particles smaller than this will contribute a marginal amount of mass to the total ejection rate anyhow. The maximum particle size is somewhat arbitrary, although it does affect the total mass production. Here, we choose $r_{\text{max}} = 15\ \mu$m, representing particles larger than anything yet encountered by CDA, but possible to be encountered at lower altitudes. Particles with radius $r \gg 15\ \mu$m will be ejected at very slow velocities due to the size-dependent speed distribution, and will neither travel far from the source location before impact nor achieve altitudes greater than a few kilometers. Because such particles will, thus, not contribute to recorded data on the plume, it is reasonable to ignore their contribution to mass production.

Let $t_r(x, y, z)$ denote a one-particle residence time for a particle of size r, that is, one particle of size r ejected from the plume is expected to spend $t_r(x, y, z)$ seconds at spatial location (x, y, z). Plume number density is then given by integrating over r and multiplying by the total number of particles ejected:

$$n(x, y, z) = N^+ \int_{r_{\text{min}}}^{r_{\text{max}}} t_r(x, y, z)p(r)\mathrm{d}r.$$

Single-particle residence time profiles as a function of particle size are exactly what we simulate and store in a discretized grid about Enceladus for discrete particle sizes r. Number density is then approximated using a quadrature rule over r:

$$n(x, y, z) \approx \sum_j N^+_{r_j} t_{r_j}(x, y, z) w_j, \qquad \text{where}$$

$$N^+_{r_j} = \begin{cases} \frac{3M^+(\alpha-4)r_j^{-\alpha}}{4\pi(r_{\text{min}}^{4-\alpha} - r_{\text{max}}^{4-\alpha})} & \alpha \neq 4 \\ \frac{3M^+}{4\pi} \log\left(\frac{r_{\text{max}}}{r_{\text{min}}}\right) r_j^{-\alpha} & \alpha = 4 \end{cases},$$

for quadrature weights $\{w_j\}$. Sample density profiles of the plume for the jets proposed in Porco et al. [135]

and the curtain model [170] are given in Figure 3.2.[1]

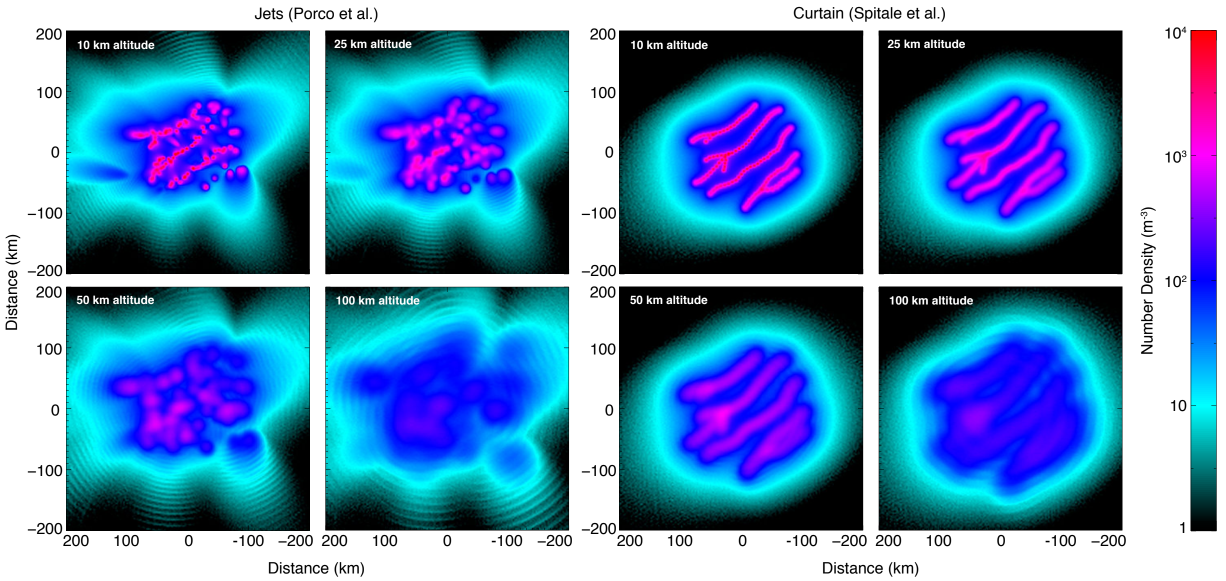


Figure 3.2: Particle number densities for the Enceladus plume at altitudes 10, 25, 50, and 100km, and corresponding to a jets configuration Porco et al. [135] and the curtain model [170]. Mass production rate is given by 20 kg/s and size-distribution slope $\alpha = 3$, and the point $(0, 0)$ corresponds to the south pole of Enceladus.

In simulations, discrete particles sizes are chosen specifically to line up with the size thresholds for CDA during low-altitude flybys, making quadrature rules that require certain quadrature nodes to be chosen inapplicable, such as high-order Newton-Cotes and variations in Gaussian quadrature. Here, we use a trapezoid rule because (i) the difference in impact rate between nodes is relatively linear, and (ii) there is insufficient data to warrant a higher-order method.

3.1.2 Size distribution slope

CDA counts impacts in size bins, giving the total number of particle impacts with radius greater than r_0, greater than r_1, and so on. Recall that in the model we assume a power-law size distribution for particles with slope α, $p(r) \sim r^{-\alpha}$. Then, the total expected number of particles larger than r_i μm for some r_i is

[1] Notice that number densities for jets in the left of Figure 3.2 have discrete lines in the density profiles like a contour map, while the curtain density profile is relatively smooth in every direction. This is an effect of the jets being simulated for a set of exact opening angles at 0.5° separation to verify that a $\cos^2$-angular distribution makes sense vs., for example, a uniform angular distribution. Curtain simulations were done *after* it was verified with jet simulations that a $\cos^2$-angular distribution is appropriate. To reduce this numerical effect, each half-angle was then simulated following a $\cos^2$ distribution over small intervals, for example at a 1° opening angle, all particles are simulated as a random variable in a $\cos^2$ distribution over $[0.75°, 1.25°]$.

given by

$$n(r > r_i)C \sim r_i^{1-\alpha},$$

for $\alpha > 1$ and some constant C. To isolate the particle size-distribution slope, we disregard leading constants and do not derive results here based on the constants. Then, if CDA collected d particles larger than radius r, we are looking for α such that $Cr^{-\alpha} = d$. Taking the logarithm of both sides gives the linear form $(1-\alpha)\log(r) + \log(C) = \log(d)$. Now we can form a linear least squares problem to fit α to k bins of CDA data by minimizing

$$\begin{pmatrix} 1 & \log(r_0) \\ \vdots & \vdots \\ 1 & \log(r_k) \end{pmatrix} \begin{pmatrix} \log(C) \\ 1-\alpha \end{pmatrix} = \begin{pmatrix} d_0 \\ \vdots \\ d_k \end{pmatrix}.$$

The least-squares solution is given by solving the normal equations, which can be explicitly solved in this case for

$$\alpha = 1 - \frac{\left(\sum_i \log(r_i)^2\right)\left(\sum_i \log(d_i)\right) - \left(\sum_i \log(r_i)\right)\left(\sum_i \log(r_i)\log(d_i)\right)}{3\sum_i \log(r_i)^2 - \left(\sum_i \log(r_i)\right)^2}.$$

In the case of simulated plume particles, we calculate an impact rate for a discrete set of particles of size $\{r_i\}$. In this case, a least-squares slope can be calculated in a similar manner, this time with respect to the probability distribution function as opposed to the cumulative distribution function used above.

It should be noted that the size distribution of particles at various altitudes does not follow a power law indefinitely. In particular, data indicates that the power law is a good approximation of the particle size distribution within some altitude-dependent size range. However, at each altitude, there is some particle size for which larger simulated particles do not achieve such altitudes. Because the logarithm of zero impacts is infinity, such a point cannot be included in the least squares fit. For the first simulated particle size that receives zero impacts, the number of impacts is instead set to $\frac{1}{n}$, where n is the number of particles simulated of that size. This provides a strict upper bound on the expected number of particle impacts for that size, and helps reduce a possible bias towards a flatter size distribution that can occur if that particle size is ignored altogether. In particular, it is possible to obtain a small number of simulated impacts for several large particle sizes; without including the fact that no impacts are obtained for the next particle size, this can bias the linear fit to suggest that the slope is flattening out for larger particles. Because CDA data only consists of three bins, each of which is a cumulative rate, such an effect is less pronounced. However, for similar reasons, we only consider size-distribution slopes as measured by CDA when all three bins receive at least one impact.

3.2 A comparison of model parameters

3.2.1 Electromagnetic field

An accurate model of particle dynamics accounts for particle charging with respect to the surrounding electromagnetic fields. A first-order approach is to consider a global electromagnetic field about the planet. A more accurate approach is to consider the global magnetic field about Saturn, while accounting for the likely perturbation of the field caused by a dust plume [157]; such effects would be particularly noticeable for an ongoing plume with a relatively large production rate, as seen on Enceladus. However, such increased physical accuracy comes at the cost of more computationally expensive simulations. Thus, it is worth considering whether such high-fidelity models are relevant to the final results.

Here, we compare simulated spacecraft impact rates near Enceladus for the three potential particle charging models: no charging, particle charging with a global Z3 magnetic field about Saturn [38], and particle charging that also accounts for a local magnetic field around the plume [157]. Because the only explicit particle data available on the Enceladus plume is CDA impact rates, we are only interested in particles larger than the smallest CDA threshold considered, in this case approximately 1.6 μm. Thus, simulations for each model are run for ten particle sizes between $1.6 - 10$ μm, for each of the eight original jets proposed in Spitale and Porco [168] as well as the modified jet proposed in Kempf et al. [91] to better reproduce CDA data. Figure 3.3 shows the expected impact rate in each scenario over the E7 and E21 flybys. In particular, notice that the local and global electromagnetic fields produce nearly identical results, and remove charging altogether remains well within the potential error of simulated data or the underlying model. We conclude that including particle charging in the equations of motion is not important for particles in the 1.6 μm and larger regime, which corresponds to particles sizes that we are interested in working with to reproduce CDA data. Note that for progressively smaller particles ($\ll 1\mu$m), charging can have significant effects on particle dynamics, but such dynamics are not relevant here.

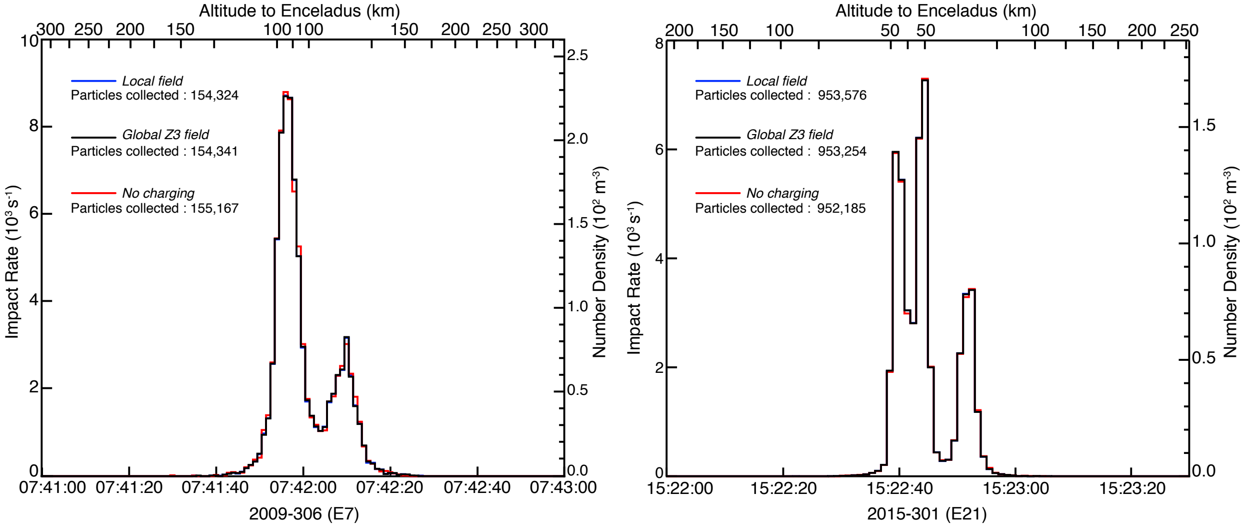


Figure 3.3: A comparison of simulated impact rates for particle charging based on a local electromagnetic field [157], a global Saturn Z3 magnetic field [38], and no particle charging, for particle sizes larger than 1.6μm (the minimum detector size of CDA data that we are considering). Impact rates are calculated for the eight original jets proposed in Spitale and Porco [168] for the E7 and E21 Cassini flybys. Note that CDA data is not considered here; this is only a comparison of impact rates for simulated models.

Because it is not critical to include particle charging for accurate dynamics, more simulations can be run at a tractable computational cost. In the next section, we take advantage of this to do a parameter study (without accounting for particle charge) of the size-dependent speed distribution introduced in Equation (1.4). This also validates the choice of not including particle charging in the Jovian magnetosphere when simulating potential Europa plumes in Chapter 4.

3.2.2 Speed distribution

The size-dependent speed distribution used for full plume simulations here are based on Monte Carlo simulations of particles being emitted through fractures [151]. Results of the Monte Carlo simulations are approximately described by the analytic distribution given in (1.4). The primary difference is that the analytic description chooses a fixed value for r_c, which may not be consistent over fractures and different vent geometries; this leads to a smooth distribution for small particles approaching gas velocities, as opposed to some speed at which probabilities greatly decrease (see Figure 1.2). In any case, we are interested in large particles, and results in Section 3.3 demonstrate that the speed distribution weights computed in Schmidt et al. [151] are largely consistent with observed data. However, large particles in the several μm and larger range are not launched at sufficiently high velocities in simulations to reproduce observed impact rates observed by CDA in the M3 (largest threshold) detector. This motivates an exploration of the speed- and size-distribution parameter space. Although there is currently not enough data from the plume, particularly with respect to large particles, to get a well-resolved profile of the speed distribution parameters, here we provide a basis for future studies of plume particle dynamics.

In a simple model that assumes fixed parameters over the entire plume, the free parameters to consider are critical radius, r_c, gas velocity, v_{gas}, and initial size-distribution slope, α. Recall from Figure 1.2 that values of $r_c = 0.2$ μm and $v_{gas} = 700$ m/s correspond relatively well to the Monte Carlo speed distribution. Here, we simulate a single plume source for $r_c = 0.1, 0.2$ and 0.4 μm and $v_{gas} = 500, 750,$ and 1000 m/s, and measure the approximate simulated size distribution slope at various altitudes above the plume. Results are shown in Figure 3.4. It is important to note that size-distribution slopes are only shown up to 75km altitudes; above, data becomes noisy because few large particles achieve such altitudes. An analysis with respect to the data from E7 and E21 flybys can be found in the following section.

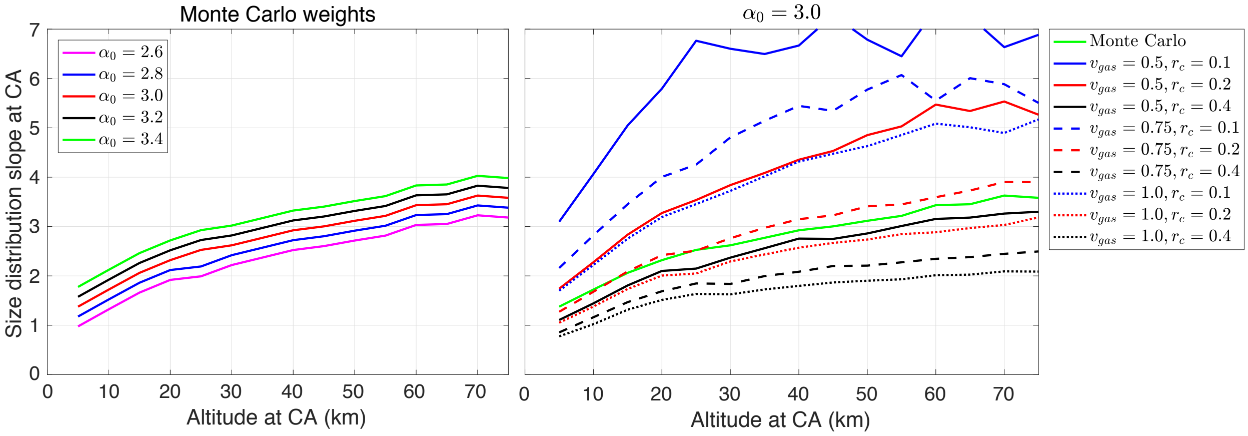


Figure 3.4: Simulated least squares size-distribution slope (Section 3.1.2) at closest approach (CA) as a function of altitude at closest approach. The left plot shows simulated slopes for multiple initial size-distribution slopes, α_0, based on the Monte Carlo weights computed in Schmidt et al. [151] and used for simulating the Enceladus plume in Section 3.3. On the right, simulated slopes at CA are shown for variations in the critical radius and gas velocity in an analytic speed distribution (see Equation (1.4)).

3.3 CDA encounters the Enceladus plume

The first and foremost result presented here is that the simulated plume model reproduces both low-altitude Cassini flybys well with a fixed set of parameters. Despite evidence for a long-term decline in total plume production between 2005 and 2016 from images [86], here we observe a plume that appears to have a very comparable mass production in 2009 and in 2015, on the order of 25 kg/s. Although some of the finer structure is missing, the simulated spacecraft impact rate for multiple particle size ranges is consistent with that experienced by CDA. Spacecraft data for three size thresholds and approximate fits using simulation data are shown in Figures 3.5 and 3.6.

Reproducing observed impact rates provides important validation for the physical models used, including the deep source plume and associated subsurface dynamics. Data from the two low-altitude flybys gives further insight to resolve our model, particularly with respect to total mass production, active areas of emission, and the size and distribution of ejecta particles.

3.3.1 Active areas of emission

Simulations are based on jet sources proposed in Porco et al. [135] and, for most jets, jets are weighted by the relative production provided in Porco et al. [135]. However, a number of proposed sources appear to be inactive during the E21 flyby. In particular, CDA did not detect a significant number of particles from the $0° - 90°$W part of Baghdad during E21 (we will refer to this part as the "upper part"). Looking at the trajectory in figure 3.1, note that, before closest approach, Cassini flew directly over the upper part of the Baghdad fracture. However, impact rates observed by CDA and shown in Figure 3.6 demonstrate a single peak impact rate when crossing over Cairo, and no noticeable increase when traversing over Baghdad. Similarly, but less noticeable, few impact were detected over Alexandria during E21. There is a slight increase in impact rate around 15 seconds after closest approach (Figure 3.6), but the number of particles encountered is significantly less than expected from the jet strengths proposed in Porco et al. [135].

To reproduce spacecraft data, it is necessary to "turn off" jets 25–34, 54–59, 89–93, and 98, corresponding to the few emissions in Alexandria and both fractures north of where Baghdad splits, in the

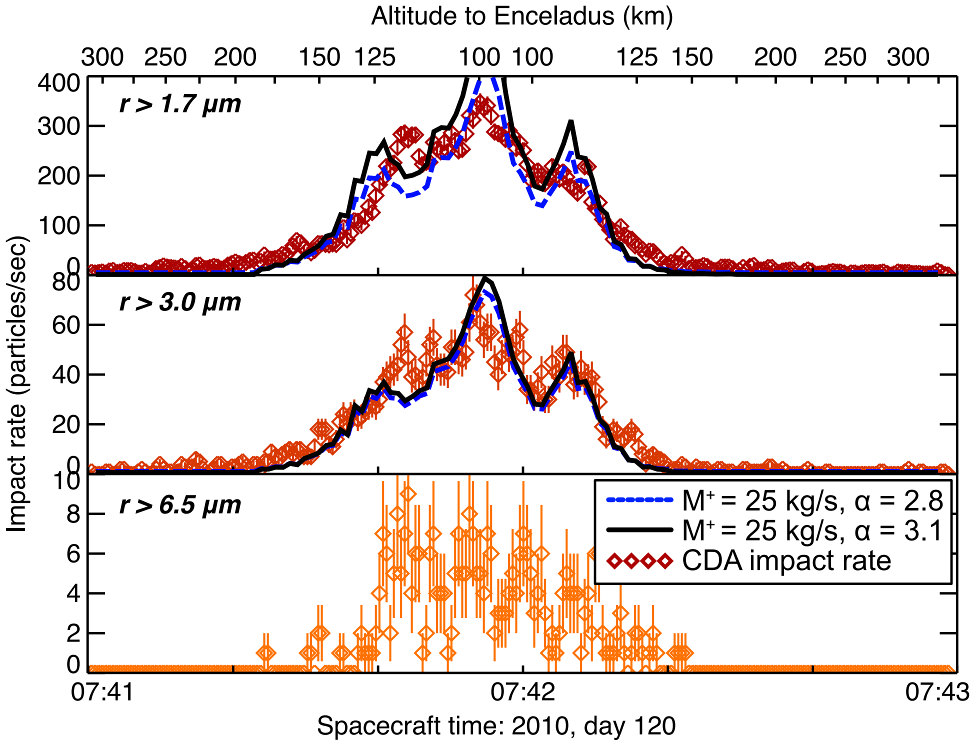


Figure 3.5: CDA data for M1, M2, and M3 sensors, and two simulated plume configurations for E7 flyby. Spacecraft data and associated error bars are shown in red/orange, and simulated plumes are shown in blue and black.

$0° - 90°$W quadrant (Figure 3.1). Weighting the rest of the jets as proposed in Porco et al. [135], simulated data reproduces both flybys well using a single set of parameters for mass production and size-distribution slope.[2] Interestingly, it is unclear whether the upper part of the Baghdad fracture was active during E7 or, similarly, whether the $180° - 270°$W ("lower part") of Baghdad was active during E21. CDA data from each can be reproduced without the aforementioned areas on the Baghdad fracture and a correspondingly lower mass production. This underscores the importance of many low-altitude flybys in resolving finer structure of the plume: altitudes of 50km or lower help to isolate which areas of the plume contribute to impact rate, and multiple flybys are needed to get an impact-rate profile across the entirety of the Tiger Stripes.

There are a number of possible explanations for why some jets proposed in Porco et al. [135] are not observed by CDA during E21. Similar to the highly tilted jets discussed in Chapter 2, it is possible that

[2] A simple adjustment of other jet weights can reproduce the impact rate of the first two detectors for each flyby near perfectly. However, such a problem is underdetermined, meaning that there are many solutions that would be a near-perfect fit, none of which would provide reliable information on emission levels across the fractures.

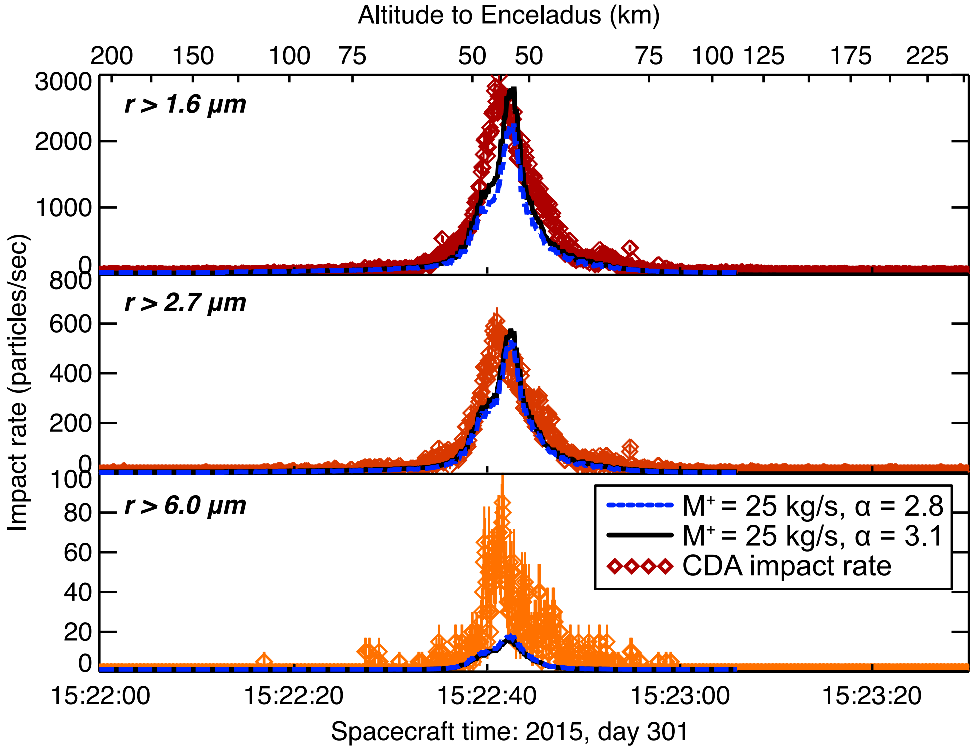


Figure 3.6: CDA data for M1, M2, and M3 sensors, and two simulated plume configurations for E21 flyby. Spacecraft data and associated error bars are shown in red/orange, and simulated plumes are shown in blue and black.

some of these jets were falsely identified as phantom jets and, thus, do not exist. However, other instruments have previously identified the upper part of the Baghdad fracture as an active area of emission [78, 135], and it is unlikely the entire region consists of phantom jets. Two additional hypothesis are: (i) the upper region of Baghdad is not emitting particles of sufficient size to be detected by CDA, or (ii) the upper region of Baghdad was not active during E21, but has been active previously. If the latter is true, it is likely not due to orbital tidal stresses often used to describe temporal variation of jets [71, 82, 148, 159] because the E7 and E21 flybys occurred at a very similar time in Enceladus' orbit about Saturn. Ingersoll and Ewald [86] proposed several explanations for long-term variation in plume production that may be relevant here, including an 11-year tidal period. Nevertheless, resolving the finer structure of plume emissions on the surface is ongoing work and requires further data.

3.3.2 Particle distributions

Two fundamental characterizations of the plume are the size and speed distribution of particles ejected. These characterize subsurface dynamics and the underlying physical mechanism, and are also important in estimating the total mass collection during a flyby for future missions. This section makes observations based on current CDA data and simulations to better understand initial particle distributions. Figure 3.7 plots a least-squares fit of the size-distribution slope based on CDA measurements and simulated data, over the E7 and E21 trajectories. Note that at some particle size, the number of impacts (simulated or observed) is zero, which cannot be included in a least-squares fit in log space. This makes the data somewhat noisy, but we can still extract some interesting information.

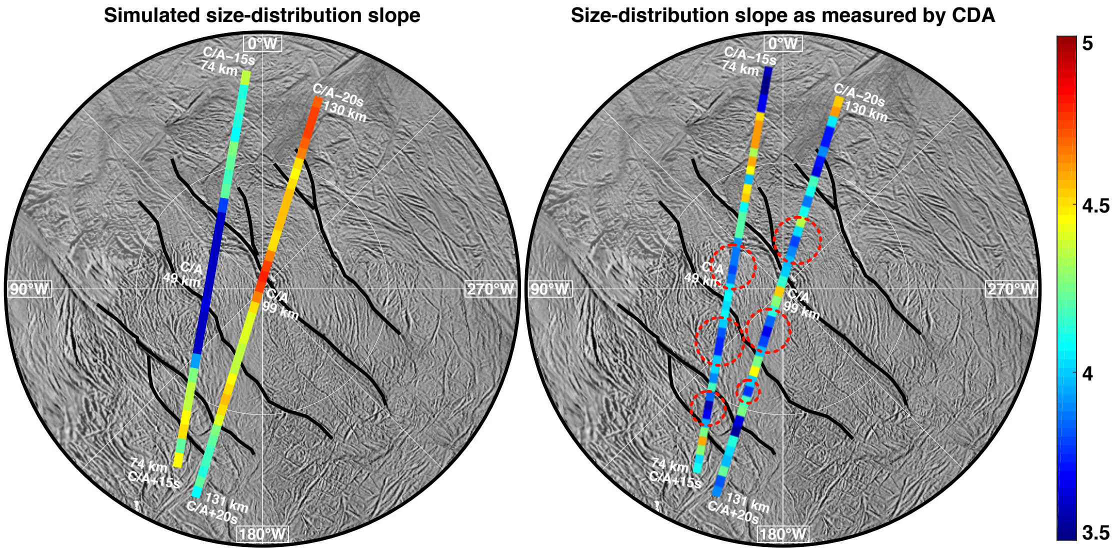


Figure 3.7: Optimal least-squares slope of power-law particle size distribution for simulated data and CDA data, along E7 and E21 trajectories.

First, note from Figure 3.7 that the observed size-distribution slope is actually quite comparable over the course of the two trajectories, despite the differing altitudes. Our speed distribution weights do not reproduce this effect: the simulated size-distribution slope grows steeper between 50km and 100km altitude. This can be seen by considering the slope as a function of altitude measured in the single-source example (Figure 3.4), as well as noting that a flatter initial distribution (dotted blue line) reproduces E7 (Figure 3.5) but is too flat for E21 (Figure 3.6), while a steeper initial distribution (black line) reproduces E21 (Figure

3.6) but is too steep for E7 (Figure 3.5). Figure 3.4 shows that modifying the initial size distribution does not affect how the slope changes as a function of altitude, and rather shifts the resulting slope profile up or down over all altitudes; thus, modifying the initial size distribution will not lead to a size distribution consistent at E7 and E21 altitudes. Appropriate modifications of size-distribution parameters can create such an effect (larger r_c, larger v_{gas}), but more data from the plume is needed to optimize the parameter space.[3]

Second, for both flybys, simulated data reproduces the first two detectors well, but underestimates the number of large particles detected in the third threshold. For E7, simulated data leads to no impacts for the third detector. Due to the number of particles simulated, the probability of a large plume particle reaching 100km or higher based on our current speed distribution is very small. This implies that *more* large particles will not address the discrepancy (due to, for example, a flatter initial size distribution); rather, large particles must be launched at higher velocities. Such a modification is also consistent with E21, where we underestimate the number of impacts in the third threshold by a factor of 3–4. In terms of the speed distribution, this corresponds to a larger critical radius, where large particles can be accelerated more efficiently by the gas. It is worth pointing out that this is inherently exciting from the perspective of mission planning. Large particles are much greater in mass (Section 3.3.3) and more likely to contain organics and signatures of biological activity. Results from E21, in particular, indicate that we can expect to collect (a potentially significant amount) more mass during low-altitude flybys than originally estimated for the Enceladus Life Finder mission.

Finally, it is interesting to note that observed CDA data in Figure 3.7 suggests the size distribution is steeper directly over the fractures and flatter between them, meaning more large particles are encountered between fractures, relative to the total impact rate. Although the simulated data has some similar behavior, it is less obvious, and appears to correspond closer to a steeper size distribution at closest approach. One possible explanation is that large particles are emitted with a larger opening angle, or, more specifically, a diffuse, curtain-like source with a flatter size distribution populates the area between fractures, while discrete jets with higher gas velocities and steeper size distributions dominate the impact rate above fractures. This

[3] It is also possible that the size distribution changed between 2009 and 2015, or varies by fracture or fracture location due to tidal stresses and fracture fill-in. However, more data is needed for such analysis as well.

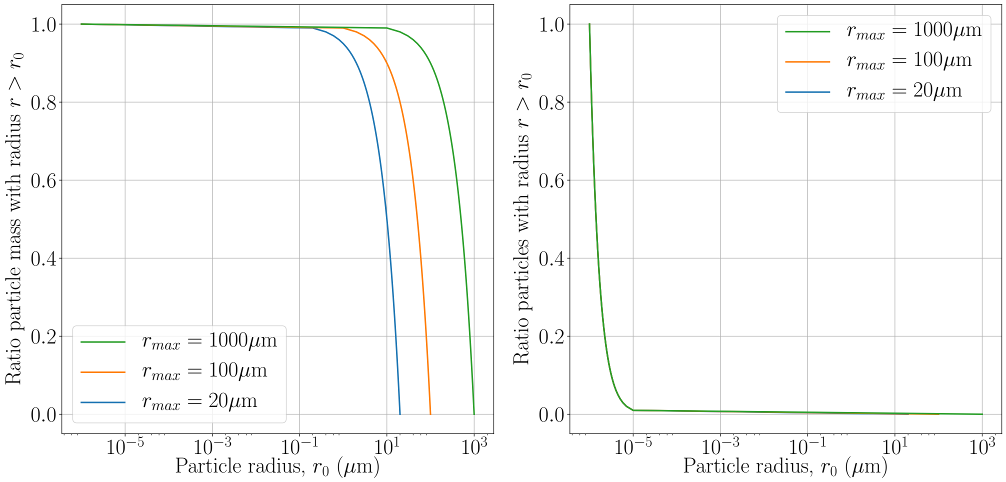
is only speculation, but such jetting activity amidst diffuse, curtain-like emissions has been seen in images, the study of which is ongoing work. Moreover, when large plume particles were first discussed in Postberg et al. [138], they were emitted with a larger opening angle for model purposes, but such an approach was not physically motivated. However, results here may support this hypothesis.

3.3.3 Mass production

The final and most significant result to discuss is total mass production of the Enceladus plume. Mass production and, in particular, the mass-to-vapor ratio, is fundamental to understanding the physical mechanism of the plume. A large mass-to-vapor ration would imply mass-loading of the emerging particle-gas flow. Plume particles could no longer be accelerated efficiently by the vapor flow, complicating the underlying physics. Mass production also has implications in the age of the rings, how long plumes have been active, and planning future flybys to collect enough particle mass for detecting biological signatures. Although temporal variability of the plume suggests that mass production is time-dependent, an average mass production is sufficient to answer many open questions.

Mass production is overwhelmingly dominated by large particles. Here, we only simulate particles of size 0.5 μm and larger. But, suppose the power-law size distribution extends back to nanograins. To demonstrate how large particles dominate the total mass production, even when small in number (due to a power-law distribution), Figure 3.8 shows the ratio of total particles and ratio of total particulate mass consisting of particles larger than a minimum radius; this can also be seen as a comparison of the average particle mass with the median particle mass. Recall from Chapter 2 that to integrate particle mass over a power-law size distribution, we must choose some minimum and maximum particle size. Here, we choose a minimum size of 10 nm, and show the results for three different maximum sizes: 20 μm, 100 μm, and 1000 μm. Even for the relatively small maximum size of 20 μm, 95% of the mass is in grains larger than one μm, while $\gg$ 99% of the particles have radius less than one μm.

Due to the significance of large particles in total mass, and because CDA is the only instrument to directly measure such particles, the most direct data available on mass production of the Enceladus plume is from low-altitude CDA flybys, which were able to count impacts from μm-sized particles. Previous estimates



(a) Average

(b) Median

Figure 3.8: A comparison of the average mass and median mass of plume particles based on a power-law size distribution. The particle radius (x-axis) corresponding to the true average mass and median mass are given at $y = 0.5$, and an equivalent measure is given for probabilities from zero to one.

of mass production have typically been through spectral analysis of images [85, 136, 181], which is an indirect measure, or modeling of nanograins [50, 112, 113], which do not make up a significant portion of the plume's mass. Kempf et al. [91] and Schmidt et al. [151] used CDA data and numerical modeling of plumes to estimate a total mass production, but, at the time, CDA had not yet performed any flybys with altitudes lower than a few hundred kilometers and, thus, had limited data on large particles. Overall, there has been a huge discrepancy in the estimated mass production rate, from a few kg/s [91] to more than 50 kg/s [85]. Here, simulations are used to reproduce data from the E7 and E21 flybys, and indicate a mass production on the order of 25 kg/s for *both* flybys. This is fairly consistent with other recent estimates of mass production [50, 112, 136], demonstrating a growing agreement between results based on multiple instruments. The growing evidence of plume variability over short time scales [71, 82, 148, 159] and long time scales [86] may help explain the variation in proposed mass production rates; that is, plume mass production likely does vary by a small factor over time.

Chapter 4

Plumes on Europa?

4.1 Evidence for plume activity

Owing to detailed data from the Cassini mission, the plume of Saturn's moon Enceladus is well studied. In contrast, not much is known about plume activity on Europa. Hubble observations of the Lyman-α brightness of the putative Europa plume in December 2012 imply an H_2O column density of $1.5 \cdot 10^{20}$ m^{-2} [144], which is comparable to the value of $9.0 \pm 2.3 \cdot 10^{19}$ m^{-2} measured at Enceladus by the Cassini Ultraviolet Imaging Spectrograph [68]. The derived plume scale height of about 200 km corresponds to a vertical gas speed at the surface of 700 m/s. This is similar to the Enceladus plume, which exhibits gas velocities of 300 to 2000 m/s [68, 175]. These similarities suggest that the Europa plume also contains at least a few mass percent of water ice grains, formed either by nucleation within the ascending vapor [151] and/or by mantle growth on frozen droplets from a subsurface water reservoir [138]. Note that although the magnitude of gas velocities is comparable, the entire range of speeds $300 - 2000$ m/s is greater than the Enceladus escape speed, while lower than the Europa escape speed, which affects the shape of the plume.

The consistency between gas velocities and H_2O column densities as seen on Enceladus and Europa suggests a similar physical mechanism driving their plumes. Here, we assume the same underlying model for a Europa plume as discussed in Chapter 1 for Enceladus. Nevertheless, plumes on Europa are expected to be different from those on Enceladus in several regards. Notably, the Europa plume is either not permanent or it is periodic in its activity [144, 166]. This sporadic nature of the Europa plumes suggests that they are activated by mechanical stress, opening cracks in the surface, which are then resealed due to either mechanical stress or water gas condensing along vent walls. Also, the scale-heights of plumes will differ significantly due to the difference in mass of Europa and Enceladus. Enceladus has a radius of only 250 km and the radius of its sphere of gravitational influence – the Hill radius – is $R_H = 950$ km, while Europa has a radius of $R_E = 1561$ km and a Hill radius of $R_H = 13,661$ km. Due to the (comparatively) massive size of Europa, charging of particles for the sizes that we are interested in (in the μm range) has a negligible effect on particle dynamics and, thus, Europa plume simulations neglect the effects of charging on a particle's trajectory.

In this work we will use spacecraft flybys from the Europa Clipper mission (see http://solarsystem.nasa.gov/missions/europaflyby/indepth) and refer to them as *Clipper flybys*. Two plume locations are taken as those suggested in Roth et al. [144], P1 and P2, and two additional locations, P3 and P4, at the footprints of the of the closest approach of two Clipper flybys (Table 4.1). Actual flybys by the EMFM will be very similar to these. Note, we define the 'closest approach' as the minimum elevation above the surface of Europa that the spacecraft attains during its trajectory. One chosen flyby, P4, is at a very low altitude in order to simulate the highest dust impact rate the spacecraft may experience. The other flyby, P3, is the planned trajectory that flies closest to one of the source locations suggested by Roth et al. [144]. Due to a lack of precise data on Europa plumes, we assume plumes to be directed orthogonally to the surface of Europa.

ID	$\lambda°$	$\phi°$	Clipper traj.	Closest approach
P1	-55	180	NA	NA
P2	-75	180	NA	NA
P3	-58.3	180.9	E26	52.2 km
P4	-60.4	356.1	E35	27.3 km

Table 4.1: Plume locations studied in this paper, given as northern latitude and eastern longitude. For P3 and P4 we also list the relevant flyby number of the Europa Clipper reference trajectory.

4.2 Plume Dynamics

Plumes on Europa are modeled in much the same way as on Enceladus (see Chapter 1), where we construct a quasi-steady state model for ejecta particles. In the case of Enceladus, the plume is believed to be continuously active and a steady state model makes sense. Because Europa plumes appear to be sporadic and much shorter lived, it should be noted that from a missions perspective, the typical plume traversal time of the spacecraft is on the order of 30 seconds. The lifetimes of particles in the plume can be approximately bounded using ballistic trajectories as $< 25v_{init}$ minutes, where generally $v_{init} \ll v_{gas}$. In this regard, the quasi-steady state model is justified for Europa plumes as well, as each of these timescales are significantly shorter than the expected duration of plume activity (at least 7 hours [144], potentially much longer).

The primary difference of Europa plume simulations with Enceladus simulations is that the speed distribution weights from Schmidt et al. [151] are not used; rather, the initial speed is sampled as an independent and identically distributed random variable from the analytic, size-dependent speed distribution in Equation (1.4). Because there is very limited data available on Europa plumes, this allows us to explore potential variations in the parameters that determine particle speeds, critical radius and gas velocity. We choose r_c between 0.3 and 0.8μm, corresponding to typical values expected for vents of decimeter width, connected to a deep source at triple point conditions for water [151].[1] Images of the Europa plumes from the Hubble telescope [144] suggest a gas velocity, and thus upper bound on the initial particle speed (Eq. (1.5)), of approximately 700 m/s. Although the estimate of 700 m/s is subject to some uncertainty based on image pixel size, we use this as the standard gas velocity for our simulations and will refer to it as the *base case*. This is comparable with the 500 m/s inferred by the ultraviolet imaging spectrograph (UVIS) as the broad emission velocity from Enceladus jets [175]. We also simulate a gas velocity of 1 km/s as a possible emission velocity for supersonic jets, a speed comparable with Enceladus supersonic jet gas velocities inferred by UVIS [68]. A gas velocity of 2.2 km/s, slightly above the Europa escape speed, is also simulated for comparative and exploratory purposes; however, we do not include the results in our primary discussion.

[1] Note that this work was done prior to the parameter analysis in Chapter 3, and the chosen values of r_c for Europa are likely larger than encountered on Enceladus. However, it is not necessarily the case that parameters on Enceladus will also be seen on Europa. Here, we simulate a range of parameters and expect (at least some of) our results to be fairly representative of potential plumes on Europa.

Recall the differential particle size distribution is assumed to follow a power law with slope α. We found $\alpha \approx 3$ to be appropriate on Enceladus (Chapter 3). A smaller α will result in a smaller plume profile and a higher probability of larger, potentially harmful particles for a spacecraft, while a larger α ejects more small particles with higher velocities, leading to significantly higher overall impact rates. To explore the parameter space, we consider $\alpha = 2.5, 3$, and 3.5. Last, the plume mass production rate is chosen to be 5 kg/s. This is smaller than estimated for Enceladus (Chapter 3) because Europa does not appear to have large fractures over which mass production is distributed; for a more localized emission source, we expect a correspondingly smaller mass production. Two dimensional particle density cuts for P1 with the base case velocity can be seen in Figure 4.1.

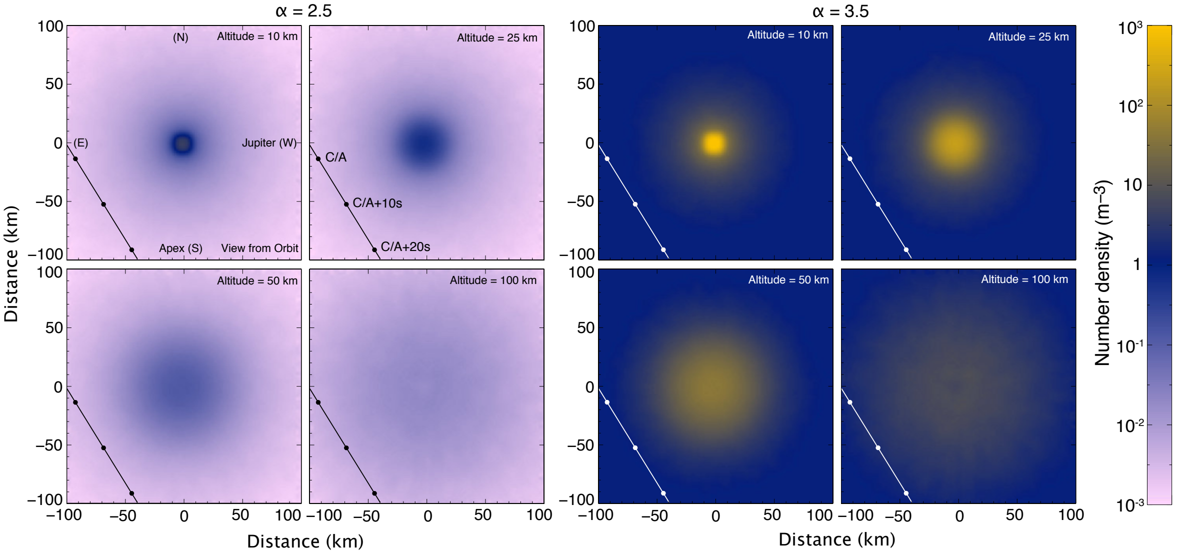


Figure 4.1: P1 particle number density cuts orthogonal to the surface of Europa at elevations 10 km, 25 km, 50 km, and 100 km, accounting for particles of size $0.6 - 5.00$ μm. Image view is from orbit looking at the surface, and the geometry is shown in the upper left image. The closest planned Clipper flyby, E26, is shown as a dotted line, with its closest approach to the surface (C/A) and location every 10 seconds thereafter marked in the second figure from top left. Plume parameters are given by $v_{gas} = 700$ m/s, $r_c = 0.8$ μm and $M^+ = 5$ kg/s, with size distribution slope α marked above the figures.

Plumes on Europa will be significantly more locally confined than the Enceladus plume, as a consequence of the different masses of the two moons. The difference in mass manifests in the two body escape velocity, which is 2.025 km/s for Europa, compared to 239 m/s for Enceladus. For a plume particle, the effective escape speed depends on the starting location and direction of ejection, due to the strong perturbations induced by the planet's gravity. A south polar map of effective escape speeds is shown in Figure 4.2, with speeds ranging from about 1.85 km/s to 2.25 km/s. If particle charging and the Lorentz force are taken into account, the effective escape speed may decrease slightly for smaller particles [91]. However, this will have only a minor effect on particle dynamics and plume shape. Overall, the gas velocities considered are is small compared to the effective escape speeds, which implies that practically all particles launched in such a plume will re-collide with the surface of Europa. Even at the extreme gas velocity of 2.2 km/s, over 99% of the ejected plume particles would re-collide with the surface of Europa. One immediate consequence of this is that Europa dust plumes do not contribute a substantial number particles to Jupiter's Galilean ring, which has been observed by the Galileo dust detector [95, 96].

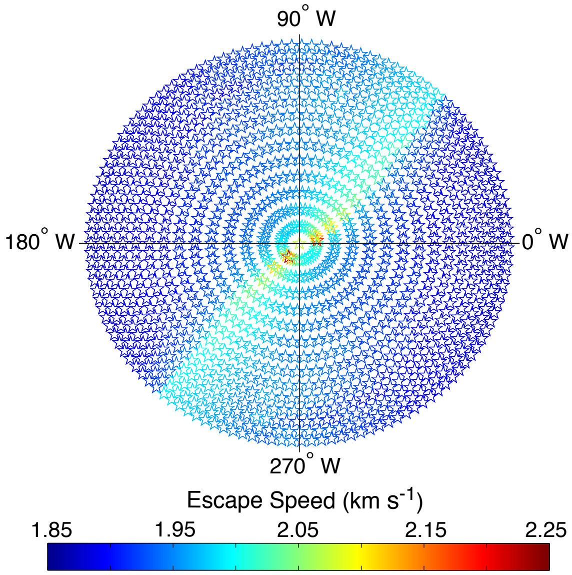


Figure 4.2: Effective particle escape speed in km/s for particles launched orthogonal to south polar terrain of Europa, 0°W points towards the planet and 90°W points in orbital direction. To determine the effective escape speed including three-body effects, we simulate particles launched orthogonally to the surface in a range of $0° - 60°$ from the pole, incrementing the initial velocity until the particle eventually escapes Europa's gravitational field. Note that because we do not consider particle charging and particle mass is negligible with respect to gravitational force, this escape speed map accounts for all particle sizes.

Due to the high escape speed of Europa, plume surface deposition must occur in a relatively small area centered about the plume. Given that v_{gas} provides an upper bound on ejection particle speeds, we can use v_{gas} and θ_{max} to determine a strict limit on the size of area resurfaced by particle deposition.

Recall that in our simulations we assume the particle ejection angle (measured from the surface normal) is bounded by $\theta \leq \theta_{max} = 45°$ and do not consider larger angles; here, we also examine possible lower values of $\theta_{max} = 15°, 30°$. We can estimate the maximal distance a particle may travel from the gravitational two-body problem. If ejected with a speed lower than the two body escape velocity $v_{esc} = \sqrt{2\mu_E/R_E}$, the particle follows a Kepler ellipse between ejection and re-impact

$$r = \frac{p}{1 + e\cos\phi} \tag{4.1}$$

where p and e are the semi-latus rectum and eccentricity of the orbit and ϕ is the true anomaly angle, measured from the pericentre. At start and re-impact we have $r = R_E$ and we obtain from equation (4.1) the values of the true anomaly at these sites

$$\phi_1 = \mathrm{acos}\left(\frac{\mathrm{p/R_E} - 1}{\mathrm{e}}\right), \quad \phi_2 = 2\pi - \phi_1\,. \tag{4.2}$$

We obtain for the distance on the surface between start and impact site

$$s = R_E\,(\phi_2 - \phi_1) = 2R_E\left[\pi - \mathrm{acos}\left(\frac{\mathrm{p/R_E} - 1}{\mathrm{e}}\right)\right]\,. \tag{4.3}$$

Using the well known expressions for the conservation of angular momentum and energy of the two body problem [40] we can express the semi-latus rectum and the eccentricity in terms of the starting speed v and ejection angle θ as

$$p = 2\sin^2\theta\left(\frac{v}{v_{esc}}\right)^2 \tag{4.4}$$

$$e^2 = 1 + 4\sin^2\theta\left(\frac{v}{v_{esc}}\right)^2\left[\left(\frac{v}{v_{esc}}\right)^2 - 1\right]\,. \tag{4.5}$$

Table 4.2 lists estimates from equation (4.3) for the farthest distance a particle may travel for given ejection parameters v_{gas}, θ_{max}. Although it is a tight bound, very few particles will achieve this distance, as the throw distance decreases for particles ejected with $\theta < \theta_{max}$ and most particles will have ejection speeds well below the gas velocity.

Equation (4.3) also provides an estimate for the surface area that may be resurfaced by a given plume. The extent to which resurfacing happens is then dependent on several other jet parameters. Mass production rate is the foremost plume parameter that will determine the rate of resurfacing, as the amount

	$\theta_{max} = 45°$	$\theta_{max} = 30°$	$\theta_{max} = 15°$
$v_{gas} = 700$ m/s	423 km	344 km	190 km
$v_{gas} = 1$ km/s	978 km	740 km	393 km

Table 4.2: Maximum distance from source plume particles will travel given maximum opening half-angle, θ_{max}, if they are ejected at gas velocity, v_{gas}.

of mass landing on the surface is approximately equal to that ejected from the plume. Critical radius r_c and the slope of the size distribution α also have notable effects on the breadth of resurfacing. A smaller critical radius means less particles will be ejected at high speeds comparable to the gas velocity, thus implying that a smaller area will be resurfaced. A flatter particle size distribution slope has a similar effect, as the plume will eject fewer small particles, which generally have higher velocities and dominate resurfacing far from the plume. These effects are demonstrated in Figure 4.3, where we show surface deposition from P1 with a higher gas velocity $v_{gas} = 1$ km/s, and several combinations of α, r_c.

For the base case, the effect of plume particles falling back onto Europa's surface is quite local, with a maximum distance from source < 423 km. This is in contrast to a collision map on Enceladus, where one sees a large, global pattern that spans more than half of Enceladus' surface [91, 150]. At $v_{gas} = 1$ km/s the area on Europa grows to < 978 km, depending on model parameters, as can be seen in Figure 4.3. Schenk et al. [150] determined a global color pattern on the surface of Enceladus and associated this with resurfacing due to plume particles. If similar techniques can demonstrate recent resurfacing on Europa due to plume activity, inspection of the dimension of the fall-back pattern may constrain the otherwise free parameters gas velocity and opening angle.

4.3 Implications for Mission Planning

Primary results from simulations are given in Table 4.3. We also constructed specific flyby trajectories to test minimum and maximum altitudes for impact rates, the results of which are discussed in this section. To perform safe and effective measurements during plume traversals, we propose maximizing the time the spacecraft spends between altitudes of 5-100 km. Given the uncertainty in parameters α, v_{gas}, and r_c, and that the location of active plumes are unknown, our results demonstrate that for most parameter sets the

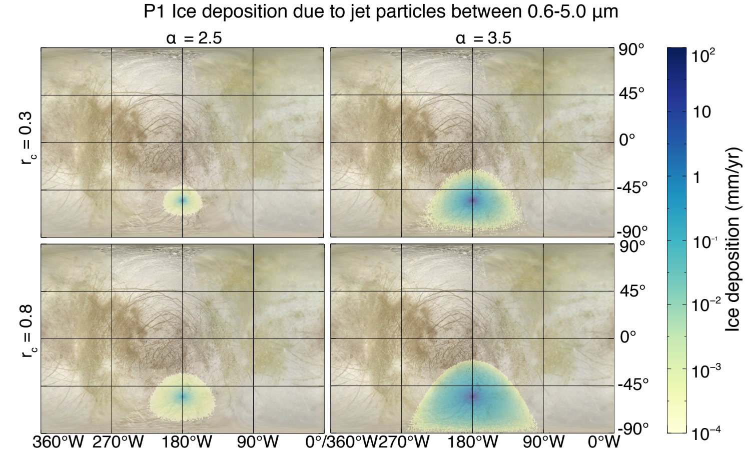

P1 Ice deposition due to jet particles between 0.6-5.0 µm
α = 2.5
α = 3.5
r_c = 0.3
r_c = 0.8
90°
45°
0°
-45°
-90°
360°W 270°W 180°W 90°W 0°/360°W 270°W 180°W 90°W 0°W
10²
10
1
10⁻¹
10⁻²
10⁻³
10⁻⁴
Ice deposition (mm/yr)

Figure 4.3: P1 surface ice deposition map due to jet particles between $0.6 - 5.0$ μm. Plume parameters are $v_{gas} = 1$ km/s, $r_c = 0.3, 0.8$ μm, and $\alpha = 2.5, 3.5$, with a plume production rate of $M^+ = 5$ kg/s.

spacecraft receives more than $O(1)$ particles / second at altitudes $\lesssim 100$ km for any plume source $\lesssim 75$ km lateral distance from the spacecraft trajectory. The effective upper altitude may be extended to 130 km (Figure 4.4) due to favorable values of α, r_c; however, with parameters that reduce plume size, e.g. a lower v_{gas} or flatter size distribution slope α, the spacecraft may not attain a significant number of impacts above around 100 km. In the unlikely scenario of a small $\alpha \leq 2.5$, this altitude may be reduced. We also must consider how many impacts the instrument can record per second. For a mass spectrometer, the length between two impacts must be shorter than the time-of-flight (TOF) signal to ensure that the spectra of separate impacts do not overlap [92]. This length is approximately 10 μs for the SUrface Dust Analyzer (SUDA), which corresponds to a maximum impact rate of less than 10^5 particles / second [92]. Flybys with altitudes ≥ 5 km have impact rates less than 10^5 for all simulated parameter sets.

Consistent with surface deposition and number density results, spacecraft impact rate is also highly sensitive on the value of α. Notice that if $\alpha = 2.5$ – simulations C13, C14, C15, and C19 – the spacecraft impact rate and length of impacts are both very small regardless of other parameters. We only included $\alpha = 2.5$ results for flybys with a closest approach at a plume location, as flybys that do not go directly over a plume at closest approach attain maximum impact rates less than 10^{-1} particles per second. Notice that in simulation pair (C19, C20) an increase from $\alpha = 2.5$ to $\alpha = 3$ with all other parameters fixed causes an

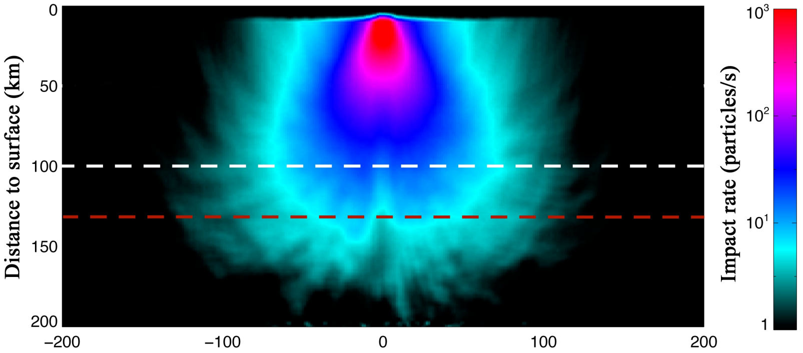


Figure 4.4: Vertical impact rate profile for P2, $v_{gas} = 700$ m/s, $r_c = 0.8$ μm, $\alpha = 3.5$. Recommended upper bound on spacecraft altitudes of 100 km is shown in white, with a parameter-specific larger threshold for detection at 130 km shown in red. At altitudes > 130 km a spacecraft will not detect more than a few particles/second.

increase in impact rate close to two orders of magnitude. Similarly, a change in slope from $\alpha = 3$ to $\alpha = 3.5$ increases the maximum impact rate by at least an order of magnitude as can be seen in pairs (C4, C6), (C5, C7), and (C20, C21). Comparing this with the effects of other parameters, aggregate plume dynamics appear to be the most sensitive to changes in α.

Next let us consider the effects of v_{gas}. A higher gas velocity increases the speed of ejected particles, and thus particles will travel farther from the plume. One consequence is that plumes could be detected by a spacecraft from larger distances. Simulations C22, C23, and C24 demonstrate this, with the spacecraft receiving several impacts per second for 1-2 minutes at total distances greater than 288 km from the plume when $v_{gas} = 1$ km/s. If we reduce v_{gas} to 700 m/s, these flybys result in a maximum impact rate less than 10^{-2} impacts per second. An increase in v_{gas} also often allows for longer periods of spacecraft detection, as can be seen in simulation pairs (C6, C7), (C8, C9). Notice that an increase from $v_{gas} = 700$ m/s to 1 km/s increases the length of time the spacecraft receives more than 1 impact per second by 15 and 40 seconds, respectively. At the same time if we assume a fixed mass production rate, particles traveling farther from the source due to a larger gas velocity also results in a lower number density close to the plume. Thus, for flybys that travel well within the detectable range of a plume with $v_{gas} = 700$ m/s, we can expect a decrease in the impact rate if we increase gas velocity to $v_{gas} = 1$ km/s. This is best seen in (C6, C7), wherein the maximum impact rate drops by about 33% with an increase in gas velocity.

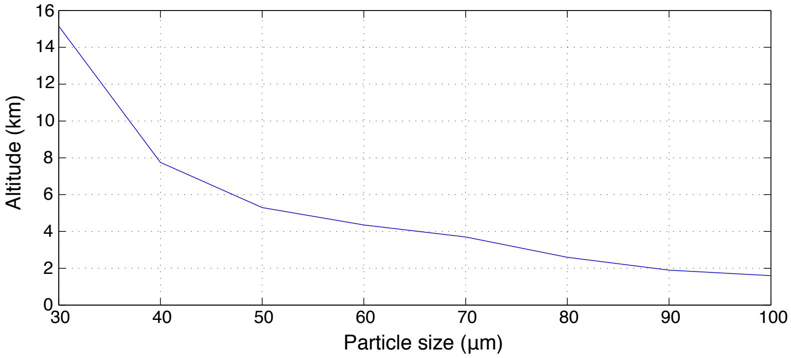
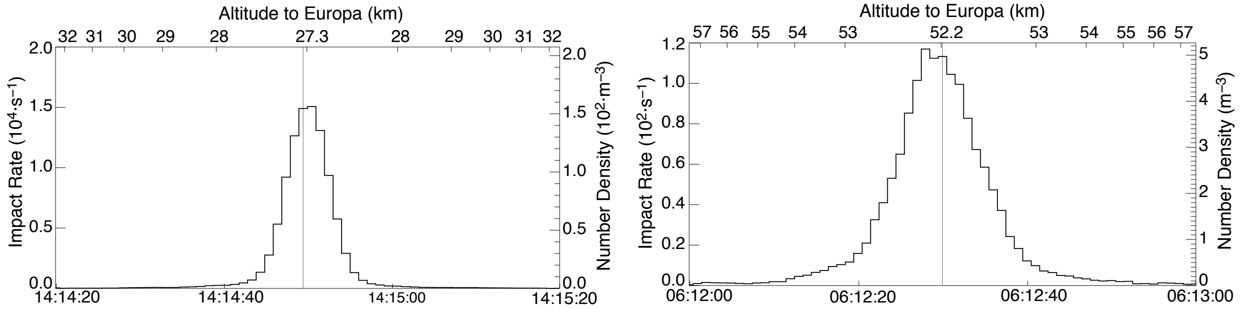


Figure 4.5: Lowest altitudes at which simulations indicate zero impacts with particles of size $\geq S$, for all given parameter sets considered.

The effects of a change in r_c on impact rate are primarily seen in flybys that do not pass directly over a plume at their closest approach. Since a change in r_c affects the average particle ejection velocity, a corresponding change in impact rate is best seen at distances where a shift in particle velocities manifests in a different particle density. Simulation sets C1 - C3, C10 - C12, and C25 - C27 each demonstrate the effect of r_c between 300 and 800 nm, with the impact rate increasing by a factor of $1.5 - 3$ with increasing r_c. For a set of flybys over a plume at closest approach, observe that the impact rate in simulations C13 - C15 is almost identical for all r_c. In general, a higher r_c can increase the spacecraft impact rate by some factor, but this factor is less significant than the change in impact rate due to a change in α or v_{gas}. It should also be noted that at very low elevations, $\lesssim$ 10 km, a small r_c can actually increase the impact rate because particles will be launched at lower velocities, leading to an increase in particle number density at low altitudes.

Spacecraft safety for a given particle impact depends on the particle size, impact speed, spacecraft orientation and spacecraft architecture. In regards to spacecraft safety, we establish the minimum altitudes at which we expect zero impacts with particles of size $\geq S$, for all given parameter sets considered in this paper. Results are shown in Figure 4.5. Note that because we sampled initial particle speeds and angles from probability distributions, zero expected impacts of size greater than S does not guarantee that the spacecraft does not encounter a particle of size greater than S, but means that such an encounter is highly unlikely. Based on Figure 4.5, we expect that recommended altitudes $5 - 100$ km should be safe for plume traversals, but we leave a full hazard analysis to spacecraft engineers.

We also show two sample impact rate profiles for a plume traversal where the closest approach of the spacecraft is directly above a plume, one at the closest approach of planned Clipper flyby E26 (April 18, 2030) with a minimum altitude of 52 km above the surface, and one at the closest approach of Clipper flyby E35 (March 26, 2031) with minimum altitude 27 km. Figure 4.6 shows the impact rate for each of these simulations over a one minute timespan surrounding the closest approach. The low-altitude E35 flyby has a more intense impact rate, on the order of 10^4 particles per second at it's peak, but still beneath the approximate upper bound on impact rates that SUDA can detect of 10^5 particles per second [92]. Although unclear in the images due to differing scales, the E35 flyby also receives at least several impacts per second for a longer period of time than the higher altitude E26 flyby. This is consistent with expectations as it spends more time at recommended altitudes less than 100 km.



(a) Clipper E35

(b) Clipper E26

Figure 4.6: Number density and impact rate for plume traversals with a plume located at the closest approach, and plume parameters $v_{gas} = 700$ m/s and $r_c = 0.8$ μm.

4.3.1 Large particles

Finally, how many particles will be collected of a given size is also of interest for planning flyby altitudes, particularly for larger particles and low-altitude flybys. This will answer the question, "if we want to collect particles of size, for example, > 20 μm, at what altitude must the spacecraft traverse a plume?" Larger particles are of particular interest in seeking organic material. Results of this section were used in the development of the Sylph free-flyer proposal, a small detachable dust detector that would eject from the spacecraft to perform a single, very low-altitude traversal of a plume on Europa to collect large plume particles [84].

Figure 4.7 and Table 4.4 show the total number of particles greater than or equal to a given size collected in a given flyby, for various flyby altitudes, particle sizes, and parameter choices. We assume 5 kg/s ejection rate, sensitive area (SA) of 4 cm^2 in Figure 4.7 and 1 m^2 in Table 4.4, and that the flyby travels directly over the plume source at closest approach. For different sensitive areas, results must be scaled accordingly. Recall low-altitude Enceladus flybys encountered significantly more large particles than the model predicts (Chapter 3). However, here values of r_c are used that more favorable for large particles compared with the weights computed in Schmidt et al. [151] and used in our Enceladus simulations. Thus, results in Table 4.4 and Figure 4.7 likely average out to be representative of particle collection on Europa.

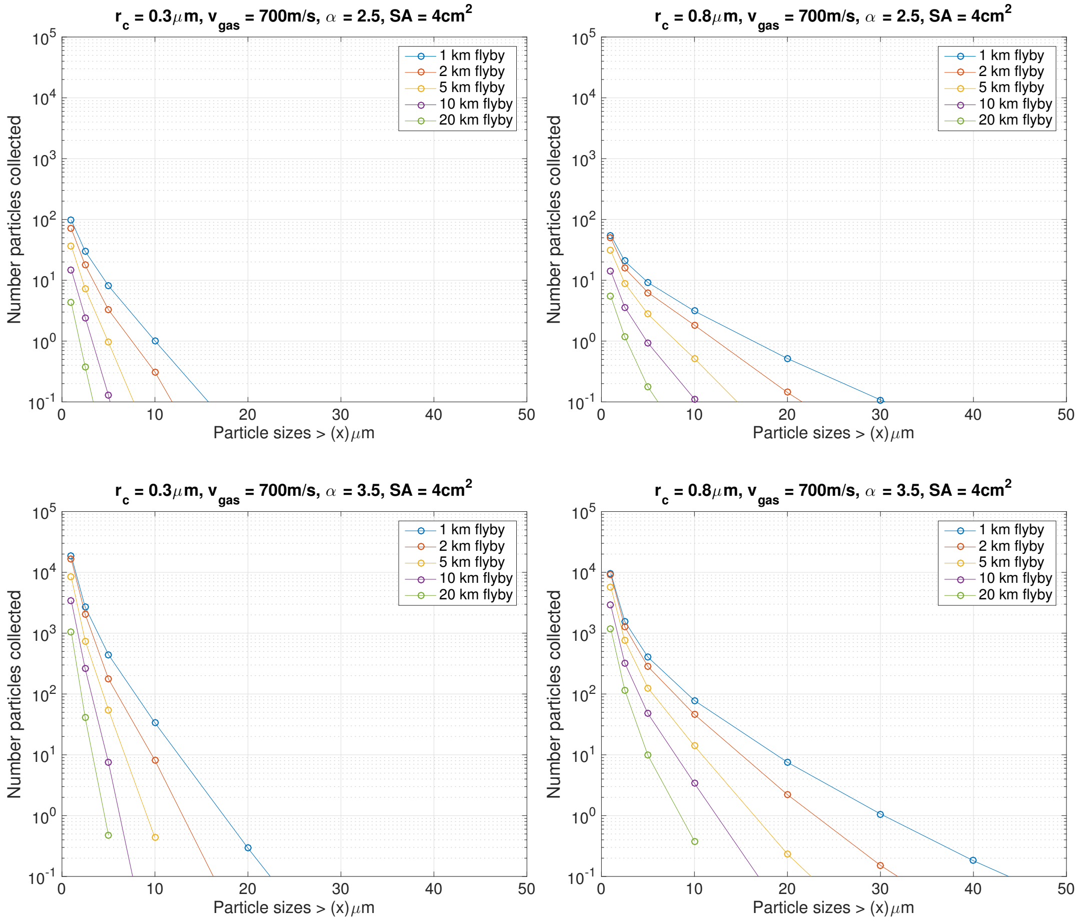


Figure 4.7: Total particles collected as a function of particle size for flyby altitudes $1, 2, 5, 10$ and 20 km. Results shown for two different critical radii, $r_c \in \{0.3, 0.8\}$, two size distribution slopes $\alpha \in \{2.5, 3.5\}$, and a gas velocity of 700 m/s.

4.4 Conclusions

We find that putative plumes on Europa will differ from the Enceladus plume in a number of ways, bearing consequences for the scientific study and prospects for discovery. The necessarily smaller size of Europa plumes provide only a short time window for detection with an in situ dust instrument. Additionally, because of Europa's comparatively larger size, a single flyby can only test for plume activity over a small fraction of Europa's surface.

Identifying sites of resurfacing due to plume activity will help resolve currently poorly known parameters of the plume models, by constraining the size of the resurfaced area as well as the gradient at which the rate of resurfacing decays away from the source location. There is no current physical evidence suggesting plume activity will occur (or not occur) in specific regions of Europa. Evidence suggests that Europa plumes are of a sporadic nature, in which case detecting plume activity requires strategically planned spacecraft flybys that optimize the probability of capturing active plumes. Our simulations indicate that flybys maintaining a spacecraft altitude of $5 - 100$ km are optimal in detecting plumes within 75 km lateral distance of the spacecraft trajectory. Note that it is likely for a spacecraft to detect plumes at greater horizontal and vertical distances, but we present these numbers as an estimated lower bound. Multiple flybys can be strategically planned as non-overlapping, 150 km wide search-stripes to cover a given region of Europa's surface. However, to collect large particles greater than a few μm in radius, the spacecraft may have to fly at 5 km or lower.

Table 4.3: Flyby simulation results referenced by C#, for a given configuration. The spacecraft's closest approach to the surface of Europa and plume source are notated C/A Surf. and C/A Pl., respectively. We define C/A Pl. as the minimal distance between the spacecraft trajectory and plume source location on the surface of Europa. Note that if the spacecraft closest approach is directly over a plume location, then C/A Surf = C/A Pl. "Max. Imp." refers to the maximum impact rate a spacecraft experiences, and Len. Imp. refers to the approximate length of time in which the spacecraft receives at least one impact per second.

Sim #	Plume	Flyby	C/A Surf. (km)	C/A Pl. (km)	v_{gas} (km/s)	r_c (μm)	α	Max. Imp. (part/s)	Len. Imp. (s)
C1	P1	E9	26.7	161.9	0.7	0.3	3.5	3.7	40
C2	P1	E9	26.7	161.9	0.7	0.55	3.5	6.4	95
C3	P1	E9	26.7	161.9	0.7	0.8	3.5	7.4	120
C4	P1	E26	52.2	99.6	0.7	0.55	3	1.2	5
C5	P1	E26	52.2	99.6	1	0.55	3	1.0	1
C6	P1	E26	52.2	99.6	0.7	0.55	3.5	21.8	105
C7	P1	E26	52.2	99.6	1	0.55	3.5	14.1	130
C8	P2	E25	53.1	220.2	0.7	0.8	3.5	3.9	70
C9	P2	E25	53.1	220.2	1	0.8	3.5	4.2	110
C10	P2	E27	52.5	152.7	0.7	0.3	3.5	22.4	60
C11	P2	E27	52.5	152.7	0.7	0.55	3.5	51.4	80
C12	P2	E27	52.5	152.7	0.7	0.8	3.5	67.5	95
C13	P3	E26	52.2	52.2	1	0.3	2.5	1.9	10
C14	P3	E26	52.2	52.2	1	0.55	2.5	2.0	12
C15	P3	E26	52.2	52.2	1	0.8	2.5	2.0	12
C16	P3	E26	52.2	52.2	1	0.3	3	43.1	40
C17	P3	E26	52.2	52.2	1	0.55	3	46.2	35
C18	P3	E26	52.2	52.2	1	0.8	3	42.0	30
C19	P4	E35	27.3	27.3	0.7	0.55	2.5	14.2	10
C20	P4	E35	27.3	27.3	0.7	0.55	3	1002	60
C21	P4	E35	27.3	27.3	0.7	0.55	3.5	20462	75
C22	P4	E37	52.8	288.5	1	0.3	3.5	3	80
C23	P4	E37	52.8	288.5	1	0.55	3.5	5	130
C24	P4	E37	52.8	288.5	1	0.8	3.5	4.5	170
C25	P4	E41	52.0	179.6	0.7	0.3	3.5	45.6	25
C26	P4	E41	52.0	179.6	0.7	0.55	3.5	51.1	50
C27	P4	E41	52.0	179.6	0.7	0.8	3.5	69.3	70

Table 4.4: Total particles collected from putative Europa plume by spacecraft with SA 1 m^2.

		$\alpha = 2.5$					$\alpha = 3.5$				
		Size $\geq$ (μm)	1 km flyby	2 km flyby	5 km flyby	25 km flyby	Size $\geq$ (μm)	1 km flyby	2 km flyby	5 km flyby	25 km flyby
$r_c = 0.8$ μm,	$v_{gas} = 0.7$ km/s	5	22321.750	18152.434	6188.656	158.24	5	978708.25	842033.83	313874.69	7781.037
		10	7779.239	5353.4054	1288.521	16.100	10	195057.41	136452.17	34871.92	398.222
		20	1290.9650	515.357	36.865	0.055	20	18851.283	7876.6487	583.925	0.595
		30	269.994	66.145	2.049	0	30	2662.361	669.006	20.823	0
		40	61.782	11.769	0.368	0	40	453.147	89.882	2.895	0
		50	16.138	1.657	0	0	50	94.593	9.895	0	0
	$v_{gas} = 1$ km/s	5	21274.206	17861.403	7443.215	428.24823	5	855635.42	766593.81	350988.16	20576.824
		10	8250.662	6264.991	2073.289	69.979	10	188899.48	150718.92	53150.456	1743.753
		20	2158.848	1118.640	210.435	1.822	20	29245.585	16134.08	3243.58	21.718
		30	675.785	249.480	24.197	0	30	6269.885	2399.06	249.407	0
		40	237.0151	72.228	2.973	0	40	1665.29	525.899	22.852	0
		50	87.091	20.248	0.331	0	50	491.961	117.798	2.058	0
$r_c = 0.55$ μm,	$v_{gas} = 0.7$ km/s	5	20748.762	16251.031	4392.777	8.572	5	971977.91	805281.71	233783.40	342.573
		10	6350.766	3733.851	668.505	2.981	10	170036.88	98558.419	18852.375	60.809
		20	606.655	155.512	4.49	0	20	9398.235	2480.263	73.002	0
		30	64.309	6.735	0	0	30	668.647	70.441	0	0
		40	6.283	0.554	0	0	40	48.678	4.362	0	0
		50	0.476	0	0	0	50	2.956	0	0	0
	$v_{gas} = 1$ km/s	5	22047.819	17972.952	6327.794	192.376	5	963488.58	830700.03	316690.48	9490.275
		10	7747.029	5376.149	1399.608	17.991	10	194015.67	137076.63	37786.006	441.818
		20	1307.485	525.861	36.508	0.291	20	19151.114	7967.296	568.973	3.110
		30	268.723	77.223	1.727	0	30	2666.403	784.21	17.987	0
		40	57.781	11.695	0	0	40	426.123	86.16	0	0
		50	14.604	3.0113	0	0	50	86.745	17.905	0	0
$r_c = 0.3$ μm,	$v_{gas} = 0.7$ km/s	5	12047.771	9530.172	1266.59	0.206	5	621106.30	518569.88	72849.089	10.647
		10	2520.479	1155.664	37.671	0	10	72724.763	28757.695	1080.962	0
		20	45.273	2.596	0	0	20	730.707	42.215	0	0
		30	0.788	0	0	0	30	8.439	0	0	0
		40	0	0	0	0	40	0	0	0	0
		50	0	0	0	0	50	0	0	0	0
	$v_{gas} = 1$ km/s	5	17917.699	14480.247	3610.997	19.256	5	873801.2	740570.04	174993.37	985.785
		10	4849.365	2849.352	405.027	0.341	10	134226.79	77097.987	9395.673	8.582
		20	277.276	61.761	0	0	20	4374.731	990.0	0	0
		30	18.62	2.074	0	0	30	194.582	22.225	0	0
		40	1.412	0	0	0	40	10.656	0	0	0
		50	0.277	0	0	0	50	1.722	0	0	0

PART II: Broadening the applicability of algebraic multigrid

Chapter 5

Algebraic multigrid

5.1 Iterative methods and algebraic multigrid

Linear systems $A\mathbf{x} = \mathbf{b}$ and their solution are ubiquitous in scientific computing and often a computationally intensive task. In many cases, the matrix A is "sparse," meaning that the total number of nonzeros in the matrix is $O(n)$. For example, A may be a $10^{10} \times 10^{10}$ matrix, but only have 10 nonzero elements per row. Such matrices are particularly common when considering the discretization of partial differential equations (PDEs), which will be the focus of this work. Explicitly inverting the matrix $A \in \mathbb{R}^{n\times n}$ through, for example, a singular value decomposition (SVD) has a computational complexity of $O(n^3)$, meaning that it takes on the order of n^3 floating-point operations (FLOPs) to invert A using the SVD. Problems of interest in high-performance simulation codes often lead to n on the order of at least billions, making a direct inversion intractable for current computing power. This has spawned a large field of methods to solve $A\mathbf{x} = \mathbf{b}$ in an iterative manner, improving an approximate solution each iteration until a desired accuracy is achieved. Krylov-type iterative methods are often guaranteed to converge in n iterations and, if each iteration costs $O(n)$ (which it does in the case of a sparse matrix), the total cost is bounded by $O(n^2)$. However, even this is too expensive for many applications requiring high-resolution simulations. Thus, there is a need in scientific computing for "fast" solvers that are linear or near-linear in complexity, that is, the cost of solving $A\mathbf{x} = \mathbf{b}$ is $O(n)$ or $O(n \log n)$. Successful fast solvers have been developed for certain classes of problems, but there remain important problems that lack an efficient and fast solver.

Algebraic multigrid (AMG) is among the fastest class of algorithms for solving sparse linear systems that result from the discretization of partial differential equations (PDEs) of elliptic type [15]. When applicable, AMG converges in linear complexity with the number of degrees-of-freedom (DOFs), and scales in parallel like $O(\log_2(P))$, up to hundreds of thousands of processors, P [9]. Originally, AMG was designed for symmetric positive definite (SPD) linear systems, and performs best when applied to the discretization of elliptic PDEs, or the spatial discretization of a parabolic PDE in time. Although this constitutes a large class of problems that arise in scientific simulations, many problems of interest remain that are difficult to solve for AMG, or any other fast solver. Problems with strongly anisotropic components, and problems arising in particle transport, advective flow calculations, and strongly varying material properties, among others, challenge the standard approaches to AMG, thus highlighting the need for more robust methods.

There have been a number of efforts in recent years to improve the convergence and scope of applicability of AMG. Adaptive methods focus on improving the multigrid hierarchy through trial cycles in the setup phase [16, 26, 26, 27, 39]. Other methods focus on modified or improved strength measures when forming coarse grids [14, 17, 19, 21, 23, 39, 102, 125, 128]. Furthermore, generalizing interpolation through energy minimization [130] and other methods [34, 43, 63, 184] is used to improve the accuracy of interpolation between grid levels. Nevertheless, many problems remain difficult for AMG to solve, while many "robust" AMG methods suffer from high computational cost. Because of this, solutions to such problems are generally obtained through Krylov methods or direct solves, each of which are computationally expensive, limiting the resolution and accuracy at which simulations can be run.

Algebraic multigrid applied to the linear system $A\mathbf{x} = \mathbf{b}$, $A \in \mathbb{R}^{n\times n}$, consists of two processes: relaxation and coarse-grid correction, typically designed to be complementary in the sense that they reduce error associated with different parts of the spectrum. Relaxation takes the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + M^{-1}(\mathbf{b} - A\mathbf{x}_k), \tag{5.1}$$

where M^{-1} is some approximation to A^{-1} such that the action of M^{-1} can be easily computed. For example, M may be the diagonal of A (Jacobi) or the upper- or lower-triangular block of A (Gauss-Seidel). Error associated with large eigenvalues of A, generally corresponding to geometrically high-frequency error that cannot be well-represented on a coarse grid, is targeted through relaxation. Coarse-grid correction is then built to reduce error not reduced by relaxation, typically algebraically smooth error, that is, modes associated with small eigenvalues. Coarse-grid correction takes the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + P(RAP)^{-1}R(\mathbf{b} - A\mathbf{x}_k), \tag{5.2}$$

where $P \in \mathbb{R}^{n \times n_c}$ and $R \in \mathbb{R}^{n_c \times n}$ are interpolation and restriction operators, respectively, between $\mathbb{R}^n$ and the next coarser grid in the AMG hierarchy, $\mathbb{R}^{n_c}$. Here, $P(RAP)^{-1}R$ acts as an approximate inverse to A (like M^{-1} in relaxation) by projecting the problem to a smaller space, $RAP \in \mathbb{R}^{n_c \times n_c}$, solving the system there, and interpolating the result back to the initial space. If the coarse-grid operator is still too large to explicitly invert, then AMG is called recursively on the coarse-grid operator.

A two-level $V(1,1)$-cycle is given by combining coarse-grid correction in (5.2) with pre- and post-relaxation steps as in (5.1), resulting in a two-grid error propagation operator of the form

$$E_{TG} = (I - M^{-1}A)(I - P(RAP)^{-1}RA)(I - M^{-T}A). \tag{5.3}$$

The goal in building an AMG solver is typically to bound $\|E_{TG}\| \ll 1$ in some norm, such that AMG iterations are guaranteed to converge quickly. In contrast to geometric multigrid (GMG), which typically uses simple interpolation routines and focuses on more robust relaxation methods, the focus in AMG is on building effective interpolation and restriction operators. AMG convergence theory is meant to bound error propagation with some constraints on P and R that guide how to construct them for a robust solver.

For a given matrix, A, the next "coarser" matrix in the hierarchy is generally defined in one of two different ways: using a CF-splitting of points or an aggregation of points. In aggregation-style multigrid, a measure of the strength-of-connection (SOC) between nodes is used to form "aggregates," which are disjoint sets of strongly connected nodes, where each aggregate represents one node on the coarse grid; transfer operators are then formed based on aggregates and typically some "smoothing" process over columns of P [88, 115, 125, 177, 178]. A CF-splitting splits the set of all DOFs of matrix A into a coarse set of C-points and a fine set of F-points, with C-points corresponding to DOFs on the coarse grid. Transfer operators are then defined using the CF-splitting, where values at C-points are restricted and interpolated by injection (that is, by value) [15, 111, 145] and values at F-points use a linear combination of connected neighboring points. Let n_f denote the number of F-points and n_c the number of C-points, and, for example, let A_{ff}

denote the block of F-to-F connections in A. Then, operators A, P, and R can be written in the following block forms:

$$A = \begin{pmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{pmatrix}, \quad P = \begin{pmatrix} W \\ I \end{pmatrix}, \quad R = \begin{pmatrix} Z & I \end{pmatrix}, \tag{5.4}$$

where $W \in \mathbb{R}^{n_f \times n_c}$ interpolates to F-points on the fine grid via linear combinations of coarse-grid DOFs, and $Z \in \mathbb{R}^{n_c \times n_f}$ restricts F-point residuals to the coarse grid. Note that (5.4) implicitly assumes the same CF-splitting for R and P, although the sparsity patterns for nonzero elements of Z^T and W may be different. For notation, define the coarse-grid operator and projection onto $\mathcal{R}(P)$, respectively, as

$$\begin{aligned} \mathcal{K} :&= RAP \\ &= ZA_{ff}W + A_{cf}W + ZA_{fc} + A_{cc}, \\ \pi_A &:= P(RAP)^{-1}RA. \end{aligned} \tag{5.5}$$

Here, $\mathcal{K}$ is used to denote the coarse-grid operator instead of the traditional notation, A_c, to avoid confusion with subscripts denoting C-points.

The remainder of numerical methods studied in this work will use a CF-splitting, with operators of the form shown in (5.4) and (5.5).

5.2 AMG: The SPD case

Let $A \in \mathbb{R}^{n \times n}$ be SPD, $\|\cdot\|$ and $\|\cdot\|_A$ represent the l^2- and A-norms, respectively, and $P : \mathbb{R}^{n_c} \to \mathbb{R}^n$ be an interpolation operator defining a coarse space of size n_c. For SPD matrices, $\|\mathbf{x}\|_A^2 := \langle A\mathbf{x}, \mathbf{x}\rangle$ defines the so-called *energy norm* or A-norm, which is used to measure the convergence of AMG and other iterative methods. In this case, defining restriction $R := P^T$ ensures that coarse-grid correction is an A-orthogonal projection. Orthogonality ensures that error in the A-norm cannot be increased through coarse-grid correction. A non-orthogonal coarse-grid correction is one of the primary difficulties with nonsymmetric AMG, and for SPD problems there is rarely good reason to choose $R \neq P^T$.

Let E_{TG} and E_{MG} denote the two-grid and multigrid error-propagation operators, respectively. Then the goal of AMG convergence theory is to develop bounds on E_{TG} and E_{MG} in the A-norm. For two-grid convergence, the *weak approximation property* (WAP) provides necessary and sufficient conditions, as well as a tight bound on $\|E_{\text{TG}}\|_A$ [57, 180]:

Theorem 1 (Weak approximation property). *Let A be SPD, $\widetilde{M} = M^T(M + M^T - A)^{-1}M$ for some relaxation scheme M, and P the interpolation operator for a two-grid method. Suppose for any $\mathbf{v} \neq 0$, there exists a $\mathbf{v}_c$ such that*

$$\frac{\|\mathbf{v} - P\mathbf{v}_c\|^2_{\widetilde{M}}}{\|\mathbf{v}\|^2_A} \leq K. \tag{5.6}$$

Then the two-grid method converges uniformly, and $\|E_{TG}\|_A \leq 1 - \frac{1}{K}$. Furthermore, the best (minimal) constant K over all P is given by

$$K_{TG} = \max_{\mathbf{v}} \frac{\|(I - \pi_{\widetilde{M}})\mathbf{v}\|^2_{\widetilde{M}}}{\|\mathbf{v}\|^2_A}, \tag{5.7}$$

in which case $\|E_{TG}\|_A = 1 - \frac{1}{K_{TG}}$.

Here $\widetilde{M}$ is the so-called "symmetrized smoother" and is primarily a theoretical tool to ensure an A-symmetric error-propagation operator [57, 180]. Equation (5.7) gives a sharp bound on two-grid convergence, but can be generalized to any SPD matrix X spectrally equivalent to $\widetilde{M}$, $X \sim_s \widetilde{M}$, that is:

$$0 \leq c_1 \mathbf{v}^T X \mathbf{v} \leq \mathbf{v}^T \widetilde{M} \mathbf{v} \leq c_2 \mathbf{v}^T X \mathbf{v}, \tag{5.8}$$

for all $\mathbf{v}$ and $0 < c_1 \leq c_2$. Denote $\pi_X := P(P^T X P)^{-1} P^T X$ the X-orthogonal projection onto $\text{Im}(P)$. Then,

$$c_1 \max_{\mathbf{v}\neq 0} \frac{\|(I - \pi_X)\mathbf{v}\|^2_X}{\|\mathbf{v}\|^2_A} \leq c_1 \max_{\mathbf{v}\neq 0} \frac{\|(I - \pi_{\widetilde{M}})\mathbf{v}\|^2_X}{\|\mathbf{v}\|^2_A} \leq K_{TG} \leq \max_{\mathbf{v}\neq 0} \frac{\|(I - \pi_X)\mathbf{v}\|^2_{\widetilde{M}}}{\|\mathbf{v}\|^2_A} \leq c_2 \max_{\mathbf{v}\neq 0} \frac{\|(I - \pi_X)\mathbf{v}\|^2_X}{\|\mathbf{v}\|^2_A}. \tag{5.9}$$

In the case of $X = I$, (5.9) simplifies to considering interpolation error in the l^2-norm [189]:

$$\lambda_{\min}(\widetilde{M}) \max_{\mathbf{v}\neq 0} \frac{\|(I - Q_P)\mathbf{v}\|^2}{\|\mathbf{v}\|^2_A} \leq K_{TG} \leq \lambda_{\max}(\widetilde{M}) \max_{\mathbf{v}\neq 0} \frac{\|(I - Q_P)\mathbf{v}\|^2}{\|\mathbf{v}\|^2_A},$$

motivating the often-used simpler form of the WAP: there exists $K \in \mathbb{R}$ such that for any vector $\mathbf{v} \in \mathbb{R}^n$,

$$\min_{\mathbf{w}_c \in \mathbb{R}^{n_c}} \|\mathbf{v} - P\mathbf{w}_c\|^2 \leq \frac{K}{\|A\|} \|\mathbf{v}\|^2_A. \tag{5.10}$$

The necessarily complementary role of relaxation and coarse-grid correction in AMG is accounted for in the WAP by requiring interpolation accuracy with respect to $\widetilde{M}$, that is, the coarse-grid correction must account for low-eigenvalue modes of $\widetilde{M}$, which are not effectively reduced through relaxation with $\widetilde{M}$. Equation (5.10) is equivalent to Theorem 1 with Richardson relaxation.

The *strong approximation property* (SAP) establishes multilevel convergence with a stronger condition than the WAP: there exists $K \in \mathbb{R}$ such that for any vector $\mathbf{v} \in \mathbb{R}^n$,

$$\min_{\mathbf{w}_c \in \mathbb{R}^{n_c}} \|\mathbf{v} - P\mathbf{w}_c\|_A^2 \leq \frac{K}{\|A\|}\|A\mathbf{v}\|^2. \tag{5.11}$$

If (5.11) holds on each level of the hierarchy, then $\|E_{MG}\|_A = 1 - \frac{1}{K_{MG}}$ and $1 \leq K_{MG} \leq 1 + K\frac{\|\widetilde{M}\|}{\|A\|}$, where $\widetilde{M}$ is the symmetrized relaxation scheme [180].

Since A is assumed to be SPD, its eigenvectors form an l^2- and A-orthonormal basis for the space $\mathbf{R}^n$. Thus, if the WAP and SAP hold for all eigenvectors, they hold for all vectors, and it follows that the WAP requires eigenvectors be interpolated with accuracy on the order of the corresponding eigenvalue, and the SAP requires interpolation accuracy on the order of the eigenvalue squared. This leads to an equivalence of satisfying the WAP based on A^2 and the SAP for A as follows:

Lemma 1 (Lemma 5.20 [179])**.** *Let $A \in \mathbb{R}^{n\times n}$ be SPD and $P \in \mathbb{R}^{n\times n_c}$. Then*

$$\min_{\mathbf{w}_c \in \mathbb{R}^{n_c}} \|\mathbf{v} - P\mathbf{w}_c\|^2 \leq \frac{K^2}{\|A^2\|}\|\mathbf{v}\|_{A^2}^2 \quad \textit{for all } \mathbf{v}, \tag{5.12}$$

if and only if

$$\min_{w_c} \|\mathbf{v} - P\mathbf{w}_c\|_A^2 \leq \frac{K}{\|A\|}\|A\mathbf{v}\|^2 \quad \textit{for all } \mathbf{v}. \tag{5.13}$$

The accuracy demands of the WAP and SAP with respect to eigenvalues indicates that the range of P should contain eigenvectors of A associated with small eigenvalues (or so-called *algebraically smooth* modes). In building AMG hierarchies, this is generally approached through some combination of, (i) ensuring that known low-energy modes are exactly represented in the range of P, and (ii) minimizing columns of P in the A-norm so that the range of P corresponds to algebraically smooth vectors. This will be explored in more detail in Chapter 6.

Now consider the case of CF-AMG (5.4). The minimizing coarse-grid vector, $\mathbf{w}_c$, in the WAP and the SAP is given by $\widetilde{M}$-orthogonal and A-orthogonal projections of the vector $\mathbf{v}$ onto the range of P, respectively.

In practice, such projections are generally too expensive to form explicitly; thus, computable measures are also of interest. One option consistent with classical AMG is to let $\mathbf{w}_c = \mathbf{v}_c$, that is, assume that the best preimage of $\mathbf{v}$ under P is the restriction of $\mathbf{v}$ to C-points, $\mathbf{v}_c$. This provides a bound on the (ℓ^2-) WAP, as $\min_{\mathbf{w}_c\in\mathbb{R}^{n_c}} \|\mathbf{v} - P\mathbf{w}_c\|^2 \leq \|\mathbf{v} - P\mathbf{v}_c\|^2$ for all $\mathbf{v}$, and thus

$$\mu(P) := \max_{\mathbf{v}\neq 0} \frac{\|\mathbf{v} - P\mathbf{v}_c\|^2}{\|\mathbf{v}\|_A^2} \geq K_{TG}. \tag{5.14}$$

Assuming $P = \begin{pmatrix} W \\ I \end{pmatrix}$, the optimal interpolation operator under $\mu(P)$ is given by

$$P_{\text{ideal}} = \operatorname*{argmin}_P \max_{\mathbf{v}} \frac{\|\mathbf{v} - P\mathbf{v}_c\|^2}{\|\mathbf{v}\|_A^2} = \begin{bmatrix} -A_{ff}^{-1}A_{fc} \\ I \end{bmatrix}, \tag{5.15}$$

where P_{ideal} is referred to as "ideal interpolation."

In addition to P_{ideal} being optimal with respect to the measure $\mu(P)$ (and thus satisfying the WAP for some $K \geq K_{TG}$), if A_{ff} is well-conditioned, then $\|A_{ff}^{-1}\|$ is bounded by a small constant, in which case P_{ideal} also satisfies the SAP:

$$\min_{\mathbf{w}_c\in\mathcal{V}_c} \|\mathbf{v} - P_{\text{ideal}}\mathbf{w}_c\|_A^2 \leq \|\mathbf{v} - P_{\text{ideal}}\mathbf{v}_c\|_A^2 \leq \|A_{ff}^{-1}\|\|A\mathbf{v}\|^2 \leq \frac{K}{\|A\|}\|A\mathbf{v}\|^2,$$

for some K. While P_{ideal} indicates an effective interpolation scheme, $A_{ff}^{-1}A_{fc}$ is often a dense matrix and difficult to compute. However, if A_{ff} is well-conditioned, its entries decay exponentially fast away from the diagonal [20], suggesting that a sparse approximation can be formed. In general, aggregation-based AMG is motivated through energy-minimization principles over the columns of the interpolation operator [28, 35, 80, 88, 120, 178] with the goal of retaining the convergence properties of ideal interpolation, while limiting coarse-grid complexity. This will be explored in more detail in Chapter 6.

However, (5.14) does not provide a sharp bound on convergence, so ideal interpolation typically does not provide optimal (two-grid) convergence factors over all P. The optimal P with respect to two-grid convergence is given in the following lemma [Lemma 1, [24]].

Lemma 2 (Optimal interpolation)**.** *Let $\widetilde{M}$ be the symmetrized relaxation scheme, and let $0 < \lambda_1 \leq ... \leq \lambda_n$ and $\mathbf{v}_1, ..., \mathbf{v}_n$ denote the eigenvalues and eigenvectors, respectively, of the generalized eigenvalue problem*

$$A\mathbf{v} = \lambda\widetilde{M}\mathbf{v}.$$

Then, the minimal convergence rate of the two-grid method $\|E_{TG}(P)\|_A$, over all P with dim$(P) = n_c$, is given by

$$\|E_{TG}(P_{\text{opt}})\|_A^2 = 1 - \lambda_{n_c+1},$$

with corresponding optimal interpolation matrix given by

$$P_{\text{opt}} = \begin{pmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_{n_c} \end{pmatrix}.$$

Although results in [24] suggest that at times a sparse approximation to P_{opt} may be feasible, it is certainly more difficult to develop a cheap, sparse approximation to P_{opt} compared with P_{ideal}. That being said, Lemma 2 does corroborate the general AMG approach of including eigenvectors of A (actually $\widetilde{M}^{-1}A$) associated with small eigenvalues in the Im(P). In fact, it follows from Lemma 2 that if the first $n_c + 1$ eigenvalues of $A\mathbf{v} = \lambda\widetilde{M}\mathbf{v}$ are all approximately zero, AMG *cannot* achieve strong convergence factors. This highlights the importance of the distribution of eigenvalues on the performance of AMG.

5.3 AMG: The nonsymmetric case

A variety of AMG methods have been proposed to generalize the AMG framework to nonsymmetric matrices as well. Perhaps the original idea, and still a common approach, is to treat a nonsymmetric matrix as if it were symmetric, where restriction, R, is given by the transpose of interpolation, P: $R := P^T$ [145]. In some circumstances, this is an effective choice, but when and why this is effective is a question that relies largely on experience. Such an approach is the extent of published research on classical AMG pointwise interpolation formulae [111, 145] for nonsymmetric problems.

In contrast, a number of works suggest that restriction should be built based on A^T, or the column-space of A, and interpolation should be based on A, or the row-space of A (for example, [27, 147]). For classical pointwise interpolation formulae, this is not immediately applicable because they are based on an appropriate strength-of-connection measure and splitting of nodes into C-points and F-points (CF-splitting). Using a different pointwise interpolation formula for P and R based on A and A^T would theoretically require an independent CF-splitting for each, introducing additional difficulties such as non-square coarse-grid operators. Of course, one could assume a fixed CF-splitting and use formulae for P and R based on A

and A^T, respectively, but simple tests indicate that this is not effective for nonsymmetric problems, and it is better to let $R := P^T$. Aggregation-based AMG is more applicable to using information from A and A^T, leading to several variations in classical smoothed aggregation (SA) for nonsymmetric problems [27, 65, 147]. There have also been recent solvers developed that use a mix of CF-splitting and aggregation-concepts and are applicable to nonsymmetric problems, typically using some form of constrained minimization on P and R to approximate the so-called "ideal" operators ([104, 108, 130, 184], Chapter 6). Although some results on this front have been encouraging, a robust AMG solver for nonsymmetric linear systems remains an open problem.

Part of the reason that nonsymmetric solvers are less robust than their SPD counterparts is that little is known about convergence theory of AMG in the nonsymmetric setting, thus limiting theoretical motivation to develop new methods. Strong theoretical results on convergence of iterative methods for non-SPD matrices are difficult to establish and few and far between in the literature. Multigrid traditionally measures convergence in the matrix-induced energy norm, $\|\mathbf{x}\|_A^2 = \langle A\mathbf{x}, \mathbf{x}\rangle$, which is not a valid norm in the non-SPD setting. An asymptotic bound on error propagation can be found by considering the spectral radius of error propagation [104, 109, 124], but asymptotic bounds are not always indicative of practical performance, and even spectral analyses can be difficult because many tools of linear algebra are not applicable; for example, the eigenvectors do not necessarily form a basis for the space and so error cannot be expanded in terms of eigenvectors. A relatively self-contained framework for nonsymmetric matrices with positive real part, $(A + A^T) > 0$, was proposed in [104], proving two-grid convergence of classical-style AMG (that is, interpolating and restricting C-points by value) in a spectral sense as well as in an appropriately derived (albeit difficult to compute) norm. The resulting theory was based on approximating the action of ideal interpolation and ideal restriction on vectors, with accuracy of approximation for a given vector based on the so-called *form absolute value* [104]. However, because the form absolute value is infeasible to compute, the practical solver motivated in [104] reduced to a generalization of constrained energy-minimization in the nonsymmetric setting, quite similar to [184], and somewhat similar to [108, 130], as explored in Chapter 6.

A generalization of the energy norm to a $\sqrt{A^*A}$-norm was introduced in [27]. *Stability* of the coarse-grid correction, $\|P(RAP)^{-1}RA\|_{QA} < C$, for some small constant $C > 1$, and the SAP on P with respect to $\sqrt{A^*A}$ were shown to be sufficient conditions for two-grid convergence of nonsymmetric AMG in [27]. Consistent with traditional AMG motivation, P is built to interpolate modes not accounted for by relaxation, in this case right singular vectors associated with small singular values. Given that the optimal (i.e., coarse-grid correction is orthogonal) R in the $\sqrt{A^*A}$-norm is given by $R = P^TVU^*$, for left and right singular vectors U and V, this suggests that the range of R^* contain left singular vectors associated with small singular values. This approach has been used in various AMG methods based on aggregation and energy minimization [27, 108, 130] and has demonstrated some success on nonsymmetric problems. However, the assumption of a stable coarse-grid correction is, to some extent, assuming away the difficulty of nonsymmetric AMG, and leads to no direct insight as to the role of R in an effective solver. Chapter 7 generalizes the work in [27], developing a complete framework for convergence of nonsymmetric AMG in the $\sqrt{A^*A}$-norm. Sufficient conditions are developed on R and P for stability, two-grid convergence, and multilevel (albeit not V-cycle) convergence.

Finally, in the symmetric case, the WAP is known to be necessary and sufficient conditions for two-grid convergence. Although the SAP provides sufficient conditions for multilevel convergence [179, 180], it is generally believed that the SAP is not attained often in practice, even if a solver is effective in the multilevel setting. Results of this work indicate a similar situation in the nonsymmetric setting. In particular, Chapters 8 and 9 introduce a new reduction-based AMG method for nonsymmetric matrices. For matrices with triangular or block-triangular structure, two-level and multilevel error propagation operators are shown to be nilpotent, that is, the solvers are guaranteed to converge asymptotically. However, convergence is obtained for transfer operators that clearly do not satisfy the conditions for convergence developed in Chapter 7, indicating that they are likely not necessary conditions for convergence. This raises the question if *necessary* conditions can be developed for convergence of nonsymmetric AMG, or if conditions can be developed that do not take the traditional form of an approximation property. A new block spectral analysis of AMG error propagation is developed in Chapter 9 that provides insight into the latter.

5.4 Computational complexity and PyAMG

The computational kernel in the multigrid setup and solve phases is a sparse matrix-vector product (SpMV). Thus, a representative measure of the cost of an AMG solver is the number of FLOPs relative to one SpMV with the initial matrix. This measure is referred to as a *work unit* (WU), where one WU is the cost of computing a SpMV on the finest level. With respect to one SpMV on the finest level, the *operator complexity* (OC) gives the cost in WUs to perform a SpMV on each level in the hierarchy, and is denoted χ_{OC}. This is equivalent to the ratio of the total number of nonzeros on all levels to the number of nonzeros on the finest-level:

$$\chi_{\text{OC}} = \sum_{\ell} \frac{|A_\ell|}{|A_0|},$$

where $|C|$ denotes the number of nonzeros in some sparse matrix C. *Cycle complexity* (CC), denoted χ_{CC}, then measures the total cost in WUs to perform one AMG iteration. Traditionally, CC is considered to scale with OC. For example, in the case of a V(2,2) cycle, $\chi_{\text{CC}} \approx 4\chi_{\text{OC}}$. However, a more detailed model for CC includes the residual computation and coarse-grid correction steps. While it is not typical to account for these parts of the solve phase, they often have non-trivial contributions to the CC, especially for richer interpolation sparsity patterns considered in Chapter 6. To this end, let A_ℓ, P_ℓ, and R_ℓ be operators on the ℓth level of the AMG hierarchy, and $|A_\ell|$ denote the number of non-zeros in matrix A_ℓ. Then, the CC for a $V(\nu_1, \nu_2)$-cycle is given as:

$$\chi_{\text{CC}} = \sum_{\ell} \frac{(\nu_1 + \nu_2 + 1)|A_\ell| + |P_\ell| + |R_\ell|}{|A_0|},$$

In addition to the CC of an AMG method, the convergence factor is also fundamental in determining total work and/or time to solution. This leads to two additional, objective measures of AMG performance: the *effective convergence factor* (ECF), denoted ρ_{eff}, which measures the residual reduction factor for the equivalent of one WU, and the *work-per-digit-of-accuracy* (WPD), denoted χ_{wpd}, which measures the WUs necessary to achieve an order-of-magnitude reduction in the residual. For convergence factor ρ and CC,

$$\rho_{eff} := \rho^{\frac{1}{\chi_{\text{CC}}}}, \qquad \chi_{wpd} := -\frac{\chi_{\text{CC}}}{\log_{10}(\rho)} = -\frac{1}{\log_{10}(\rho_{eff})}.$$

Finally, classical AMG methods often yield minimal setup costs. However, as more features are introduced, such as improved strength-of-connection methods, energy minimization, and adaptivity, these costs can become significant. *Setup complexity* (SC) is thus defined as the total WUs in building the AMG hierarchy.

Detailed estimates of complexity measures are often neglected in numerical results. One contribution of this work is that precise complexity measures are provided for numerical results. Coupled with the convergence factor, this information is used to assess the effectiveness of the solver. The SC estimates have been used to expose the expensive parts of the algorithm and motivated the complexity reduction techniques introduced in Section 6.3.1. Note that although the proposed complexity measures are good indicators of serial performance, they do not necessarily reflect parallel efficiency of the algorithm.

All numerical methods studied and developed here are implemented in the PyAMG library (`https://github.com/pyamg/pyamg`) [11]. We have also implemented many existing AMG algorithms into PyAMG as well as a framework to track the setup and solve costs for each method. The goal is to develop a test suite of AMG solvers to (i) compare performance of different AMG methods, and (ii) determine whether there is an effective AMG method for a given linear system. Development of PyAMG remains ongoing work.

Chapter 6

Energy-minimization in AMG

6.1 The role of energy minimization

Interpolation operators in AMG methods are often constructed with the goal of minimizing some functional with a theoretical relation to convergence, such as the WAP (5.6) and the SAP (5.11). However, there are two important factors that must be considered in practice, but are generally absent from theory – (i) the process used to form interpolation operators must remain linear in complexity in keeping with $O(n)$ AMG methods, and (ii) interpolation operators must remain sparse in order to construct a sparse coarse-grid matrix.

AMG is a popular solver largely because of its linear complexity in the setup and solve phase. However, this constraint can prove difficult when designing AMG methods, and makes approximating some theorems and functionals more tractable than others. Operators used in convergence theory can prove problematic in the practical setting because they cannot be easily computed, such as π_A, P_{ideal}, and P_{opt}. It is also important to note that convergence results such as the WAP and the SAP are typically required to hold for all $\mathbf{v}$, or equivalently for some orthogonal basis for the current space such as the eigenvectors for SPD matrices. Constructing interpolation or coarsening based on a full basis of vectors is generally not tractable in linear complexity, and thus there are two forms of approximation that can be used, (i) work with a candidate set of k vectors, where $k \ll n$, or (ii) (approximately) work in an operator norm, which is a supremum over all vectors.

The first approach is to directly satisfy some theorem or functional based on a set of candidate vectors of dimension $k \ll n$. This is a standard approach for satisfying the WAP or SAP. As in (5.6), the (two-grid) convergence rate is bounded by the maximum K_{TG} over $\mathbf{v}$, which typically occurs for $\mathbf{v}$ associated with small eigenvalues of A, that is, when $\|\mathbf{v}\|_A$ is very small and interpolation must be most accurate. For differential operators, it is common to have a zero or near-zero row sum, making the constant a good representation of low-energy modes. Developing and using additional candidate vectors is the basis for adaptive approaches, developed for difficult linear systems beyond the scope of classical SA or AMG [16, 25, 39]. In such solvers, an adaptive process is used to develop a set of target vectors representative of low-energy modes of A. Interpolation is then constrained to exactly interpolate these modes, and the process repeated on coarse grids.

An alternative approach to accounting for all $\mathbf{v}$ that does not require a set of candidate vectors is to formulate a minimization of some functional over all vectors $\mathbf{v}$. Notationally, let $\widehat{R}$ define the coarse-grid. Since we are interested in AMG based on a CF-splitting here, this means $\widehat{R}$ takes the block form $\widehat{R} = (\mathbf{0}, I)$, defining the C-points as the coarse grid DOFs. Then, we can bound

$$K_{TG} \leq C\|P\widehat{R}\|_A^2, \tag{6.1}$$

with a constant C that depends on how effective relaxation is on A_{ff} [55, 179]. This can be seen as an energy-stability constraint coupled with a "compatibility" measure of the fine and coarse grids. The constant C depends on how well relaxation captures information on F-points, consistent with the idea of compatible relaxation [14, 19], which ensures that the relaxation scheme is able to effectively reduce error on F-points. Then, assuming a "compatible" choice of grids, interpolation must be "stable" in energy, that is, $\|P\widehat{R}\mathbf{v}\|_A \ll \|\mathbf{v}\|_A$.[1] Note that the energy-stability constraint is equivalent to the WAP; however, the differing explicit conditions make for different approaches to constructing multigrid hierarchies.

As an induced A-norm, $\|P\widehat{R}\|_A$ is a supremum over all $\mathbf{v}$. However $\|P\widehat{R}\|_A$ can be bounded in the Frobenius norm, which gives an indirect approach to minimizing $\|P\widehat{R}\|_A$ independent of a set of candidate vectors. Recall by using a CF-splitting, we define the coarse grid via $\widehat{R} = (\mathbf{0}, I)$. Then,

$$\|P\widehat{R}\|_A^2 = \|A^{\frac{1}{2}}P\widehat{R}A^{-\frac{1}{2}}\|^2 \leq \|A^{\frac{1}{2}}P\widehat{R}A^{-\frac{1}{2}}\|_F^2 = \mathrm{tr}(P^TAPS_A),$$

[1] For a more thorough treatment of this result, see [55, 179].

where $S_A := \widehat{R}A^{-1}\widehat{R}^T = (A_{cc} - A_{cf}A_{ff}^{-1}A_{fc})^{-1}$, is the Schur complement of A in A_{cc}. Although an interesting equivalence, the Schur complement is difficult to form in practice. A more tractable approach can be obtained by pulling out an $A^{-\frac{1}{2}}$,

$$\|P\widehat{R}\|_A^2 \leq \|A^{-1}\|\|A^{\frac{1}{2}}P\widehat{R}\|^2 \leq \|A^{-1}\|\|A^{\frac{1}{2}}P\widehat{R}\|_F^2 = \|A^{-1}\|\operatorname{tr}(\widehat{R}^T P^T A P \widehat{R}) = \|A^{-1}\|\operatorname{tr}(P^T A P). \tag{6.2}$$

Minimizing $\operatorname{tr}(P^T AP)$ has been proposed in this form in [20], and is equivalent to minimizing columns of P in the A-norm. This approach can be seen in smoothed aggregation (SA) [178], the general energy-minimization framework proposed in [130], and so-called root-node AMG [108, 152], which is further developed in Section 6.3.

It is worth considering the leading constant that appeared in (6.2), $\|A^{-1}\| = \frac{1}{\lambda_{\min}(A)}$, as this is likely large and could lead to a poor bound on $\|P\widehat{R}\|_A^2$. Note that the energy constraint in (6.1) can also be formulated as

$$\mathbf{v}^T\widehat{R}^T P^T A P\widehat{R}\mathbf{v} \leq \eta\mathbf{v}^T A\mathbf{v} \qquad \Longleftrightarrow \qquad \mathbf{v}_c^T\mathcal{K}\mathbf{v}_c \leq \eta\mathbf{v}^T A\mathbf{v}, \tag{6.3}$$

for all vectors $\mathbf{v}$ [Theorem 5.2, [57]]. Here, (6.3) was achieved for all $\mathbf{v}$ by noting that $\|P\widehat{R}\|_A \leq \|P\widehat{R}\|_A\|\mathbf{v}\|_A$, in which case $\eta = \|P\widehat{R}\|_A$. Because, we do not know if a low-energy $\mathbf{v}_c$ on the coarse grid corresponds to a low-energy $\mathbf{v}$, and vice-versa, the $\|A^{-1}\|$ in (6.2) accounts for the possible scenario that $\mathbf{v}$ is the smallest eigenvector of A and $\mathbf{v}_c$ the largest eigenvector of A_c. In practice, we make the heuristic assumption that the energy of $\mathbf{v}, \mathbf{v}_c$ with respect to A, A_c correlate and, thus, minimizing $\operatorname{tr}(P^T AP) = \sum_i \lambda_i(A_c)$ is an effective way to minimize $\|P\widehat{R}\|_F$.

Remark 1. *It is worth noting that due to the possible energy distinction between* $\mathbf{v}$ *and* $\mathbf{v}_c$*, minimizing* $\operatorname{tr}(A_c)$ *as opposed to* $\|A_c\| = \rho(A_c)$ *makes sense heuristically. The trace will minimize all eigenvalues, with emphasis on large ones, which will help satisfy* (6.3) *for all* $\mathbf{v}$. *Minimizing* $\rho(A_c)$ *may not actually reduce* η*, particularly if the smallest eigenvector in* A *restricts to a* $\mathbf{v}_c$ *in the middle of the spectrum.*

In fact, energy minimization is directly related to approximating the ideal interpolation operator. Observe that $AP_{\text{ideal}} = \begin{pmatrix} 0 \\ S_A \end{pmatrix}$. Given that $AP_{\text{ideal}} = 0$ over F-points, this motivates minimizing columns of P in the A-norm to approximate the action of P_{ideal}. The identity block over C-points along with any constraints enforced ensures that columns of W are nonzero (the solution to minimizing a general P in the A-norm without constraints is $P = \mathbf{0}$). Let B be a set of column-wise constraint vectors to be in the range of P and B_c given by B restricted to the C-points. Then, energy minimization, coupled with a predetermined sparsity pattern and constraints $PB_c = B$, is exactly the conjugate gradient variant of energy minimization proposed in [130]. Lemma 3 shows the relationship between P_{ideal} and energy minimization. That is, the CG variant of energy minimization used to form P over a given sparsity pattern is equivalent to minimizing the difference between columns of P and P_{ideal} in the A-norm, over a given sparsity pattern.

Lemma 3. *Let $A \in \mathbb{R}^{n\times n}$ be SPD, $P_{\text{ideal}} \in \mathbb{R}^{n\times n_c}$ be given by (5.15), and $\mathbf{e}_\ell$ the ℓth canonical basis vector, where $\mathbf{p}_\ell = P\mathbf{e}_\ell$ is the ℓth column of P. Denote by $\mathcal{N}^F$ a sparsity pattern for any matrix $W \in \mathbb{R}^{n_f\times n_c}$ where $W_{ij} = 0$ if $(i,j) \notin \mathcal{N}^F$. Then, define the set $\mathcal{P}_\ell$ as the ℓth column of any matrix with the structure of P_{ideal}, restricted to sparsity pattern $\mathcal{N}^F$, that is*

$$\mathcal{P}_\ell = \left\{ P\mathbf{e}_l \,:\, P = \begin{bmatrix} W \\ I \end{bmatrix}, \text{where } W \in \mathbb{R}^{n_f\times n_c} \text{ and } W_{ij} = 0 \text{ if } (i,j) \notin \mathcal{N}^F_{ij} \right\}. \tag{6.4}$$

Then for $l = 1, \ldots, n_c$,

$$\operatorname*{argmin}_{\mathbf{p}_\ell\in\mathcal{P}_\ell} \|\mathbf{p}_\ell\|_A = \operatorname*{argmin}_{\mathbf{p}_\ell\in\mathcal{P}_\ell} \|\mathbf{p}_\ell - P_{\text{ideal}}\mathbf{e}_\ell\|_A\,. \tag{6.5}$$

Equivalently, minimizing the columns of P in the A-norm is equivalent to minimizing the difference between the columns of P and P_{ideal} in the A-norm.

Proof. Equivalence is established by demonstrating identical weak forms for the two minimization problems in (6.5). Consider $\mathbf{p}_\ell \in \mathcal{P}_\ell$ and define N_ℓ to be the diagonal matrix that enforces sparsity pattern $\mathcal{N}^F$ on the ℓth column of W, $\mathbf{w}_\ell$. That is, $\mathbf{w}_\ell = W\mathbf{e}_\ell = N_\ell W\mathbf{e}_\ell$, where the kth entry of $\mathbf{w}_\ell$ equals zero if $(k,\ell) \notin \mathcal{N}^F$.

(1) Consider minimizing the lth column of P, given by $\mathbf{p}_\ell = \begin{bmatrix} \mathbf{w}_\ell \\ \mathbf{e}_\ell \end{bmatrix}$, in the A-norm. To this end, define the functional $G(\mathbf{w}_\ell) = \Big\langle A\mathbf{p}_\ell, \mathbf{p}_\ell \Big\rangle = \Big\langle A \begin{bmatrix} \mathbf{w}_\ell \\ \mathbf{e}_\ell \end{bmatrix}, \begin{bmatrix} \mathbf{w}_\ell \\ \mathbf{e}_\ell \end{bmatrix} \Big\rangle$, with first variation

$$G'(\mathbf{w}_\ell; \mathbf{v}) = 2\Big\langle N_\ell A_{ff} N_\ell \mathbf{w}_\ell - N_\ell A_{fc}\mathbf{e}_\ell, \mathbf{v} \Big\rangle$$

for $\mathbf{v} = N_\ell \mathbf{v}$. The weak form for minimizing G is then given by

$$N_\ell A_{ff} N_\ell \mathbf{w}_\ell = N_\ell A_{fc}\mathbf{e}_\ell.$$

(2) Consider minimizing the difference between the lth column of P and P_{ideal} in the A-norm. That is, define the functional

$$\begin{aligned} H(\mathbf{w}_\ell) &= \Big\langle A(P - P_{\text{ideal}})\mathbf{e}_\ell, (P - P_{\text{ideal}})\mathbf{e}_\ell \Big\rangle \\ &= \Big\langle A_{ff}(N_\ell \mathbf{w}_\ell - A_{ff}^{-1} A_{fc}\mathbf{e}_\ell), N_\ell \mathbf{w}_\ell - A_{ff}^{-1} A_{fc}\mathbf{e}_\ell \Big\rangle. \end{aligned}$$

Taking the first variation yields

$$H'(\mathbf{w}_\ell; \mathbf{v}) = \Big\langle N_\ell A_{ff} N_\ell \mathbf{w}_\ell - N_\ell A_{fc}\mathbf{e}_\ell, \mathbf{v} \Big\rangle,$$

with $\mathbf{v} = N_\ell \mathbf{v}$, resulting in the weak form

$$N_\ell A_{ff} N_\ell \mathbf{w}_\ell = N_\ell A_{fc}\mathbf{e}_\ell.$$

□

Remark 2. *A similar result to Lemma 3 can be found in [20] in the Frobenius norm.*

Although P_{ideal} in (5.15) is motivated through (and optimal in the sense of (5.14)) the WAP (5.6), and with an appropriate CF-splitting satisfies the SAP (5.11), P_{ideal} is not optimal in any sense with respect to the SAP. However, a similar derivation leads to an equivalent "ideal interpolation" operator with respect to the SAP, which is introduced in Lemma 4.

Lemma 4. *Let $A \in \mathbb{R}^{n\times n}$ be SPD and $P \in \mathbb{R}^{n\times n_c}$ take the form $P = \begin{pmatrix} W \\ I \end{pmatrix}$. Then, consider the SAP under the assumption that the pre-image of any vector $\mathbf{v}$ under P is given by $\mathbf{v}_c$,*

$$\hat{\mu}(P) := \max_{\mathbf{v}\neq 0} \frac{\|\mathbf{v} - P\mathbf{v}_c\|_A^2}{\|A\mathbf{v}\|^2}.$$

Then, for any smoothing scheme, M, $K_{MG} \leq 1 + \hat{\mu}(P)\frac{\|M\|}{\|A\|}$, and the optimal P with respect to minimizing $\hat{\mu}$ is given by

$$\operatorname*{argmin}_P \hat{\mu}(P) = \begin{bmatrix} (A_{ff}^2 + A_{fc}A_{cf})^{-1}(A_{ff}A_{fc} + A_{fc}A_{cc}) \\ I \end{bmatrix},$$

which is exactly ideal interpolation (5.15) for A^2.

Proof. First note from the SAP,

$$
\begin{aligned}
K_{MG} &\leq 1 + \left(\max_{\mathbf{v} \neq 0} \min_{\mathbf{w}_c} \frac{\|\mathbf{v} - P\mathbf{w}_c\|_A^2}{\|A\mathbf{v}\|^2} \right) \frac{\|M\|}{\|A\|} \\
&\leq 1 + \left(\max_{\mathbf{v} \neq 0} \frac{\|\mathbf{v} - P\mathbf{v}_c\|_A^2}{\|A\mathbf{v}\|^2} \right) \frac{\|M\|}{\|A\|}.
\end{aligned}
$$

Based on the proof of Theorem 3.1 and Corollary 3.2 in [55], it follows that

$$
\operatorname*{argmin}_P \max_{\mathbf{v} \neq 0} \frac{\|\mathbf{v} - P\mathbf{v}_c\|_A^2}{\|A\mathbf{v}\|^2} = \operatorname*{argmin}_P \max_{\mathbf{v} \neq 0} \frac{\|\mathbf{v} - P\mathbf{v}_c\|^2}{\|A\mathbf{v}\|^2} = \operatorname*{argmin}_P \max_{\mathbf{v} \neq 0} \frac{\|\mathbf{v} - P\mathbf{v}_c\|^2}{\|\mathbf{v}\|_{A^2}^2}.
$$

The final equation is $\mu(P)$ (5.14) as applied to A^2, and thus the minimum is attained by ideal interpolation (5.15) as applied to A^2. □

Thus far we have been considering the relation of energy minimization to certain ideal operators to motivate the use of energy minimization in building transfer operators. This is a global process over P, working in the operator norm as opposed to based on a set of candidate vectors. In practice, energy minimization is typically coupled with a set of constraint vectors representative of low-energy modes and interpolation is built with these modes exactly in the range of P [20, 107, 108, 130, 152, 183]. For scalar PDEs, the constant vector is a common choice of constraint because discretizations of differential operators typically have zero row sums. A constrained minimization is then performed to build transfer operators.

6.1.1 The nonsymmetric setting

The concept of energy minimization applied to AMG transfer operators is also compatible with non-symmetric problems, where we have a distinct restriction operator R not necessarily equal to P.[2] Motivated by two-grid convergence results in [27], R (P) is designed to focus on left (right) singular vectors associated with small singular values. This is done by minimizing columns of R^* (P) in the AA^*-norm (A^*A-norm) (for example, fGMRES [130, (2.7)] and CGNR [130, (2.34)]). Consistent with the solution of energy-minimization in the A-norm (Lemmas 3 and 4), an equivalent result holds for A^*A and $\sqrt{A^*A}$ (as opposed to A^2 and A). Building on Lemmas 1 and 3 gives the following result in Lemma 5. Coupled with Conjecture 1, two-grid convergence follows from [27]. Lemma 5 provides a meaningful theoretical motivation for energy minimization as applied to non-symmetric problems.

[2] Note that R is not the same as $\widehat{R}$.

Lemma 5. *The solution to energy-minimization in the A^*A- and AA^*-norms satisfy the non-symmetric strong approximation property in the $\sqrt{A^*A}$- and $\sqrt{AA^*}$-norms, respectively, that is, for all $\mathbf{v}$ there exists a $\mathbf{v}_{c_1}, \mathbf{v}_{c_2}$ such that*

$$\left\| \mathbf{v} - P_{\text{ideal}}^{A^*A} \mathbf{v}_{c_1} \right\|_{\sqrt{A^*A}}^2 \leq \frac{K}{\|A^*A\|} \|\mathbf{v}\|_{A^*A}^2,$$
$$\left\| \mathbf{v} - P_{\text{ideal}}^{AA^*} \mathbf{v}_{c_2} \right\|_{\sqrt{AA^*}}^2 \leq \frac{K}{\|AA^*\|} \|\mathbf{v}\|_{AA^*}^2.$$

Proof. The proof follows immediately from the equivalence of the WAP(A^2) and SAP(A) (Lemma 1) and the convergence of energy-minimization to P_{ideal}, in this case P_{ideal} for A^*A and AA^* (Lemma 3). □

Conjecture 1 (Stability)**.** *Let A be nonsingular. The non-orthogonal coarse-grid correction given by transfer operators $R = \left(P_{\text{ideal}}^{AA^*}\right)^T$ and $P = P_{\text{ideal}}^{A^*A}$ is stable, that is*

$$\left\| P(RAP)^{-1}RA \right\|_{\sqrt{A^*A}} = \left\| I - P(RAP)^{-1}RA \right\|_{\sqrt{A^*A}} = C, \tag{6.6}$$

for some constant C, independent of mesh spacing.

Remark 3. *In the symmetric case, $R = P^T$ and $C = 1$, as the coarse-grid correction is an A-orthogonal projection. In the non-symmetric case, the stability assumption necessary for two-grid convergence as shown in [27] and discussed in Chapter 7 is primarily to ensure a non-singular and reasonably conditioned coarse-grid operator, RAP. Conjecture 1 appears to hold in general, but expanding the ideal operators and forming RAP or the full projection (6.6) does not provide a clear method to bound its norm. However, in practice the root-node approach over traditional aggregation offers greater stability of the non-orthogonal projection through enforcing the identity over C-points in transfer operators. Specifically, when forming the coarse grid, the identity block in R and P help ensure the non-singularity of RAP.*

6.2 Weighted energy-minimization

6.2.1 Minimization theory

As discussed in Section 6.1 and can be seen in other AMG methods, AMG interpolation operators are often constructed based on some combination of ensuring that a given set of candidate vectors is interpolated exactly, and minimizing the energy of the coarse grid. We propose forming P through minimizing a general weighted functional of these two approaches using the WAP as in (5.6) and energy minimization as in (6.2):

$$\mathcal{G}(P) = (1-\tau)\frac{\|(I-P\widehat{R})\mathbf{v}\|^2_{\widetilde{M}}}{\|\mathbf{v}\|^2_A} + \tau\,\mathrm{tr}(P^TAP), \tag{6.7}$$

for $\tau \in [0,1)$ and candidate vector $\mathbf{v}$. If multiple candidate vectors $\{\mathbf{v}_i\}$ are available a priori, for example, rigid body modes in elasticity, then we minimize over the maximum $\mathbf{w} \in \mathrm{Span}\{\mathbf{v}_i\}$. This is a complementary approach, focusing on accurate interpolation of low-energy modes as well as energy stability on the coarse grid. It is also complementary in the sense that the first term is defined over a candidate set of vectors, $\{\mathbf{v}_i\}$, while the second term is defined over P, and should improve interpolation regardless of the provided candidate vectors.

Let P take the block form in (5.4) and consider minimizing (6.7). Define a set of n_B candidate vectors as A-orthonormalized columns of a matrix B, and let $X \sim_s M$ as in (5.8). Then, consider minimizing K_{TG} from (5.7), restricted to unit linear combinations of $\mathbf{v} \in \mathrm{Im}(B)$ ($\|\mathbf{v}\|_A = 1$):

$$\begin{aligned}
\max_{\mathbf{v}\in\mathrm{Im}(\mathbf{B})} \|(I-\pi_{\widetilde{M}})\mathbf{v}\|^2_{\widetilde{M}} &\leq c_2 \max_{\mathbf{v}\in\mathrm{Im}(\mathbf{B})} \|(I-\pi_X)\mathbf{v}\|^2_X \\
&\leq c_2 \max_{\mathbf{v}\in\mathrm{Im}(\mathbf{B})} \|(I-P\widehat{R})\mathbf{v}\|^2_X \\
&= c_2 \max_{\mathbf{v}\in\mathrm{Im}(\mathbf{B})} \left\langle X\begin{pmatrix}\mathbf{v_f}-W\mathbf{v_c}\\0\end{pmatrix}, \begin{pmatrix}\mathbf{v_f}-W\mathbf{v_c}\\0\end{pmatrix}\right\rangle \\
&= c_2 \max_{\mathbf{v}\in\mathrm{Im}(\mathbf{B})} \Big\langle X_{ff}W\mathbf{v_c}, W\mathbf{v_c}-2\mathbf{v}_f\Big\rangle + c_2\|\mathbf{v_f}\|^2_{X_{ff}} \\
&\leq c_2\Big\langle X_{ff}WB_c, WB_c - 2B_f\Big\rangle_F + c_2\|B_f\|^2_{X_{ff}} \\
&= c_2\Big\langle X_{ff}WB_cB_c^T, W\Big\rangle_F - 2c_2\Big\langle X_{ff}W, B_fB_c^T\Big\rangle_F + c_2\|B_f\|^2_{X_{ff}}.
\end{aligned} \tag{6.8}$$

This approximates the WAP in the X-norm using an l^2-projection onto $\mathrm{Im}(P)$ (as opposed to the optimal $\pi_{\widetilde{M}}$-orthogonal projection). Recall the second term in (6.7) corresponds to minimizing the columns of P in the A-norm. Expanding $\mathrm{tr}(P^TAP)$ gives

$$\begin{aligned}
\mathrm{tr}(P^TAP) &= \mathrm{tr}\left[\begin{pmatrix}W^T & I\end{pmatrix}\begin{pmatrix}A_{ff} & A_{fc}\\ A_{cf} & A_{cc}\end{pmatrix}\begin{pmatrix}W\\ I\end{pmatrix}\right] \\
&= \mathrm{tr}(W^TA_{ff}W) + 2\,\mathrm{tr}(A_{cf}W) + \mathrm{tr}(A_{cc}) \\
&= \Big\langle A_{ff}W, W\Big\rangle_F + 2\Big\langle W, A_{fc}\Big\rangle_F + \mathrm{tr}(A_{cc})
\end{aligned} \tag{6.9}$$

Plugging equations (6.8) and (6.9) into (6.7) gives a functional of W to minimize in forming P. Dropping terms independent of W and pulling out factor of two for a more familiar form, define

$$\begin{aligned}\mathcal{F}(W) = \frac{\tau}{2}\Big\langle A_{ff}W, W\Big\rangle_F + \frac{c_2(1-\tau)}{2}\Big\langle X_{ff}WB_cB_c^T, W\Big\rangle_F - ...\\ \Big\langle W, c_2(1-\tau)X_{ff}B_fB_c^T - \tau A_{fc}\Big\rangle_F.\end{aligned} \tag{6.10}$$

Observe that (6.10) is a quadratic functional in W. Define a bounded linear operator, $\mathcal{L}$, and right-hand-side, $\mathcal{B}$, as

$$\mathcal{L}W = \tau A_{ff}W + c_2(1-\tau)X_{ff}WB_cB_c^T \tag{6.11}$$

$$\mathcal{B} = c_2(1-\tau)X_{ff}B_fB_c^T - \tau A_{fc}, \tag{6.12}$$

in which case $\mathcal{F}(W) = \frac{1}{2}\langle \mathcal{L}W, W\rangle_F - \langle W, \mathcal{B}\rangle_F$. Note that if A_{ff} and X_{ff} are SPD, $\mathcal{L}$ is self-adjoint and positive definite in the Frobenius norm:

$$\begin{aligned}\Big\langle \mathcal{L}W, Z\Big\rangle_F &= \Big\langle \tau A_{ff}W, Z\Big\rangle_F + c_2(1-\tau)\Big\langle X_{ff}WB_cB_c^T, Z\Big\rangle_F\\ &= \Big\langle \tau W, A_{ff}Z\Big\rangle_F + c_2(1-\tau)\Big\langle W, X_{ff}ZB_cB_c^T\Big\rangle_F\\ &= \tau\Big\langle W, \mathcal{L}Z\Big\rangle_F\\ \Big\langle \mathcal{L}W, W\Big\rangle_F &= \tau\Big\langle A_{ff}W, W\Big\rangle_F + c_2(1-\tau)\Big\langle X_{ff}WB_cB_c^T, W\Big\rangle_F\\ &= \tau\Big\langle A_{ff}W, W\Big\rangle_F + c_2(1-\tau)\Big\langle X_{ff}WB_c, WB_c\Big\rangle_F\\ &> 0\end{aligned}$$

Using the symmetry of $\mathcal{L}$, the first and second Frechét derivative of $\mathcal{F}$ are given by:

$$\begin{aligned}\mathcal{F}'(W)[V] &= \lim_{\alpha\to 0}\frac{\mathcal{F}(W+\alpha V) - \mathcal{F}(W)}{\alpha}\\ &= \Big\langle \mathcal{L}W - \mathcal{B}, V\Big\rangle_F\\ \mathcal{F}''(W)[V] &= \Big\langle \mathcal{L}, V\Big\rangle_F.\end{aligned}$$

Since $\mathcal{L}$ is self-adjoint and positive definite, $\mathcal{F}''(W)[V] \geq 0 \;\forall\; V$. Thus, the minimum of $\mathcal{F}$ in W is achieved at W such that $\mathcal{F}'(W) = 0$, and $\mathcal{F}'(W) = 0 \;\forall\; V$ if and only if $\mathcal{L}W = \mathcal{B}$. This has a unique solution, $W = \mathcal{L}^{-1}\mathcal{B}$. However, it is likely that $\mathcal{L}^{-1}\mathcal{B}$ is dense and not practical, motivating a constrained sparsity pattern for W.

In practice, the sparsity pattern of W must be fixed a priori in order to control the operator complexity of W and A_c. Define a vector space

$$\mathcal{X} = \Big\{ W \,:\, W \in \mathbb{R}^{N_f \times N_c}, W_{ij} = 0 \text{ if } (i,j) \notin \mathcal{N} \Big\},$$

for a set of indices $\mathcal{N}$ denoting a fixed sparsity pattern for W. A Hilbert space $\mathcal{H}$ can be defined over $\mathcal{X}$ with the Frobenius inner product, $\langle A, B\rangle_F = \sum_{ij} A_{ij}B_{ij}$. It is easily verified that $\mathcal{X}$ is complete over the norm induced by $\langle \cdot, \cdot, \rangle_F$ due to the completeness of $\mathbb{R}$. Now define the bounded linear functional $\hat{\mathcal{L}} : \mathcal{H} \to \mathcal{H}$ as

$$(\hat{\mathcal{L}}W)_{ij} = \begin{cases} (\mathcal{L}W)_{ij} & (i,j) \in \mathcal{N} \\ \\ 0 & (i,j) \notin \mathcal{N} \end{cases},$$

and a corresponding bilinear form

$$a(W, V) = \Big\langle \hat{\mathcal{L}}W, V \Big\rangle_F.$$

A quadratic form as in (6.10) restricted over $\mathcal{N}$ can then be defined as

$$\hat{\mathcal{F}}(W) = \frac{1}{2}\Big\langle \hat{\mathcal{L}}W, W \Big\rangle_F - \Big\langle W, \hat{\mathcal{B}} \Big\rangle_F, \tag{6.13}$$

where $\hat{\mathcal{B}} \in \mathcal{H}$ is $\mathcal{B}$ restricted to $\mathcal{N}$. Note that in $\mathcal{H}$, $\langle W, \mathcal{B}\rangle_F = \langle W, \hat{\mathcal{B}}\rangle_F$. A similar derivation as shown for $\mathcal{L}$ confirms that $\hat{\mathcal{L}}$ is self-adjoint and $a(W, V)$ symmetric. Then, observe that for $W \in \mathcal{H}, W \neq 0$, $\hat{\mathcal{L}}$ and $a(W, V)$ are positive:

$$\Big\langle \hat{\mathcal{L}}W, W \Big\rangle_F = \Big\langle \mathcal{L}W, W \Big\rangle_F > 0,$$

The following lemma of functional analysis can then be invoked to find a solution to (6.13).

Lemma 6. *Let $a(x, y)$ be a bounded, symmetric, positive bilinear form on a Hilbert space $\mathcal{H}$, and $\mathcal{G}(x)$ a bounded linear functional on $\mathcal{H}$. Then the following are equivalent*

$$x = \min_{x \in \mathcal{H}} \ \frac{1}{2}a(x, x) - \mathcal{G}(x) + C \tag{6.14}$$

$$x = x \text{ such that } a(x, y) = \mathcal{G}(y) \quad \textit{for all } y \in \mathcal{H} \tag{6.15}$$

Furthermore, there exists a unique solution $x \in \mathcal{H}$ satisfying (6.14)*,* (6.15)*.*

Based on Lemma 6, we seek the unique solution to

$$\hat{\mathcal{L}}W = \hat{B}, \quad W \in \mathcal{H}, \tag{6.16}$$

which can be iterated towards using a preconditioned conjugate gradient method

Because $\hat{\mathcal{L}}$ is self-adjoint and positive in $\mathcal{H}$, conjugate gradient (CG) in the Hilbert space setting is a competitive approach to solving (6.16) in an iterative fashion. It is generally advisable to precondition CG iterations for optimal convergence. Here, we construct a diagonal preconditioner for (6.16) to make iterations more robust when $\hat{\mathcal{L}}$ is poorly conditioned at a marginal increase of computational cost.

Unlike with matrices, however, it is not clear what the "diagonal" of $\hat{\mathcal{L}}$ is. Let $W \in \mathbb{R}^{N_f \times N_c}$, and define the operator $\overline{(W)}$ as the columns of W stacked in a column-vector. Note that $\overline{(W^T)}$ then gives the rows of W stacked as a column-vector. Then, let Y be the permutation matrix such that $\overline{(W)} = Y\overline{(W^T)}$ and $YY^T = Y^TY = I$, which can be thought of as a mapping of W from row-major format to column-major format. First note the following lemma with regards to Kronecker products and the action of Y.

Lemma 7. *Let Y be a permutation matrix mapping $W \in \mathbb{R}^{N_f \times N_c}$ from row-major format to column-major format, that is, $\overline{(W)} = Y\overline{(W^T)}$. Then, for any $P \in \mathbb{R}^{N_f \times N_f}$ and $Q \in \mathbb{R}^{N_c \times N_c}$,*

$$Y(P \otimes Q)Y^T = Q \otimes P$$

Proof. First consider the structure of Y. Note the following relations between $W, \overline{(W)}$, and $\overline{(W^T)}$, i.e. W stored as a standard dense matrix, a column-major matrix, and a row-major matrix, respectively,

$$\overline{(W)}_{i+jN_f} = W_{ij}$$
$$\overline{(W^T)}_{j+iN_c} = W_{ij}.$$

Defining Y such that $Y\overline{(W^T)} = \overline{(W)}$, it follows that

$$Y_{i+jN_f, j+iN_c} = 1, \quad \text{for } i \in [0, N_f], j \in [0, N_c],$$

and the action of YAY^T is then given as

$$[YAY^T]_{i+jN_f, k+lN_f} = A_{j+iN_c, l+kN_c}. \tag{6.17}$$

Now consider the element-wise Kronecker products of P and Q:

$$[P \otimes Q]_{j+iN_c, l+kN_c} = P_{ik}Q_{jl}, \tag{6.18}$$

$$[Q \otimes P]_{i+jN_f, k+lN_f} = P_{ik}Q_{jl}, \tag{6.19}$$

for $i, k \in [0, N_f], j, l \in [0, N_c]$. Combining (6.17), (6.18), and (6.19) gives

$$\begin{aligned}
[Y(P \otimes Q)Y^T]_{i+jN_f, k+lN_f} &= (P \otimes Q)_{j+iN_c, l+kN_c} \\
&= P_{ik}Q_{jl} \\
&= [P \otimes Q]_{i+jN_f, k+lN_f}.
\end{aligned}$$

It follows that $Y(P \otimes Q)Y^T = Q \otimes P$. □

Remark 4. *Lemma 7 is a known result that we arrived at inadvertently, where Y is known as the "Perfect Shuffle" matrix [41]. Its relation to row-major and column-major storage of matrices is interesting and not something that we have seen in the literature.*

Now consider finding the diagonal of $\mathcal{L}$ by looking at $\overline{\mathcal{L}}$ as an operator on $\overline{(W)}$. To do so, represent the action of $A_{ff}W$ through $(I_{N_c} \otimes A_{ff})\overline{(W)}$, where $(I_{N_c} \otimes A_{ff})$ gives a block diagonal matrix of N_c A_{ff}'s, each to be multiplied by one column of W. Recalling the identity $(A \otimes B)(C \otimes D) = (AB \otimes CD)$ and Lemma 7,

$$\begin{aligned}
\overline{\mathcal{L}}\overline{(W)} &= \tau(I_{N_c} \otimes A_{ff})\overline{(W)} + c_2(1-\tau)(I_{N_c} \otimes X_{ff})\overline{(WB_cB_c^T)} \\
&= \tau(I_{N_c} \otimes A_{ff})\overline{(W)} + c_2(1-\tau)(I_{N_c} \otimes X_{ff})YY^T\overline{(WB_cB_c^T)} \\
&= \tau(I_{N_c} \otimes A_{ff})\overline{(W)} + c_2(1-\tau)(I_{N_c} \otimes X_{ff})Y\overline{(B_cB_c^TW^T)} \\
&= \tau(I_{N_c} \otimes A_{ff})\overline{(W)} + c_2(1-\tau)(I_{N_c} \otimes X_{ff})Y(I_{N_f} \otimes B_cB_c^T)Y^TY\overline{(W^T)} \\
&= \tau(I_{N_c} \otimes A_{ff})\overline{(W)} + c_2(1-\tau)(I_{N_c} \otimes X_{ff})(B_cB_c^T \otimes I_{N_f})\overline{(W)} \\
&= \tau(I_{N_c} \otimes A_{ff})\overline{(W)} + c_2(1-\tau)(B_cB_c^T \otimes X_{ff})\overline{(W)} \\
&= \Big[\tau(I_{N_c} \otimes A_{ff}) + c_2(1-\tau)(B_cB_c^T \otimes X_{ff})\Big]\overline{(W)}
\end{aligned}$$

This derivation can be naturally extended to $\mathcal{H}$, where $W \in \mathcal{H}$ has a specified sparsity pattern, $\mathcal{N}$, by setting the kth row and column of $\overline{\mathcal{L}}$ equal to zero for all k such that $\overline{(W)}_k := W_{ij}$, and $(i,j) \notin \mathcal{N}$. Because $\overline{\mathcal{L}}$ is a block operator with block size $N_f \times N_f$, it follows that there is a distinct "diagonal" in $\mathcal{L}$ corresponding to each jth column of W,

$$D_j = \tau \cdot \mathrm{diag}(A_{ff}) + c_2(1-\tau)(B_c B_c^T)_{jj} \cdot \mathrm{diag}(X_{ff}). \tag{6.20}$$

A diagonal preconditioning for $\hat{\mathcal{L}}$ is then given by taking the Hadamard product with $\mathcal{D} \in \mathcal{H}$, where the jth column of $\mathcal{D}$ is given by the element-wise inverse of (6.20):

$$\mathcal{D}_{ij} = \frac{1}{\tau(A_{ff})_{ii} + c_2(1-\tau)(B_c B_c^T)_{jj}(X_{ff})_{ii}}, \qquad \text{for } (i,j) \in \mathcal{N}. \tag{6.21}$$

In the case of A_{ff} having a constant or near-constant diagonal, and letting X_{ff} be the diagonal of A_{ff} (a common practical choice), $\mathcal{D}$ is constant or near-constant. In practice, preconditioning with $\mathcal{D}$ is important for problems in which diagonal elements of A or target vectors B consist of a wide range of values.

Remark 5 (Sylvester and Lyapunov equations)**.** *In fact, this preconditioner is applicable to solving general systems of the form*

$$AWB + CWD = F, \tag{6.22}$$

for solution matrix W, where A, B, C and D need not be symmetric (of course an appropriate Krylov solver must be chosen based on properties of the functional). A general diagonal preconditioner for (6.22) *is given by taking the Hadamard product with*

$$\widehat{\mathcal{D}}_{ij} = \frac{1}{B_{jj}A_{ii} + D_{jj}C_{ii}}. \tag{6.23}$$

Systems of the form in (6.22) *arise often in the context of optimal control theory. Letting $B = C = I$,* (6.22) *is a Sylvester equation; letting $B = A^T$, $C = -I$, and $D = I$,* (6.22) *is a discrete Lyapunov equation; and letting $B = C = I$ and $D = A^T$,* (6.22) *is a continuous Lyapunov equation. There have been many efforts at developing Krylov preconditioners for such systems; for example, see [44, 48, 75, 158, 186]. Here we develop a simple preconditioner for problems of the form* (6.23)*, that is easy to form and apply.*

6.2.2 Comparison with constrained energy minimization

In this section, we present numerical results for a variety of problems, comparing a weighted energy minimization and constrained energy minimization, and analyzing the choice of constraint vector. The method proposed here is implemented in the PyAMG library [11]; AMG methods such as strength-of-connection (SOC), coarsening, etc., follow that of [108], and the reader is referred there for details. In figures, RN refers to a constrained energy minimization using root-node AMG (RN AMG) [108] and TM_{10^k} refers to weighted energy minimization proposed here with weight $\tau = 10^k$. Unless otherwise specified, a V-cycle is applied with two iterations of Jacobi pre- and post-relaxation as a preconditioner for CG and the constant vector with several Jacobi smoothing iterations applied is chosen as the constraint vector. Weighted energy-minimization uses the diagonal preconditioning of Section 6.2.1; and constrained energy-minimization uses the diagonal preconditioning of [130]. The test problems considered are:

(1) Anisotropic diffusion: 2-dimensional rotated anisotropic diffusion, discretized with linear finite elements, on an unstructured triangular mesh:

$$-\nabla \cdot Q^T D Q \nabla u = f \quad \text{for } \Omega = [0,1]^2, \tag{6.24}$$

$$u = 0 \quad \text{on } \partial\Omega, \tag{6.25}$$

where

$$Q = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}, \qquad D = \begin{bmatrix} 1 & 0 \\ 0 & \epsilon \end{bmatrix}.$$

Here, ϵ represents anisotropy and ψ the angle of rotation from the coordinate axis, which both contribute to the difficulty of this problem [17, 19, 21, 23, 35, 39, 43, 63, 107, 125, 127, 130, 152]. To see this, consider the non-rotated case of $\psi = 0$, which has a spectrum of the form

$$\begin{aligned} -\nabla \cdot D\nabla u_{jk} &= \lambda_{jk} u_{jk}, \qquad \text{where} \\ u_{jk} &= \sin(j\pi x)\sin(k\pi y), \\ \lambda_{jk} &= \pi^2(j^2 + \epsilon k^2), \end{aligned}$$

for $j, k \in \mathbb{Z}^+$. If $\epsilon = 1$, then the lowest energy mode is the lowest Fourier mode: $u_{11} = \sin(\pi x)\sin(\pi y)$, which is locally representative of all low-energy modes. However, if $\epsilon \approx 0$, for small j there are high-frequency eigenfunctions in the y-direction ($k \gg 0$) with relatively small eigenvalues that are no longer represented locally by the lowest Fourier mode. As a result, relaxation schemes are unable to capture these modes, while coarse-grid correction is not equipped to handle such *hidden* high frequency error.

For angles ψ aligned with the mesh, line relaxation or semi-coarsening along the direction of anisotropy can be used [149]. However, for strong anisotropies, $\epsilon \approx 0$, with angles that are not aligned with the mesh, efficient and effective multigrid solvers remain elusive. Here we consider a finite element discretization of strongly and *totally* anisotropic diffusion, $\epsilon = 0.001$ and $\epsilon = 0$, respectively, on a unit square with Dirichlet boundary conditions. Totally anisotropic diffusion is particularly challenging as the problem is effectively reduced to a sequence of 1D problems on a 2D domain. Due to the unstructured mesh, all angles $\theta \in (0, {}^{\pi}/_{2})$ are effectively equivalent from a solver perspective; thus, moving forward we (arbitrarily) let $\theta = {}^{3\pi}/_{16}$.

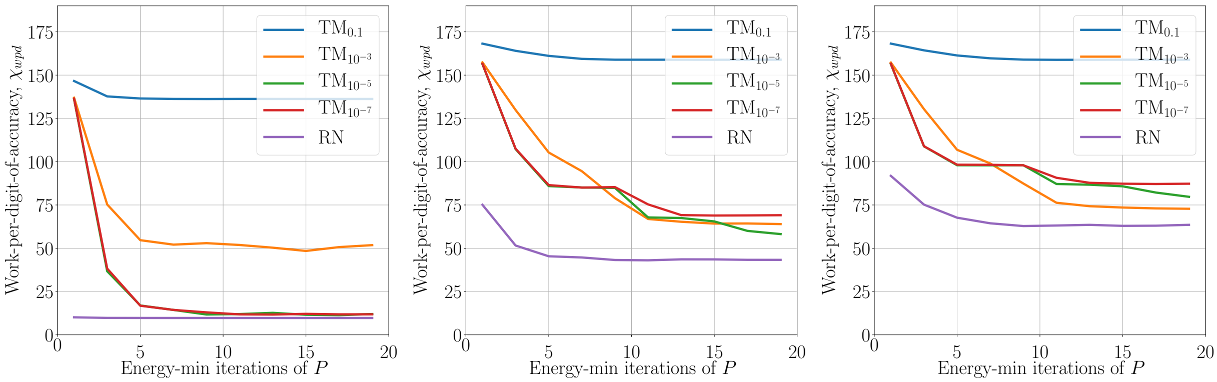
(2) Jump-coefficient Poisson: 2-dimensional diffusion problem (as in equations (6.24-6.25)), discretized with linear finite elements on a structured, regular triangular mesh, with a jump-coefficient that oscillates every other grid point:

$$Q = I \text{ and } D = \begin{bmatrix} f(x,y) & 0 \\ 0 & f(x,y) \end{bmatrix}, \text{ where } f(x,y) = \begin{cases} K & \text{if } \operatorname{mod}(\frac{x}{h},2) = 1 \text{ AND } \operatorname{mod}(\frac{y}{h},2) = 0 \\ K & \text{if } \operatorname{mod}(\frac{x}{h},2) = 0 \text{ AND } \operatorname{mod}(\frac{y}{h},2) = 1 \\ 1 & \text{otherwise} \end{cases}.$$

Here, h is the mesh spacing in both directions and, thus, x/h and y/h yield the integer index of each mesh point in the x- and y-direction, respectively. This then implies that the number of coefficient jumps grows proportionally with the number of mesh points, resulting in a checkerboard like pattern where the large and small coefficient alternate.

6.2.2.1 Effect of smoothing P

Figure 6.1 shows the WPD as a function of smoothing iterations of P, for variations in energy minimization applied to anisotropic Poisson (Problem # 1), with anisotropy $\epsilon \in \{1, 0.001, 0\}$. Interpolation uses a degree-four sparsity pattern, that is, the sparsity pattern for each column of P reaches out to neighbors within graph distance four from the corresponding C-point (see [108, 152]). This wider sparsity pattern often leads to better convergence rates for difficult problems [152], but also requires more iterations of energy-minimization. Essentially, wider sparsity patterns create more interpolation coefficients in P, which are then determined through energy-minimization.



(a) $\epsilon = 1$

(b) $\epsilon = 0.001$

(c) $\epsilon = 0$

Figure 6.1: WPD as a function of number of iterations of energy-minimization applied to P for problem #1 and (a) isotropic diffusion ($\epsilon = 1$), (b) anisotropic diffusion ($\epsilon = 0.001$), and (c) totally anisotropic diffusion ($\epsilon = 0$).

Several immediate results follow from Figure 6.1. First, there is a limit at which additional smoothing iterations of P no longer improve convergence. For the isotropic case ($\epsilon = 1$), the best convergence rates are obtained by simply enforcing the constraint with a single constrained smoothing pass; additional energy-minimization steps do not improve convergence. As the level of anisotropy increases ($\epsilon \to 0$), the number of smoothing iterations of P required to achieve the best performance increases. However, convergence of the AMG solver based on a given constraint vector and coarsening scheme remains bounded below, regardless of further energy minimization of P. Second, it is clear that enforcing the constraint exactly or near-exactly is fundamental to good convergence, even for the simplest isotropic problem. Although theory tells us that interpolating low-energy modes is necessary for good convergence, the fact that this cannot be achieved through weighted energy minimization is slightly non-intuitive. Energy-minimization reduces the columns of P in the A-norm, which should thus build P to include low-energy modes in its range. Heuristically, it seems that after a handful of CG iterations, the range of P would contain sufficient low-energy modes for good convergence. However, it is clear in Figure 6.1 that even in the isotropic case, using a large $\tau = 0.1$ to focus on energy minimization over constraints leads to very poor performance.

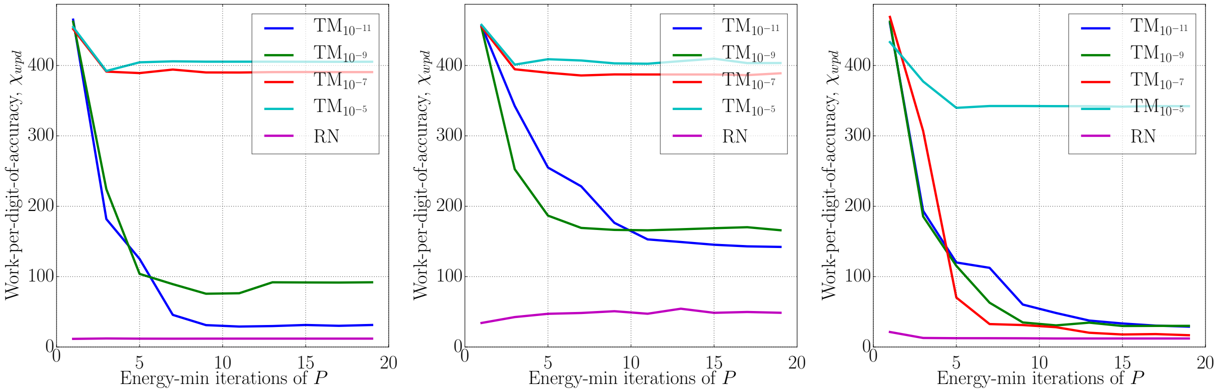
Together, these points underline the role of energy minimization in AMG convergence as an acceleration technique. For some difficult problems, energy minimization is critical to achieving scalable convergence. Strongly anisotropic diffusion is one such example that typically proves difficult for standard AMG methods, but can be solved effectively with constrained energy minimization [108]. Nevertheless, regardless of energy minimization, strong convergence cannot be obtained without enforcing or nearly-enforcing an appropriate constraint vector (Figure 6.1).

Figure 6.2 shows the WPD as a function of smoothing iterations of P for variations in energy minimization applied to the jump-coefficient problem (Problem # 2), with jump coefficients $K = 10^6$ and $K = 10^3$. For $K = 10^6$, Figure 6.2a shows results for diagonal preconditioning of energy-minimization and Figure 6.2b shows the case of preconditioning turned off. Figure 6.2c shows the case of diagonal preconditioning with $K = 10^3$. Comparing Figures 6.2a and 6.2b, we see that using preconditioning in weighted energy minimization reduces the number of iterations necessary to achieve good convergence. Moreover, preconditioning actually improves the best achievable AMG convergence factor in practice. For constrained energy-minimization, energy minimization iterations without preconditioning increases the WPD by $3-5\times$ within a reasonable number of iterations on P (of course, asymptotically the preconditioned and non-preconditioned results are equivalent but, in practice, only $O(1)$ iterations are done.) This raises an interesting question as to if better preconditioners for energy minimization can actually improve the AMG solver's performance in a way that additional iterations with a diagonal preconditioner cannot; however, this is a topic for future study.

Focusing on the more practical solvers in Figure 6.2a, we also see that the results mirror those in Figure 6.1. Overall, the constrained energy-minimization case performs best, with weighted energy-minimization able to approach the constrained case only for the right τ values and enough energy-minimization iterations on P. Again, there is a limit beyond which additional energy-minimization iterations no longer improve AMG convergence. For constrained energy-minimization, relatively few iterations are needed. Lastly, enforcing the constraint exactly or near-exactly is fundamental to good convergence. Using energy-minimization with larger τ values leads to poor performance.

The effects of changing the jump coefficient K can be seen by comparing Figures 6.2a and 6.2c. Interestingly, the larger jump value leads to a need for smaller τ values for the weighted case, (compare the curves for $\tau = 10^{-7}$). Overall, apart from changing the size of beneficial τ values, the size of the coefficient jump does not noticeably affect either the weighted or constrained energy minimization.

A final note of interest is that larger interpolation sparsity patterns do not help here. Thus, a moderate sparsity pattern of degree three is chosen for the results.



(a) With diag. precon., $K = 10^6$ (b) Without diag. precon., $K = 10^6$ (c) With diag. precon., $K = 10^3$

Figure 6.2: Work-per-digit of accuracy, comparing weighted and constrained energy-minimization, the use of diagonal preconditioning and two different coefficient jumps.

Remark 6. *We did not find tracking the CG residual norm during energy-minimization to be useful, and hence omit plots of this information. The key difficulty is that it is not clear how to connect the residual norm to the eventual multigrid convergence rate. In other words, it is not clear how to use the residual norm to halt the energy-minimization process. For instance, taking the cases of constrained energy-min from Figures 6.1 and 6.2, it is clear that at most five iterations of energy-minimization are needed. However, the residual norm continues to decrease monotonically for multiple orders of magnitude over iteration five to iteration 19. Yet, this extra residual reduction does not speed up convergence of the resulting multigrid solver. In practice, the number of iterations typically equals the degree of the sparsity pattern of P plus some small number, usually two or three. This number of iterations is required to first fill the allowed sparsity pattern, and then to provide two or three iterations of additional smoothing.*

6.2.2.2 Constraint vectors and adaptivity

The previous section demonstrated two things: (i) for good convergence, it is important that P exactly or almost exactly interpolates an appropriate constraint vector, and (ii) coupled with a good constraint, energy minimization can improve convergence, but only a fixed amount. This leads to the natural idea of adding an additional constraint vector when further energy minimization of P no longer improves convergence. Such an approach is the basis of adaptive work, where a set of constraint vectors are developed that are then included or approximately included in the range of P [16, 25, 39]. There are multiple ways to generate constraint vectors; here we take the simple approach of generating a random vector $\mathbf{x}_0$ and applying some form of improvement iterations (either relaxation or V-cycles) to reduce $\|\mathbf{x}_0\|_A$. Table 6.3 shows results for constrained energy minimization AMG applied to anisotropic Poisson, with varying numbers of improvement iterations and varying numbers of constraint vectors.

Constraints	Imp. Iters	OC	CC	CF
1	2	1.52	5.97	0.75
2	2	1.55	6.01	0.74
3	2	1.56	6.02	0.79
1	5	1.51	5.95	0.64
2	5	1.54	5.99	0.70
3	5	1.55	6.01	0.73
1	10	1.50	5.95	0.53
2	10	1.54	5.98	0.67
3	10	1.55	5.99	0.69
1	25	1.50	5.95	0.49
2	25	1.54	5.97	0.67
3	25	1.55	5.98	0.65
1	100	1.50	5.95	0.48
2	100	1.50	5.95	0.50
3	100	1.50	5.95	0.51

(a) Two-grid

Constraints	Imp. Iters	OC	CC	CF
1	2	1.64	9.39	0.76
2	2	1.67	9.52	0.81
3	2	1.67	9.50	0.85
1	5	1.63	9.29	0.64
2	5	1.66	9.44	0.78
3	5	1.66	9.47	0.83
1	10	1.62	9.27	0.54
2	10	1.64	9.36	0.76
3	10	1.66	9.42	0.82
1	25	1.62	9.23	0.51
2	25	1.66	9.32	0.69
3	25	1.66	9.42	0.78
1	100	1.62	9.28	0.50
2	100	1.62	9.28	0.54
3	100	1.62	9.27	0.54

(b) Multigrid

Figure 6.3: Constrained energy minimization applied to strongly anisotropic diffusion ($\epsilon = 0.001$) in a two-grid and multigrid method. Constraints are initialized as a random vector; for the first constraint, Jacobi iterations are applied as improvement iterations. After an AMG hierarchy has been formed with one target, a new random vector is generated and V-cycles are applied as improvement iterations to generate a second target. The hierarchy is rebuilt using the new constraints, and so on.

Several interesting things follow from the results in Table 6.3. First, the difference in convergence factor between two-grid and multigrid is very small. This indicates that we are solving our coarse-grid problem well in V-cycles, and that convergence is limited by how "good" the coarse grid is, and not how accurately we are solving it. Moreover, naively adding constraint vectors that were not accounted for in the range of P does not improve convergence and, in fact, degrades convergence in all cases, while increasing the setup complexity.

6.3 Constrained energy minimization and root-node AMG

In Section 6.3 it was shown that a constrained energy minimization for building AMG interpolation operators consistently outperforms a weighted energy minimization. Constrained energy minimization has been proposed in various forms [20, 108, 130, 152, 183]. In particular, in [152], a general algorithm for constrained energy minimization was shown to contribute to a robust AMG solver for strongly anisotropic diffusion, referred to as *root-node AMG* (RN AMG). Although an elliptic operator and SPD matrix, strongly anisotropic diffusion remains a challenge for existing AMG solvers, particularly on unstructured meshes and for non-grid aligned anisotropies. Unfortunately, RN AMG as proposed in [152] suffers from SC and CC that are intractably high for a “fast” solver. Section 6.3.1, develops a new filtering routine for building interpolation operators in RN AMG that leads to a robust and scalable solver for strongly anisotropic diffusion, as well as other difficult SPD and nonsymmetric matrix equations. Although algorithmically simple, the proposed filtering is fundamental to the construction of a practical solver. The following sections study the performance of RN AMG compared with SA and classical AMG applied to several test problems.

6.3.1 Sparsity pattern filtering

The first step in building an interpolation or restriction operator for energy-minimization-based AMG is to determine a sparsity pattern. Typically, this is done by taking an initial matrix consisting of, for example, only the identity over C-points or a tentative prolongation operator like in SA [115]. The initial sparsity pattern is then expanded to include neighbors of degree two, three, or more, typically by multiplying by a SOC matrix [108]. For difficult problems, large sparsity patterns up to degree four or five may be necessary for scalable convergence [152]; however, such large sparsity patterns result in relatively dense transfer operators and, thus, dense coarse-grid operators, which can significantly increase the setup and cycle complexity of AMG.

Although a degree-four sparsity pattern may be necessary for good convergence, many of the nodes distance three or four away from a given F-point are not critical for good interpolation. Here we propose pre-filtering and post-filtering algorithms that eliminate nonzeros deemed unimportant from the sparsity pattern before energy-minimization is applied to P, and again after, respectively. This can be thought of as a semi-adaptive process to determine the sparsity pattern and allows for long-distance interpolation in the direction of strong connections, while limiting complexity. Conceptually, each node should interpolate from its strongly connected neighbors. If we build the sparsity pattern based on powers of a SOC matrix, the value of an entry in the sparsity pattern should reflect (to some extent) its importance. Thus, given an initial sparsity pattern, entries are filtered as in [43] by either retaining the k largest values in a row or by applying a drop tolerance θ. Algorithm 1 describes this process in detail, where $\max(G, i, k)$ is the kth largest off-diagonal entry in row i. The idea of pre-filtering has shown to be effective for model problems using a polynomial approximation to A_{ff}^{-1} in [20]. The pre-filtering used here is less expensive, relying on values already computed by the root-node algorithm.

Algorithm 1: `filter`(G)

```
Input:  $G$:       matrix to be filtered
        $\theta$:  filtering drop-tolerance
        $k$:       filtering threshold
Output: $G$
if $k$ then
    for $|G(i,j)| < \max(G,i,k)$ do
        $G(i,j) \leftarrow 0$
if $\theta$ then
    for $|G(i,j)| < \theta \max(G,i,1)$ do
        $G(i,j) \leftarrow 0$
```

After P (R) is formed, a post-filtering process is applied to reduce complexity, removing elements directly from P (R) after smoothing. However, this leads to a P (R) that (i) no longer exactly interpolates constraint vectors, and (ii) may have large increases in column-energy caused by eliminating entries. Thus, following post-filtering, the constraints are re-enforced and an additional iteration of energy minimization is applied to smooth columns.

Post-filtering generally results in a lower complexity in the Galerkin coarse-grid operator and all subsequent coarser grid operations. A similar filtering approach is also effective for classical AMG methods [43].

Because pre-filtering is only based on SOC and not the fully formed interpolation operator, it is possible that influential entries are inadvertently removed, thus degrading convergence. One advantage of post-filtering is that element removal is based on the smoothed interpolation stencil entries and is thus less likely to inadvertently degrade convergence by removing important entries from P compared with pre-filtering. However, post-filtering does not reduce the OC and SC as effectively as pre-filtering, in particular because energy-minimization (one of the dominant costs in SC) is applied to a larger sparsity pattern, which is then trimmed afterwards. Trimming the sparsity pattern of P and R before initiating the construction significantly lowers the SC in many cases, with minimal impact on AMG convergence. In practice, the best results are obtained using a combination of pre- and post-filtering.

6.3.1.1 Filtering in practice

Here we demonstrate the effectiveness of the proposed filtering strategy in reducing the cycle and setup complexity of a RN AMG solver. Although filtering is a key component of RN AMG on nearly all problems, it is especially applicable in 3D, where there is high connectivity between nodes, resulting in relatively dense operators. Consider the anisotropic diffusion problem

$$u_{xx} + u_{yy} + 0.001u_{zz} = f. \tag{6.26}$$

Linear finite elements are used to discretize (6.26) on an unstructured tetrahedral mesh of the unit cube, with homogeneous, Dirichlet boundaries, yielding a matrix with approximately 2.65M DOFs. While the anisotropy is aligned with the coordinate axis, the unstructured mesh yields a variety of non-grid-aligned anisotropies, known to be more difficult for AMG than the grid-aligned case (see Section 6.2.2). Table 6.1 shows complexities and average convergence factors (ρ) for solving (6.26) using various combinations of pre- and post-filtering on P. A V$(1,1)$-cycle with symmetric Gauss-Seidel relaxation and CG acceleration is used. The evolution SOC measure is used [128], with a drop tolerance of 4.0, and CG energy minimization with $d = 4$.

Table 6.1 shows up to an order-of-magnitude reduction in SC and more than a 50% reduction in CC by using filtering. While filtering does not guarantee a reduction in complexity, significant savings are often

Pre-filter θ	–	0.1	0.2	–	–	0.1	0.2
Post-filter θ	–	–	–	0.1	0.2	0.1	0.2
SC	1157.9	161.7	123.2	581.5	432.9	156.2	129.8
OC	3.0	1.4	1.3	1.8	1.5	1.4	1.3
CC	18.9	8.0	7.1	10.1	8.0	7.7	6.9
ρ	0.52	0.60	0.62	0.51	0.53	0.59	0.61

Table 6.1: Impact of filtering for the 3D-anisotropic diffusion problem.

observed with only a marginal impact on convergence. In some situations, filtering has been observed to not only reduce cost, but also improve convergence [108].

The SC in Table 6.1 is broken down into four main categories in Table 6.2. “Aggregation” is the cost of computing the strength matrix S and forming aggregates $\mathcal{A}$, most of which is due to using the evolution measure. “Candidates” refers to relaxing candidate set B and restricting B to a coarse level. Column “P” refers to the cost of forming the tentative interpolation operator and applying energy-minimization smoothing iterations to construct P, while column “RAP” represents a measure of the triple-matrix product. Each column gives the total cost for the given processes over all levels, measured in WUs.

Pre-filter θ	Post-filter θ	Aggregation	Candidates	P	RAP	Total SC
–	–	478.7	24.1	469.7	271.1	1243.6
0.2	–	48.5	10.4	55.1	9.2	123.2
–	0.2	64.1	11.7	338.7	18.5	432.9
0.2	0.2	46.3	10.1	65.4	7.9	129.8

Table 6.2: Break down of setup cost in WUs for 3D-anisotropic diffusion.

Filtering P has a direct impact on all setup components that use R and P matrix operations (see “RAP” and “Candidates” in Table 6.2). Consequently, this reduces the cost of restricting the residual and the coarse-grid correction as measured in the CC, along with the cost of smoothing P, which is the focus of pre-filtering. Table 6.2 also highlights the high cost of the strength measure in cases when filtering is not used and, thus, coarse-grid complexity is not contained. Thus filtering provides the additional benefit of reduced costs on all subsequent grids through a sparser coarse-grid operator.

Remark 7. *Although GMRES energy-minimization targets the SAP and CG the WAP, when applied to SPD problems, there is not a notable difference in convergence. For instance if GMRES is used in Table 6.1, then the convergence rates change by no more than 0.005 and operator complexities remain essentially the same.*

6.3.1.2 Modified evolution SOC

In considering Table 6.2, one possible area for cost reduction is the aggregation phase, where the cost of the SOC computation dominates. Traditional SOC measures require only a few work units, usually two or three times the operator complexity. However, Table 6.2 shows that evolution-based SOC is a substantial part of the setup phase, which is attributed to the global spectral radius estimate used in weighted Jacobi [128]. This estimate is calculated with an Arnoldi/Lanczos process and costs roughly 15 matrix-vector multiplies on each level. However, alternative methods [8] use ℓ_1-Jacobi relaxation to provide an inexpensive local row-wise weight. This alternative is explored here for the evolution SOC measure.

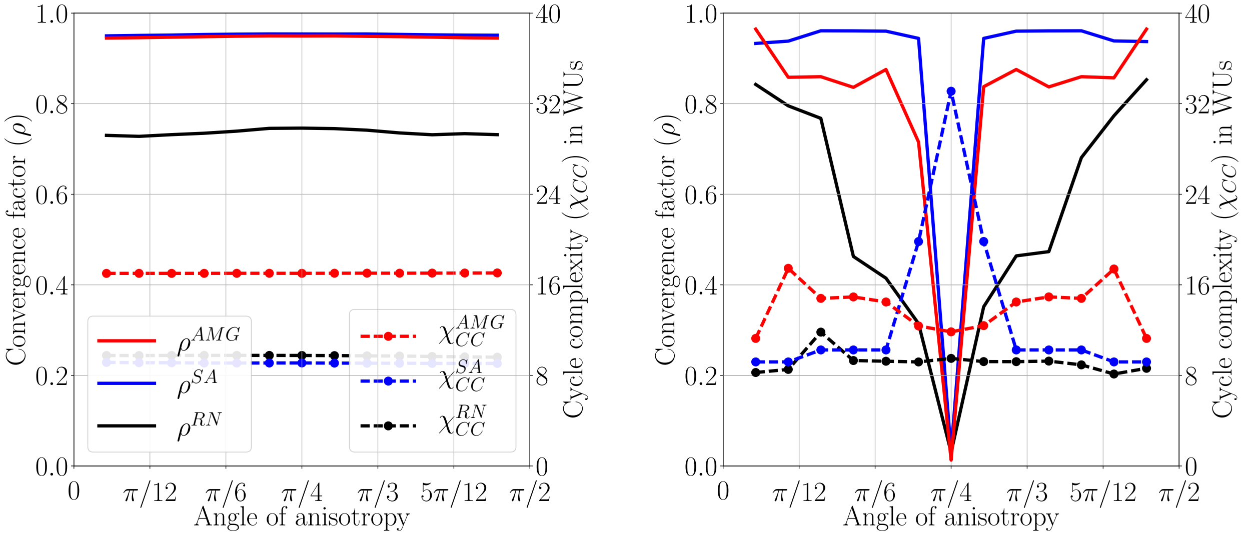
Table 6.3 depicts detailed SC results for using the modified ℓ_1-Jacobi evolution measure (cf. Table 6.2). This change results in similar operator and cycle complexities and nearly identical convergence rates as the original evolution measure. When comparing Tables 6.2 and 6.3, it is apparent that the "Aggregation" phase has been significantly reduced in cost. Additionally, when examining the most efficient solvers, where pre-filtering uses $\theta = 0.2$, the overall savings in the SC are roughly 20%.

Pre-filter θ	Post-filter θ	Aggregation	Candidates	P	RAP	Total SC
–	–	449.1	24.4	473.0	249.5	1195.9
0.2	–	25.5	10.4	53.6	7.6	97.1
–	0.2	43.2	11.7	326.7	16.9	398.6
0.2	0.2	22.7	10.2	63.3	6.9	103.1

Table 6.3: Break down of setup cost in WUs for 3D-anisotropic diffusion.

6.3.2 Totally anisotropic diffusion

Diffusion-like operators are prototypical AMG problems, as they are elliptic and SPD. However, with strong anisotropy, these problems still pose a challenge to multilevel solvers. A 3D example is used in Section 6.3.1 to demonstrate filtering, and variations in the 2D problem in Section 6.2.2. Here, we return to the 2D problem introduced in Section 6.2.2 to consider a structured and unstructured mesh, angles between zero and $^{\pi}/_{2}$, and a comparison with SA and classical AMG. As a test problem, we choose the hardest case of $\epsilon = 0$, or "totally anisotropic diffusion."



(a) Unstructured mesh: $n = 5 \times 10^6$.

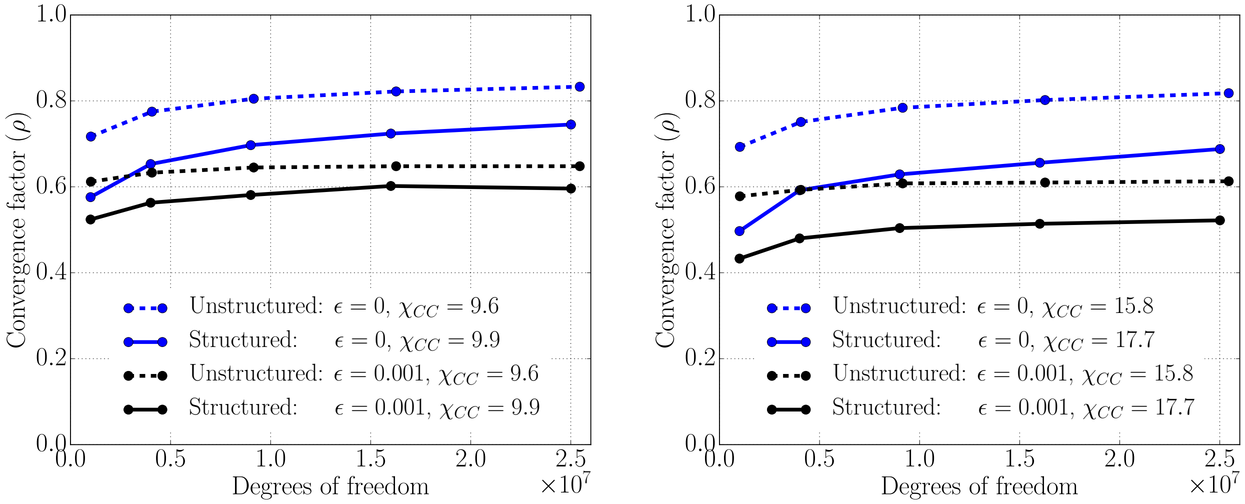
(b) Structured mesh: $n = 8 \times 10^6$.

Figure 6.4: Convergence factor (ρ) and CC (χ_{CC}) for totally anisotropic diffusion ($\epsilon = 0$), with angles in $(0, ^{\pi}/_{2})$. A structured grid of size 2000 × 2000 and an unstructured mesh with resolution $h \approx 1/2000$ are used. Legend entries in (a) apply to (b) as well. RN AMG, shown in black, outperforms SA and classical AMG in all cases.

Figure 6.4 compares RN AMG, classical AMG, and SA applied to totally anisotropic diffusion for various angles. All solves use a symmetric V(1,1)-cycle with symmetric Gauss-Seidel relaxation and CG acceleration. Grid-aligned anisotropies are generally easier to solve than non-grid aligned, hence the excellent convergence factors at $\theta = ^{\pi}/_{4}$ on the structured mesh. In the case of an unstructured mesh, all angles are effectively non-grid aligned resulting in consistent performance across angles.

Classical AMG uses a classical strength measure with drop tolerance of 0.5, and standard CF-splitting and interpolation [145]. SA uses a symmetric strength measure (with a drop tolerance of 0.0, that is, the strength matrix is given by A with each row normalized) and two iterations of weighted Jacobi interpolation smoothing [177]. The root-node solver in this case uses two steps of an evolution strength measure (with a drop tolerance of 4.0), along with six iterations of CG energy-minimization smoothing of P. For the energy minimization, a degree $d = 4$ sparsity pattern is used with filtering, $\theta_{\text{pre}} = \theta_{\text{post}} = 0.1$. Two and three Jacobi smoothing iterations were applied to the SA AMG solver in an attempt to mimic the expanded sparsity pattern used in RN AMG, but the convergence with respect to CC did not improve.

Figure 6.4 highlights the effectiveness of RN AMG over all angles with moderate operator and cycle complexities. For time-to-solution with respect to floating point operations and wall-clock time, RN AMG achieves between 3–30× speed-up in comparison to SA and classical AMG on a structured mesh and unstructured mesh, with moderate cycle complexities in all cases. It should be noted that performance of SA improves when using the modified evolution SOC measure introduced in Section 6.3.1.2, but RN AMG still performs 2–3× faster with respect to time and complexity. This is a notable achievement in performance for this problem, as anisotropic diffusion remains a significant challenge to most solvers.



(a) V(1, 1)-cycle.

(b) W(1, 1)-cycle.

Figure 6.5: Scaling results for RN AMG for anisotropic diffusion with $\epsilon = 0.001, \epsilon = 0$, and $\theta = 3\pi/16$. Cycle complexity, χ_{CC}, is shown and is constant for all problem sizes.

Figure 6.5 demonstrates the scaling of convergence factors as problem size increases for $V(1,1)$-

and $W(1,1)$-cycles. In the case of $\epsilon = 0.001$, V-cycle convergence factors asymptote and scale perfectly, independent of h, on structured and unstructured meshes, up to 25 million unknowns; the SC and CC also scale, but are not shown. However, in the case of $\epsilon = 0$ there is a slow growth in the convergence factor as the problem size increases for both V- and W-cycles. To analyze, consider a convergence factor, $\rho(h)$, dependent on spatial step size h:

$$\rho = \bar{\rho}(1 - ah^q), \tag{6.27}$$

where $q = 1$ in the case of linear finite elements, a is some constant, and $\bar{\rho}$ is the asymptotic convergence factor, $\lim_{h\to 0} \rho = \bar{\rho}$. A log of (6.27) and an expansion yields $\log(1 - ah) = -ah + O(h^2)$ or $-\log(\rho) = -\log(c) + ah$. A linear fit on the three smallest step sizes in Figure 6.5 for $\epsilon = 0$ leads to the following asymptotic convergence factors

$$\text{structured:} \quad \rho \to 0.82, \quad \text{and unstructured:} \quad \rho \to 0.88.$$

Although an asymptotic convergence factor of approximately 0.88 is relatively slow, *scalable* convergence of totally anisotropic diffusion on unstructured meshes has not been achieved by other AMG methods.

In practice, using W-cycles over V-cycles should increase the accuracy of the coarse-grid correction. However, Figure 6.5 reveals W-cycles offer only minor improvements. This indicates that the algebraically smooth error is not well represented by the coarse grid, and that improved strength measures and coarsening routines may be needed to improve the coarse-grid, thereby improving convergence for this problem [17, 21, 23, 39, 125, 128].

6.3.3 Recirculating flow (non-symmetric)

One of the benefits of the root-node approach is the ability to handle a variety of problems, including non-symmetric problems and systems, without redesign of methodology and implementation. In this section, a standard recirculating flow example known as the *double-glazing problem* is used [52], which models temperature distribution over a domain when an external wall is hot. The governing PDE is given by

$$-\epsilon\nabla \cdot \nabla u + \mathbf{b}(\mathbf{x}) \cdot \nabla u = f, \tag{6.28}$$

where $\epsilon = 0.005$ and wind is given by $\mathbf{b}(\mathbf{x}) = [2x_1(1 - x_0^2), -2x_0(1 - x_1^2)]$. Dirichlet boundaries are imposed on the domain $[0, 1] \times [0, 1]$, where $u = 0$ on the north, south and west sides of the domain, and $u = 1$ on the east, leading to boundary layers near corners with discontinuities.

A standard Galerkin finite element method (GFEM) based on a regular triangular mesh is used, resulting in a non-symmetric discrete linear system. Multigrid theory for non-symmetric problems is less developed in comparison to the symmetric case; however, SA has been extended to non-symmetric problems [147] and is used in this section.

Each example uses a V$(1, 1)$-cycle of weighted Jacobi with GMRES acceleration. While Jacobi relaxation is not guaranteed to converge for a non-symmetric problem, it remains around half the cost of using relaxation on the normal equations.[3] For SA, a classical strength measure (with drop tolerance 0.25) is used along with one and three steps of Jacobi smoothing steps applied to P, labeled SA$_1$ and SA$_3$, respectively. The symmetric traditional SA strength measure is not used due to the non-symmetry of the problem. Two steps of the evolution strength measure (with drop tolerance 3.0) is used for RN AMG, along with two iterations of GMRES energy minimization for P with $d = 1$ (labeled RN$_1$), and five iterations of GMRES energy minimization for P with $d = 3$ (labeled RN$_3$).

		2000 × 2000				3000 × 3000				4000 × 4000			
θ	d	SC	OC	CC	ρ	SC	OC	CC	ρ	SC	OC	CC	ρ
–	SA$_1$	72	1.4	5.2	0.74	71	1.4	5.2	0.82	71	1.4	5.2	0.88
	SA$_3$	229	2.3	11.2	0.96	230	2.3	11.2	0.96	227	2.3	11.2	0.93
–	RN$_1$	98	1.4	5.1	0.46	96	1.4	5.0	0.50	95	1.4	4.9	0.45
	RN$_3$	403	2.5	9.7	0.68	407	2.4	9.4	0.63	405	2.3	9.2	0.76
0.1	RN$_1$	126	1.4	5.1	0.52	125	1.4	5.0	0.56	124	1.4	4.9	0.56
	RN$_3$	284	1.8	6.7	0.53	287	1.8	6.7	0.52	285	1.8	6.6	0.63

Table 6.4: Non-symmetric SA and RN AMG for the recirculating flow problem.

Table 6.4 demonstrates that optimal results are achieved for this example with no filtering and a small sparsity pattern ($d = 1$). This agrees with practical experience: generally it is effective to increase the filtering tolerance as the degree of the sparsity pattern for P increases, or as the connectivity of matrix A

[3] Note that RN AMG does require normal-equation relaxation for some highly nonsymmetric systems [108].

increases. This is observed in the 3D-anisotropic diffusion problem, where high connectivity and a $d = 4$ sparsity pattern allows for a large $\theta = 0.2$. If the sparsity pattern increases in distance from the root-node, or the matrix is highly connected, then it is likely there are entries that are not critical to performance and are candidates for removal.

Classical CF AMG is not designed for non-symmetric problems, making RN AMG the clear choice for a problem such as this. Root-node AMG achieves more than a 6× speed-up over SA AMG for the largest problem size considered, *with a lower CC*, and only slightly larger SC. Furthermore, RN AMG convergence factors with degree $d = 1$ appear to have reached an asymptote, while SA AMG is still demonstrating a steady increase with problem size.

Remark 8. *With the recirculating flow, as the grid-size (h) approaches zero, there are two competing factors that contribute to the numerical difficulty of the problem. As $h \to 0$, the diffusive part of the problem, $-\epsilon\nabla \cdot \nabla$, becomes increasingly dominant because the diffusion discretization scales like $\frac{1}{h^2}$, while $\mathbf{b}(\mathbf{x}) \cdot \nabla$ scales like $\frac{1}{h}$. The resulting linear system is, thus, more symmetric and diffusion-like, which is preferable for AMG. However, as $h \to 0$, convergence factors often increase to an asymptotic value (see (6.27) and Figures 6.5a and 6.5b), due to smaller eigenvalues and an increasing number of levels in the AMG hierarchy. Together, these factors correspond to the so-called half-grid Reynolds number or cell Reynolds number, $R_h = \frac{|\mathbf{b}|h}{2\epsilon}$, where convergence factors are expected to be consistent for a fixed R_h, and degrade for $R_h \gg 1$. Figure 6.6 demonstrates this phenomenon. Convergence factors tend to asymptote for a fixed half-grid Reynolds number as $h \to 0$. In this case, convergence degrades by a factor of 10 when increasing from $R_h = 1.25$ to $R_h = 2.5$; thus it is faster to solve a refined problem with $R_h = 1.25$ (and several times as many DOFs), rather than a system with $R_h = 2.5$.*

For many nonsymmetric problems, there will be a reasonable h for which the half-grid Reynolds number is approximately one or less, in which case RN AMG is a likely to be a robust choice of solver. However, in truly advection-dominated cases or the limit of a purely advective problem such as steady state transport, RN AMG does not perform well (typically converges, but with convergence factors $\rho \gg 0.8$ [108]). This suggests that there is still something missing in how we are handling highly nonsymmetric matrices,

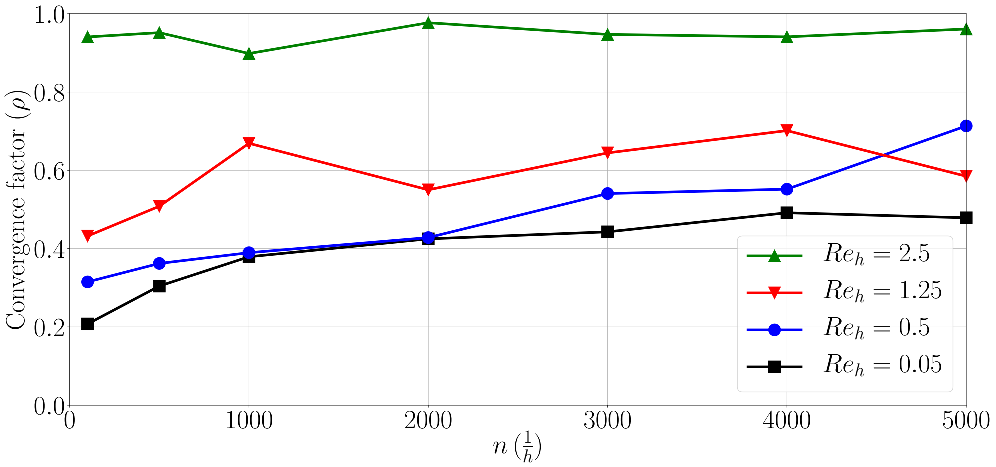


Figure 6.6: RN AMG convergence factor as a function of h, for fixed half-grid Reynolds numbers, $R_h \in \{0.05, 0.5, 1.25, 2.5\}$.

motivating the theoretical framework developed in Chapter 7 and the practical solver developed in Chapters 8 and 9. In particular, the solver developed in Chapters 8 and 9 offers a substantial improvement over RN AMG and other state-of-the-art AMG solvers when applied to highly nonsymmetric problems.

Chapter 7

Convergence of nonsymmetric algebraic multigrid

7.0.1 Motivation

The $\sqrt{A^*A}$-norm and $\sqrt{AA^*}$-norm are introduced in [27] as generalizations of the A-norm. In the geometric MG setting with self-adjoint elliptic differential operators, the A-norm corresponds with the $\mathcal{H}^1$-Sobolev norm, which enforces accuracy of solution values *and* derivatives. This avoids approximate solutions with large oscillations and non-physical behavior that can occur when minimizing, for example, the l^2-norm, which is desirable as we generalize to the nonsymmetric algebraic setting. The $\sqrt{A^*A}$-norm and $\sqrt{AA^*}$-norm also reduce to the A-norm in the case of SPD matrices, in which case classical AMG theory applies, making this approach a true generalization of existing theory, as opposed to an entirely new framework.

Thus, let $L \in \mathbb{R}^{n\times n}$ be some nonsingular matrix with singular value decomposition (SVD) $L = U\Sigma V^*$ and singular values ordered such that $0 < \sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_n$. *For ease of notation in later proofs, we scale L such that $\sigma_n = 1$.* Defining the unitary operator $Q := VU^*$, we can write $\sqrt{L^*L} = QL$ and $\sqrt{LL^*} = LQ$. Each of these matrices are SPD and, to solve $L\mathbf{x} = \mathbf{b}$, we can consider applying classical AMG techniques to either of the equivalent systems

$$QL\mathbf{x} = Q\mathbf{b}, \tag{7.1}$$

$$LQ\mathbf{y} = \mathbf{b} \text{ for } \mathbf{x} = Q\mathbf{y}. \tag{7.2}$$

Note that (7.1) has some similarity to a normal-equation formulation of the problem; however, AMG is typically applied to large, sparse, ill-conditioned matrices, and solving the normal equations squares the condition number of the problem, making it much more difficult to solve. Because Q is unitary, the condition number of QL and LQ are the same as the condition number of L. Then, let $P \in \mathbb{R}^{n\times n_c}$ and $R \in \mathbb{R}^{n\times n_c}$ denote AMG interpolation and restriction operators, respectively. If we define $R := Q^*P$, then NS-AMG applied to $L\mathbf{x} = \mathbf{b}$ has some similarity to classical AMG applied to (7.1), in which case classical AMG convergence theory applies. In practice, Q cannot be easily formed because it is generally dense and requires computing the SVD of L, an $O(n^3)$ procedure. However, these systems provide a framework with which to consider NS-AMG convergence. Although (7.1) is not, in general, computable, one can still apply tools used in conventional AMG convergence theory. In particular, the strong approximation property (SAP) will be important in the development here. Assume the current grid has been partitioned into fine and coarse grid points.

Definition 1 (SAP on P with respect to A)**.** *An interpolation operator, P, satisfies the SAP with respect to SPD matrix A, with constant K_P if, for any $\mathbf{v}$ on the fine grid, there exists a $\mathbf{v}_c$ on the coarse grid such that*

$$\|\mathbf{v} - P\mathbf{v}_c\|_A^2 \leq \frac{K_P}{\|A\|}\|A\mathbf{v}\|^2. \tag{7.3}$$

In the symmetric setting, satisfying the SAP on all levels in a hierarchy is sufficient for multilevel convergence [179]. Nonsymmetric matrices typically lead to a non-orthogonal coarse-grid correction and even two-grid convergence is no longer ensured by the SAP [27]. Because coarse-grid correction is not orthogonal in the nonsymmetric setting, it is important that coarse-grid correction be *stable*, that is, coarse-grid correction can only increase error by some small constant $C \geq 1$:

Definition 2 (Stability of Π)**.**

$$\|\Pi\|_{QL}^2 \leq C, \tag{7.4}$$

where $C \geq 1$ is a small constant, independent of the grid size.

Note that for orthogonal projections such as in the symmetric setting, $C = 1$ and error cannot be increased. For non-orthogonal projections, this constant must be bounded independent of grid level and problem size for scalable convergence. In [27], stability and the SAP on P with respect to the QL-norm, along with additional relaxation to account for potential increases in error from coarse-grid correction, are shown to be sufficient conditions for two-grid convergence in the QA-norm:

Theorem 2 (Two-Grid QL-Convergence (Theorem 2.3, [27])). *Let G be the error-propagation operator for ν iterations of Richardson-relaxation on the normal equations (L^*L), $G := \left(I - \frac{L^*L}{\|L\|^2}\right)^\nu$, and $(I - \Pi)$ the (non-orthogonal) coarse-grid correction defined by interpolation and restriction operators, P and R, respectively* (5.2)*. If P satisfies a SAP with respect to the QL-norm with constant K_P and coarse-grid correction is stable with constant C, then*

$$\|(I - \Pi)G\mathbf{e}\|_{QL} \leq \frac{16CK_P}{25\sqrt{4\nu + 1}}\|\mathbf{e}\|_{QL}.$$

Two-grid convergence of NS-AMG in the QL-norm follows by performing sufficient iterations of relaxation, ν, such that $\|(I - \Pi)G\mathbf{e}\|_{QL} < \|\mathbf{e}\|_{QL}$.

Defining a stable coarse-grid operator is a crux of NS-AMG because it is possible to build P that satisfies a SAP with respect to QL and R that satisfies a SAP with respect to LQ and still get a coarse-grid operator, R^*LP, that is singular or near-singular (see Lemma 9). Coarse-grid correction can then increase error significantly, leading to divergence. However, stability alone is not a very practical constraint; in particular, it does not give us useful information on what to consider when forming R. This is the fundamental part of convergence theory lacking for NS-AMG: conditions on P *and* R that give insight into their respective roles in AMG convergence and can be used in practice to form effective transfer operators.

This work builds on the framework developed in [27], considering convergence of NS-AMG in the $\sqrt{A^*A}$-norm. Section 7.1 introduces further background information and results on the relation between various approximation properties and stability, in particular, showing that a SAP on P with respect to QL and a SAP on R with respect to LQ are not sufficient conditions for convergence. New bases for P and R are then introduced and used to develop sufficient conditions on P and R for stability and two-grid convergence of NS-AMG. Two-grid results are extended in Section 7.2 to develop sufficient conditions for multigrid convergence of NS-AMG in the $\sqrt{A^*A}$-norm. Although the conditions are relatively stringent, this is the first result published on convergence of multilevel methods for nonsymmetric matrices. A discussion on the results of this work and their relation to recently developed, effective NS-AMG solvers is given in Section 7.3, along with a look at future directions of research for NS-AMG.

7.1 Two-grid convergence

7.1.1 Approximation properties

Building on the framework developed in [27], define the following three projections:

$$I - \Pi_1 = I - P(P^*QLP)^{-1}P^*QL,$$
$$I - \Pi_2 = I - R(R^*LQR)^{-1}R^*LQ$$
$$I - \Pi = I - P(R^*LP)^{-1}R^*L,$$

corresponding to a QL-orthogonal projection onto $\mathcal{R}(P)$, an LQ-orthogonal projection onto $\mathcal{R}(R)$, and error-propagation for the Petrov-Galerkin coarse-grid correction used in NS-AMG, respectively. Here, we generalize the use of the SAP, weak approximation property (WAP), and super strong approximation property (SSAP) to the nonsymmetric setting.

Definition 3 (WAP on P with respect to A)**.** *An interpolation operator, P, satisfies the WAP with respect to SPD matrix A, with constant K_w if, for any $\mathbf{v}$ on the fine grid, there exists a $\mathbf{v}_c$ on the coarse grid such that*

$$\|\mathbf{v} - P\mathbf{v}_c\|^2 \leq \frac{K_w}{\|A\|}\|\mathbf{v}\|_A^2. \tag{7.5}$$

Definition 4 (SSAP on P with respect to A)**.** *An interpolation operator, P, satisfies the SSAP with respect to SPD matrix A, with constant K_s if, for any $\mathbf{v}$ on the fine grid, there exists a $\mathbf{v}_c$ on the coarse grid such that*

$$\|\mathbf{v} - P\mathbf{v}_c\|^2 \leq \frac{K_s}{\|A\|^2}\|A\mathbf{v}\|^2. \tag{7.6}$$

Lemma 8 generalizes relations of approximation properties in the SPD setting to the nonsymmetric setting.

Lemma 8 (Equivalence of approximation properties)**.** *Let A be SPD.*

(1) If P satisfies the SSAP with respect to A with constant K_s, then P also satisfies the WAP with respect to A with constant $K_w = K_s$.

(2) If P satisfies the SSAP with respect to A with constant K_s, then P also satisfies the SAP with respect to A with constant $K_P = K_s$.

(3) If P satisfies the SAP with respect to A with constant K_P, then P also satisfies the SSAP with respect to A with constant $K_s = K_P^2$.

(4) If P satisfies the SAP with respect to A with constant K_P, then P also satisfies the WAP with respect to A with constant $K_w = K_P^2$.

Proof.

(1) Assume P satisfies the SSAP with respect to A with constant K_s. For some vector $\mathbf{v}$, let $\mathbf{v}_c$ be a corresponding coarse-grid vector that satisfies the SSAP. Then,

$$\|\mathbf{v} - P\mathbf{v}_c\|^2 \le \frac{K_s}{\|A\|^2}\|A\mathbf{v}\|^2 = \frac{K_s}{\|A\|^2}\|(A)^{1/2}(A)^{1/2}\mathbf{v}\|^2 \le \frac{K_s}{\|A\|^2}\|(A)^{1/2}\|^2\|(A)^{1/2}\mathbf{v}\|^2 = \frac{K_s}{\|A\|}\|\mathbf{v}\|_A^2,$$

which satisfies the WAP on P with respect to A with constant $K_w = K_s$.

(2) Assume P satisfies the SSAP with respect to A with constant K_s. For some vector $\mathbf{v}$, let $\mathbf{v}_c$ be a corresponding coarse-grid vector that satisfies the SSAP. Then,

$$\|\mathbf{v} - P\mathbf{v}_c\|_A^2 \le \|A\|\|\mathbf{v} - P\mathbf{v}_c\|^2 \le \frac{K_s}{\|A\|}\|A\mathbf{v}\|^2,$$

which satisfies the SAP on P with respect to A and constant $K_P = K_s$.

(3) Assume P satisfies the SAP with respect to A and constant K_P. For some vector $\mathbf{v}$, let $\mathbf{v}_c :=$ argmin$\|\mathbf{v} - P\mathbf{v}_c\|_A = \|(I - \Pi_1)\mathbf{v}\|_A$. Orthogonality implies

$$\langle A(\mathbf{v} - P\mathbf{v_c}), P\mathbf{z}\rangle = \langle A(I - \pi_1)v, P\mathbf{z}\rangle = 0, \forall \mathbf{z}.$$

Next, consider $\mathbf{w} = (A)^{-1}(\mathbf{v} - P\mathbf{e}_c)$ and let $\mathbf{w}_c$ be a corresponding coarse-grid vector that satisfies the SAP. Then $\mathbf{v} - P\mathbf{v}_c$

$$\begin{aligned}
\|\mathbf{v} - P\mathbf{v}_c\|^2 &= \langle A(\mathbf{v} - P\mathbf{v}_c), (A)^{-1}(\mathbf{v} - P\mathbf{v}_c)\rangle \\
&= \langle A(\mathbf{v} - P\mathbf{v}_c), (A)^{-1}(\mathbf{v} - P\mathbf{v}_c) - P\mathbf{w}_c\rangle \qquad (\mathit{orthogonality}) \\
&\leq \|\mathbf{v} - P\mathbf{v}_c\|_A \|(A)^{-1}(\mathbf{v} - P\mathbf{v}_c) - P\mathbf{w}_c\|_A \qquad (SAP) \\
&\leq \|\mathbf{v} - P\mathbf{v}_c\|_A \sqrt{\frac{K_P}{\|A\|}} \|A(A)^{-1}(\mathbf{v} - P\mathbf{v}_c)\| \\
&= \|\mathbf{v} - P\mathbf{v}_c\|_A \sqrt{\frac{K_P}{\|A\|}} \|\mathbf{v} - P\mathbf{v}_c\|.
\end{aligned}$$

This implies

$$\|\mathbf{v} - P\mathbf{v}_c\|^2 \leq \frac{K_P}{\|A\|}\|\mathbf{v} - P\mathbf{v}_c\|_A^2 \leq \frac{K_P^2}{\|A\|^2}\|A\mathbf{v}\|^2.$$

Thus, P satisfies the SSAP with respect to A and constant $K_w = K_P^2$.

(4) Assume P satisfies the SAP with respect to A with constant K_P. From 3 above, the SAP on P implies the SSAP on P with respect to A and constant $K_s = K_P^2$. From 2 above, P satisfies a WAP with respect to A with constant $K_w = K_2 = K_P^2$.

□

The next corollary is a result of the proof of part 3 in Lemma 8. It will be useful later.

Corollary 1. *Let A be SPD. Assume P satisfies a SAP with respect to A with constant K_P. Then,*

$$\|(I - \pi_1)\mathbf{v}\|^2 = \frac{K_P}{\|A\|}\|(I - \pi_1)\mathbf{v}\|_A^2.$$

Proof. This result follows from the proof of part 3 in Lemma 8. □

In the non-symmetric development that follows, we consider P that satisfies a SAP with respect to QL and R that satisfies a SAP with respect to LQ. Approximation properties on P with respect to QL ensure that right singular vectors with small singular values well represented in $\mathcal{R}(P)$. Likewise, approximation properties of R with respect to LQ ensure that the left singular vectors with small singular values are well represented in $\mathcal{R}(R)$. Recall, if we define $R := Q^*P$, we have an orthogonal coarse-grid correction and classical (SPD) AMG theory implies. In this case, the optimal P with respect to two-grid convergence is given by letting columns of P be the first n_c right singular vectors, where n_c is the size of the coarse grid [57]. It follows that the optimal R then consists of the first n_c left singular vectors. This provides motivation on building R.

In practice, stability alone is not practical for the development of a solver because it does not give conditions directly on R and P to motivate building a solver. Given that the optimal P and R are given by columns consisting of the first n_c right and left singular vectors, respectively, a natural idea for convergence of NS-AMG is to use approximation properties for P *and* R. Unfortunately, approximation properties alone are not sufficient for stability, as shown in Lemma 9.

Lemma 9.

A SAP on P with respect to QL and the SAP on R with respect to LQ $\not\Rightarrow$ stability of Π.

Proof.

Let $L \in \mathbb{R}^{4\times 4}$ with associated singular value triplets, $(\sigma_i, \mathbf{u}_i, \mathbf{v}_i)$. Assume $\sigma_i \leq 1\,\forall i$, $\sigma_1, \sigma_2 < 1/2$, and $\sigma_3, \sigma_4 \geq 1/2$. Define $P = \left[\mathbf{v}_1/\sqrt{\sigma_1}, \mathbf{v}_2/\sqrt{\sigma_2}, \mathbf{v}_3/\sqrt{\sigma_3}\right]$ and $R = \left[\mathbf{u}_1/\sqrt{\sigma_1}, \mathbf{u}_2/\sqrt{\sigma_2}, \mathbf{u}_4/\sqrt{\sigma_4},\right]$. Then $P^*QLP = R^*LQR = I$ and the SAP on P and R are satisfied with $K_P = K_R = 2$. However, the coarse-grid operator

$$R^*LP = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0 \end{bmatrix}$$

is singular, implying that $\|\Pi\|_{QL}^2$ is unbounded. □

7.1.2 General conditions for stability and two-grid convergence

The crux of convergence theory for NS-AMG is in dealing with a non-orthogonal coarse-grid correction that can increase error. In particular, if R^*LP ends up being a singular or near-singular matrix, then the resulting solver will likely diverge. In [27] this was handled by assuming that the oblique projection Π is stable, with its norm bounded by some small constant greater than one. However, such conditions are not practical in the development of a solver. In the previous section, approximation properties alone were shown to be insufficient conditions for stability. Here, we develop conditions on R and P for stability of coarse-grid correction and, thus, convergence. The basic premise is that Π is invariant over any change of basis for P

and R. If we let B_P and B_R be nonsingular $n_c \times n_c$ square matrices such that $\hat{P} := PB_P$ and $\hat{R} := RB_R$, then,

$$\begin{aligned}\Pi &= P(R^*LP)^{-1}R^*L \\ &= \hat{P}B_P^{-1}(R^*LP)^{-1}B_R^{-*}R^*L \\ &= \hat{P}(\hat{R}^*L\hat{P})^{-1}\hat{R}^*L. \end{aligned} \tag{7.7}$$

Here, we introduce a certain quality measure on P and R that gives sufficient conditions for stability and two-grid convergence when coupled with approximation properties. In particular, we will introduce basis matrices W and Z based on Π_1 and Π_2 and corresponding to the action of P and R, respectively. After proving several results on W and Z, we can establish a stability bound for $\|\Pi\|_{QL}$ accordingly.

Moving forward, we will call two operators A and B *spectrally equivalent* and *norm equivalent* if there exist constants α_s, η_s and α_n, η_n, respectively, such that

$$\alpha_s \leq \frac{\langle A\mathbf{x}, \mathbf{x}\rangle}{\langle B\mathbf{x}, \mathbf{x}\rangle} \leq \eta_s, \qquad \alpha_n \leq \frac{\langle A\mathbf{x}, A\mathbf{x}\rangle}{\langle B\mathbf{x}, B\mathbf{x}\rangle} \leq \eta_n,$$

denoted $A \sim_s B$ and $A \sim_n B$. For self-adjoint, compact operators, $A \sim_n B \implies A \sim_s B$ with the same constants [53].More results on the equivalence of operators in a Hilbert space can be found in [53].

Lemma 10 (Basis for $\mathcal{R}(P)$)**.** *Let $P \in \mathbb{R}^{n \times n_c}$ such that $P^*P \sim_s I_{n_c}$ and P satisfies the SAP with respect to QL, with constant $K_P \geq 1$. Then, there exists a basis for $\mathcal{R}(P)$, $\hat{P} = PB_P$, such that*

$$\hat{P}^*QL\hat{P} \sim_n \begin{pmatrix} \Sigma_{1..k} & \mathbf{0} \\ \mathbf{0} & I \end{pmatrix},$$

for change of basis matrix $B_P \in \mathbb{R}^{n_c \times n_c}$, where $B_P^*B_P \sim_s I$, and similarity constants depend only on K_P.

Proof. To build the basis, we develop a block decomposition of $\mathcal{R}(P)$ in the form $\hat{P} = \begin{pmatrix} W_1 & W_2 \end{pmatrix}$. Pick k such that $K_P\sigma_k < \frac{1}{2}$ and let $V_1 = [v_1, ..., v_k]$ denote the first k right singular vectors. Then, define

$$W_1 := \Pi_1 V_1, \qquad \mathcal{N}_1 := (I - \Pi_1)V_1 = V_1 - W_1.$$

To complete the basis, let $W_2 = [w_{k+1}, ..., w_{n_c}] \subset \mathcal{R}(P)$ be a QL-orthonormal basis for $\mathcal{R}(P)\backslash\mathcal{R}(W_1)$, that is, $W_2^*QLW_2 = I_{n_c-k}$ and $\langle QLw_j, w_\ell\rangle = 0$ for all $j \leq k, \ell > k$. Then

$$\hat{P}^*QL\hat{P} = \begin{pmatrix} W_1^*QLW_1 & \mathbf{0} \\ \mathbf{0} & I_{n_c-k} \end{pmatrix}.$$

Expanding, $W_1^* QLW_1 = V_1^* QLV_1 - \mathcal{N}_1^* QLV_1 - V_1^* QL\mathcal{N}_1 + \mathcal{N}_1^* QL\mathcal{N}_1$. By symmetry of $I - \Pi_1$ in the QL-norm,

$$\langle \mathcal{N}_1^* QLV_1 \mathbf{x}, \mathbf{y} \rangle = \langle QLV_1 \mathbf{x}, (I - \Pi_1) V_1 \mathbf{y} \rangle = \langle QL(I - \Pi_1) V_1 \mathbf{x}, (I - \Pi_1) V_1 \mathbf{y} \rangle =$$
$$\langle QL\mathcal{N}_1 \mathbf{x}, \mathcal{N}_1 \mathbf{y} \rangle = \langle \mathcal{N}_1^* QL\mathcal{N}_1 \mathbf{x}, \mathbf{y} \rangle,$$

and, thus, $\mathcal{N}_1^* QLV_1 = \mathcal{N}_1^* QL\mathcal{N}_1 = V_1^* QL\mathcal{N}_1$. Noting that $V_1^* QLV_1 = \Sigma_{1..k}$,

$$W_1^* QLW_1 = \Sigma_{1..k} - \mathcal{N}_1^* QL\mathcal{N}_1.$$

Due to the identity block, proving the norm equivalence is reduced to proving $W_1^* QLW_1 \sim_s \Sigma_{1..k}$ and $W_1^* QLW_1 \sim_n \Sigma_{1..k}$.

$\mathbf{W_1^* QLW_1 \sim_n \Sigma_{1..k}}$: By the SAP,

$$\begin{aligned} \|\mathcal{N}_1^* QL\mathcal{N}_1 \mathbf{x}\| &= \sup_{\mathbf{y} \neq \mathbf{0}} \frac{\langle QL\mathcal{N}_1 \mathbf{x}, \mathcal{N}_1 \mathbf{y} \rangle}{\|\mathbf{y}\|} \\ &\leq \|(QL)^{\frac{1}{2}} \mathcal{N}_1 \mathbf{x}\| \cdot \frac{\|(QL)^{\frac{1}{2}} \mathcal{N}_1 \mathbf{y}\|}{\|\mathbf{y}\|} \\ &\leq \sqrt{K_P} \|\Sigma_{1..k} \mathbf{x}\| \cdot \sigma_k \sqrt{K_P} \\ &= (1 - \sigma_k K_P) \|\Sigma_{1..k}\|. \end{aligned}$$

By the reverse triangle inequality,

$$\|(\Sigma_{1..k} - \mathcal{N}_1^* QL\mathcal{N}_1) \mathbf{x}\| \geq \Big| \|\Sigma_{1..k} \mathbf{x}\| - \|\mathcal{N}_1^* QL\mathcal{N}_1 \mathbf{x}\| \Big| \geq (1 - \sigma_k K_P) \|\Sigma_{1..k} \mathbf{x}\|.$$

A similar result using the triangle inequality gives, for $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2]^T$,

$$(1 - K_p \sigma_k)^2 \|\Sigma_{1..k} \mathbf{x}_1\|^2 + \|\mathbf{x}_2\|^2 \leq \|\hat{P}^* QL\hat{P} \mathbf{x}\|^2 \leq (1 + K_p \sigma_k)^2 \|\Sigma_{1..k} \mathbf{x}_1\|^2 + \|\mathbf{x}_2\|^2,$$

giving norm equivalence $\hat{P}^* QL\hat{P} \sim_n \begin{pmatrix} \Sigma_{1..k} & \mathbf{0} \\ \mathbf{0} & I_{nc-k} \end{pmatrix}$ with constants $((1 - K_p \sigma_k)^2, (1 + K_p \sigma_k)^2)$.

$\mathbf{B_P^* B_P \sim_s I}$: Note that this is equivalent to

$$\alpha \leq \frac{\|B_P \mathbf{x}\|}{\|\mathbf{x}\|} \leq \beta,$$

where $0 < \alpha \leq \beta$ are independent of level. By assumption, $\frac{\langle P\mathbf{x}, P\mathbf{x}\rangle}{\langle \mathbf{x},\mathbf{x}\rangle}$ is bounded above and below. Suppose that $\frac{\langle \hat{P}\mathbf{x}, \hat{P}\mathbf{x}\rangle}{\langle \mathbf{x},\mathbf{x}\rangle}$ is also bounded above and below. Then,

$$\frac{\langle \hat{P}\mathbf{x}, \hat{P}\mathbf{x}\rangle}{\langle \mathbf{x},\mathbf{x}\rangle} = \frac{\langle PB_P\mathbf{x}, PB_P\mathbf{x}\rangle}{\langle \mathbf{x},\mathbf{x}\rangle} = \frac{\langle PB_P\mathbf{x}, PB_P\mathbf{x}\rangle}{\langle B_P\mathbf{x}, B_P\mathbf{x}\rangle}\frac{\langle B_P\mathbf{x}, B_P\mathbf{x}\rangle}{\langle \mathbf{x},\mathbf{x}\rangle},$$

and it follows that $\alpha < \frac{\langle B_P\mathbf{x}, B_P\mathbf{x}\rangle}{\langle \mathbf{x},\mathbf{x}\rangle} < \beta$, where α and β depend on the bounds for P and $\hat{P}$. Thus it remains to show that $\hat{P}^*\hat{P} \sim_s I$.

$\hat{\mathbf{P}}^*\hat{\mathbf{P}} \sim_{\mathbf{s}} \mathbf{I}$: First we consider the diagonal blocks. By assumotion P satisfies the SAP and, equivalently, the SSAP. Then, for any vector $\alpha \in \mathbb{R}^k$,

$$\begin{aligned}\|\mathcal{N}_1\alpha\|_{QL}^2 &= \Big\|(I-\Pi_1)\sum_{i=1}^k \alpha_i\mathbf{v}_i\Big\|_{QL}^2 \leq K_P\sum_{i=1}^k \alpha_i^2\sigma_1^2 \leq K_P\sigma_k^2\|\alpha\|^2,\\ \|\mathcal{N}_1\alpha\|^2 &= \Big\|(I-\Pi_1)\sum_{i=1}^k \alpha_i\mathbf{v}_i\Big\|^2 \leq K_P^2\sum_{i=1}^k \alpha_i^2\sigma_1^2 \leq K_P^2\sigma_k^2\|\alpha\|^2.\end{aligned}$$

Also note that for all $j \leq k$ and $\ell > k$,

$$0 = \langle QL\mathbf{w}_j, \mathbf{w}_\ell\rangle = \langle QL\Pi_1\mathbf{v}_j, \mathbf{w}_\ell\rangle = \langle QL\mathbf{v}_j, \Pi\mathbf{w}_\ell\rangle = \sigma_j\langle \mathbf{v}_j, \mathbf{w}_\ell\rangle.$$

Thus, $V_1^*W_2 = \mathbf{0}$ and $W_1^*W_2 = -\mathcal{N}_1^*W_2$. Given $K_P\sigma_k < 1$,

$$\begin{aligned}\|W_1\|^2 &= \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\langle (V_1-\mathcal{N}_1)\mathbf{x}, (V_1-\mathcal{N}_1)\mathbf{x}\rangle}{\|\mathbf{x}\|^2}\\ &\geq \sup_{\mathbf{x}\neq\mathbf{0}} 1 - \frac{2\|V_1\mathbf{x}\|\|\mathcal{N}_1\mathbf{x}\|}{\|\mathbf{x}\|^2} + \frac{\|\mathcal{N}_1\mathbf{x}\|^2}{\|\mathbf{x}\|^2}\\ &= \sup_{\mathbf{x}\neq\mathbf{0}} \Big(1 - \frac{\|\mathcal{N}_1\mathbf{x}\|}{\|\mathbf{x}\|}\Big)^2\\ &\geq (1-K_P\sigma_k)^2.\end{aligned}$$

A similar argument based on $\|W_1\|^2 \leq \sup_{\mathbf{x}\neq\mathbf{0}} 1 + \frac{2\|V_1\mathbf{x}\|\|\mathcal{N}_1\mathbf{x}\|}{\|\mathbf{x}\|^2} + \frac{\|\mathcal{N}_1\mathbf{x}\|^2}{\|\mathbf{x}\|^2}$ leads to the upper bound $\|W_1\|^2 \leq (1+K_P\sigma_k)^2$.

Denoting $V_3 := [\mathbf{v}_{k+1}, ..., \mathbf{v}_n]$ and $\Sigma_3 =$diag$(\sigma_{k+1}, ..., \sigma_n)$,

$$\langle W_2\mathbf{x}, W_2\mathbf{x}\rangle = \langle V^*W_2\mathbf{x}, V^*W_2\mathbf{x}\rangle = \langle V_3^*W_2\mathbf{x}, V_3^*W_2\mathbf{x}\rangle.$$

Recall L is scaled such that the largest singular value is equal to one. Then, by QL-orthonormality of columns of W_2,

$$\begin{aligned}
\langle \mathbf{x}, \mathbf{x} \rangle &= \langle (V_3 \Sigma_3 V_3^*) W_2 \mathbf{x}, W_2 \mathbf{x} \rangle \\
&\leq \langle W_2 \mathbf{x}, W_2 \mathbf{x} \rangle \\
&\leq \frac{1}{\sigma_{k+1}} \langle (V_3 \Sigma_3 V_3^*) W_2 \mathbf{x}, W_2 \mathbf{x} \rangle \\
&= \frac{1}{\sigma_{k+1}} \langle \mathbf{x}, \mathbf{x} \rangle.
\end{aligned}$$

Now let $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2]^T$ and consider $\hat{P}^* \hat{P}$ using the above results on $W_1^* W_1$ and $W_2^* W_2$. Using the definition $W_1 = V_1 - \mathcal{N}_1$, orthogonality relation $V_1^* W_2 = \mathbf{0}$, and an ϵ-inequality,

$$\begin{aligned}
\langle \hat{P}^* \hat{P} \mathbf{x}, \mathbf{x} \rangle &= \langle W_1^* W_1 \mathbf{x}_1, \mathbf{x}_1 \rangle - 2 \langle \mathcal{N}_1 \mathbf{x}_1, W_2 \mathbf{x}_2 \rangle + \langle W_2^* W_2 \mathbf{x}_2, \mathbf{x}_2 \rangle \\
&\geq (1 - K_P \sigma_k)^2 \|\mathbf{x}_1\|^2 - 2 K_p \sqrt{\sigma_k} \|\mathbf{x}_1\| \|\mathbf{x}_2\| + \|\mathbf{x}_2\|^2 \\
&\geq \left[(1 - K_P \sigma_k)^2 - \frac{K_P^2 \sigma_k^2}{\epsilon_1} \right] \|\mathbf{x}_1\|^2 + (1 - \epsilon_1) \|\mathbf{x}_2\|^2
\end{aligned}$$

for any $\epsilon_1 > 0$. To bound in $\|\mathbf{x}\|$ for spectral equivalence, we set the constants for $\mathbf{x}_1$ and $\mathbf{x}_2$ equal and solve for ϵ_1 as the positive root of the quadratic function $f_1(\epsilon) = \epsilon^2 + (K_P^2 \sigma_k^2 - 2 K_P \sigma_k) \epsilon - K_P^2 \sigma_k^2$, given by

$$\epsilon_1(K_P \sigma_k) = \frac{K_P \sigma_k}{2} \left(2 - K_P \sigma_k + \sqrt{4 + (2 - K_P \sigma_k)^2} \right).$$

Here $\epsilon_1(0) = 0$ and $\epsilon_1(0.5) = 1$. Calculus shows that ϵ_1 is monotonically increasing over the interval $[0, 1]$ with an approximately linear slope, and thus $\frac{\langle \hat{P}^* \hat{P} \mathbf{x}, \mathbf{x} \rangle}{\langle \mathbf{x}, \mathbf{x} \rangle} \geq 1 - \epsilon_1 > 0$ under the assumption that $0 < K_P \sigma_k < \frac{1}{2}$.

Similarly, an upper bound is obtained as $\frac{\langle \hat{P}^* \hat{P} \mathbf{x}, \mathbf{x} \rangle}{\langle \mathbf{x}, \mathbf{x} \rangle} \leq \frac{1 + \epsilon_2}{\sigma_k}$, where ϵ_2 is the positive root of the quadratic function $f_2(\epsilon) = \epsilon^2 + (1 - \sigma_k (1 + K_P \sigma_k)^2) \epsilon - K_P^2 \sigma_k^3$, given by

$$\epsilon_1(K_P \sigma_k) = \frac{-(1 - \sigma_k (1 + K_P \sigma_k)^2) + \sqrt{(1 - \sigma_k (1 + K_P \sigma_k)^2)^2 + 4 K_P^2 \sigma_k^3}}{2}.$$

□

Corollary 2 (Basis for $\mathcal{R}(R)$)**.** *Let $R \in \mathbb{R}^{n \times n_c}$ such that $R^* R \sim_s I_{n_c}$ and R satisfies the SAP with respect to LQ, with constant $K_R \geq 1$. Then, there exists a basis for $\mathcal{R}(R)$, $\hat{R} = R B_R$, such that*

$$\hat{R}^* L Q \hat{R} \sim_n \begin{pmatrix} \Sigma_{1..k} & \mathbf{0} \\ \mathbf{0} & I \end{pmatrix}$$

for change of basis matrix $B_R \in \mathbb{R}^{n_c \times n_c}$, where $B_R^* B_R \sim_s I$, and similarity constants depend only on K_R.

Proof. The proof is equivalent to that of Lemma 10, but this time using the LQ-norm and left singular vectors. Pick k such that $K_R\sigma_k < \frac{1}{2}$ and let $U_1 = [u_1, ..., u_k]$ denote the first k left singular vectors. Recall that $\Pi_2 = R(R^*LQR)^{-1}R^*LQ$ is the LQ-orthogonal projection onto $\mathcal{R}(R)$ and define $\hat{R} := \begin{pmatrix} Z_1 & Z_2 \end{pmatrix}$, where

$$Z_1 = \Pi_2 U_1 = \Pi_2 Q^* V_1,$$

and $Z_2 = [z_{k+1}, ..., z_{n_c}] \subset \mathcal{R}(R)$ is an LQ-orthonormal basis for $\mathcal{R}(R)\backslash\mathcal{R}(Z_1)$. Denote

$$\mathcal{M}_1 := (I - \Pi_2)U_1 = U_1 - Z_1 = Q^*V_1 - Z_1. \tag{7.8}$$

The remainder of the proof follows as in Lemma 10. □

Corollary 3 (Spectral equivalence)**.** *Under the assumptions and bases developed in Lemma 10 and Corollary 2,*

$$\hat{P}^*QL\hat{P} \sim_s \begin{pmatrix} \Sigma_{1..k} & \mathbf{0} \\ \mathbf{0} & I \end{pmatrix},$$
$$\hat{R}^*LQ\hat{R} \sim_s \begin{pmatrix} \Sigma_{1..k} & \mathbf{0} \\ \mathbf{0} & I \end{pmatrix},$$

with the same constants as for norm equivalence.

Proof. Given that each of the above operators are compact and self-adjoint, norm-equivalence from Lemma 10 and Corollary 2 imply spectral equivalence with the same constants [53]. □

Corollary 4. *Given basis vectors W_2 and Z_2, as defined in Lemma 10 and Corollary 2 and the first k left and right singular vectors U_1 and V_1, respectively,*

$$U_1^*LW_2 = Z_2^*LV_1 = \mathbf{0}.$$

Proof. From the definition of W_1 and W_2, we have $W_1^* QLW_2 = \mathbf{0}$ and $\Pi_1 W_2 = W_2$. Then

$$\begin{aligned}
\mathbf{0} &= W_1^* QLW_2 \\
&= (\Pi_1 V_1)^* QLW_2 \\
&= (V_1 - (I - \Pi_1)V_1)^* QLW_2 \\
&= V_1^* QLW_2 - V_1^*(I - \Pi_1)^* QLW_2 \\
&= U_1^* LW_2 - V_1^* QL(I - \Pi_1)W_2 \\
&= U_1^* LW_2.
\end{aligned}$$

By a similar argument, we have $Z_2^* LV_1 = \mathbf{0}$. □

The next goal is to prove the stability of Π under an additional assumption. Recall that if P satisfies the SAP with respect to QL, then $\hat{P} = PB_P$ is decomposed as $\hat{P} = [W_1, W_2]$, where $W_2^* QLW_2 = I_{n_c - k}$ and $\|W_2 \mathbf{x}\|^2 \leq \frac{1}{\sigma_k}\|\mathbf{x}\|^2$. Likewise, if R satisfies the SAP with respect to LQ, then $\hat{R} = RB_R$ is decomposed as $\hat{R} = [Z_1, Z_2]$, where $Z_2^* QLZ_2 = I_{n_c - k}$ and $\|Z_2 \mathbf{x}\|^2 \leq \frac{1}{\sigma_k}\|\mathbf{x}\|^2$. In what follows, we must make an additional assumption on $Z_2^* LW_2 = (QZ_2)^* QLW_2$. It is clear that

$$\|Z_2^* LW_2\|^2 = \sup_{\mathbf{x},\mathbf{y} \neq \mathbf{0}} \frac{\langle QLW_2 \mathbf{x}, QZ_2 \mathbf{y}\rangle}{\|\mathbf{x}\|\|\mathbf{y}\|} \leq \frac{\|W_2 \mathbf{x}\|_{QL} \|Z_2 \mathbf{y}\|_{QL}}{\|\mathbf{x}\|\|\mathbf{y}\|} \leq 1.$$

The assumption is on the lower bound:

$$\frac{\|Z_2^* LW_2 \mathbf{x}\|}{\|\mathbf{x}\|} \geq \delta,$$

for some δ to be specified. With the SAP on P and R and this additional assumption, we can establish the bound

$$\|\Pi\|_{QL}^2 \leq C_\Pi,$$

where C_Π depends only on K_P, K_R and K_2. In particular, this means that it is independent of problem size and grid level.

We first need to prove the following technical lemma, bounding a block matrix in norm from above and below (Lemma 11), which will then be used in the proof of stability in Theorem 3 and multilevel convergence in Section 7.2.

Lemma 11. *Consider the block matrix* $\begin{pmatrix} A & -B \\ -C & D \end{pmatrix}$ *and suppose*

$$a_0\|\mathbf{x}\| \leq \|A\mathbf{x}\| \leq a_1\|\mathbf{x}\|, \qquad \|B\mathbf{x}\| \leq b\|\mathbf{x}\|,$$

$$d_0\|\mathbf{x}\| \leq \|D\mathbf{x}\| \leq d_1\|\mathbf{x}\|, \qquad \|C\mathbf{x}\| \leq c\|\mathbf{x}\|,$$

for $a_0, a_1, b, c, d_0, d_1 > 0$ *and* $d_0 > \frac{bc}{a_0}$. *Then*

$$\eta_0 \leq \frac{\left\| \begin{pmatrix} A & -B \\ -B & D \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \right\|^2}{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2} \leq \eta_1,$$

where

$$\begin{aligned} \eta_0 &= \frac{a_0^2 + b^2 + c^2 + d_0^2 - \sqrt{(a_0^2 + b^2 - c^2 - d_0^2)^2 + 4(a_0c + bd_0)^2}}{2} > 0, \\ \eta_1 &= \frac{a_1^2 + b^2 + c^2 + d_1^2 - \sqrt{(a_1^2 + b^2 - c^2 - d_1^2)^2 + 4(a_1c + bd_1)^2}}{2} > 0. \end{aligned}$$

Proof. Starting with the lower bound, an ϵ-inequality can be used to bound (7.13) below in norm:

$$\begin{aligned} \left\| \begin{pmatrix} A & -B \\ -C & D \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \right\|^2 &= \|A\mathbf{x} - B\mathbf{y}\|^2 + \|C\mathbf{x} - D\mathbf{y}\|^2 \\ &= \|A\mathbf{x}\|^2 - 2\langle A\mathbf{x}, B\mathbf{y}\rangle + \|B\mathbf{y}\|^2 + \|C\mathbf{x}\|^2 - 2\langle C\mathbf{x}, D\mathbf{y}\rangle + \|D\mathbf{y}\|^2 \\ &\geq (1 - \epsilon_1)\|A\mathbf{x}\|^2 - ({}^1\!/_{\epsilon_1} - 1)\|B\mathbf{y}\|^2 + (1 - \epsilon_2)\|D\mathbf{y}\|^2 - ({}^1\!/_{\epsilon_2} - 1)\|C\mathbf{x}\|^2 \\ &\geq \Big[a_0^2(1 - \epsilon_1) - c^2({}^1\!/_{\epsilon_2} - 1)\Big]\|\mathbf{x}\|^2 + \Big[d_0^2(1 - \epsilon_2) - b^2({}^1\!/_{\epsilon_1} - 1)\Big]\|\mathbf{y}\|^2 \end{aligned}$$

for any $\epsilon_1, \epsilon_2 \in (0, 1]$. Note that the upper bound on ϵ_1 and ϵ_2 is necessary to keep the leading constants on $\|A\mathbf{x}\|^2$ and $\|D\mathbf{y}\|^2$ positive because we bounded these from below, and vice versa for $\|B\mathbf{y}\|^2$ and $\|C\mathbf{x}\|^2$. This leads to a system of constraints

$$\begin{aligned} C_1(\epsilon_1, \epsilon_2) &:= a_0^2(1 - \epsilon_1) - c^2({}^1\!/_{\epsilon_2} - 1) > 0, \\ C_2(\epsilon_1, \epsilon_2) &:= d_0^2(1 - \epsilon_2) - b^2({}^1\!/_{\epsilon_1} - 1) > 0, \end{aligned} \tag{7.9}$$

for some $\epsilon_1, \epsilon_2 \in (0, 1]$. The boundary of these constraints in the (ϵ_1, ϵ_2)-plane is given by the functions

$$\begin{aligned} \widehat{\epsilon_2}(\epsilon_1) &= \frac{c^2}{c^2 + a_0^2(1 - \epsilon_1)}, \\ \widetilde{\epsilon_2}(\epsilon_1) &= 1 + \frac{b^2}{d_0^2} - \frac{b^2}{d_0^2\epsilon_1}, \end{aligned}$$

with the region of points satisfying the constraints bounded below by $\widehat{\epsilon_2}$ and above by $\widetilde{\epsilon_2}$. A little algebra shows that $\widehat{\epsilon_2}$ is concave up, $\widetilde{\epsilon_2}$ concave down, and both functions are monotonically increasing over $(0, 1]$

with a crossover point at $\widehat{\epsilon_2}(1) = \widetilde{\epsilon_2}(1) = 1$. It follows that there exists some region within $(0,1) \times (0,1)$ (constraints on ϵ_1 and ϵ_2) that satisfies (7.9) if and only if $\widehat{\epsilon_2}'(1) > \widetilde{\epsilon_2}'(1)$, which reduces to

$$d_0 > \frac{bc}{a_0}.$$

The maximum bound is obtained by setting the leading constants on $\|\mathbf{x}\|^2$ and $\|\mathbf{y}\|^2$ equal. Thus we will consider a constrained maximization over C_1 such that $C_1 = C_2$ (or vice versa). Since we are maximizing the intersection of two convex functionals, which is also convex, the maximum is unique. Thus consider $\epsilon_2(\epsilon_1)$ and denote $\epsilon_2' := \frac{\partial \epsilon_2}{\partial \epsilon_1}$. Then, at the maximum, we must have $\frac{\partial}{\partial \epsilon_1} C_1(\epsilon_1, \epsilon_2(\epsilon_1)) = \frac{\partial}{\partial \epsilon_1} C_1(\epsilon_1, \epsilon_2(\epsilon_1)) = 0$:

$$\begin{aligned} -a_0^2 + \frac{c^2}{\epsilon_2^2}\epsilon_2' = 0 \qquad &\Longrightarrow \qquad \epsilon_2' = \frac{a_0^2}{c^2}\epsilon_2^2, \\ -d_0^2\epsilon_2' + \frac{b^2}{\epsilon_1^2} = 0 \qquad &\Longrightarrow \qquad \epsilon_2' = \frac{b^2}{d_0^2\epsilon_1^2}. \end{aligned}$$

Setting the functions for ϵ_2' equal leads to the constraint $\epsilon_2 = \frac{bc}{a_0 d_0 \epsilon_1}$, and plugging into C_1 and C_2 gives

$$\begin{aligned} C_1(\epsilon_1) &= a_0^2 + c^2 - \epsilon_1\left(a_0^2 + \frac{a_0 c d_0}{b}\right), \\ C_2(\epsilon_1) &= d_0^2 + b^2 - \frac{1}{\epsilon_1}\left(\frac{bcd_0}{a_0} + b^2\right). \end{aligned}$$

Setting $C_1 = C_2$ leads to a quadratic function in ϵ_1:

$$\epsilon_1^2\left(a_0^2 + \frac{a_0 c d_0}{b}\right) + \epsilon_1\left(b^2 + d_0^2 - a_0^2 - c^2\right) - b\left(\frac{cd_0}{a_0} + b\right) = 0.$$

Because $a_0, b, c, d_0 > 0$, we have $-b\left(\frac{cd_0}{a_0} + b\right) < 0$ and, thus, there exists exactly one positive root, given by

$$\epsilon_1 = \frac{(a_0^2 + c^2 - b^2 - d_0^2) + \sqrt{(a_0^2 + c^2 - b^2 - d_0^2)^2 + 4(a_0 b + cd_0)^2}}{2\left(a^2 + \frac{a_0 c d_0}{b}\right)}.$$

Plugging into C_1 gives

$$C_1(\epsilon_1) = C_2(\epsilon_1) = \frac{a_0^2 + b^2 + c^2 + d_0^2 - \sqrt{(a_0^2 + c^2 - b^2 - d_0^2)^2 + 4(a_0 b + cd_0)^2}}{2}, \tag{7.10}$$

where $\eta_0 := C_1(\epsilon_1)$.

A similar derivation can be used for an upper bound. We bound in norm from above, again using an ϵ-inequality, and seek to minimize the intersection of

$$C_3(\epsilon_1, \epsilon_2) := a_1^2(1 + \epsilon_1) + c^2(1 + {}^{1}/_{\epsilon_2}),$$

$$C_4(\epsilon_1, \epsilon_2) := d_1^2(1 + \epsilon_2) + b^2(1 + {}^{1}/_{\epsilon_1}).$$

Using the same process as above, the minimum is obtained for the single positive root of the quadratic equation

$$\epsilon_1^2\left(a_1^2+\frac{a_1cd_1}{b}\right)+\epsilon_1\left(a_1^2+c^2-b^2-d_1^2\right)-b\left(\frac{cd_1}{a_1}+b\right)=0,$$

given by

$$\epsilon_1=\frac{(b^2+d_1^2-a_1^2-c_1^2)+\sqrt{(a_1^2+c^2-b^2-d_1^2)^2+4(a_1b+cd_1)^2}}{2\left(a_1^2+\frac{a_1cd_1}{b}\right)}.$$

Plugging into C_3 and C_4, we solve for an upper bound

$$\eta_1=\frac{a_1^2+b^2+c^2+d_1^2-\sqrt{(a_1^2+c^2-b^2-d_1^2)^2+4(a_1b+cd_1)^2}}{2}. \tag{7.11}$$

□

Theorem 3 (Stability)**.** *Assume that P satisfies the SAP with respect to QL for the first k right singular vectors, with constant K_P, and R satisfies the SAP with respect to LQ for the first k left singular vectors, with constant K_R. Let $K_1:=\max\{K_P,K_R\}$, and decompose P and R in their change of bases (Lemma 10, Corollary 2) as*

$$\hat{P}=[W_1,W_2],\qquad \hat{R}=[Z_1,Z_2],$$

where $dim(W_1)=dim(Z_1)=k$, and k is chosen such that for singular vectors $k+1,...,n_c$, assume $\exists\ \delta$

$$0<\frac{K_1\sigma_k}{1-3K_1\sigma_k}<\delta\leq 1.$$

such that $\delta\|\mathbf{x}\|\leq\|Z_2^*LW_2\mathbf{x}\|\leq\|\mathbf{x}\|$. Then

$$\|\Pi\|_{QL}^2\leq C_\Pi,$$

where C_Π is specified in the proof.

Proof. First recall from (7.7) that $\Pi=P(R^*LP)^{-1}R^*L=\hat{P}(\hat{R}^*L\hat{P})^{-1}\hat{R}^*L$. Denote $\hat{L}_c:=\hat{R}^*L\hat{P}$. Taking an orthogonal decomposition of the space $\mathbf{x} = Q\hat{R}\mu + (Q\hat{R})^{\perp_{QL}}\nu$, where $(Q\hat{R})^{\perp_{QL}} := I - Q\hat{R}(\hat{R}^*LQ\hat{R})^{-1}\hat{R}^*L$,

$$\begin{aligned}
\|\Pi\|_{QL}^2 &= \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\Pi\mathbf{x}\|_{QL}^2}{\|\mathbf{x}\|_{QL}^2} \\
&= \sup_{\mu,\nu\neq\mathbf{0}} \frac{\left\langle QL\hat{P}\hat{L}_c^{-1}\hat{R}L(Q\hat{R}\mu + (Q\hat{R})^{\perp_{QL}}\nu), \hat{P}\hat{L}_c^{-1}\hat{R}L(Q\hat{R}\mu + (Q\hat{R})^{\perp_{QL}}\nu)\right\rangle}{\left\langle QL(Q\hat{R}\mu + (Q\hat{R})^{\perp_{QL}}\nu), Q\hat{R}\mu + (Q\hat{R})^{\perp_{QL}}\nu\right\rangle} \\
&= \sup_{\mu\neq\mathbf{0}} \frac{\left\|(\hat{P}^*QL\hat{P})^{\frac{1}{2}}\hat{L}_c^{-1}(\hat{R}^*LQ\hat{R})\mu\right\|^2}{\left\|(\hat{R}^*LQ\hat{R})^{\frac{1}{2}}\mu\right\|^2 + \left\|(Q\hat{R})^{\perp_{QL}}\nu\right\|_{QL}^2}.
\end{aligned}$$

Let $\eta := (\hat{R}^*LQ\hat{R})^{\frac{1}{2}}\mu$. Then

$$\|\Pi\|_{QL}^2 \leq \sup_{\eta\neq\mathbf{0}} \frac{\left\|(\hat{P}^*QL\hat{P})^{\frac{1}{2}}\hat{L}_c^{-1}(\hat{R}^*LQ\hat{R})^{\frac{1}{2}}\eta\right\|^2}{\|\eta\|^2}.$$

From Lemma 10,

$$\begin{aligned}
0 < 1 - K_1\sigma_k \leq \frac{\langle \hat{P}^*QL\hat{P}\mathbf{x}, \mathbf{x}\rangle}{\left\langle \begin{pmatrix} \Sigma_{1..k} & \mathbf{0} \\ \mathbf{0} & I_{n_c-k}\end{pmatrix}\mathbf{x}, \mathbf{x}\right\rangle} \leq 1, \\
0 < 1 \leq \frac{\langle (\hat{R}^*LQ\hat{R})^{-1}\mathbf{x}, \mathbf{x}\rangle}{\left\langle \begin{pmatrix} \Sigma_{1..k} & \mathbf{0} \\ \mathbf{0} & I_{n_c-k}\end{pmatrix}\mathbf{x}, \mathbf{x}\right\rangle} \leq \frac{1}{1-K_1\sigma_k},
\end{aligned}$$

in which case

$$\left\|(\hat{P}^*QL\hat{P})^{\frac{1}{2}}\hat{L}_c^{-1}(\hat{R}^*LQ\hat{R})^{\frac{1}{2}}\right\|^2 \leq \left\|\begin{pmatrix} \Sigma_{1..k}^{\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & I_{nc-k}\end{pmatrix}\hat{L}_c^{-1}\begin{pmatrix} \Sigma_{1..k}^{\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & I_{nc-k}\end{pmatrix}\right\|^2. \tag{7.12}$$

Instead of bounding in norm, we bound the inverse away from zero by proving appropriate bounds on each block and invoking Lemma 11. Recall from Lemma 10 and Corollary 2 that we can expand $W_1 = V_1 - \mathcal{N}_1$ and $Z_1 = U_1 - \mathcal{M}_1$. It then follows from Corollary 4 that $Z_2^*LW_1 = -Z_2^*L\mathcal{N}_1$ and $Z_1^*LW_2 = -\mathcal{M}_1^*LW_2$, and the inverse of (7.12) is given by

$$\begin{aligned}
\begin{pmatrix} \Sigma_{1..k}^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & I_{n_c-k}\end{pmatrix}\hat{L}_c\begin{pmatrix} \Sigma_{1..k}^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & I_{n_c-k}\end{pmatrix} &= \begin{pmatrix} \Sigma_{1..k}^{-\frac{1}{2}}Z_1^*LW_1\Sigma_{1..k}^{-\frac{1}{2}} & \Sigma_{1..k}^{-\frac{1}{2}}Z_1^*LW_2 \\ Z_2^*LW_1\Sigma_{1..k}^{-\frac{1}{2}} & Z_2^*LW_2\end{pmatrix} \\
&= \begin{pmatrix} \Sigma_{1..k}^{-\frac{1}{2}}Z_1^*LW_1\Sigma_{1..k}^{-\frac{1}{2}} & -\Sigma_{1..k}^{-\frac{1}{2}}\mathcal{M}_1^*LW_2 \\ -Z_2^*L\mathcal{N}_1\Sigma_{1..k}^{-\frac{1}{2}} & Z_2^*LW_2\end{pmatrix}.
\end{aligned} \tag{7.13}$$

We will now bound the diagonal blocks in norm from below and the negative off-diagonal blocks from above.

First consider the upper right block:

$$\begin{aligned}\|\Sigma_1^{-\frac{1}{2}}\mathcal{M}_1^*LW_2\| &= \sup_{\mathbf{x},\mathbf{y}\neq\mathbf{0}} \frac{\langle \Sigma_1^{-\frac{1}{2}}\mathcal{M}_1^*LW_2\mathbf{x},\mathbf{y}\rangle}{\|\mathbf{x}\|\|\mathbf{y}\|} \\ &= \sup_{\mathbf{x},\mathbf{y}\neq\mathbf{0}} \frac{\langle (QL)^{\frac{1}{2}}W_2\mathbf{x},(QL)^{\frac{1}{2}}\mathcal{M}_1\Sigma_1^{-\frac{1}{2}}\mathbf{y}\rangle}{\|\mathbf{x}\|\|\mathbf{y}\|} \\ &\leq \sup_{\mathbf{x},\mathbf{y}\neq\mathbf{0}} \frac{\|(QL)^{\frac{1}{2}}W_2\mathbf{x}\|\|(LQ)^{\frac{1}{2}}\mathcal{M}_1\Sigma_1^{-\frac{1}{2}}\mathbf{y}\|}{\|\mathbf{x}\|\|\mathbf{y}\|}.\end{aligned}$$

Columns of W_2 are QL-orthonormal, so $\frac{\|(QL)^{\frac{1}{2}}W_2\mathbf{x}\|}{\|\mathbf{x}\|} = 1$. Defining $\mathbf{z} = \Sigma_1^{-\frac{1}{2}}\mathbf{y}$ and applying the SAP on R with respect to LQ,

$$\begin{aligned}\|\Sigma_1^{-\frac{1}{2}}\mathcal{M}_1^*LW_2\|^2 &\leq \sup_{\mathbf{z}\neq\mathbf{0}} \frac{\langle LQ(I-\Pi_2)U_1\mathbf{z},(I-\Pi_2)U_1\mathbf{z}\rangle}{\langle\Sigma_1\mathbf{y},\mathbf{y}\rangle} \\ &\leq \sup_{\mathbf{y}\neq\mathbf{0}} \frac{\frac{K_R}{\|LQ\|}\mathbf{y}^*U_1^*U\Sigma^2U^*U_1\mathbf{y}}{\mathbf{y}^*\Sigma_1\mathbf{y}} \\ &= \frac{K_R\sigma_k}{\sigma_n}.\end{aligned} \tag{7.14}$$

An identical derivation bounds $\|Z_2^*L\mathcal{N}_1\Sigma_{1..k}^{-\frac{1}{2}}\|^2 \leq \frac{K_P\sigma_k}{\sigma_n}$. Recall $K_1 := \max\{K_P, K_R\}$ and L is scaled such that $\sigma_n = 1$. Expanding the upper left block and using the reverse triangle inequality along with a similar proof to the off-diagonal blocks gives

$$\|I - \Sigma_{1..k}^{-\frac{1}{2}}\mathcal{M}_1^*L\mathcal{N}_1\Sigma_{1..k}^{-\frac{1}{2}}\|^2 \geq (1-3\sigma_kK_1)^2.$$

Given bounds on each block, we can now bound the inverse (7.13) through Lemma 11. In this case, $a = 1-3\sigma_kK_1$, $b = c = \sqrt{\sigma_kK_1}$, and $d = \delta$. Recall from Lemma 11, we must have $d > \frac{bc}{a}$, in this case leading to the constraint $\delta > \frac{\sigma_kK_1}{1-3\sigma_kK_1}$. Plugging in to (7.10) gives the bound

$$\left\|\begin{pmatrix}\Sigma_{1..k}^{-\frac{1}{2}} & \mathbf{0}\\ \mathbf{0} & I_{n_c-k}\end{pmatrix}\hat{L}_c\begin{pmatrix}\Sigma_{1..k}^{-\frac{1}{2}} & \mathbf{0}\\ \mathbf{0} & I_{n_c-k}\end{pmatrix}\begin{pmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{pmatrix}\right\|^2 \geq \frac{1}{C_\Pi}\|\mathbf{x}\|^2 > 0,$$

where

$$C_\Pi := \frac{2}{(1-3\sigma_kK_1)^2 + 2\sigma_kK_1 + \delta^2 - \sqrt{((1-3\sigma_kK_1)^2-\delta^2)^2 + 4\sigma_k(1-3\sigma_kK_1+\delta)^2}}. \tag{7.15}$$

We then end up with the following chain of inequalities showing stability of Π:

$$\begin{aligned}
\|\Pi\|_{QL}^2 &\leq \left\| (\hat{P}^*QL\hat{P})^{\frac{1}{2}} \hat{L}_c^{-1} (\hat{R}^*LQ\hat{R})^{\frac{1}{2}} \right\|^2 \\
&\leq \left\| \begin{pmatrix} \Sigma_{1..k}^{\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & I_{nc-k} \end{pmatrix} \hat{L}_c^{-1} \begin{pmatrix} \Sigma_{1..k}^{\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & I_{nc-k} \end{pmatrix} \right\|^2 \\
&= \sup_{\mathbf{y}\neq\mathbf{0}} \frac{\|\mathbf{y}\|^2}{\left\| \begin{pmatrix} \Sigma_{1..k}^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & I_{n_c-k} \end{pmatrix} \hat{L}_c \begin{pmatrix} \Sigma_{1..k}^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & I_{n_c-k} \end{pmatrix} \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} \right\|^2} \\
&\leq C_\Pi.
\end{aligned}$$

□

Under the additional assumption that the SAP on P holds for all vectors, two-grid convergence follows from Theorem 2 [27].

7.1.3 Discussion

In the SPD case, stability of coarse-grid correction is trivial. For general nonsymmetric matrices, stability and overcoming the typical loss of orthogonality is the crux in developing convergence theory as well as a convergent algorithm. Despite the change of basis from transfer operators used in practice, Theorem 3 provides important information on how to build R and P for a stable coarse-grid correction. As typical in AMG, R and P must satisfy the SAP, particularly for the lowest-energy left and right singular vectors, respectively. For larger singular values, the SAP holds trivially and, for SPD matrices, this effectively means that one only must pay attention to singular vectors with small singular values. In the nonsymmetric setting, stability requires the additional constraint there exists some δ such that $0 < \frac{\sigma_k K_1}{1-3\sigma_k K_1}\|\mathbf{x}\| < \delta\|\mathbf{x}\| \leq \|Z_2^* L W_2 \mathbf{x}\|$. Broadly, this means that the action of R and P on vectors associated with large singular values is also important.

By QL- and LQ-orthonormality of W_2 and Z_2, we know that

$$(Z_2^* U \Sigma^{\frac{1}{2}}) \Sigma^{\frac{1}{2}} U^* Z_2 = (W_2^* V \Sigma^{\frac{1}{2}}) \Sigma^{\frac{1}{2}} V^* W_2 = I.$$

For stability, we are interested in the action of $Z_2^* L W_2 = (Z_2^* U \Sigma^{\frac{1}{2}})(\Sigma^{\frac{1}{2}} V^* W_2)$. In the SPD case, $U = V$, $Z_2 = W_2$, and we have $Z_2^* L W_2 = I$, implying $\delta = 1$. In the nonsymmetric setting, we want the action of

$U^* Z_2 \mathbf{x} \sim V^* W_2 \mathbf{x}$. This is saying that we need left and right singular vectors associated with large singular values to have a similar action on R and P, respectively. For example, if $v_\ell \notin \mathcal{R}(P)$, for some ℓ in the middle or upper part of the spectrum, it is important that $u_\ell \notin \mathcal{R}(R)$. Unlike in SPD problems when eigenvectors associated with large eigenvalues trivially satisfy the SAP (with the zero vector) and can be largely ignored when constructing a solver, the nonsymmetric setting requires attention in all parts of the spectrum for a bounded coarse-grid correction.

Although singular values and vectors are typically not available in practice, it is worth considering the bounds proved in Theorem 3 to make sure that they are reasonable; after all, potential increases in error due to a non-orthogonal coarse-grid correction must be overcome by effective relaxation. For example, if the best we can do is bound $C_\Pi < 100$, this does not suggest that two-grid convergence is attainable in practice. Fortunately, if the assumptions of Theorem 3 are satisfied well, we attain nice bounds on C_Π, as shown in Figure 7.1.

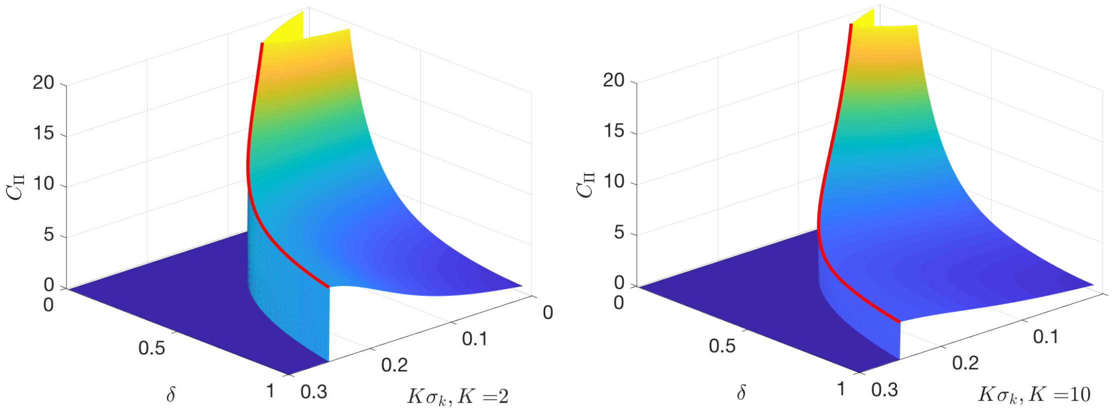


Figure 7.1: Explicit bounds $\|\Pi\|_{QL}^2 \leq C_\Pi$ as given in (7.15) and Theorem 3. The red line gives the boundary of the constraint $\delta > \frac{\sigma_k K_1}{1-3\sigma_k K_1}$. For an appropriate choice of k and sufficiently satisfied constraint δ, the non-orthogonal coarse-grid correction can be bounded in norm with reasonable values, e.g., 1–5. Although the bounds appear tighter for $K = 10$, larger K means that k, the dimension of W_1 and Z_1, is likely smaller. In turn, the dimension of W_2 and Z_2 is larger and, thus, the constraint on δ must be satisfied for a larger set of vectors, which is likely more difficult.

It is worth pointing out that, consistent with traditional (SPD) AMG theory, the difficulty of a problem depends largely on the distribution of singular values. In the SPD case, many small eigenvalues means that more modes must be included in the range of interpolation with greater accuracy. Intuitively, a subspace correction is difficult when a significant proportion of eigenmodes must be represented on the coarse space. In fact, for SPD matrices, it is well-known that the optimal two-grid convergence rate in the A-norm for a coarse grid of size n_c is given by $\|E_{TG}\|_A^2 = 1 - \lambda_{n_c+1}$ [24, 57]. Here, λ_i is the ith eigenvector of the generalized eigenvalue problem $A\mathbf{v} = \lambda_i M\mathbf{v}$, for symmetric relaxation scheme M. Thus, if the first n_c+1 eigenvalues are all approximately zero, AMG *cannot* achieve strong convergence factors. One practical example is elasticity with a Poisson ratio approaching 0.5, which can prove difficult for AMG because there are many low-energy modes that are difficult to all be represented on a coarse grid. In the nonsymmetric setting, we have the additional δ constraint that must also be satisfied for stability. Above, it is somewhat implicitly assumed that the range of W_2 and Z_2 correspond to singular values that are not "too" small. However, accounting for very small singular values, $\sigma_i \approx 0$, in the δ constraint poses additional difficulties, due to scaling by Σ in $Z_2^*LW_2$. Of course one can always decrease K by making the SAP more accurate, thereby allowing for smaller δ. Nevertheless, many near-zero singular values typically makes for a difficult problem for NS-AMG.

Finally, the bound on $B_P^*B_P \sim_s I$ (and similarly for B_R) in Lemma 10 is a subtle but important result. For stability, we must make some assumption on $Z_2 * QLW_2$ being nicely bounded below, independent of mesh. However, we want to be able to say something about how P and R relate on the high modes. The W_2 and Z_2 depend on B_P and B_R. If the change-of-basis matrices are not also nicely bounded, then an assumption on W_2 and Z_2 does not translate to a reasonable assumption on the action of P and R on high modes.

7.2 Multigrid convergence

Recall from (7.7) that coarse-grid correction is invariant under a change of basis $P = \hat{P}B_P^{-1}$ and $R = \hat{R}B_R^{-1}$, for change of basis matrices B_P and B_R as designed in Lemma 10. In this section we will work entirely in the basis $\hat{P}, \hat{R}$. For multilevel convergence, the key step is to prove the equivalence between inner products in the orthogonal coarse-grid, $\hat{P}^*QL\hat{P}$, and the norm measured in practice, $(L_c^*L_c)^{\frac{1}{2}}$, which is done in Lemma 12 and the following corollaries.

Lemma 12 (Norm-equivalence of coarse-grid operators)**.** *Under the same assumptions as Theorem 3, $L_c \sim_n \hat{P}^* QL\hat{P}$, with constants specified below.*

Proof. From Lemma 10 we have $\hat{P}^* QL\hat{P} \sim_n \begin{pmatrix} \Sigma_{1..k} & 0 \\ 0 & I \end{pmatrix}$ so, by transitivity of norm equivalence [53], we can reduce the problem to proving $\begin{pmatrix} \Sigma_{1..k} & 0 \\ 0 & I \end{pmatrix} \sim_n L_c$. We again invoke Lemma 11 and bound each block of L_c individually, where

$$L_c = \begin{pmatrix} Z_1^* L W_1 & Z_1^* L W_2 \\ Z_2^* L W_1 & Z_2^* L W_2 \end{pmatrix} = \begin{pmatrix} \Sigma_{1..k} - U_1^* L \mathcal{N}_1 - cM_1^* L V_1 + \mathcal{M}_1^* L \mathcal{N}_1 & -\mathcal{M}_1^* L W_2 \\ -Z_2^* L \mathcal{N}_1 & Z_2^* L W_2 \end{pmatrix}.$$

Recall from Lemma 8 that the SAP with constant K implies the NWAP with constant K^2. Then,

$$\begin{aligned}
\|\mathcal{M}_1^* L \mathcal{N}_1 \mathbf{x}\| &= \sup_{\mathbf{y}\neq\mathbf{0}} \frac{\langle (QL)^{\frac{1}{2}} \mathcal{N}_1 \mathbf{x}, (LQ)^{\frac{1}{2}} \mathcal{M}_1 \mathbf{y} \rangle}{\|\mathbf{y}\|} = \\
&\qquad \|(I - \Pi_1) V_1 \mathbf{x}\|_{QL} \frac{\|(I - \Pi_2) U_1 \mathbf{y}\|_{LQ}}{\|\mathbf{y}\|} \leq K_1 \sigma_k \|\Sigma_1 \mathbf{x}\|, \\
\|\mathcal{M}_1^* L V_1 \mathbf{x}\| &= \sup_{\mathbf{y}\neq\mathbf{0}} \frac{\langle \Sigma_1 \mathbf{x}, \mathcal{M}_1 \mathbf{y} \rangle}{\|\mathbf{y}\|} \leq \|\Sigma_1 \mathbf{x}\| \frac{\|(I - \Pi_2) U_1 \mathbf{y}\|}{\|\mathbf{y}\|} \leq K_1 \sigma_k \|\Sigma_1 \mathbf{x}\|, \\
\|U_1^* L \mathcal{N}_1 \mathbf{x}\| &= \sup_{\mathbf{y}\neq\mathbf{0}} \frac{\langle \mathcal{N}_1 \mathbf{x}, V \Sigma_1 \mathbf{y} \rangle}{\|\mathbf{y}\|} \leq \|(I - \Pi_1) V_1 \mathbf{x}\| \frac{\|V \Sigma_1 \mathbf{y}\|}{\|\mathbf{y}\|} \leq K_1 \sigma_k \|\Sigma_1 \mathbf{x}\|, \\
\|\mathcal{M}_1^* L W_2 \mathbf{x}\| &= \sup_{\mathbf{y}\neq\mathbf{0}} \frac{\langle (QL)^{\frac{1}{2}} W_2 \mathbf{x}, (LQ)^{\frac{1}{2}} \mathcal{M}_1 \mathbf{y} \rangle}{\|\mathbf{y}\|} \leq \\
&\qquad \|W_2 \mathbf{x}\|_{QL} \frac{\|(I - \Pi_2) U_1 \mathbf{y}\|_{LQ}}{\|\mathbf{y}\|} \leq \sqrt{K_1} \|\Sigma_1 \mathbf{x}\|, \\
\|Z_2^* L \mathcal{N}_1 \mathbf{x}\| &= \sup_{\mathbf{y}\neq\mathbf{0}} \frac{\langle (QL)^{\frac{1}{2}} \mathcal{N}_1 \mathbf{x}, (LQ)^{\frac{1}{2}} Z_2 \mathbf{y} \rangle}{\|\mathbf{y}\|} \leq \\
&\qquad \|(I - \Pi_1) V_1 \mathbf{x}\|_{QL} \frac{\|Z_2 \mathbf{y}\|_{LQ}}{\|\mathbf{y}\|} \leq \sigma_k \sqrt{K_1} \|\mathbf{x}\|.
\end{aligned}$$

From above we have $(1 - 3K_1\sigma_k)\|\Sigma_1 \mathbf{x}\| \leq \|Z_1^* L W_1 \mathbf{x}\| \leq (1 + 3K_1\sigma_k)\|\Sigma_1 \mathbf{x}\|$ and by assumption and construction $\delta \leq \|Z_2^* L W_2\| \leq 1$. Invoking Lemma 11, $a_0 = 1 - 3K_1\sigma_k$, $a_1 = 1 + 3K_1\sigma_k$, $b = \sqrt{K_1}$, $c = \sigma_k\sqrt{K_1}$, $d_0 = \delta$, and $d_1 = 1$. This leads to the same constraint on δ as used in Theorem 3, $\delta > \frac{K_1\sigma_k}{1 - 3K_1\sigma_k}$. Then

$$0 < C_{\text{lower}} \leq \frac{\|L_c \mathbf{x}\|^2}{\left\| \begin{pmatrix} \Sigma_{1..k} & 0 \\ 0 & I \end{pmatrix} \mathbf{x} \right\|^2} \geq C_{\text{upper}},$$

where

$$\begin{aligned}
2C_{\text{lower}} &= (1-3K_1\sigma_k)^2 + K_1 + \sigma_k^2 K_1 + \delta^2 - ... \\
&\qquad \sqrt{((1-3K_1\sigma_k)^2 + K_1 - \sigma_k^2 K_1 - \delta^2)^2 + 4K_1((1-3K_1\sigma_k)\sigma_k + \delta)^2}, \\
2C_{\text{upper}} &= (1+3K_1\sigma_k)^2 + K_1 + \sigma_k^2 K_1 + 1 - ... \\
&\qquad \sqrt{((1+3K_1\sigma_k)^2 + K_1 - \sigma_k^2 K_1 - 1)^2 + 4K_1((1+3K_1\sigma_k)\sigma_k + 1)^2}.
\end{aligned}$$

Including the equivalence constants of $\hat{P}^*QL\hat{P} \sim_n \begin{pmatrix} \Sigma_{1..k} & 0 \\ 0 & I \end{pmatrix}$ from Lemma 10,

$$0 < C_{\text{lower}}(1-K_1\sigma_k)^2 \leq \frac{\|L_c\mathbf{x}\|^2}{\|P^*QL\hat{P}\mathbf{x}\|^2} \leq C_{\text{upper}}(1+K_1\sigma_k)^2.$$

□

Corollary 5 (Spectral-equivalence of coarse-grid operators)**.** *Under the same assumptions as Theorem 3, $(L_c^*L_c)^{\frac{1}{2}} \sim_s \hat{P}^*QL\hat{P}$ with the same constants as in Lemma 12.*

Proof. Given that $(L_c^*L_c)^{\frac{1}{2}}$ and $\hat{P}^*QL\hat{P}$ are both compact self-adjoint operators, norm equivalence implies spectral equivalence (with the same constants) [53]. Noting that $\|(L_c^*L_c)^{\frac{1}{2}}\mathbf{y}\|^2 = \langle (L_c^*L_c)^{\frac{1}{2}}\mathbf{y}, (L_c^*L_c)^{\frac{1}{2}}\mathbf{y}\rangle = \langle L_c\mathbf{y}, L_c\mathbf{y}\rangle = \|L_c\mathbf{y}\|^2$, the result follows from Lemma 12. □

Corollary 6 (Spectral-equivalence of inner products)**.** *Let $Q_cL_c := (L_c^*L_c)^{\frac{1}{2}}$. Then for any coarse-grid vector $\mathbf{y}_c$,*

$$C_{\text{lower}}(1-K_1\sigma_k)^2 \leq \frac{\|\mathbf{y}_c\|^2_{Q_cL_c}}{\|\hat{P}\mathbf{y}_c\|^2_{QL}} \leq C_{\text{upper}}(1+K_1\sigma_k)^2.$$

Proof. The proof follows from the spectral equivalence shown in Corollary 5 and noting that $\|\hat{P}\mathbf{y}_c\|^2_{QL} = \langle \hat{P}^*QL\hat{P}\mathbf{y}_c, \mathbf{y}_c\rangle$. □

So far we have considered the the relation between the orthogonal coarse-grid operator and the coarse grid attained in practice. The final piece of the puzzle in showing multilevel convergence is considering how an error vector $\mathbf{e}$ can be decomposed in norm over the subspaces $\mathcal{R}(\Pi)$ and $\mathcal{R}(I-\Pi)$. For an orthogonal projection, say $\widehat{\Pi}$, in some norm $\|\cdot\|$, $\|\widehat{\Pi}\mathbf{e}\|^2 + \|\widehat{\Pi}\mathbf{e}\|^2 = \|\mathbf{e}\|^2$. Because Π as used here is an oblique projection, this equality does not hold. However, bounds on the decomposition are closely related to stability as proved in Section 7.1.2, each related to the angle between the subspaces $\mathcal{R}(\Pi)$ and $\mathcal{R}(I-\Pi)$. We start be reviewing the following results connecting the angle between subspaces of a Hilbert space, the norm of an oblique projection, and a strengthened Cauchy-Schwarz inequality [45, 172].

Lemma 13 (Strengthened Cauchy Schwarz)**.** *Define the minimal canonical angle between $\mathcal{R}(\Pi)$ and $\mathcal{R}(I-\Pi)$ in the QL inner product by*

$$\cos\left(\theta_{min}^{(\Pi)}\right) := \inf_{\substack{x\in\mathcal{R}(P),\|x\|_{QL}=1,\\ y\in\mathcal{R}(I-P),\|y\|_{QL}=1}} |\langle x, r\rangle_{QL}|.$$

Then for all $\mathbf{x}\in\mathcal{R}(\Pi)$ and $\mathbf{y}\in\mathcal{R}(I-\Pi)$,

$$\|\Pi\|_{QL} = \|I-\Pi\|_{QL} = \frac{1}{\sin\left(\theta_{min}^{(\Pi)}\right)},$$
$$\left|\langle\mathbf{x},\mathbf{y}\rangle_{QL}\right| \leq \cos\left(\theta_{min}^{(\Pi)}\right)\|\mathbf{x}\|_{QL}\|\mathbf{y}\|_{QL}.$$

Corollary 7. *Suppose $1\leq\|\Pi\|_{QL}^2\leq C_\Pi$. Then a decomposition of any vector $\mathbf{e}$ over $\mathcal{R}(P)$ and $\mathcal{R}(I-P)$ can be bounded as*[1]

$$\begin{aligned}\left(C_\Pi-\sqrt{C_\Pi^2-C_\Pi}\right)\|\mathbf{e}\|_{QL}^2 &\leq \left(\|\Pi\mathbf{e}\|_{QL}^2+\|(I-\Pi)\mathbf{e}\|_{QL}^2\right)\\ &\leq \left(C_\Pi+\sqrt{C_\Pi^2-C_\Pi}\right)\|\mathbf{e}\|_{QL}^2.\end{aligned}$$

Proof. From Lemma 13, for all $\mathbf{x},\mathbf{y}\in\mathbb{R}^n$

$$\begin{aligned}\left|\langle\Pi\mathbf{x},(I-\Pi)\mathbf{y}\rangle_{QL}\right| &\leq \cos\left(\arcsin\left(\frac{1}{\|\Pi\|_{QL}}\right)\right)\|\Pi\mathbf{x}\|_{QL}\|(I-\Pi)\mathbf{y}\|_{QL}\\ &= \sqrt{1-\frac{1}{\|\Pi\|_{QL}^2}}\|\Pi\mathbf{x}\|_{QL}\|(I-\Pi)\mathbf{y}\|_{QL}\\ &\leq \sqrt{1-\frac{1}{C_\Pi}}\|\Pi\mathbf{x}\|_{QL}\|(I-\Pi)\mathbf{y}\|_{QL}.\end{aligned}$$

Then,

$$\begin{aligned}\|\mathbf{e}\|_{QL}^2 &= \|\Pi\mathbf{e}\|_{QL}^2+\langle\Pi\mathbf{e},(I-\Pi)\mathbf{e}\rangle_{QL}+\|(I-\Pi)\mathbf{e}\|_{QL}^2\\ &\geq \|\Pi\mathbf{e}\|_{QL}^2-2\sqrt{1-\frac{1}{C_\Pi}}\|\Pi\mathbf{e}\|_{QL}\|(I-\Pi)\mathbf{e}\|_{QL}+\|(I-\Pi)\mathbf{e}\|_{QL}^2.\end{aligned}$$

[1] Note that we can also bound $(\|\Pi\mathbf{e}\|_{QL}^2+\|(I-\Pi)\mathbf{e}\|_{QL}^2)\leq 2C_\Pi\|\mathbf{e}\|_{QL}^2$. Although Corollary 7 only decreases this bound by $\frac{1}{2}$ to 1, depending on C_Π, it does make the bound tight as $C_\Pi\to 1$, corresponding to an orthogonal projection.

Using an ϵ-inequality on the middle term with $\epsilon = 1$ and noting that $\frac{1}{\sqrt{1-\frac{1}{C_\Pi}}} = C_\Pi - \sqrt{C_\Pi^2 - C_\Pi}$ for $C_\Pi > 1$,

$$\left(C_\Pi - \sqrt{C_\Pi^2 - C_\Pi}\right) \|\mathbf{e}\|_{QL}^2 \leq \left(\|\Pi\mathbf{e}\|_{QL}^2 + \|(I-\Pi)\mathbf{e}\|_{QL}^2\right).$$

For $C_\Pi = 1$, $\left(C_\Pi - \sqrt{C_\Pi^2 - C_\Pi}\right) \|\mathbf{e}\|_{QL}^2 = \|\mathbf{e}\|_{QL}^2 = \left(\|\Pi\mathbf{e}\|_{QL}^2 + \|(I-\Pi)\mathbf{e}\|_{QL}^2\right)$. A similar proof leads to the upper bound. □

We now have all of the tools to introduce multilevel convergence of NS-AMG. Note that due to the non-orthogonal coarse-grid correction, accuracy of the recursive coarse-grid solve must be progressively better on coarser levels in the hierarchy. This necessitates some form improved multilevel cycle with additional relaxation iterations or multigrid cycles on coarser levels in the hierarchy. Although the desired uniform V-cycle convergence does not hold in this setting, Theorem 4 remains the first proof of multilevel AMG convergence for nonsymmetric matrices.[2]

Theorem 4 (Multilevel convergence)**.** *Suppose we have an AMG hierarchy with ℓ levels, and assume the conditions for Theorem 3 hold on each level. Further, on level $\ell - 1$ (the coarsest level in the hierarchy that is not a direct solve), assume that*

$$\|I - B_{\ell-1}^{-1}L_{\ell-1}\|_{(L_{\ell-1}^*L_{\ell-1})^{\frac{1}{2}}} \leq \delta_{\ell-1} < 1,$$

that is we have a convergent two-grid method on level $\ell - 1$. Then there exists a convergent, multilevel cycle in the QL norm, with the number of relaxation and multigrid iterations per level that depend on the constants in Corollaries 6 and 7.

Proof. First consider the difference between the exact projection, Π, and the inexact projection, $\widetilde{\Pi}$, corresponding to the recursive application of a multilevel AMG cycle to the coarse-grid problem. From Corollary

[2] In [109], the multilevel error propagation operator was shown to be nilpotent, that is, asymptotically guaranteed to converge, for triangular or block-triangular matrices. However, [109] only applied to a small class of nonsymmetric problems and only proved asymptotic convergence

6,

$$\begin{aligned}
\|(\Pi_0 - \widetilde{\Pi}_0)G_0\mathbf{e}^{(i)}\|_{QL}^2 &= \left\|P_0(L_1^{-1} - B_1^{-1})L_1(L_1^{-1}R_0^*L_0G_0)\mathbf{e}^{(i)}\right\|_{QL}^2 \\
&\leq \frac{1}{C_{\text{lower}}(1-K_1\sigma_k^2)}\left\|(L_1^{-1} - B_1^{-1})L_1(L_1^{-1}R_0^*L_0G_0)\mathbf{e}^{(i)}\right\|_{Q_cL_c}^2 \\
&\leq \frac{\delta_1}{C_{\text{lower}}(1-K_1\sigma_k^2)}\left\|L_1^{-1}R_0^*L_0G_0\mathbf{e}^{(i)}\right\|_{Q_cL_c}^2 \\
&\leq \frac{\delta_1 C_{\text{upper}}(1+K_1\sigma_k^2)}{C_{\text{lower}}(1-K_1\sigma_k^2)}\left\|P_0A_1^{-1}R_0^*A_0G_0\mathbf{e}^{(i)}\right\|_{QL}^2 \\
&= \frac{\delta_1 C_{\text{upper}}(1+K_1\sigma_k^2)}{C_{\text{lower}}(1-K_1\sigma_k^2)}\|\Pi_0G_0\mathbf{e}^{(i)}\|_{QL}^2.
\end{aligned}$$

Denote the leading constant $\lambda_1 := \frac{\delta_i C_{\text{upper}}(1+K_1\sigma_k^2)}{C_{\text{lower}}(1-K_1\sigma_k^2)}$. Then error can be expanded as

$$\begin{aligned}
\|\mathbf{e}^{(i+1)}\|_{QL}^2 &= \|(I-\widetilde{\Pi}_0)G_0\mathbf{e}^{(i)}\|_{QL} \\
&\leq \|(I-\Pi_0)G_0\mathbf{e}^{(i)}\|_{QL}^2 + 2\left\langle (I-\Pi_0)G_0\mathbf{e}^{(i)}, (\Pi_0-\widetilde{\Pi}_0)G_0\mathbf{e}^{(i)}\right\rangle_{QL} + ... \\
&\qquad \|(\Pi_0-\widetilde{\Pi}_0)G_0\mathbf{e}^{(i)}\|_{QL}^2.
\end{aligned}$$

Because $\mathcal{R}(\Pi) = \mathcal{R}(P) = \mathcal{R}(\widetilde{\Pi})$, we can use a strengthened Cauchy-Schwarz on the middle term (Lemma 13):

$$\begin{aligned}
\|\mathbf{e}^{(i+1)}\|_{QL}^2 &\leq \|(I-\Pi_0)G_0\mathbf{e}^{(i)}\|_{QL}^2 + \lambda_1\|\Pi_0G_0\mathbf{e}^{(i)}\|_{QL}^2 + ... \\
&\qquad 2\cos\left(\theta_{min}^{(\Pi_0)}\right)\|(I-\Pi_0)G_0\mathbf{e}^{(i)}\|_{QL}\|(\Pi_0-\widetilde{\Pi}_0)G_0\mathbf{e}^{(i)}\|_{QL} \\
&\leq \|(I-\Pi_0)G_0\mathbf{e}^{(i)}\|_{QL}^2 + \lambda_1\|\Pi_0G_0\mathbf{e}^{(i)}\|_{QL}^2 + ... \\
&\qquad 2\cos\left(\theta_{min}^{(\Pi_0)}\right)\sqrt{\lambda_1}\|(I-\Pi_0)G_0\mathbf{e}^{(i)}\|_{QL}\|\Pi_0G_0\mathbf{e}^{(i)}\|_{QL} \\
&= \left(1 + \frac{4\lambda_1\cos^2\left(\theta_{min}^{(\Pi_0)}\right)}{\epsilon}\right)\|(I-\Pi_0)G_0\mathbf{e}^{(i)}\|_{QL}^2 + (\lambda_1+\epsilon)\|\Pi_0G_0\mathbf{e}^{(i)}\|_{QL}^2,
\end{aligned}$$

for any $\epsilon > 0$. We choose ϵ such that the leading constants are equal, which corresponds to solving for the single positive root of $\epsilon^2 + \epsilon(\lambda_1 - 1) - 4\lambda_1\cos^2\left(\theta_{min}^{(\Pi_0)}\right) = 0$, given by

$$\epsilon_{\lambda_1} = \frac{1 - \lambda_1 + \sqrt{(\lambda_1-1)^2 + 16\lambda_1\cos^2\left(\theta_{min}^{(\Pi_0)}\right)}}{2}.$$

Finally, we use the result from Corollary 7 to bound error propagation as

$$\|\mathbf{e}^{(i+1)}\|_{QL}^2 \leq (\lambda_1 + \epsilon_{\lambda_1})\left(C_\Pi + \sqrt{C_\Pi^2 - C_\Pi}\right)\|G\mathbf{e}^{(i)}\|_{QL}^2. \tag{7.16}$$

□

There are two important things to note on the bound in (7.16). First, for non-orthogonal coarse-grid correction, the bound is likely greater than one, and sufficient iterations of relaxation must be performed to overcome this constant and achieve a convergent method. Conceptually, this makes sense – non-orthogonal coarse-grid correction will likely increase error of components not in the range of P, and additional relaxation must then be performed to attenuate this error. Such a result is consistent with the two-grid theory developed in [27]. Second, the convergence factor on level zero is a function of (and greater than) the convergence factor on level one. This implies that a normal V-cycle will typically not be sufficient for convergence, or at least not scalable convergence as the problem size (and, thus, number of levels in the hierarchy) increases. This can be remedied by performing additional iterations of relaxation on coarser levels or by using a multigrid cycling strategy with multiple coarse-grid iterations such as F-, K-, or W-cycles. Unfortunately, the constants in (7.16) are not easily measured in practice, so experience and experimentation may be necessary to choose the appropriate cycle for a given problem.

Theorem 4 also sheds some light on why using $R := P$ works for some nonsymmetric problems. Typically, nonsymmetry in PDE discretizations arises from advection, which becomes progressively smaller compared with diffusion as $h \rightarrow 0$. If we have only a slightly nonsymmetric matrix with a large symmetric component, the constants in (7.16) are likely small and reasonable convergence factors attainable.

Remark 9 (The symmetric case)**.** *Note that in the case of SPD matrices (or any orthogonal coarse-grid correction), $C_\Pi = 1$, $\lambda_1 = \delta_1 < 1$, and the projections are orthogonal, so $\cos\left(\theta_{min}^{(\Pi_0)}\right) = 0$. Then, from above,*

$$\|\mathbf{e}^{(i+1)}\|_{QL}^2 \leq \|(I - \Pi_0)G_0\mathbf{e}^{(i)}\|_{QL}^2 + \delta_1\|\Pi_0 G_0\mathbf{e}^{(i)}\|_{QL}^2.$$

With minor modifications, a classical V-cycle proof follows from the assumption that relaxation is effective on modes that are not in the range of interpolation (and vice versa). That is, the proof is relatively tight in the multilevel case because it breaks down to an SPD V-cycle proof when we have an orthogonal coarse-grid correction. It is possible that some of the constants derived with respect to equivalence, stability, etc. are not tight, but they do provide reasonable bounds.

7.3 Discussion

Here we have established conditions on R and P for two-grid and multigrid convergence of NS-AMG in the $\sqrt{A^*A}$-norm. As common with convergence theory, the constants and bounds are not easily measured in practice. Nevertheless, they do provide a fundamental understanding of what a NS-AMG hierarchy needs to converge. The criteria for convergence developed here can largely be broken down into two simple guidelines:

(1) It is not enough for R and P to include low-energy left and right singular vectors in their range (classical approximation-property-based AMG approach). The action of R and P must lead to a non-singular (and reasonably conditioned) coarse-grid operator, which can be achieved by ensuring that their action on left and right singular vector, respectively, associated with larger singular values are similar.

(2) Multilevel convergence of NS-AMG may require additional iterations of relaxation or recursive multigrid cycles on coarser levels of the hierarchy to converge. It is almost invariably the case that an AMG hierarchy based on a nonsymmetric matrix (regardless of the chosen AMG framework) will consist of non-orthogonal coarse-grid corrections. In practice, such potential increases in error may need to be accounted for through additional work. That is, a divergent V-cycle for NS-AMG does *not* mean that an appropriately modified V-cycle or modified AMG cycle will also not converge.

The latter point on cycle-type does not provide explicitly useful information such as what kind of cycle is needed for convergence. However, it does inform a user that uniform V-cycle convergence of NS-AMG is not expected in theory and, moreover, that if a hierarchy does not converge, it may be remedied by some additional relaxation or multigrid cycles on coarser levels in the hierarchy.

One interesting aspect of this work is that it was developed around the same time as several very successful NS-AMG solvers [109, 110]. However, a brief look at some of the new NS-AMG solvers make it fairly clear that they do *not* satisfy the conditions for two-grid or multigrid convergence presented here. In the extreme case of triangular and block-triangular matrices considered in Chapter 8 [109], strong convergence rates are obtained using one-point interpolation (interpolate each F-point from its strongest C-neighbor) or injection (interpolate C-points by value; do not interpolate to F-points). This is generalized to other nonsymmetric matrices in Chapter 9 [110], but interpolation appears to still not be of fundamental importance. In both of these cases, it is almost certainly the case that the transfer operators do not satisfy the SAP or conditions presented here for convergence with respect to any norm. For [109, 110], the focus is on constructing a good approximation to the ideal restriction operator, and results suggest that, in terms of convergence, a good restriction operator allows for a lousy interpolation operator. Furthermore, results in [109, 110] also did not require multigrid cycles with additional smoothing or recursive cycles. The following chapters take a more practical approach, developing this reduction-based AMG method for highly nonsymmetric problems.

The disconnect between successful results in [109, 110] and the relatively complete theory developed here suggest that there is still more to study on both the theoretical and practical side. In [110], a spectral analysis was used to motivated the AMG transfer operators. However, proving spectral- or norm-based convergence of [110] is ongoing work. Conversely, can the theory developed here lead to a new, robust AMG algorithm for nonsymmetric problems? The optimal restriction and interpolation operators with respect to the $\sqrt{A^*A}$-norm are the first n_c left and right singular vectors, respectively. Recently, [24] looked at sparse approximations to optimal interpolation for SPD matrices with some success; perhaps such an approach can be extended to the nonsymmetric setting.

Chapter 8

Ideal restriction part I: reduction and upwind discretizations

8.1 Motivation

Iterative methods and AMG for elliptic and parabolic PDEs are well-studied and, in many cases, effective choices in practice. However, hyperbolic PDEs make up a large class of problems of interest and are typically difficult for solvers. In contrast to elliptic and parabolic PDEs, the solution of hyperbolic PDEs lies on characteristic curves, and the solution at any point depends only on the solution upwind along its given characteristic. This allows for very steep gradients or "fronts" to propagate through the domain in the direction of characteristics. Typical continuous finite-element or finite-difference discretizations often struggle to capture such behavior because fronts are effectively discontinuities in the discrete setting, and not necessarily grid-aligned. Due to the discontinuous-like behavior and the flow of information in a single direction along characteristics, discontinuous upwind discretizations are a natural approach to discretizing many hyperbolic PDEs [29, 116, 117, 141]. For a fully upwinded discretization, the resulting matrix has a block-triangular structure in some (although potentially unknown) ordering. Although solving a block-triangular matrix is an easy task in the serial setting, direct solves are limited by scalability in the parallel setting and fast iterative methods have not been developed for such problems.

Solving triangular systems in a traditional context using a forward or backward solve is a strictly sequential operation and, thus, does not scale well in parallel. In the dense matrix setting, this offers limited opportunity for parallelism. However, for sparse matrices, scheduling algorithms have been developed that can add some level parallelism to this process [3, 5, 99, 101]. In a sparse triangular matrix, there exists at least one DOF, say i_0, that is independent of other DOFs and can be eliminated from the system. Then, by nature of a sparse matrix, we expect multiple DOFs to depend only on i_0, and can then be computed in parallel. In a discretization sense, these would be DOFs immediately downwind from i_0. Once new DOFs have been computed, DOFs downwind of them can be computed, and so on until the solution is obtained. This algorithm requires only $2 \cdot nnz$ floating point operations for a sparse matrix with nnz nonzero entries, but the parallelism depends on the structure of the matrix. Much of the research on increased parallel performance for such problems is focused on shared memory environments such as GPUs [99, 101].

One problem and discretization that requires many parallel sparse triangular solves is the so-called "diffusion synthetic acceleration" (DSA) approach to solving the linear Boltzmann transport equation (so-called "synthetic acceleration" is simply preconditioning) [1, 37, 69, 141]. The full linear problem is seven dimensional, necessitating highly parallel solvers for even moderately refined grids, and is fundamental to studying neutron and radiation transport. DSA is effectively a two-level multigrid algorithm. A "transport sweep" acts as a surrogate for relaxation, and consists of discretizing in angle and solving a steady state transport equation over all angles in the spatial domain. An accurate transport sweep puts error in the range of a surrogate "coarse-grid" diffusion solve, and the process iterates back and forth [1]. AMG is a fast and scalable solver for the diffusion solve, but the transport sweep consists of solving many discretizations of a hyperbolic PDE, often based on an upwind discretization [29, 116, 117, 141], and is difficult for most iterative solvers [154].

Similar scheduling algorithms to those used for sparse triangular solves on GPUs have been developed in distributed memory environments to perform "transport sweeps" [2, 7, 37, 69]. On a structured mesh, these algorithms were shown to have an optimal theoretical scaling over all angles of $O(dP^{\frac{1}{d}})$, for P processors and a problem of dimension d [7]. Although this is reasonable scaling in parallel, it is suboptimal on the so-called "road to exascale," as a steadily increasing number of processors become available. On irregular meshes, scheduling for the transport sweep becomes more difficult and this theoretical bound cannot be achieved. The sweeping algorithm has been shown to be a bottleneck in large simulation codes [133]. On the other hand, AMG scales like $O(\log_2(P))$ in parallel [9] and $O(n)$ with respect to DOFs, even on irregular meshes. An effective AMG algorithm for triangular systems could prove faster than known scheduling algorithms for transport sweeps in a highly parallel setting. For example, for $P = 10^6$ and $d = 3$, we have $dP^{\frac{1}{d}} = 300$, while $\log_2(P) \approx 20$, yielding a factor of 15 improvement in parallel efficiency. Of course the leading constants may change this balance in either direction, especially on irregular meshes, but this suggests there is a reasonable P for which AMG will outperform a traditional transport sweep.

This chapter develops a novel, reduction-based AMG method (AMGir) to solve systems with triangular structure, with a particular focus on upwind discretizations of hyperbolic PDEs. Although geometric multigrid has been applied to upwind discretizations and other hyperbolic-type problems [4, 188] using the well-known line-smoother approach [131], an effective AMG algorithm does not need geometric information, a structured grid, or a fixed/known direction over which to relax. For these reasons, an effective AMG solver is more robust and less intrusive because it can easily be used by software packages, regardless of their meshing, discretization, etc.. AMGir significantly out-performs current state-of-the-art iterative solvers in the context of steady state transport [154] and a space-time discretization of the wave equation. Even for high-order, discontinuous discretizations on unstructured grids, AMGir is able to achieve an order-of-magnitude residual reduction in 1–2 iterations.

Theoretical motivation for the new method and its relation to existing algorithms is discussed in Sections 8.1.1 and 8.1.2. The underlying premise is that certain "ideal" operators can lead to a reduction-based AMG method; in particular, we develop an AMG algorithm based on approximating ideal restriction. Section 8.2 introduces a natural way to approximate ideal operators for matrices with triangular structure, as well as theoretical results on the corresponding multigrid error propagation. The algorithm and possible variations are presented in Section 8.3, and numerical results and scaling studies in Section 8.4.

8.1.1 Ideal restriction

In a classical AMG sense, suppose that a CF-splitting has been performed on the current grid, that is, all degrees-of-freedom (DOFs) have been designated as either a coarse-point (C-point) or a fine-point (F-point) (see Section 5.1 and (5.4)).

Lemma 14 (Schur-complement coarse grid)**.** *Let A take the block form as in* (5.4) *and A_{ff} be nonsingular.*

Define

$$R_{ideal} := \begin{pmatrix} -A_{cf}A_{ff}^{-1} & I \end{pmatrix}, \qquad P_{ideal} := \begin{pmatrix} -A_{ff}^{-1}A_{fc} \\ I \end{pmatrix}.$$

Then, for all R and P of the form in (5.4)*,*

$$RAP_{ideal} = R_{ideal}AP = R_{ideal}AP_{ideal} = S_A,$$

where $S_A := A_{cc} - A_{cf}A_{ff}^{-1}A_{fc}$ is the Schur-complement of A in A_{cc}. That is, the coarse-grid operator, $\mathcal{K}$, is given by the Schur complement of A.

Proof. Suppose $R = R_{ideal}$. Then, $Z = -A_{cf}A_{ff}^{-1}$ and

$$\begin{aligned} R_{ideal}AP &= ZA_{ff}W + A_{cf}W + ZA_{fc} + A_{cc} \\ &= -A_{cf}W + A_{cf}W - A_{cf}A_{ff}^{-1}A_{fc} + A_{cc} \\ &= S_A. \end{aligned}$$

Identical steps show the equivalent result for P_{ideal}. □

The operator R_{ideal} defined in Lemma 14 is referred to here as "ideal restriction," a natural extension of the well-known "ideal interpolation" operator, P_{ideal}, also defined in Lemma 14. Note that if A is symmetric, $R_{ideal} = P_{ideal}^T$. Ideal interpolation is well-motivated under classical AMG theory (for example, see [55]). Here, we show that ideal restriction and ideal interpolation each have significance in the nonsymmetric setting as well. In Theorem 5, ideal restriction is shown to be ideal in a certain sense, albeit not the same sense for which ideal interpolation acquired its name. Thus, consider error-propagation for a coarse-grid correction applied to the current error vector, $(\mathbf{e}_f, \mathbf{e}_c)^T$:

$$\begin{aligned} (I-\pi)\mathbf{e} &= \left[\begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} - \begin{pmatrix} W \\ I \end{pmatrix} \mathcal{K}^{-1} \begin{pmatrix} Z & I \end{pmatrix} \begin{pmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{pmatrix} \right] \begin{pmatrix} \mathbf{e}_f \\ \mathbf{e}_c \end{pmatrix} \\ &= \begin{pmatrix} I - W\mathcal{K}^{-1}(ZA_{ff} + A_{cf}) & -W\mathcal{K}^{-1}(ZA_{fc} + A_{cc}) \\ -\mathcal{K}^{-1}(ZA_{ff} + A_{cf}) & I - \mathcal{K}^{-1}(ZA_{fc} + A_{cc}) \end{pmatrix} \begin{pmatrix} \mathbf{e}_f \\ \mathbf{e}_c \end{pmatrix} \\ &= \begin{pmatrix} \left[I - W\mathcal{K}^{-1}(ZA_{ff} + A_{cf}) \right] \mathbf{e}_f - W\mathcal{K}^{-1}(ZA_{fc} + A_{cc})\mathbf{e}_c \\ \left[I - \mathcal{K}^{-1}(ZA_{fc} + A_{cc}) \right] \mathbf{e}_c - \mathcal{K}^{-1}(ZA_{ff} + A_{cf})\mathbf{e}_f \end{pmatrix}. \end{aligned} \tag{8.1}$$

Now, let $\mathbf{e}_f = W\mathbf{e}_c + \delta_f$, that is, let $\mathbf{e}_f$ be interpolated from $\mathbf{e}_c$ with some error term δ_f, and recall $\mathcal{K} = ZA_{ff}W + A_{cf}W + ZA_{fc} + A_{cc}$. Then, plugging $W\mathbf{e}_c + \delta_f$ into the fine-grid block, and $ZA_{fc} + A_{cc} =$

$\mathcal{K} - ZA_{ff}W - A_{cf}W$ into the coarse-grid block gives:

$$\begin{aligned}(I-\pi)\mathbf{e} &= \begin{pmatrix} \Big[I - W\mathcal{K}^{-1}(ZA_{ff}+A_{cf})\Big](W\mathbf{e}_c + \delta_f) - W\mathcal{K}^{-1}(ZA_{fc}+A_{cc})\mathbf{e}_c \\ \mathcal{K}^{-1}(ZA_{ff}+A_{cf})(W\mathbf{e}_c - \mathbf{e}_f) \end{pmatrix} \\ &= \begin{pmatrix} \Big[I - W\mathcal{K}^{-1}(ZA_{ff}+A_{cf})\Big]\delta_f + W\mathbf{e}_c - W\mathcal{K}^{-1}\mathcal{K}\mathbf{e}_c \\ -\mathcal{K}^{-1}(ZA_{ff}+A_{cf})\delta_f \end{pmatrix} \\ &= \begin{pmatrix} \Big[I - W\mathcal{K}^{-1}(ZA_{ff}+A_{cf})\Big]\delta_f \\ -\mathcal{K}^{-1}(ZA_{ff}+A_{cf})\delta_f \end{pmatrix}. \end{aligned} \tag{8.2}$$

The following theorem follows from (8.2).

Theorem 5 (Ideal restriction)**.** *For a given CF-splitting, assume that A_{ff} is full rank and let A take the block form as given in (5.4). Assume that C-points are interpolated and restricted by injection in a classical-AMG sense (5.4). Then, an exact coarse-grid correction at C-points is attained for all $\mathbf{e}$ if and only if*

$$R = R_{ideal} := \begin{pmatrix} -A_{cf}A_{ff}^{-1} & I \end{pmatrix}. \tag{8.3}$$

Furthermore, the error in coarse-grid correction is given by

$$(I-\pi)\mathbf{e} = \begin{pmatrix} \mathbf{e}_f - W\mathbf{e}_c \\ \mathbf{0} \end{pmatrix}, \tag{8.4}$$

where $P = \begin{pmatrix} W \\ I \end{pmatrix}$. Finally, a coarse-grid correction using R_{ideal} followed by an exact solve on F-points results in an exact two-grid method, independent of W.

Proof. From (8.2), coarse-grid correction as applied to some vector $\mathbf{e} = (\mathbf{e}_f, \mathbf{e}_c)^T$ and restricted to C-points is given by $\mathcal{K}^{-1}(ZA_{ff}+A_{cf})(W\mathbf{e}_c - \mathbf{e}_f)$. Noting that there does not exist a W such that $W\mathbf{e}_c = \mathbf{e}_f$ for all vectors $\mathbf{e} = (\mathbf{e}_f, \mathbf{e}_c)^T$, it follows that $\mathcal{K}^{-1}(ZA_{ff}+A_{cf})(W\mathbf{e}_c - \mathbf{e}_f) = 0$ if and only if $ZA_{ff} + A_{cf} = 0$. Given that A_{ff} is nonsingular, $ZA_{ff} + A_{cf} = 0$ if and only if $Z = -A_{cf}A_{ff}^{-1}$. The error shown in (8.4) follows directly from plugging R_{ideal} into (8.2). An exact solve on F-points would eliminate this error, providing an exact two-grid method. □

Corollary 8 (Ideal interpolation)**.** *An exact solve on F-points followed by a coarse-grid correction using so-called "ideal interpolation," $P := P_{ideal}$, as defined in Lemma 14 gives an exact two-level method, independent of Z.*

Proof. An initial exact solve on F-point rows of the equation $A\mathbf{x} = \mathbf{b}$ means that the residual at F-point rows is zero. For given error vector $\mathbf{e}$, this is equivalent to updating $\mathbf{e}$ such that $A_{ff}\mathbf{e}_f + A_{fc}\mathbf{e}_c = 0$, or $\mathbf{e}_f = -A_{ff}^{-1}A_{fc}\mathbf{e}_c$. Plugging this into the coarse-grid correction expansion in (8.1), along with $W = -A_{ff}^{-1}A_{fc}$ and $\mathcal{K} = S_A$ (see Lemma 14) gives

$$\begin{aligned}(I-\pi)\mathbf{e} &= \begin{pmatrix} -A_{ff}^{-1}A_{fc}\mathbf{e}_c \\ \mathbf{e}_c \end{pmatrix} - \begin{pmatrix} -A_{ff}^{-1}A_{fc} \\ I \end{pmatrix} \mathcal{S}_A{}^{-1} \begin{pmatrix} Z & I \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ S_A\mathbf{e}_c \end{pmatrix} \\ &= \mathbf{0}.\end{aligned}$$

□

It follows from Theorem 5 and Corollary 8 that R_{ideal} and P_{ideal} each lead to an exact two-level method, independent of the accompanying interpolation and restriction operators, respectively, when coupled with an exact solve on F-points. Such a two-level method is in fact a reduction algorithm, as solving $A\mathbf{x} = \mathbf{b}$ is reduced to solving two smaller systems. Note that the ordering of solving the coarse- and fine-grid problems is fundamental to achieving reduction. That is, to achieve reduction, the F-point solve must *follow* coarse-grid correction with R_{ideal}, while the F-point solve must *precede* coarse-grid correction with P_{ideal}. Background on reduction algorithms and ideal restriction in the multigrid context is given in Section 8.1.2.

A natural corollary of Theorem 5 is that if we define "trivial interpolation" as $P_0 = \begin{pmatrix} 0 \\ I \end{pmatrix}$, then error propagation of coarse-grid correction with R_{ideal} and P_0 is an ℓ^2-orthogonal projection. Similarly, if we define "trivial restriction" as $R_0 = \begin{pmatrix} 0 & I \end{pmatrix}$, then residual propagation of coarse-grid correction with R_0 and P_{ideal} is an ℓ^2-orthogonal projection. This is consistent with the proofs of Theorem 5 and Corollary 8, that is, P_{ideal} is based on eliminating residuals and R_{ideal} on eliminating error. Despite orthogonality, in practice, choosing $W = 0$ for interpolation of F-points does not always provide the best convergence factors. This is discussed in greater detail in Section 8.3.

8.1.2 Relation to existing AMG methods

The goal of a reduction-based algorithm is simply to take a problem and split its solving into smaller sub-problems. One way to enable reduction is through block row-elimination, where A can be written as

$$\begin{pmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{pmatrix} = \begin{pmatrix} I & 0 \\ A_{cf}A_{ff}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{ff} & 0 \\ 0 & S \end{pmatrix} \begin{pmatrix} I & A_{ff}^{-1}A_{fc} \\ 0 & I \end{pmatrix}. \tag{8.5}$$

Using (8.5), solving $A\mathbf{x} = \mathbf{b}$ is reduced to solving two systems of size $n_F \times n_F$ and $n_C \times n_C$, the latter of which is the Schur complement, $S := A_{cc} - A_{cf}A_{ff}^{-1}A_{fc}$. Unfortunately, (8.5) is generally impractical as shown because of the need to compute the inverse of A_{ff} and solve the typically dense Schur-complement system. However, such block decompositions have been used to motivate many algorithms, and a new approximation to inverting (8.5) can be found in Section 8.2.4.

The goal of this paper is to develop a reduction-based AMG algorithm for highly nonsymmetric linear systems. However, Schur-complement coarse-grid operators and reduction-based multigrid are not new to the multigrid literature. Reduction-based multigrid has been considered in many contexts, starting in the geometric multigrid setting [60, 143]. A high-performance variation of geometric multigrid based on a reduction point-of-view is the SMG solver in the Hypre library, which uses a sparse approximation to P_{ideal} that reproduces the action of P_{ideal} on the constant vector [56]. The SMG solver, however, assumes knowledge of an underlying structured grid, and forms symmetric, Galerkin coarse grids. More recently, an adaptive, two-level reduction-based AMG method was proposed in [22, 105], where an approximation to P_{ideal} is developed through an adaptive process. This work provided interesting theoretical results; however, they are based in the A-norm, and thus again confined to the SPD setting. As seen in Section 8.1.1, a reduction-based algorithm is invariably linked to a Schur-complement coarse-grid operator. Other literature exists that is not explicitly marketed as reduction-based, but targets a Schur-complement coarse grid, for example, [33, 106, 114, 118, 123, 146]. Finally, the multigrid reduction-in-time parallel-in-time algorithm [49, 58, 59] is also based on (8.5), and in [58, 61] it was shown that the well-known *parareal* parallel-in-time method [61] is equivalent to a two-level multigrid reduction-in-time method.

Algebraic multigrid methods have also been developed that target nonsymmetric systems, but the theory is limited and the development of effective solvers is ongoing. The reduction-based decomposition given in (8.5) was considered in [123] for nonsymmetric M-matrices resulting from convection-diffusion discretizations. An AMG algorithm for nonsymmetric systems was proposed in [184] based on (8.5), where it is shown that, coupled with an exact solve on F-points, convergence in the spectral sense is bounded based on how well the coarse grid approximates the Schur complement.

Ideal restriction and interpolation are approximated in [184] by performing a constrained minimization over a fixed sparsity pattern for R and P, where known near-null space modes are interpolated exactly, and the remaining DOFs used to approximate the ideal operators. Such an approach is similar to the constrained minimization approach for nonsymmetric systems used in [108, 130], which approximates the ideal operators of A^*A and AA^* for P and R, respectively. Although reduction is not achieved for ideal operators based on A^*A and AA^*, such operators are well-motivated under traditional AMG convergence theory [108]. In fact, in the symmetric setting, approximating ideal operators for A vs. A^2 through constrained energy-minimization produce nearly identical performance [108]. As an extension to the nonsymmetric setting, it is hypothesized that the AMG solvers in [108, 184] would achieve similar performance, although such a comparison has not yet been done. Regardless, the constrained approximation of ideal operators such as in [108, 184] is a broadly applicable solver that relies heavily on constraints for good convergence [108, 163].

This leads to the novelty of this work. Here, we show that matrices with triangular structure are amenable to a reduction-type algorithm (Section 8.2), and we develop a natural way to approximate ideal operators for triangular matrices. The resulting solver does not rely on constraints for strong convergence, and is based only on an appropriate CF-splitting and the approximation of R_{ideal}, which are discussed in Section 8.2.

8.2 Algebraic multigrid based on ideal restriction (AMGir)

8.2.1 Triangular structure and approximate ideal restriction

In Section 8.1.1, theory was developed to motivate ideal restriction and ideal interpolation for nonsymmetric linear systems. However, ideal operators are typically not formed in practice due to the complexity of forming A_{ff}^{-1}. This section shows that for block-triangular matrices, there is a natural way to approximate

A_{ff}^{-1}. Furthermore, approximating ideal operators in coarse-grid correction, coupled with Jacobi F-relaxation, gives a nilpotent multigrid error-propagation operator, that is, convergence to the exact solution is ultimately guaranteed (Section 8.2.2). Coupled with an appropriate CF-splitting (Section 8.2.3), this leads to a fast and practical solver for matrices with triangular structure.

Although direct solves of triangular matrices do not scale well in parallel, direct solve considerations suggest that a triangular matrix is amenable to a reduction-based algorithm, because each step in a forward or backward solve is effectively reduction by eliminating one DOF. Thus, the goal here is to develop a reduction-based solver for triangular systems, independent of matrix ordering, based on the concept of "approximate ideal restriction" (AIR). Moving forward, assume that A is lower triangular with unit diagonal in some ordering. For theoretical purposes, we assume that A is actually ordered to be lower triangular, but it is important to note that results presented here are independent of this ordering, and it is only used to simplify proofs. An in-depth discussion on handling of block-triangular matrices is given in Section 8.3.

Let $A_{ff} = I - L_{ff}$, where L_{ff} is the strictly lower triangular part of A_{ff}, and is, thus, also nilpotent. Hence, A_{ff}^{-1} can be written as a finite Neumann expansion:

$$A_{ff}^{-1} = (I - L_{ff})^{-1} = \sum_{i=0}^{n} L_{ff}^{i}. \tag{8.6}$$

An order-k approximation to A_{ff}^{-1} is then given by truncating (8.6): $\Delta^{(k)} := \sum_{i=0}^{k} L_{ff}^{i}$, for some $0 \leq k \leq n$. The error in $\Delta^{(k)}$ can be measured as $I - \Delta^{(k)} A_{ff} = L_{ff}^{k+1}$. Noting that $R_{\text{ideal}} A = \begin{pmatrix} 0 & S \end{pmatrix}$, let $R = \begin{pmatrix} -A_{cf}\Delta^{(k)} & I \end{pmatrix}$ and consider its action on A:

$$\begin{aligned} RA &= \begin{pmatrix} A_{cf}(I - \Delta^{(k)} A_{ff}) & A_{cc} - A_{cf}\Delta^{(k)} A_{fc} \end{pmatrix} \\ &= \begin{pmatrix} A_{cf} L_{ff}^{k+1} & A_{cc} - A_{cf}\Delta^{(k)} A_{fc} \end{pmatrix}. \end{aligned}$$

Since L_{ff} is nilpotent, approximating ideal restriction via a truncated Neumann expansion of A_{ff}^{-1} can be seen as equivalent to approximating the action of R_{ideal} on A, namely, trying to set RA equal to zero within F-point columns. For certain structured matrices, the kth-order Neumann expansion is exactly eliminating the contribution of F-points within distance k for a given C-point (row of R). In fact, this was the original motivation for AIR – eliminating the contribution of error at F-points to the coarse-grid right-hand side.

Numerical results presented in Section 8.4 are based on two properties of the AMGir error-propagation operator. In a reduction context, the error-propagation operator in the multilevel setting is nilpotent, ensuring an (eventually) exact method (Section 8.2.2). In addition to the nilpotency, an appropriate CF-splitting such that off-diagonal elements of A_{ff} are small additionally ensures effective error reduction at indices that are not "reduced" in a given iteration (Section 8.2.3). The latter is likely the primary reason for strong convergence, although both are important.

8.2.2 Nilpotency of error propagation

Lemma 15 and Corollary 9 below are presented as important pieces in motivating AMGir. Corollary 9 is fundamental to a multilevel implementation of AMGir, showing that as long as A_{ff}^{-1} in ideal restriction and interpolation is approximated with a lower-triangular matrix, the coarse-grid operator retains the same triangular structure as the fine grid. Thus, the same algorithms for forming P and R are appropriate in a recursive and multilevel fashion. For given ideal restriction and interpolation approximations, coupled with Jacobi relaxation on F-points, Lemma 16 shows that the two-grid error propagation operator is nilpotent. Theorem 6 is the primary convergence result of this section, showing that a full multilevel implementation has a nilpotent error-propagation operator. A trivial corollary is that convergence to the exact solution is guaranteed within n iterations. Although bounds of $O(n)$ iterations of $O(n)$ complexity is far removed from the typical linear or near-linear complexity of AMG, such a bound is at least of theoretical interest.

Lemma 15. *Let A be lower triangular and suppose the DOFs of A have been partitioned into C-points and F-points. Then, if $\Delta \in \mathbb{R}^{n_f \times n_f}$ is lower triangular, $A_{cf}\Delta A_{fc}$ is nilpotent and strictly lower triangular. The same holds for $A_{fc}\hat{\Delta}A_{cf}$, where $\hat{\Delta} \in \mathbb{R}^{n_c \times n_c}$.*

Proof. Let C-points be given by $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_{n_c}\}$, where $\mathcal{C}_i$ denotes the global index of the ith C-point, and likewise for F-points. Furthermore, assume $\mathcal{C}$ and $\mathcal{F}$ are in increasing order, that is, $\mathcal{C}_1 < \mathcal{C}_2 < ... < \mathcal{C}_{n_c}$, and that global indices of A are ordered such that A is lower triangular, that is, $A_{ij} = 0$ if $i \leq j$. In terms of paths in the graph of A, this means that from a given node i, there only exist paths to nodes $j \leq i$.

For $A_{cf}\Delta A_{fc}$ to be strictly lower triangular, we must have $(A_{cf}\Delta A_{fc})_{jk} = 0$ for all $0 \leq j \leq k \leq n_c$.

This can be shown by considering paths between the jth and kth C-points in the graph of $A_{cf}\Delta A_{fc}$.

(1) Starting at $\mathcal{C}_j$, A_{fc} can only contain paths from $\mathcal{C}_j$ to F-points $\mathcal{F}_i < \mathcal{C}_j$.

(2) Then, Δ can only contain paths from $\mathcal{F}_i$ to F-points $\mathcal{F}_\ell \leq \mathcal{F}_i$.

(3) Finally, A_{cf} maps from F-points $\mathcal{F}_\ell$ to C-points $\mathcal{C}_k < \mathcal{F}_\ell$.

Thus, the only possible paths in the graph of $A_{cf}\Delta A_{fc}$ are from points $\mathcal{C}_j$ to $\mathcal{C}_k < \mathcal{C}_j$. It follows that if $\mathcal{C}_k \geq \mathcal{C}_j$, or, equivalently, $k \geq j$, then $(A_{cf}\Delta A_{fc})_{jk} = 0$. □

Corollary 9. *Let A be lower triangular and Δ_R and Δ_P be lower triangular approximations to A_{ff}^{-1}, and define $P := \begin{pmatrix} -\Delta_P A_{fc} \\ I \end{pmatrix}$ and $R := \begin{pmatrix} -A_{cf}\Delta_R & I \end{pmatrix}$. Then, the coarse-grid operator $\mathcal{K} := RAP$ (5.5) is lower triangular with diagonal given by that of A_{cc}.*

Proof. First note that

$$\mathcal{K} := RAP = A_{cc} - A_{cf}(\Delta_R + \Delta_P - \Delta_R A_{ff}\Delta_P)A_{fc}.$$

We know that A_{cc} is lower triangular. It follows from Lemma 15 that $A_{cf}(\Delta_R + \Delta_P - \Delta_R A_{ff}\Delta_P)A_{fc}$ is strictly lower triangular. Thus, $\mathcal{K}$ is lower triangular with the same diagonal as A_{cc}. □

Lemma 16 (Nilpotent two-grid AMG)**.** *Let A be lower triangular in some ordering and Δ_R and Δ_P be lower triangular approximations to A_{ff}^{-1}, and define $P := \begin{pmatrix} -\Delta_P A_{fc} \\ I \end{pmatrix}$ and $R := \begin{pmatrix} -A_{cf}\Delta_R & I \end{pmatrix}$. Then, the two-grid AMG preconditioner with coarse-grid correction and Jacobi F-relaxation is strictly lower triangular, and thus nilpotent.*

Proof. Define C-points as the set $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_{n_c}\}$, where $\mathcal{C}_i$ is the global index of the ith C-point, and likewise for F-points. Furthermore, let $\mathcal{C}$ and $\mathcal{F}$ be in increasing order, that is, $\mathcal{C}_1 < \mathcal{C}_2 < ... < \mathcal{C}_{n_c}$, and let global indices of A be ordered such that A is lower triangular, that is, $A_{ij} = 0$ if $i < j$. Then, for a given node i in A, there only exists paths to nodes $j \leq i$. It follows that off-diagonal block A_{cf} only maps from F_i to $C_j < F_i$, and similarly for A_{fc}.

The two-grid AMG error-propagation operator with relaxation scheme M^{-1} looks like $E_{TG} = (I - M^{-1}A)(I - P(RAP)^{-1}RA)$. First consider the coarse-grid correction term:

$$\begin{aligned} I - P(RAP)^{-1}RA &= \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} - \begin{pmatrix} -\Delta_P A_{fc} \\ I \end{pmatrix} \mathcal{K}^{-1} \begin{pmatrix} -A_{cf}\Delta_R & I \end{pmatrix} \begin{pmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{pmatrix} \\ &= \begin{pmatrix} I + \Delta_P A_{fc}\mathcal{K}^{-1}A_{cf}(I - \Delta_R A_{ff}) & \Delta_P A_{fc}\mathcal{K}^{-1}(A_{cc} - A_{cf}\Delta_R A_{fc}) \\ -\mathcal{K}^{-1}A_{cf}(I - \Delta_R A_{ff}) & I - \mathcal{K}^{-1}(A_{cc} - A_{cf}\Delta_R A_{fc}) \end{pmatrix}. \end{aligned} \tag{8.7}$$

Note that $I - \Delta_R A_{ff}$ is strictly lower triangular, and $\mathcal{K}$ and $A_{cc} - A_{cf}\Delta_R A_{fc}$ are lower triangular. Then, the action of the lower-left block in (8.7) can be seen as three steps, mapping $F_i \to F_j \to C_k \to C_\ell$, where $C_\ell \leq C_k < F_j < F_i$. A similar result holds for the upper-right block, showing that, in the global ordering, the off-diagonal blocks of (8.7) are strictly lower triangular. For the lower-right block, it follows from Corollary 9 and Lemma 15 that $\mathcal{K}^{-1}(A_{cc} - A_{cf}\Delta_R A_{fc})$ is lower triangular with unit diagonal. Subtracting from the identity gives a strictly lower triangular block. Finally, given that the product of a strictly lower triangular matrix and lower triangular matrix is strictly lower triangular, it follows from Lemma 15 that $\Delta_P A_{fc}\mathcal{K}^{-1}A_{cf}(I - \Delta_R A_{ff})$ is strictly lower triangular. Then, the upper left block is lower triangular with unit diagonal.

Thus, the error-propagation operator for coarse-grid correction is lower triangular with unit diagonal in the FF-block, and strictly lower triangular in other blocks. Furthermore, the error-propagation matrix for Jacobi relaxation on F-points is strictly lower triangular in the FF-block and lower triangular elsewhere. It follows that the AIR two-grid error-propagation operator given by the product of F-Jacobi relaxation and coarse-grid correction is a strictly lower triangular and, thus, nilpotent, operator. □

Theorem 6 (Nilpotent AMG)**.** *Let A be lower triangular in some ordering and Δ_R and Δ_P be lower triangular approximations to A_{ff}^{-1}, and define $P := \begin{pmatrix} -\Delta_P A_{fc} \\ I \end{pmatrix}$ and $R := \begin{pmatrix} -A_{cf}\Delta_R & I \end{pmatrix}$. Then, the multilevel AMG preconditioner with coarse-grid correction and Jacobi F-relaxation is strictly lower triangular, and thus nilpotent.*

Proof. In proving a nilpotent error-propagation for a multilevel algorithm, we proceed in a recursive fashion. Consider the error-propagation of coarse-grid correction with an inexact coarse-grid solve. Error propagation of coarse-grid correction takes the form $\mathbf{e}_{new} = \mathbf{e}_{old} - P\mathbf{x}_c$, for some coarse-grid correction $\mathbf{x}_c$. In the case of an exact coarse-grid solve, $P\overline{\mathbf{x}}_c = P\mathcal{K}^{-1}RA\mathbf{e}_{old}$ and $\mathbf{e}_{new} = (I - P\mathcal{K}^{-1}RA)\mathbf{e}_{old}$. For an inexact coarse-grid solve, the correction can be written as some perturbation of the exact solve, $\mathbf{x}_c = \overline{\mathbf{x}}_c + (\mathbf{x}_c - \overline{\mathbf{x}}_c)$, where the error in the correction is given by $\mathbf{e}_c = \mathbf{x}_c - \overline{\mathbf{x}}_c$. The error propagation of the inexact coarse-grid solve, that is, the coarse-grid V-cycle, then operates on $\mathbf{e}_c$, that is, $E_c(\mathbf{e}_c) = E_c(\mathbf{x}_0 - \overline{\mathbf{x}})$, for some initial guess $\mathbf{x}_0$. Assuming a zero initial guess on the coarse-grid problem as is standard in AMG, then

$$\begin{aligned}
\mathbf{e}_{new} &= \mathbf{e}_{old} - P(\overline{\mathbf{x}}_c - E_c\overline{\mathbf{x}}_c) \\
&= \mathbf{e}_{old} - P(I - E_c)\overline{\mathbf{x}}_c \\
&= \mathbf{e}_{old} - P(I - E_c)\mathcal{K}^{-1}RA\mathbf{e}_{old} \\
&= \Big[(I - P\mathcal{K}^{-1}RA) + PE_c\mathcal{K}^{-1}RA\Big]\mathbf{e}_{old}.
\end{aligned}$$

Lemma 16 showed that $(I - P\mathcal{K}^{-1}RA)$ is strictly lower triangular when coupled with Jacobi F-relaxation. Now, we must show that the error term $PE_c\mathcal{K}^{-1}RA$ is also strictly lower triangular. Given that A is lower triangular, it is sufficient to show that $PE_c\mathcal{K}^{-1}R$ is strictly lower triangular. Expanding, we get

$$PE_c\mathcal{K}^{-1}R = \begin{pmatrix} \Delta_P A_{fc} E_c \mathcal{K}^{-1} A_{cf} \Delta_R & -\Delta_P A_{fc} E_c \mathcal{K}^{-1} \\ -E_c \mathcal{K}^{-1} A_{cf} \Delta_R & E_c \mathcal{K}^{-1} \end{pmatrix}. \tag{8.8}$$

An analogous result to Lemma 16 confirms that if $E_c\mathcal{K}^{-1}$ is strictly lower triangular, then so is (8.8).

Let E_i be the error-propagation operator for a V-cycle starting on the ith level of the hierarchy. On the coarsest level of the hierarchy, say $\ell = 0$, $E_0 = \mathbf{0}$ because the solve is exact. Then, on level $\ell = 1$, $E_c\mathcal{K}^{-1} = E_0\mathcal{K}^{-1} = \mathbf{0}$ is strictly lower triangular. It follows that the error-propagation operator on level $\ell = 1$, E_1, is strictly lower-triangular. On level $\ell = 2$, $E_c\mathcal{K}^{-1} = E_1\mathcal{K}^{-1}$ is again strictly lower triangular, because E_1 is strictly lower triangular and $\mathcal{K}$ lower triangular. By induction, the error-propagation operator E_ℓ for a multilevel hierarch with ℓ levels is strictly lower triangular and, thus, nilpotent. □

For triangular systems, it is easy to see that simple Jacobi relaxation also produces a nilpotent error-propagation operator. The significance of Lemma 16 and Theorem 6 is in showing analytically that (at least in the spectral sense) coarse-grid correction does not cause divergent behavior, and AMGir is asymptotically robust for triangular systems.

8.2.3 CF-splitting and L_{ff}

Although nilpotency of error-propagation is a nice property to have, it does not necessarily imply fast convergence. To understand what else is contributing to good convergence, we further examine the error-propagation operator. First note that error propagation for Jacobi F-relaxation is given by $I - D_{ff}^{-1}A_{ff} = L_{ff}$. Letting $\Delta_P = \Delta_R := \Delta^{(k)}$ for some $k \geq 0$, then the following identities hold, which we will need later:

$$\begin{aligned} \mathcal{K} &= A_{cc} - A_{cf}\Delta^{(2k+1)}A_{fc}, \\ \left(\Delta^{(2k+1)}\right)^{-1} &= \left(I + L_{ff}^{k+1}\right)^{-1}\left(\Delta^{(k)}\right)^{-1}, \\ \left(\Delta^{(2k+1)}\right)^{-1}\Delta^{(k)} &= I + \sum_{j=1}^{n}\left(-L_{ff}^{k+1}\right)^{j}. \end{aligned} \tag{8.9}$$

For ease of notation, let $\Delta := \Delta^{(k)}$ and $\tilde{\Delta} := \Delta^{(2k+1)}$. Then, using (8.9) and a Woodbury inverse expansion for $\mathcal{K}^{-1}$, we get

$$\begin{aligned} \mathcal{K}^{-1}(A_{cc} - A_{cf}\Delta A_{fc}) &= \left(A_{cc}^{-1} + A_{cc}^{-1}A_{cf}\left(\tilde{\Delta}^{-1} - A_{fc}A_{cc}^{-1}A_{cf}\right)^{-1}A_{fc}A_{cc}^{-1}\right)(A_{cc} - A_{cf}\Delta A_{fc}) \\ &= \left(I + A_{cc}^{-1}A_{cf}\left(\tilde{\Delta}^{-1} - A_{fc}A_{cc}^{-1}A_{cf}\right)^{-1}A_{fc}\right)\left(I - A_{cc}^{-1}A_{cf}\Delta A_{fc}\right) \\ &= I + A_{cc}^{-1}A_{cf}\left(\tilde{\Delta}^{-1} - A_{fc}A_{cc}^{-1}A_{cf}\right)^{-1}(I - \tilde{\Delta}^{-1}\Delta)A_{fc} \\ &= I - A_{cc}^{-1}A_{cf}\left(\tilde{\Delta}^{-1} - A_{fc}A_{cc}^{-1}A_{cf}\right)^{-1}\sum_{j=1}^{n}\left(-L_{ff}^{k+1}\right)^{j}A_{fc}. \end{aligned} \tag{8.10}$$

Plugging (8.10) and the identity $I - \Delta^{(k)}A_{ff} = L_{ff}^{k+1}$ into the coarse-grid correction (8.7) shows that error propagation for coarse-grid correction can be written as

$$\begin{aligned} \mathbf{e}_f^{(1)} = &\left(I + \Delta A_{fc}\mathcal{K}^{-1}A_{cf}L_{ff}^{k+1}\right)\mathbf{e}_f^{(0)} + \\ &\Delta A_{fc}\left(I - A_{cc}^{-1}A_{cf}\left(\tilde{\Delta}^{-1} - A_{fc}A_{cc}^{-1}A_{cf}\right)^{-1}\sum_{j=1}^{n}\left(-L_{ff}^{k+1}\right)^{j}A_{fc}\right)\mathbf{e}_c^{(0)}, \\ \mathbf{e}_c^{(1)} = &-\mathcal{K}^{-1}A_{cf}L_{ff}^{k+1}\mathbf{e}_f^{(0)} + A_{cc}^{-1}A_{cf}\left(\tilde{\Delta}^{-1} - A_{fc}A_{cc}^{-1}A_{cf}\right)^{-1}\sum_{j=1}^{n}\left(-L_{ff}^{k+1}\right)^{j}A_{fc}\mathbf{e}_c^{(0)}. \end{aligned}$$

Define $\mathcal{A} := \Delta A_{fc}$, $\mathcal{B} := \mathcal{K}^{-1}A_{cf}$, and $\mathcal{C} := A_{cc}^{-1}A_{cf}\left(\tilde{\Delta}^{-1} - A_{fc}A_{cc}^{-1}A_{cf}\right)^{-1}$. Then, if we consider the highest-order term of L_{ff}, error propagation takes the form

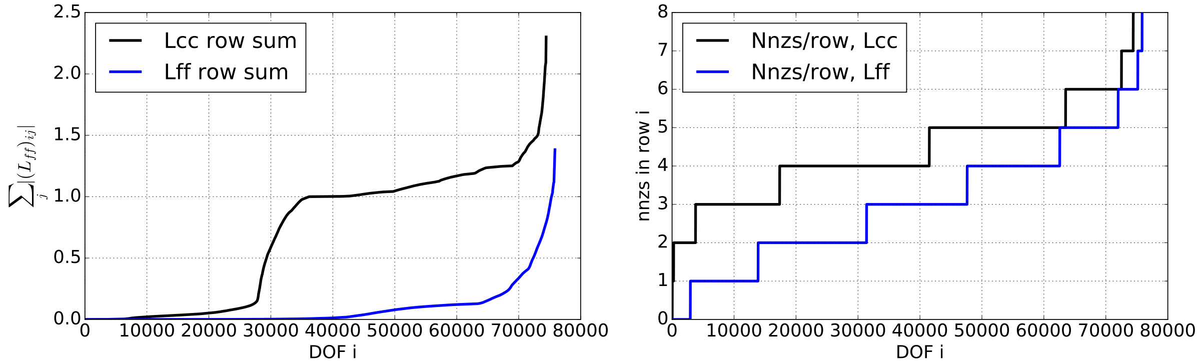
$$\begin{aligned} \mathbf{e}_f^{(i+1)} &= \left(I + \mathcal{A}\mathcal{B}L_{ff}^{k+1}\right)\mathbf{e}_f^{(i)} + \mathcal{A}\left(I - \mathcal{C}L_{ff}^{k+1}A_{fc}\right)\mathbf{e}_c^{(i)}, \\ \mathbf{e}_c^{(i+1)} &= -\mathcal{B}L_{ff}^{k+1}\mathbf{e}_f^{(i)} + \mathcal{C}L_{ff}^{k+1}A_{fc}\mathbf{e}_c^{(i)}. \end{aligned} \tag{8.11}$$

The key thing to notice in (8.11) is that each error update on C-points is hit with L_{ff}^{k+1}. Note that ℓ iterations of F-Jacobi relaxation has error propagation

$$\begin{pmatrix} \mathbf{e}_f^{(i+\ell)} \\ \mathbf{e}_c^{(i+\ell)} \end{pmatrix} = \begin{pmatrix} L_{ff}^{\ell} & -\Delta^{(\ell-1)}A_{fc} \\ 0 & I \end{pmatrix} \begin{pmatrix} \mathbf{e}_f^{(i)} \\ \mathbf{e}_c^{(i)} \end{pmatrix}. \tag{8.12}$$

Then, coupling coarse-grid correction (8.11) with $k+1$ iterations of pre- or post-F-relaxation scales the error at *all* points by L_{ff}^{k+1}.

Such a scaling of error by L_{ff}^{k+1} is a complementary contribution to strong convergence factors achieved by AMGir. Ensuring that L_{ff} is not only nilpotent, but also has very small elements, is key to achieving a fast reduction in error. Here, we focus on matrices that result from the discretization of differential operators, in which case off-diagonal entries are typically expected to be smaller in magnitude than the diagonal. In fact, the traditional AMG motivation for a CF-splitting, picking F-points such that A_{ff} is well-conditioned, is also appropriate for AMGir. A classical AMG coarsening approach targets A_{ff} to be strictly diagonally dominant, which is equivalent to minimizing the size of elements in L_{ff}. Figure 8.1 demonstrates that for a model problem introduced in Section 8.4, L_{ff} has very small row sums, although very few rows of A_{ff} are diagonal (in which case the row of L_{ff} is empty). Similar results hold on all levels of the hierarchy, as well as for other discretizations and finite element orders.



(a) Absolute row sum

(b) Nonzeros per row

Figure 8.1: Data on CF-splitting and L_{ff} for the first level in a hierarchy built on a second-order finite element discretization of the *inset* model problem (see Section 8.4).

Small row sums of L_{ff} reduce error efficiently for most rows. For rows without small row sums in L_{ff}, we rely on the redistribution of error through operators $\mathcal{A}, \mathcal{B}$, and $\mathcal{C}$, as well as the nilpotency of error propagation to reduce error. Specifically, larger row sums in L_{ff} typically occur near the boundary of the domain. Near the boundary, a nilpotent error-propagation operator reaches the exact solution in $O(1)$ iterations.

8.2.4 Schur-complement preconditioning

This work focuses on a traditional AMG framework. However, preconditioners are often developed by approximating the block decomposition of A in (8.5). An analogous result to Theorem 6 holds for the block preconditioner

$$\widehat{M} = \begin{pmatrix} I & 0 \\ A_{cf}\Delta_R & I \end{pmatrix} \begin{pmatrix} \Delta^{-1} & 0 \\ 0 & \widehat{\mathcal{K}} \end{pmatrix} \begin{pmatrix} I & \Delta_P A_{fc} \\ 0 & I \end{pmatrix}, \tag{8.13}$$

where Δ, Δ_R, and Δ_P are lower-triangular approximations to A_{ff}^{-1} and $\widehat{\mathcal{K}}$ some approximation to the Schur complement (not necessarily RAP), that is, the multilevel error-propagation operator, $I-\widehat{M}^{-1}A$, is nilpotent. Letting $\Delta = \Delta_P = \Delta_R$ be an order-k truncated Neumann expansion as in (8.6) and $\widehat{\mathcal{K}} := A_{cc} - A_{cf}\Delta A_{fc}$ gives the relation $A = \widehat{M} + N$, where $N = \begin{pmatrix} A_{ff} - \Delta^{-1} & 0 \\ 0 & 0 \end{pmatrix}$. Some linear algebra then shows that

$$I - \widehat{M}^{-1}A = \begin{pmatrix} I + \Delta A_{fc}\mathcal{K}^{-1}A_{cf} & 0 \\ -\mathcal{K}^{-1}A_{cf} & 0 \end{pmatrix} \begin{pmatrix} (D_{ff}^{-1}L_{ff})^{k+1} & 0 \\ 0 & 0 \end{pmatrix}. \tag{8.14}$$

Note that here we have only assumed A_{ff} to be nonsingular. An interesting result of (8.14) is that the error after $\ell > 0$ iterations only depends on previous error at F-points, that is, at each iteration, C-point error is eliminated and then acquired based on error at F-points. In the case of lower triangular A_{ff}, (8.13) can act as a preconditioner with very similar properties to AMGir. Although we do not numerically explore preconditioning with $\widehat{M}$ here, its theoretical similarity to AMGir is worth demonstrating for future work.

Remark 10. *An infinite Neumann expansion* (8.6) *can be used to approximate A_{ff}^{-1} for general non-triangular matrices, and similar algorithms developed* (8.11) *and* (8.14)*. However, in the case of triangular matrices, an order-k Neumann approximation to A_{ff}^{-1} is exact on the diagonal and first k sub-diagonals. For general matrices, such accuracy cannot be guaranteed. Even in the case of symmetric positive definite matrices, where off-diagonal entries decay exponentially fast for well-conditioned A_{ff} [20], a truncated Neumann expansion does not necessarily provide a good approximation to A_{ff}^{-1} (or its diagonal for that matter). The application of AMGir to matrices with non-triangular components is on-going work.*

8.3 Algorithm

Algorithmically, AMGir takes on the traditional structure of an AMG method. Building on the theoretical motivation provided in Section 8.2, here we develop a practical AMGir algorithm, with particular attention paid to limiting the setup and solve complexity, while retaining strong convergence.

Blocks: In certain cases, matrix equations solved by AMG admit a natural block structure, the most common of which are from systems of PDEs, where a single spatial node has several variables, say k, discretized over it. Two traditional approaches to handling such a system are (i) to treat each variable as a block, resulting in a decomposition of the matrix A into ${}^{n}/{}_{k} \times {}^{n}/{}_{k}$ blocks, or (ii) to treat each node over which multiple variables are discretized as a block or "super-node," resulting in a decomposition of the matrix A into $k \times k$ blocks. For the super-node approach, coarsening, interpolation, etc., are all done in a block sense, e.g., each block of k DOFs is designated either a C-block or an F-block [145]. A conceptually similar approach was used for discontinuous discretizations of elliptic PDEs in [129]. Typically, adjacent elements in a discontinuous discretization each have an independent DOF defined on the same spatial node and, thus, [129] defines a block in the initial matrix as all nodes defined at a given spatial location.

The problems considered here are not systems and do not contain super-nodes in the traditional sense, but discontinuous discretizations do lead to an inherent block structure. Recall that AMGir is designed for matrices with a triangular structure. Discontinuous upwind discretizations result in a block lower-triangular matrix in some ordering, where each block corresponds to a discontinuous finite element. Three approaches for AMGir applied to $A\mathbf{x} = \mathbf{b}$ are then as follows:

(a) Ignore block structure and treat $A\mathbf{x} = \mathbf{b}$ as a scalar problem.

(b) Let D_b be the block diagonal of A, and solve $D_b^{-1}A\mathbf{x} = D_b^{-1}\mathbf{b}$, that is, scale the block diagonal of A to consist of identity blocks.

(c) Treat $A\mathbf{x} = \mathbf{b}$ as a block system, that is, perform coarsening, relaxation, interpolation, and restriction in a block setting.

Ignoring block structure altogether achieves reasonable convergence factors for some problems, but also diverges for others. As it turns out, some consideration of the element-wise block structure is generally necessary for strong convergence of AMGir. Such results highlight the significance of A being triangular for successful reduction. This leaves options (b) and (c) for handling the block structure. With respect to convergence, all tests have indicated comparable convergence rates for these two options. However, setup and cycle complexities increase when using the full block-approach (c) as opposed to scaling by the block-inverse and solving as a scalar problem (b). For this reason, the first step of AMGir scales by the block-inverse of A, after which everything is done in a scalar setting. It is worth pointing out that in terms of finite-element order, the cost of constructing high-order discretizations becomes intractable far before the cost of computing D_b^{-1}, that is, computing D_b^{-1} is not a limiting factor for finite element order.

Remark 11. *Depending on the discretization, scaling by D_b^{-1} can increase the number of nonzeros in the matrix by up to 1.5 times, thus increasing complexities. This is accounted for in numerical results presented in Section 8.4, that is, presented complexities are based on the number of nonzeros in the initial matrix.*

Matrix truncation: For scalable convergence of AMGir in serial and in parallel, it is important to limit the complexity of coarse-grid operators. This motivates building R only based on strong connections and, thus, the truncating or lumping of weak connections. The idea is very simple: remove entries from a matrix in the hierarchy, A_ℓ, that are smaller than some threshold, typically with respect to the diagonal element of the given row. Such methods have been used in AMG for symmetric problems with diffusive components (see, for example, [13, 54, 176]). Heuristically, eliminating small entries is even more appropriate in the hyperbolic setting, because the solution at any given point only depends on the solution at other points upwind along the characteristic. In the discrete setting, small entries that arise in matrix operations are often not aligned with the characteristic and are more of a numerical effect, suggesting that some can be eliminated without a significant degradation of convergence. In AMGir, elements $\{a_{ij} \mid j \neq i, |a_{ij}| \leq \varphi |a_{ii}|\}$ are eliminated (that is, set to zero) for each row i of matrix A_ℓ, and some drop-tolerance φ.

Remark 12. *In a more traditional AMG approach, one can also eliminate small off-diagonal entries from a matrix by “lumping” or adding them to the diagonal. The purpose of lumping is to preserve the row-sum of the matrix; however, numerical tests indicate no improvement in performance by lumping entires over* *eliminating them, and thus it is not explored further in this work.*

Coarsening: Recall that the staple of AMGir is approximating $-A_{cf}A_{ff}^{-1}$ in ideal restriction and coupling coarse-grid correction with relaxation on F-points. The effectiveness of each of these processes depends heavily on A_{ff} being well-conditioned and diagonally dominant. Such goals are consistent with the goals of coarsening in classical AMG [15]. Thus, the so-called classical strength of connection (SOC) and classical AMG CF-splitting are used as the coarsening procedure for AMGir [15, 145]. For coarsening, the SOC takes the hard minimum approach, that is, the set of strong connections for a given row consists of negative entries that are large with respect to the largest negative off-diagonal in that row:

$$\mathcal{N}_\ell = \{a_{\ell k} \mid -a_{\ell k} \geq \psi \max_{j\neq\ell} -a_{\ell j}\}.$$

Various alternative SOC measures and coarsening routines were tested as well, such as an absolute value as opposed to a hard minimum, a symmetric smoothed aggregation measure, and an evolution measure [128], but the classical measure emphasizing large negative values consistently performs best. Once a SOC matrix has been built, this is used to compute a classical AMG CF-splitting of DOFs [15], and the algorithm proceeds to approximating R_{ideal}.

Relaxation: Because ideal restriction is designed to eliminate error at C-points (Theorem 5), coarse-grid correction was coupled with Jacobi F-relaxation (8.12) in designing and analyzing AMGir (Section 8.2). One F-Jacobi iteration takes the form

$$\mathbf{x}_f^{i+1} = \mathbf{x}_f^i + D_{ff}^{-1}(\mathbf{b}_f - A_{ff}\mathbf{x}_f^i - A_{fc}\mathbf{x}_c^i), \tag{8.15}$$

where D_{ff} is the diagonal of A_{ff}. Interestingly, error propagation for ℓ F-Jacobi iterations (8.12) exactly takes the form of using an ℓth-order Neumann approximation to interpolate a coarse-grid correction. In both cases, an accurate coarse-grid correction at C-points is used to reduce error at F-points. In fact, given a correction term $(RAP)^{-1}RA\mathbf{e}$ to be interpolated to the fine grid, the following two algorithms are equivalent:

- Interpolate the correction using $\hat{P} = \begin{pmatrix} -\Delta^{(\ell)}A_{fc} \\ I \end{pmatrix}$.

- Interpolate the correction to C-points only using $\hat{P} = \begin{pmatrix} 0 \\ I \end{pmatrix}$, but perform ℓ iterations of F-Jacobi relaxation on the correction before updating the solution.

Stronger still in terms of error reduction is performing ℓ iterations of F-Jacobi relaxation on the system, following the correction. Given this equivalence, AMGir relies on F-relaxation to reduce error at F-points, and uses a very sparse and low-order approximation to P_{ideal} to limit OC and CC.

Restriction: Controlling the sparsity-pattern fill-in of coarse-grid operators is important in controlling complexity of the method, and this fill-in depends on the sparsity of R and P. Thus, in practice, restriction operators are built in AMGir by using a truncated Neumann expansion (8.6) on a SOC matrix, as opposed to directly on A, to reduce the number of nonzeros in R. Here, strong connections are defined through a classical SOC based on the absolute value, that is, strong connections to node i are given by the set

$$\widehat{\mathcal{N}}_i = \{a_{ij} \mid |a_{ij}| \geq \phi \max_{\ell \neq j} |a_{\ell j}|\}. \tag{8.16}$$

Restriction is then built as follows: let C be the SOC matrix associated with (8.16). Then $A_{cf}A_{ff}^{-1}$ is approximated by $C_{cf}\widehat{\Delta}^{(k)}$, where $\widehat{\Delta}^{(k)}$ is a kth-order Neumann approximation to C_{ff}^{-1} (8.6).

Interpolation: As mentioned previously, AMGir can and does rely primarily on relaxation to reduce error at F-points. Thus, the main role of interpolation is in defining the coarse-grid operator, $\mathcal{K} := RAP$. Continuing with the goal of limiting coarse-grid fill in, we want to keep P as sparse as possible, while retaining certain properties important to AMGir. Specifically, we want to maintain the constant vector as an algebraically smooth mode in $\mathcal{K}$, because this is a fundamental assumption to the chosen SOC and coarsening routine. This motivates a *one-point* interpolation. One-point interpolation determines each F-point value from, and only from, the value at its strongest-connected C-point neighbor (like a truncated A_{fc}). In this case, each row of P has exactly one nonzero, equal to one.

Remark 13. *Although using a more accurate approximation to P_{ideal} for P also gives a more accurate coarse-grid operator with respect to the Schur complement, in practice the improvement in convergence of such an approach does not overcome the increase in CC.*

Parameter choice: More robust AMG solvers such as root-node AMG [108], which are designed to solve a wider class of problems than traditional AMG methods, have many parameters to tune for optimal convergence. One of the novel features of AMGir is its relative insensitivity to parameter choice, that is, a "good" set of parameters has proved effective on all discretizations and problems that AMGir has been applied to. To demonstrate this robustness, all results are presented with a fixed set of parameter values. The matrix A is initially scaled by a block-diagonal inverse, after which the system is treated as a scalar problem. No Krylov acceleration is used in the solve phase, and $V(0, 2_F)$-cycles are applied, with two iterations of Jacobi F-relaxation applied as post-relaxation. Classical AMG coarsening is used to generate a CF-splitting, based on a classical SOC with threshold $\psi = 0.3$. The truncation tolerance is chosen as $\varphi = 10^{-3}$, which is numerically motivated in Section 8.4.0.2. A degree-one Neumann expansion (8.6) is used to build R with SOC tolerance $\phi = 0.025$, and so-called one-point interpolation is used.

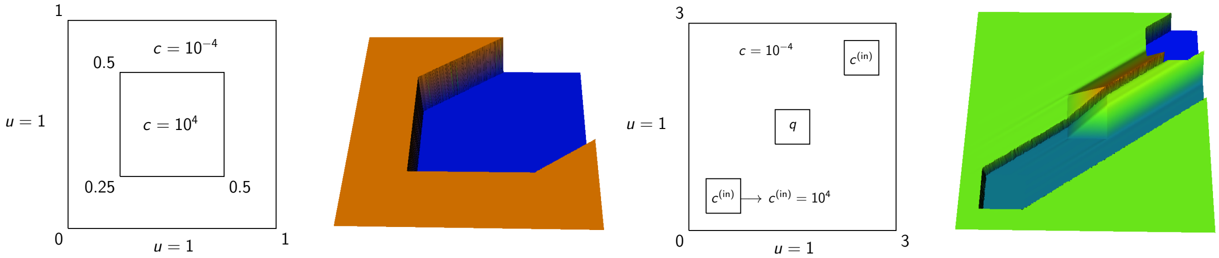
8.4 Steady state transport

The model problem used here is the steady state transport equation:

$$\begin{aligned} \mathbf{b}(x,y) \cdot \nabla u + c(x,y)u &= q(x,y) \qquad \Omega, \\ u &= g(x,y) \qquad \Gamma_{\text{in}}, \end{aligned} \tag{8.17}$$

for domain Ω and inflow boundary Γ_{in}. Multiple cases are studied that encompass spatially dependent source terms, $q(x,y)$, discontinuities in the material coefficient, $c(x,y)$, and constant and non-constant flow direction, $\mathbf{b}(x,y)$, over structured and unstructured meshes. Two domains and the respective solution for a constant flow are given in Figure 8.2. Several variations of the *inset* domain with non-constant flow are shown in Figure 8.4 in Section 8.4.0.1.

To accompany the different domains considered, multiple upwind discretizations are implemented. A first-order lumped finite element discretization [116, 117] is applied on structured and unstructured meshes. Standard fully upwinded discontinuous Galerkin (DG) discretizations [98, 142] are also tested, with finite element orders $0-6$, and the boundary conditions strongly- and weakly-imposed. A comprehensive introduction can be found in [46]. Standard upwinded DG methods arise as special cases in [29] and for almost-scattering-free problems in [141]. In all problems tested, AMGir performs near identically on strongly

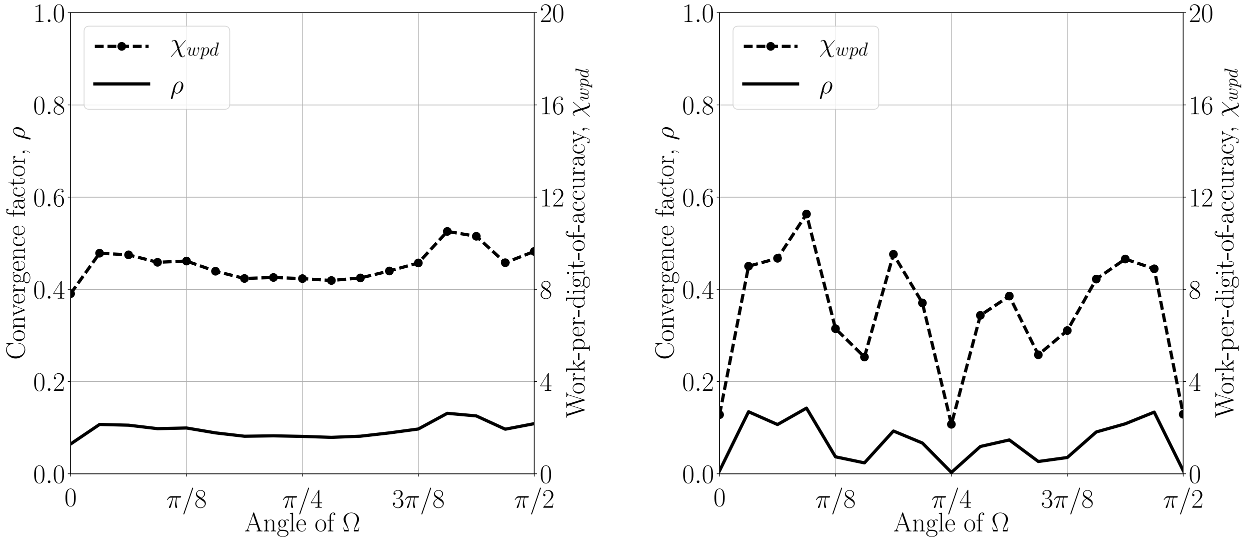


(a) *Inset* domain

(b) *Block-source* domain

Figure 8.2: Two domains for the steady state transport equation. Inflow boundaries consist of the south and west boundaries with inflow $u = 1$. Material coefficient $c(x, y)$ is piecewise constant in both cases, with changes of eight orders of magnitude. The *block-source* domain has an interior source $q(x, y) = 1$ in the interior block and $q(x, y) = 0.5 \cdot 10^4$ in the lower left block.

and weakly enforced boundary conditions; thus, moving forward, weakly-enforced boundary conditions are (arbitrarily) used.



(a) LCB unstructured mesh

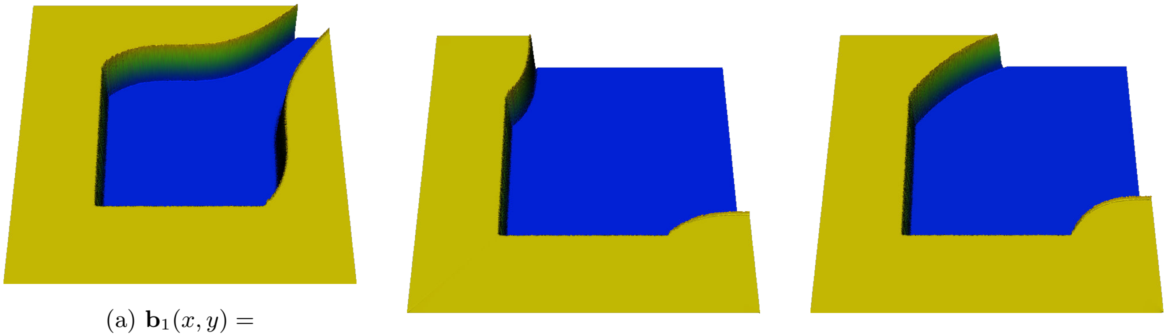
(b) LCB structured mesh

Figure 8.3: Convergence factors and work-per-digit-of-accuracy for AMGir applied to LCB discretizations of the *inset* problem, with angles between 0 and $\pi/2$, on unstructured and structured meshes, and ≈ 2.25M DOFs.

8.4.0.1 Non-constant flow and angular variation

Anisotropies on unstructured meshes and strong non-grid-aligned anisotropies can prove difficult for AMG solvers [108, 152]. An additional novel feature of AMGir is its robustness with respect to problems defined on structured and unstructured meshes. Figure 8.3 demonstrates this robustness for LCB discretizations of the *inset* problem on structured and unstructured meshes, with fixed angle $\Omega := \mathbf{b}(x, y) = (\cos(\theta), \sin(\theta))$, for angles $\theta \in [0, {}^{\pi}/_{2}]$.

Because unstructured meshes are often used in practice and typically more difficult from a solver perspective, further results in Section 8.4 are presented for unstructured meshes. By nature of the relative invariance of convergence and complexity with respect to angle, further tests in Section 8.4 arbitrarily fix $\theta = {}^{3\pi}/_{16}$ for the *inset* and block-source problems.



(a) $\mathbf{b}_1(x, y) = (\cos(\pi y)^2, \cos(\pi x)^2)$.

(b) $\mathbf{b}_2(x, y) = (\sin(\pi y)^2, \sin(\pi x)^2)$.

(c) $\mathbf{b}_3(x, y) = (y^4, \cos({}^{\pi x}/_{2})^2)$.

Figure 8.4: Solution of three different moving flows defined on the *inset* domain.

Remark 14. *For some angles, AMGir performs better in terms of convergence factor and work per digit of accuracy on an unstructured mesh compared with a structured mesh. However, even when these measures favor an unstructured mesh, the wall-clock time of the setup and solve phase is at least* 2× *faster in all cases for a structured mesh over an unstructured mesh. It is possible that a structured mesh makes for a more structured matrix amenable to matrix-vector operations, but a detailed analysis is outside the scope of this work.*

In addition to being robust with respect to angular variations, AMGir is generally insensitive to flow direction $\mathbf{b}(x, y)$. Figure 8.4 shows the solution to three different non-constant $\mathbf{b}(x, y)$ defined on the *inset* domain. Table 8.1 then shows the convergence factor, CC, and work-per-digit-of-accuracy of AMGir applied to a constant flow, and each of the moving flows shown in Figure 8.4.

$\mathbf{b}(x,y)$	Ω	$\mathbf{b}_1(x,y)$	$\mathbf{b}_2(x,y)$	$\mathbf{b}_3(x,y)$
ρ	0.10	0.11	0.09	0.10
CC	7.58	7.49	7.54	7.51
χ_{WPD}	7.56	7.86	7.17	7.63

Table 8.1: Convergence factor, CC, and work-per-digit-of-accuracy for AMGir applied to variations in flow direction, $\mathbf{b}(x,y)$, and constant flow direction $\mathbf{b}(x,y) = \Omega = (\cos(^{3\pi}/_{16}), \sin(^{3\pi}/_{16}))$. Discretizations are defined on the *inset* domain with an unstructured mesh, using upwind DG with linear elements and $\approx$ 2.7M DOFs.

8.4.0.2 Truncation and building restriction

Truncating coarse-grid operators as introduced in Section 8.3 offers a significant improvement in AMGir performance when considering work-per-digit-of-accuracy. One of the difficulties with scaling AMGir to progressively larger problems is the fill-in of the coarse-grid operators, which can greatly increase the CC. However, as discussed in Section 8.3, many of the non-zero entries in A_i for i lower in the AMGir hierarchy are likely to be relatively small with respect to diagonal elements, and can be removed without substantial degradation in convergence. Figure 8.5 shows work-per-digit-of-accuracy as a function of truncation tolerance for many combinations of problems considered here. All discretizations, all domains, and finite elements of order $0-6$ are tested and shown in Figure 8.5. Despite matrices with very different element magnitude and connectivity, a truncation tolerance of $\varphi = 10^{-3}$ (shown in dotted black) is an effective choice in all cases, leading to a $20-50\%$ reduction in iteration cost. Setup cost is not measured in detail, but setup wall-clock times are reduced by comparable amounts.

Remark 15. *It is worth pointing out that $\varphi = 10^{-3}$ is not the best choice of truncation tolerance for all problems considered, only that it is an effective choice in all cases. For matrices with high connectivity such as higher order finite elements, a larger φ reduces the work-per-digit-of-accuracy. Conversely, matrices with very low connectivity, such as a degree zero DG discretization, favor smaller φ.*

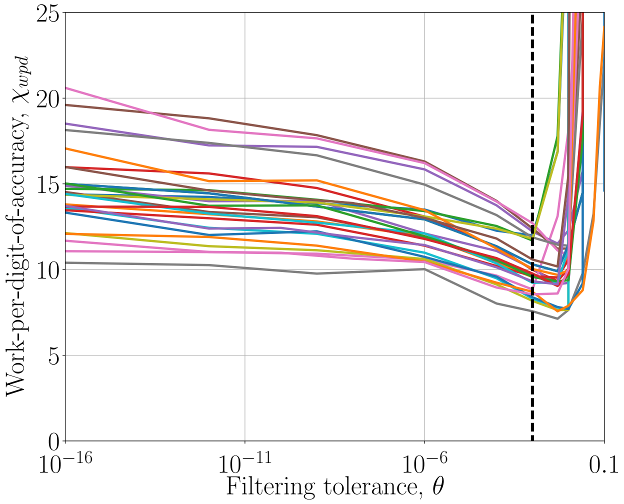
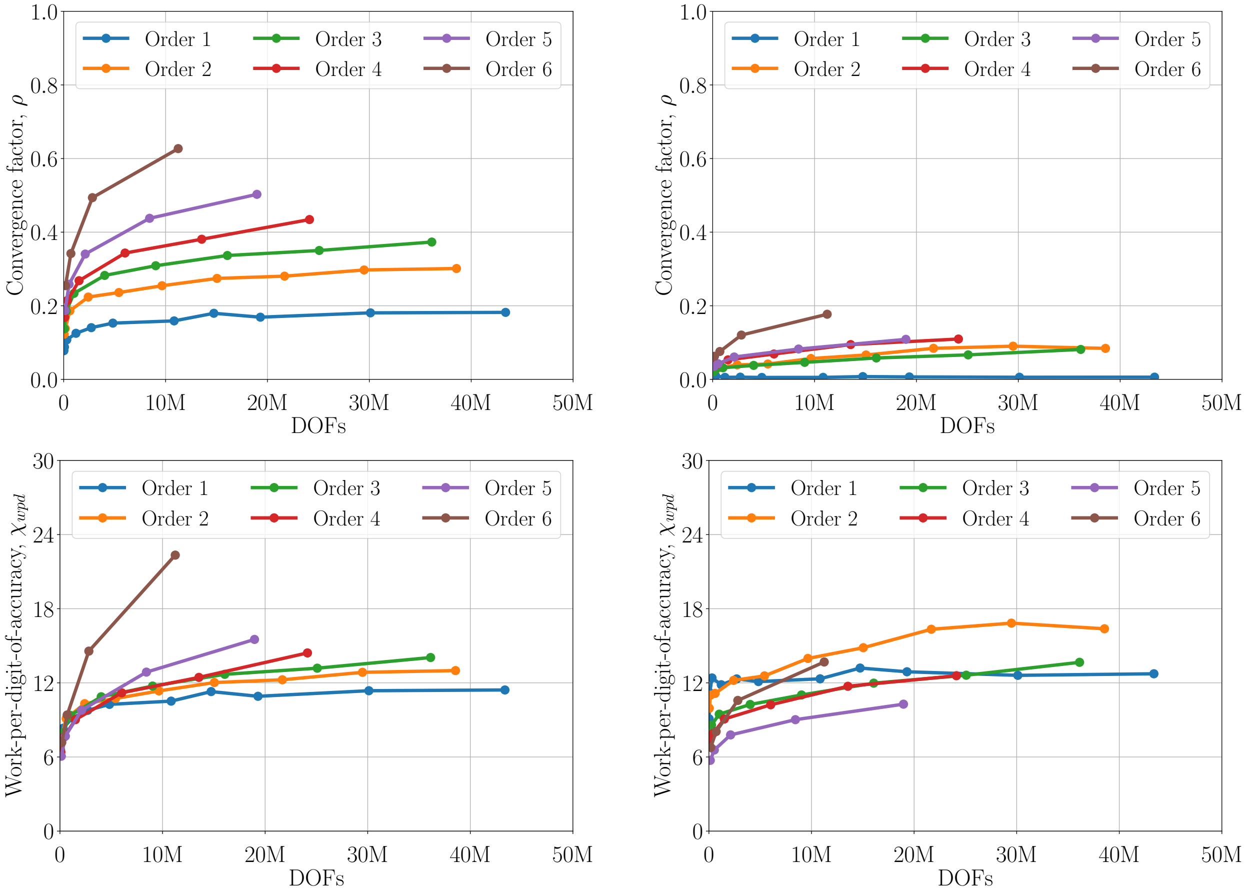


Figure 8.5: Work-per-digit of accuracy as a function of truncation tolerance for AMGir applied to various discretizations of the steady state transport equation. The dotted block line shows $\varphi = 10^{-3}$, which is an effective choice for all problems tested.

8.4.0.3 Scaling in h and element order

The final numerical study presented here is the scaling of AMGir with respect to DOFs, as well as finite-element degree. One of the exciting features of AMGir is its ability to solve high-order finite-element discretizations, something that AMG methods often struggle with. This is done in two parts: first, a finite-element degree is fixed and the performance of AMGir is examined as spatial resolution tends to zero; and, second, the spatial resolution is fixed and AMGir is applied to finite element discretizations of degree $d = 1, ..., 6$. Two cycle types will be considered: a V-cycle and an F-cycle. An F-cycle consists of first restricting all the way to the coarsest grid (as in a V-cycle) without relaxing along the way, followed by performing an additional V-cycle at each level of the hierarchy when traversing back to the finest grid. The complexity of an F-cycle is typically 1.5–3 times that of a V-cycle. Although F-cycles originate in full multigrid and full AMG, which focus on achieving discretization-level accuracy in a single F-cycle, accuracy with respect to discretization is not considered in this work. Instead, the F-cycle is used as it can provide more robust convergence and scaling than a V-cycle, at a much lower cost than alternatives such as W-cycles and K-cycles.

Figure 8.6 shows AMGir scaling of work-per-digit-of-accuracy and convergence factor as a function of

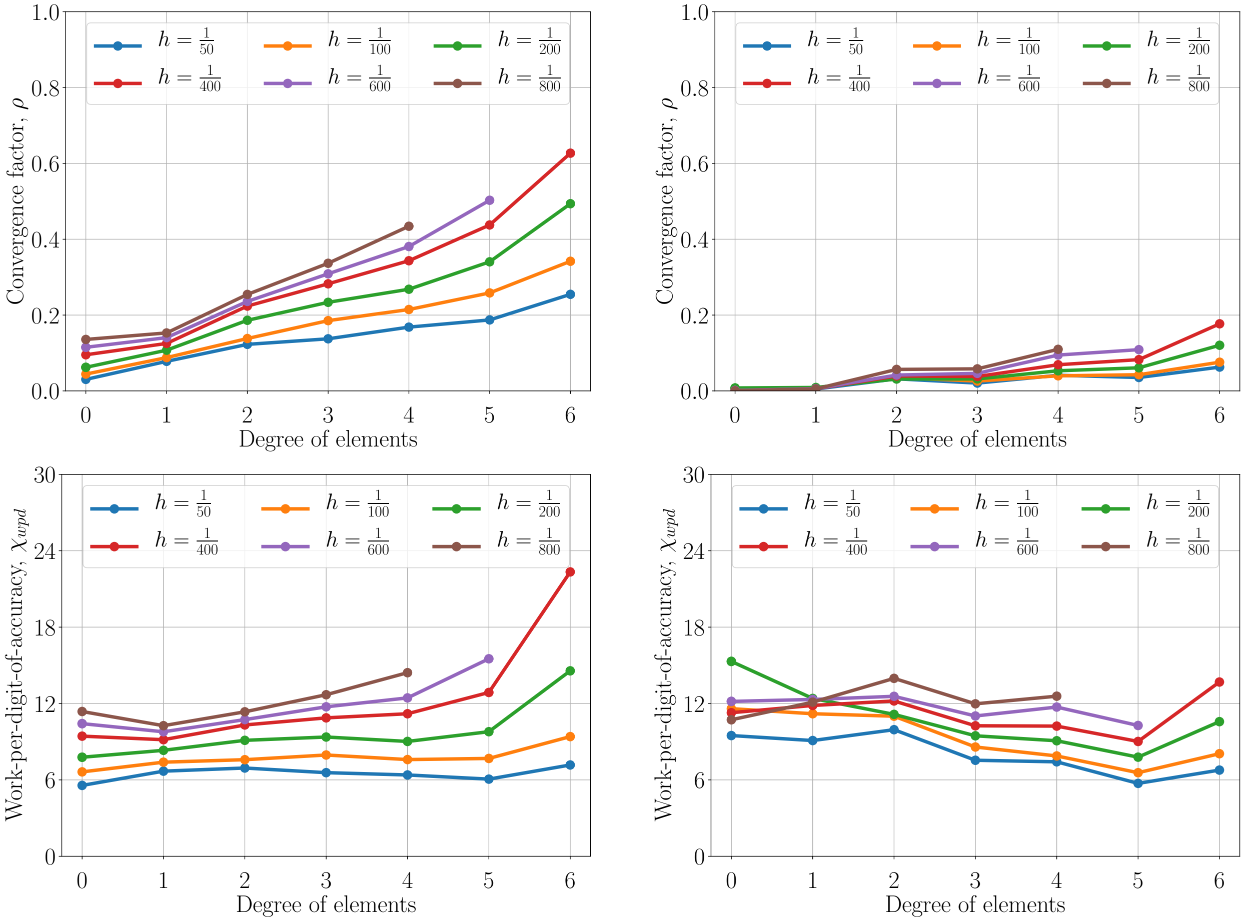


(a) V-cycle

(b) F-cycle

Figure 8.6: Scaling of convergence factor and work-per-digit-of-accuracy as a function of DOFs, for AMGir applied to upwind DG discretizations of the block-source problem, with finite-element degrees 1–6.

number of DOFs. Low-order elements have reached asymptotic behavior in V- and F-cycles, thus showing AMGir to be a scalable method up to ≈ 40M DOFs. Convergence factors and work-per-digit-of-accuracy are still slowly increasing for higher-order elements; however, memory constraints in the serial setting prevented scaling to larger problems. Figure 8.7 shows scaling of convergence and work-per-digit-of-accuracy as a function of finite element order. Notice that for F-cycles in particular, scaling with respect to work is near-perfect for all spatial resolutions tested. Based on Figure 8.6, it is unlikely for such perfect scaling to continue as $h \to 0$; nevertheless, it is encouraging that AMGir performs so well on high-order elements.



(a) V-cycle

(b) F-cycle

Figure 8.7: Scaling of convergence factor and work-per-digit-of-accuracy as a function of finite-element degree, for AMGir applied to upwind DG discretizations of the block-source problem, with spatial resolution $h = \frac{1}{50}$ to $h = \frac{1}{800}$.

Chapter 9

Ideal restriction part II: extensions and applications

9.1 Ideal transfer operators and AMG

In Chapter 8, a reduction-based AMG method was introduced based on approximating the ideal restriction operator. The approximation is based on a truncated Neumann expansion to A_{ff}^{-1}, which is effective for triangular or block-triangular matrices, but whose accuracy decreases when non-triangular components are introduced to the matrix. Ideal interpolation and restriction have come up regularly in this work so far in general theory, energy minimization, and AMGir. This chapter looks at the role of ideal transfer operators with respect to error and residual reduction, and develops a block spectral analysis of AMG error propagation that suggests F-relaxation with a good approximation to the ideal transfer operators should lead to a generally robust solver. A new approximation to ideal restriction is then developed, referred to as AIR, which limits to the Neumann expansion in the case of triangular matrices, but provides robust approximations for symmetric matrices as well. Numerical results on advection-diffusion-reaction equations, from strictly advective to diffusion dominated, demonstrate AIR to be a robust solver for symmetric and nonsymmetric problems.

9.1.1 Advection-diffusion-reaction

In terms of solver development, a good model problem to study AMG for nonsymmetric systems is the advection-diffusion-reaction equation, which comes up in fluid flow and particle transport equations, among others. Let κ be the diffusion coefficient, β a measure of the size of advection, and h the mesh spacing. Then $R_h := \frac{\beta h}{\kappa}$, often called the grid Reynold's number, is a measure of the numerical balance between advection and diffusion. For $R_h > 1$, the problem is advection-dominated. Under appropriate boundary conditions, the limit of the weak form as $\kappa \to 0$ is the purely hyperbolic steady-state transport equation, which is well-discretized by upwinding, resulting in a triangular or block-triangular matrix. Conversely, for $R_h < 1$, the resulting discretization is diffusion dominated, converging to an operator that numerically looks like a diffusion discretization as κ grows (i.e., the advection component is arbitrarily small and the matrix is effectively SPD). AMG (and many other iterative methods) are designed for elliptic, diffusion-like problems, and in many cases achieve excellent convergence rates for the diffusion-dominated case. Recently, a reduction-based AMG method was developed that is highly effective on the hyperbolic limit [109], but deteriorates when significant diffusion is introduced. The goal of this work is to bridge this gap and develop an AMG solver that is robust across the spectrum of diffusivity and, in particular, well-suited for a high-performance parallel implementation. Note that a scalar model problem is chosen here intentionally to isolate the effects of nonsymmetry. Linear systems with block structure resulting from a system of PDEs can be difficult for AMG, often requiring individual attention that is outside the scope of this work (for example, [36]).

There have been many efforts at developing robust iterative methods specifically for advection-diffusion-type problems. Geometric multigrid (GMG) methods for problems with advection are often based on the concept of semi-coarsening or line-relaxation in the direction of advection [131, 149, 187, 188]; however, such approaches have several drawbacks, including requiring a priori knowledge of the underlying problem and discretization, as well as limitations with respect to unstructured meshes and parallelism. Overall, robust convergence was obtained in [187] for a range of advection and diffusion, but the method employed a line Gauss-Seidel smoother in the x- and y-direction, which are strictly serial algorithms and not well-suited for parallel environments. Several efforts have been made to develop AMG algorithms that are robust for advection-dominated problems [10, 65, 94, 126, 185]. Generally strong convergence was obtained in [10] and [126]; however, the results in each were based on multigrid K- or W-cycles, which come with a substantially higher communication cost in parallel than a normal V-cycle, thus limiting their parallel efficiency. Analysis has also been done in the scope of Krylov methods, for example, [100, 160], and other algebraic solvers such as [12] and [97]. However, a solver that is robust across the spectrum of diffusivity and, in particular, scalable and parallelizable, has proven difficult to achieve.

9.1.2 Error and residual

Let $\overline{\mathbf{x}}$ be the exact solution to $A\mathbf{x} = \mathbf{b}$ and $\mathbf{x}$ some approximation. Two measures of convergence used in iterative methods are the error, $\mathbf{e} := \overline{\mathbf{x}} - \mathbf{x}$, and residual, $\mathbf{r} := \mathbf{b} - A\mathbf{x} = A\mathbf{e}$. Although eliminating error is typically the true goal of an iterative method, for nonsingular A, $\mathbf{e} = \mathbf{0}$ if and only if $\mathbf{r} = \mathbf{0}$. Since $\mathbf{r}$ is measurable in practice and $\mathbf{e}$ is not, it is worth considering both. Error-propagation and residual-propagation of coarse-grid correction take on the following operator form:

$$\mathbf{e}_{new} = \mathcal{E}\mathbf{e}_{old} := (I - P(RAP)^{-1}RA)\mathbf{e}_{old}, \tag{9.1}$$

$$\mathbf{r}_{new} = \mathcal{R}\mathbf{e}_{old} := (I - AP(RAP)^{-1}R)\mathbf{r}_{old}. \tag{9.2}$$

Note that these operators are similar: $\mathcal{E} = A^{-1}\mathcal{R}A$. In the case of a symmetric matrix, A, and Galerkin coarse grid, $R := P^T$, then $\mathcal{E} = \mathcal{R}^T$. In general, for symmetric A, the roles of restriction and interpolation are more-or-less interchangeable. Suppose A is symmetric and P_0 and R_0 are effective transfer operators at reducing the error in the ℓ^2-norm: $1 \gg \|I - P(RAP)^{-1}RA\| = \|I - AR^T(P^TAR^T)^{-1}P^T\|$. Then defining $R_1 := P_0^T$ and $P_1 := R_0^T$ make for effective transfer operators in reducing the residual. A similar result based on symmetry holds for the spectral radius. Although convergence is not necessarily measured in the ℓ^2- or spectral-sense, this is indicative that the same principles can be effective for building R *and* P for symmetric matrices. For nonsymmetric matrices, the relation between restriction and interpolation is less clear. In particular, for a nonsymmetric problem, if an AMG solver based on P_0 and R_0 effectively reduces the error, then an AMG solver based on $P_1 := R_0^T$ and $R_1 := P_0^T$ will effectively reduce the residual with respect to A^T. However, this does *not* indicate that a solver based on P_1 and R_1 will be effective when applied to A.

This work focuses on so-called ideal interpolation and ideal restriction operators. When considered in a reduction setting [90, 105, 109, 143], each of these operators is tightly coupled with an exact solve or effective relaxation on F-points. The idea behind F-relaxation is to improve the solution at F-points, and then distribute this accuracy to C-points via coarse-grid correction (ideal interpolation), or get an accurate coarse-grid correction at C-points and distribute this accuracy to F-points via F-relaxation (ideal restriction).

Here we consider some F-relaxation scheme where Δ is an approximation to A_{ff}^{-1}. Assuming F-points have been chosen such that A_{ff} is well-conditioned, then F-relaxation should be effective at reducing F-point residuals and/or errors. Residual propagation and error propagation for F-relaxation are given respectively by

$$\mathbf{r}^{(i+1)} = \begin{pmatrix} I - A_{ff}\Delta & 0 \\ -A_{cf}\Delta & I \end{pmatrix} \begin{pmatrix} \mathbf{r}_f \\ \mathbf{r}_c \end{pmatrix}, \tag{9.3}$$

$$\mathbf{e}^{(i+1)} = \begin{pmatrix} I - \Delta A_{ff} & -\Delta A_{fc} \\ 0 & I \end{pmatrix} \begin{pmatrix} \mathbf{e}_f \\ \mathbf{e}_c \end{pmatrix}. \tag{9.4}$$

For symmetric A, residual and error propagation are adjoints of each other. For nonsymmetric matrices, we can note that in relaxing *only* on A_{ff}, error propagation and residual propagation are similar: $I - \Delta A_{ff} = \Delta(I - A_{ff}\Delta)\Delta^{-1}$. This ensures that, asymptotically, our relaxation scheme on A_{ff} provides similar behavior on errors and residuals. However, the connection between error and residual reduction in practice is less clear, particularly when considering C-points and F-points, as in (9.3) and (9.4). Residual reduction is based on the column scaling of A and error reduction on the row-scaling. For results presented here, Δ corresponds to 1–2 iterations of Jacobi F-relaxation.

Moving forward, recall we assume a block form of A, P, and R, as given in Section 5.1 and (5.4).

9.1.3 Ideal interpolation and residual reduction

Suppose that the error after relaxation is in the range of interpolation, that is, $\mathbf{e}_{old} = P\mathbf{v}_c$ for some coarse-grid vector $\mathbf{v}_c$. Then coarse-grid correction yields

$$\mathbf{e}_{new} = \mathbf{e}_{old} - P\mathcal{K}^{-1}RAP\mathbf{v}_v = \mathbf{e}_{old} - P\mathbf{v}_c = \mathbf{0}.$$

Obviously, it is advantageous for relaxation to put error in the range of interpolation or, conversely, for interpolation to accurately represent relaxed error. This is the motivation for the interpolation definition used in classical AMG [111, 145]. A basic assumption in AMG is that C-points and F-points are chosen such that F-point relaxation can efficiently reduce the residual at F-points, that is, A_{ff} is well-conditioned, which is also the basis of compatible relaxation [19, 102]. Returning to the error, let $\mathbf{e}_c$ and $\mathbf{e}_f$ be the current error restricted to C-points and F-points, respectively, and $\mathbf{r} = \mathbf{b} - A\mathbf{x}$ the current residual. Since A_{ff} is assumed to be well-conditioned, at convergence of F-relaxation we have $\mathbf{r}_f = \mathbf{0}$, which implies

$$A_{ff}\mathbf{e}_f + A_{fc}\mathbf{e}_c = \mathbf{0} \quad \implies \quad \mathbf{e}_f = -A_{ff}^{-1}A_{fc}\mathbf{e}_c.$$

This is the basis for so-called *ideal interpolation,*

$$P_{\text{ideal}} = \begin{pmatrix} -A_{ff}^{-1}A_{fc} \\ I \end{pmatrix}, \tag{9.5}$$

which exactly represents the error in the range of interpolation.

Noting that ideal interpolation is based on an assumption of zero residuals at F-points, consider the effect of ideal interpolation on residual propagation of coarse-grid correction (9.2):

$$\begin{aligned} \mathbf{r}_f^{(i+1)} &= \mathbf{r}_f^{(i)} - (A_{ff}W + A_{fc})(RAP)^{-1}(Z\mathbf{r}_f^{(i)} + \mathbf{r}_c)^{(i)}, \\ \mathbf{r}_c^{(i+1)} &= \mathbf{r}_c^{(i)} - (A_{cf}W + A_{cc})(RAP)^{-1}(Z\mathbf{r}_f^{(i)} + \mathbf{r}_c)^{(i)}. \end{aligned}$$

For $P = P_{\text{ideal}}$, $W := -A_{ff}^{-1}A_{fc}$ and $RAP = A_{cc} - A_{cf}A_{ff}^{-1}A_{fc}$, independent of Z (Lemma 1, [109]). Then (9.2) reduces to

$$\mathbf{r}_{new} = \begin{pmatrix} \mathbf{r}_f \\ -Z\mathbf{r}_f \end{pmatrix}. \tag{9.6}$$

That is, ideal interpolation (i) eliminates the contribution of coarse-grid correction to the F-point residual, and (ii) eliminates the contribution of the previous C-point residual to the updated residual. This is consistent with the notion of preceding coarse-grid correction based on ideal interpolation with an exact F-point solve: we use F-relaxation to make $\mathbf{r}_f$ small (or zero for an exact solve) and follow with coarse-grid correction that does not change $\mathbf{r}_f$, but updates $\mathbf{r}_c$ with the new $\mathbf{r}_f$.

Looking at (9.6) suggests that $Z = \mathbf{0}$ is a good choice for restriction when coupled with P_{ideal}, as the residual at C-points is then eliminated with coarse-grid correction. In fact, if $Z = 0$, $RAP = A_{cf}W + A_{cc}$ and (9.2) results in an ℓ^2-orthogonal coarse-grid correction, meaning that $Z = \mathbf{0}$ is optimal in an ℓ^2-sense. In this case, the ℓ^2-orthogonal coarse-grid correction also eliminates C-point residuals. However, from a practical perspective, $Z = \mathbf{0}$ is often not the best choice. This is because $Z = \mathbf{0}$ does not focus on a decomposition of the error such that coarse-grid correction focuses on algebraically smooth modes, and, additionally, P_{ideal} is typically not obtained in practice. In this case, it is important that Z contributes to making RAP "close" to the Schur complement [184].

A theoretical understanding of ideal interpolation in two-grid convergence for SPD matrices and exactly how it is ideal can be found in [55]. In general, ideal interpolation may result in a dense W, and is impractical to form. However, the goal in much of AMG literature (aside from choosing a good C/F splitting) is to build a sparse approximation to P_{ideal}. If F-relaxation converges quickly, A_{ff} should be well-conditioned, and a sparse approximation to A_{ff}^{-1} is possible [20].

9.1.4 Ideal restriction and error reduction

The focus in AMG (and most multigrid methods) has long been the accuracy of interpolation. The above seems to suggest that the choice of restriction is less important than interpolation. For symmetric matrices, there are good reasons to take $R = P^T$: the coarse-grid matrix remains symmetric and, moreover, coarse-grid correction is then an orthogonal projection in the A-norm, which is typically what AMG convergence is measured in. In this setting, interpolation really is the determining factor in AMG convergence. For nonsymmetric problems, A no longer defines a valid norm, and the loss of orthogonality in coarse-grid correction is difficult to avoid. In this case, the choice $R = P^T$ is somewhat arbitrary, although in some cases this can be effective [145]. However, other choices are possible; for example, see [104, 108, 109, 130, 147, 184]. Here, we consider the fact that, as with interpolation, there is in a sense an "ideal" restriction operator. Recall that ideal interpolation was based on the residual; *ideal restriction* is instead based on the error.

One role of restriction can be seen as attenuating the effect of error components not in the range of interpolation. In geometric MG, full weighting acts as a filter, thereby reducing the high-frequency contamination of the resulting coarse grid correction [18]. When residuals are smooth, injection approximates full weighting. When red-black relaxation is used with a 5-point stencil, residuals are often smooth at red points and zero at black points. There, half injection approximates full weighting. In any case, let P be interpolation as in (5.4). Then the error $\mathbf{e}$ can be written as

$$\mathbf{e} = \begin{pmatrix} \mathbf{e}_f \\ \mathbf{e}_c \end{pmatrix} = \begin{pmatrix} W\mathbf{e}_c + \delta\mathbf{e}_f \\ \mathbf{e}_c \end{pmatrix},$$

where $\delta\mathbf{e}_f$ is the "contamination" of the F-point error. If $\delta\mathbf{e}_f = \mathbf{0}$, then the F-point error is in the range of interpolation, and coarse-grid correction is exact (see Section 9.1.3). Here, we focus on the effect of restriction on error-propagation of coarse-grid correction (9.1):

$$\begin{aligned}
\mathbf{e}^{(i+1)} &= \begin{pmatrix}\mathbf{e}_f^{(i)}\\ \mathbf{e}_c^{(i)}\end{pmatrix} - P(RAP)^{-1}RA\left[\begin{pmatrix}W\mathbf{e}_c^{(i)}\\ \mathbf{e}_c^{(i)}\end{pmatrix} + \begin{pmatrix}\delta\mathbf{e}_f^{(i)}\\ \mathbf{0}\end{pmatrix}\right]\\
&= \begin{pmatrix}\mathbf{e}_f^{(i)}\\ \mathbf{e}_c^{(i)}\end{pmatrix} - P(RAP)^{-1}RA\left[P\mathbf{e}_c^{(i)} + \begin{pmatrix}\delta\mathbf{e}_f^{(i)}\\ \mathbf{0}\end{pmatrix}\right]\\
&= \begin{pmatrix}\mathbf{e}_f^{(i)} - W\mathbf{e}_c^{(i)}\\ \mathbf{0}\end{pmatrix} - P(RAP)^{-1}RA\begin{pmatrix}\delta\mathbf{e}_f^{(i)}\\ \mathbf{0}\end{pmatrix}. \qquad (9.7)
\end{aligned}$$

Eliminating the contribution of $\delta\mathbf{e}_f$ to the coarse-grid right-hand side consists of setting $RA\begin{pmatrix}\delta\mathbf{e}_f\\ \mathbf{0}\end{pmatrix} = \mathbf{0}$ for all $\delta\mathbf{e}_f$. Doing so gives an exact correction at C-points (9.7) independent of the current error vector, and F-points are updated as $\mathbf{e}_f^{(i+1)} = \mathbf{e}_f^{(i)} - W\mathbf{e}_c^{(i)}$. Expanding RA, we have

$$\mathbf{0} = \begin{pmatrix}Z & I\end{pmatrix}\begin{pmatrix}A_{ff} & A_{fc}\\ A_{cf} & A_{cc}\end{pmatrix}\begin{pmatrix}\delta\mathbf{e}_f\\ \mathbf{0}\end{pmatrix} = (ZA_{ff} + A_{cf})\delta\mathbf{e}_f, \qquad (9.8)$$

which is satisfied by $Z = -A_{cf}A_{ff}^{-1}$. This leads to the *ideal restriction* operator,

$$R_{\text{ideal}} = \begin{pmatrix}-A_{cf}A_{ff}^{-1} & I\end{pmatrix}. \qquad (9.9)$$

In fact, R_{ideal} is the unique operator that gives an exact correction at C-points, independent of the interpolation operator [109]. Similar to the case of ideal interpolation, letting $W := \mathbf{0}$ gives an ℓ^2-orthogonal coarse-grid correction, although there are typically more effective choices of W in practice. Regardless, the exact correction at C-points obtained with R_{ideal} is best followed by F-relaxation to distribute the new accuracy at C-points to F-points.

9.1.5 Block-Analysis of AMG

Let Δ be some approximation to A_{ff}^{-1} defining our F-relaxation scheme, and W and Z some interpolation and restriction operators over F-points, respectively. Then error propagation of a two-level scheme with post F-relaxation takes the form

$$\begin{aligned}
\mathcal{E}\mathbf{e} &= \underbrace{\begin{pmatrix}I - \Delta A_{ff} & -\Delta A_{fc}\\ 0 & I\end{pmatrix}}_{F-relaxation}\underbrace{\begin{pmatrix}I - W\mathcal{K}^{-1}(ZA_{ff} + A_{cf}) & -W\mathcal{K}^{-1}(ZA_{fc} + A_{cc})\\ -\mathcal{K}^{-1}(ZA_{ff} + A_{cf}) & I - \mathcal{K}^{-1}(ZA_{fc} + A_{cc})\end{pmatrix}}_{Coarse-grid\ correction}\begin{pmatrix}\mathbf{e}_f\\ \mathbf{e}_c\end{pmatrix}\\
&= \begin{pmatrix}I - \Delta A_{ff} - \widehat{W}\mathcal{K}^{-1}(ZA_{ff} + A_{cf}) & -\Delta A_{ff} - \widehat{W}\mathcal{K}^{-1}(ZA_{fc} + A_{cc})\\ -\mathcal{K}^{-1}(ZA_{ff} + A_{cf}) & I - \mathcal{K}^{-1}(ZA_{fc} + A_{cc})\end{pmatrix}\begin{pmatrix}\mathbf{e}_f\\ \mathbf{e}_c\end{pmatrix},
\end{aligned}$$

where $\widehat{W} := (I - \Delta A_{ff})W - \Delta A_{fc}$ and $\mathcal{K} := RAP = ZA_{ff}W + ZA_{fc} + A_{cf}W + A_{cc}$. We refer to $\widehat{W}$ as the *effective interpolation*, because a little bit of algebra shows that we can expand error-propagation to take the form of an approximate LDU preconditioner for A:

$$\begin{aligned}\mathcal{E} &= I - M^{-1}A \\ &= \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} - \begin{pmatrix} I & \widehat{W} \\ 0 & I \end{pmatrix} \begin{pmatrix} \Delta & 0 \\ 0 & \mathcal{K}^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ Z & I \end{pmatrix} A.\end{aligned}$$

Here, the LDU preconditioner, M, can be collapsed:

$$\begin{aligned}M &= \begin{pmatrix} I & 0 \\ -Z & I \end{pmatrix} \begin{pmatrix} \Delta^{-1} & 0 \\ 0 & \mathcal{K} \end{pmatrix} \begin{pmatrix} I & -\widehat{W} \\ 0 & I \end{pmatrix} \\ &= \begin{pmatrix} \Delta^{-1} & A_{fc} + (A_{ff} - \Delta^{-1})W \\ -Z\Delta^{-1} & (Z\Delta^{-1} + A_{cf})W + A_{cc} \end{pmatrix}.\end{aligned}$$

In looking at convergence of preconditioners, it is useful to let $A = M + N$ and notice that $I - M^{-1}A = -M^{-1}N$. Here, N reduces to the following outer product:

$$\begin{aligned}N &= \begin{pmatrix} A_{ff} - \Delta^{-1} & -(A_{ff} - \Delta^{-1})W \\ Z\Delta^{-1} + A_{cf} & -(A_{cf} + Z\Delta^{-1})W \end{pmatrix} \\ &= \begin{pmatrix} A_{ff} - \Delta^{-1} \\ Z\Delta^{-1} + A_{cf} \end{pmatrix} \begin{pmatrix} I & -W \end{pmatrix}.\end{aligned}$$

Then,

$$\begin{aligned}(I - M^{-1}A)\mathbf{e} &= -\begin{pmatrix} I & \widehat{W} \\ 0 & I \end{pmatrix} \begin{pmatrix} \Delta & 0 \\ 0 & \mathcal{K}^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ Z & I \end{pmatrix} \begin{pmatrix} A_{ff} - \Delta^{-1} \\ Z\Delta^{-1} + A_{cf} \end{pmatrix} \begin{pmatrix} I & -W \end{pmatrix} \mathbf{e} \\ &= \begin{pmatrix} I - \Delta A_{ff} - \widehat{W}\mathcal{K}^{-1}(ZA_{ff} + A_{cf}) \\ -\mathcal{K}^{-1}(ZA_{ff} + A_{cf}) \end{pmatrix} \begin{pmatrix} I & -W \end{pmatrix} \mathbf{e}.\end{aligned}$$

Denoting $H := I - \Delta A_{ff} - \widehat{W}\mathcal{K}^{-1}(ZA_{ff} + A_{cf})$ and $G := -\mathcal{K}^{-1}(ZA_{ff} + A_{cf})$, then the null space and range of $I - M^{-1}A$ are given respectively by

$$\begin{aligned}\mathcal{N}(I - M^{-1}A) &= \begin{pmatrix} W \\ I \end{pmatrix} \overline{\zeta} \quad \forall\, \overline{\zeta} \in \mathbb{R}^{n_c}, \\ \mathcal{R}(I - M^{-1}A) &= \begin{pmatrix} H \\ G \end{pmatrix} \overline{\eta} \quad \forall\, \overline{\eta} \in \mathbb{R}^{n_f}.\end{aligned}$$

Note that the null space is just the range of interpolation because we have an exact coarse-grid solve. Eigenvectors of $I - M^{-1}A$ with nonzero eigenvalues must then take the form $\begin{pmatrix} H \\ G \end{pmatrix} \eta$ for some $\eta \in \mathbb{R}^{n_f}$.

Then the eigenvalue problem $(I - M^{-1}A)\mathbf{v} = \lambda\mathbf{v}$ takes the form

$$\begin{aligned}
\begin{pmatrix} H \\ G \end{pmatrix} \begin{pmatrix} I & -W \end{pmatrix} \mathbf{v} &= \lambda \mathbf{v}, \\
\begin{pmatrix} H \\ G \end{pmatrix} \begin{pmatrix} I & -W \end{pmatrix} \begin{pmatrix} H \\ G \end{pmatrix} \eta &= \lambda \begin{pmatrix} H \\ G \end{pmatrix} \eta, \\
\begin{pmatrix} H \\ G \end{pmatrix} (H - WG)\eta &= \begin{pmatrix} H \\ G \end{pmatrix} \lambda\eta.
\end{aligned}$$

Thus, eigenvalues of $I - M^{-1}A$ corresponding to the range are given by eigenvalues of

$$\begin{aligned}
H - WG &= I - \Delta A_{ff} - \widehat{W}\mathcal{K}^{-1}(ZA_{ff} + A_{cf}) + W\mathcal{K}^{-1}(ZA_{ff} + A_{cf}) \\
&= I - \Delta A_{ff} - (\widehat{W} - W)\mathcal{K}^{-1}(ZA_{ff} + A_{cf}) \\
&= (I - \Delta A_{ff}) + \Delta(A_{ff}W + A_{fc})\mathcal{K}^{-1}(ZA_{ff} + A_{cf}). \qquad (9.10)
\end{aligned}$$

Terms $(A_{ff}W + A_{fc})$ and $(ZA_{ff} + A_{cf})$ represent how close to ideal our given interpolation and restriction operators are. If E_{TG} is the two-grid error-propagation operator. then if W *or* Z are ideal, $\rho(E_{TG}) = 1 - \Delta A_{ff}$. Similarly, for an exact solve on F-points ($\Delta := A_{ff}^{-1}$) and general W and Z,

$$\rho(E_{TG}) = \rho\Big((A_{ff}W + A_{fc})\mathcal{K}^{-1}(Z + A_{cf}A_{ff}^{-1})\Big). \qquad (9.11)$$

An exact solve on F-points coupled with ideal interpolation *or* ideal restriction results in $\rho(E_{TG}) = 0$.

Looking at (9.10), the first term corresponds to F-relaxation and the second to coarse-grid correction. For SPD matrices, it is natural to make the coarse-grid correction term small when relaxation error is large, and vice versa. In the nonsymmetric setting, such an approach is not necessarily well motivated because error modes with large positive eigenvalues with respect to relaxation (i.e., error not effectively reduced by relaxation), could hypothetically be cancelled out by large negative eigenvalues in coarse-grid correction. Thus all we can definitively glean from (9.10) for convergence of AMG is: (i) it is important that A_{ff} be well conditioned and Δ a good approximation to A_{ff}^{-1}, and (ii) it is important to approximate the ideal operators. If $\rho(I - \Delta A_{ff}) \ll 1$, then (9.10) and (9.11) show that how well we approximate the ideal operators largely governs convergence. This motivates the development of a local approximation to the ideal restriction operator (AIR) that is cheap to compute, parallelizable, and, moreover, an accurate approximation to the action of R_{ideal}, which is introduced in Section 9.2.

Remark 16. *As will be seen in Section 9.3, focusing on approximations to both ideal restriction and ideal interpolation is not effective for diffusion-dominated problems. Instead, good convergence requires that we approximate ideal restriction, while taking a more traditional approach to interpolation where the range of interpolation contains low-energy modes not attenuated by relaxation.*

Due to the accuracy of a Neumann approximation for approximating the inverse of a triangular matrix [109] as well as the fact that the condition number of an advection discretization typically scales like $1/h$ *while diffusion scales like* $1/h^2$*, it makes sense that more accurate approximations to ideal operators are attained for advection-dominated problems. As can be seen in* (9.10)*, a sufficiently accurate approximation to* R_{ideal} *makes the choice of interpolation largely irrelevant, and vice versa. For diffusion-dominated problems, we cannot rely on such accurate approximations to ideal transfer operators. However, since diffusion-dominated also corresponds to increasingly symmetric matrices, it makes sense to bring in traditional AMG techniques, in particular including low-energy modes in the range of interpolation.*

9.2 Local approximate Ideal restriction (AIR)

9.2.1 Local approximate ideal restriction

As shown in Section 9.1.4, ideal restriction can be motivated through eliminating the contribution of error at F-points to the coarse-grid right-hand side. Thus, we will try to do this "locally" for each C-point (corresponding to a given row of R). For each ith C-point, a restriction neighborhood $\mathcal{R}_i$ (consisting of some "nearby" F-points) is chosen, and the idea is to choose restriction weights z_{ik} for each $k \in \mathcal{R}_i$ so that the effect of perturbing error at any $j \in \mathcal{R}_i$ on the residual at i is zero. Note that a unit change in error at point j changes the residual at point i by a_{ij} and the residual at each point k in $\mathcal{R}_i$ by a_{kj}. Requiring that restriction weights be defined so that the total effect of these changes on the residual at point i is zero gives the following equation:

$$a_{ij} + \sum_{k \in \mathcal{R}_i} z_{ik} a_{kj} = 0. \tag{9.12}$$

Solving (9.12) for all $j \in \mathcal{R}_i$ determines the ith row of Z, where $R = (Z, I)$, and is equivalent to setting $(RA)_{ik} = 0$ for all k such that $(i, k) \in \mathcal{R}_i$. This is simply setting RA equal to zero within a pre-determined

F-point sparsity pattern for R. Note that this can also be seen as directly approximating the action of R_{ideal} on F-points, where $R_{ideal}A = (\mathbf{0}, S_A)$, for Schur complement S_A. In either case, denoting indices of the sparsity pattern for the ith row of R as $\mathcal{R}_i = \{\ell_1, ..., \ell_{S_i}\}$, where $S_i = |\mathcal{R}_i|$ is the size of the sparsity pattern, the resulting linear system takes the form

$$\begin{pmatrix} a_{\ell_0\ell_0} & a_{\ell_1\ell_0} & \cdots & a_{\ell_{S_i}\ell_0} \\ a_{\ell_0\ell_1} & a_{\ell_1\ell_1} & \cdots & a_{\ell_{S_i}\ell_1} \\ \vdots & & \ddots & \vdots \\ a_{\ell_0\ell_{S_i}} & a_{\ell_1\ell_{S_i}} & \cdots & a_{\ell_{S_i}\ell_{S_i}} \end{pmatrix} \begin{pmatrix} z_{i\ell_0} \\ z_{i\ell_1} \\ \vdots \\ z_{i\ell_{S_i}} \end{pmatrix} = - \begin{pmatrix} a_{i\ell_0} \\ a_{i\ell_1} \\ \vdots \\ a_{i\ell_{S_i}} \end{pmatrix}. \tag{9.13}$$

For matrices with a $k \times k$ block structure, an equivalent system to (9.13) can be formed based on block connections, where a_{ℓ_i,ℓ_j} is a $k \times k$ block in the matrix A and z_{i,ℓ_j} a $k \times k$ block in R. Solving (9.13) for k right-hand sides determines all elements for R in block form. If A_{ff} is diagonally dominant, which should be the case given an appropriate CF-splitting, then (9.13) is nonsingular and has a unique solution. However, there have been situations on coarse levels in the hierarchy where a local linear system is singular, and (9.13) is formulated as a least-squares problem to pick the minimal norm solution. Although other solutions could be chosen, such as a solution with the minimum number of nonzeros, singular local systems are sufficiently rare that the choice of solution has a negligible effect on the resulting solver. We refer to the proposed method for building R as *local approximate ideal restriction* (AIR).

Remark 17 (Row scaling)**.** *In some cases, the row scaling of a matrix can cause problems for classical AMG interpolation formulae, leading to negative diagonal entries in the coarse-grid operator. Although such discretizations are not commonplace, it is worth pointing out that AIR is insensitive to row scaling. Suppose that the fine-grid matrix is scaled by some diagonal matrix, D: $\tilde{A} := DA$. Then let $\tilde{R}$ and $\tilde{Z}$ denote the corresponding local approximate ideal restriction operator and its F-block, respectively. Weights $\tilde{z}_{ik}$ are given by solving*

$$\tilde{a}_{ij} + \sum_{k\in\mathcal{R}_i} \tilde{z}_{ik}\tilde{a}_{kj} = d_i a_{ij} + \sum_{k\in\mathcal{R}_i} \tilde{z}_{ik} d_k a_{kj} = 0,$$

for all $(i,k) \in \mathcal{R}_i$, $i = 0, ..., n_c - 1$. This is satisfied by $\tilde{z}_{ik} := d_i z_{ik} d_k^{-1}$, where z_{ik} are the weights for A satisfying (9.12) and (9.13). It follows that $\tilde{Z} := D_c Z D_f^{-1}$ and $\tilde{R} := D_c R D^{-1}$. The resulting coarse-grid *operator is then defined as*

$$\tilde{R}\tilde{A}P = D_cRD^{-1}DAP = D_cRAP,$$

which is simply maintaining the fine-grid row scaling in the coarse-grid operator. In fact, looking at the error-propagation operator,

$$\begin{aligned} I - P(\tilde{R}\tilde{A}P)^{-1}\tilde{R}\tilde{A} &= I - P(D_cRAP)^{-1}D_cRA \\ &= I - P(RAP)^{-1}RA, \end{aligned}$$

we see that error propagation of two-grid coarse-grid correction is independent of row-scaling. The same idea can be applied in a recursive manner for a full multilevel hierarchy. This is a subtle feature of AIR that makes it robust for a wider class of problems and discretizations.

9.2.2 Comparison with Neumann series

In Chapter 8, ideal restriction was approximated in the context of an AMG algorithm targeting triangular or block-triangular matrices. For ease of of notation, assume that A is lower triangular with unit diagonal. Then $A_{ff} = I - L_{ff}$, for strictly lower triangular matrix L_{ff}, and A_{ff}^{-1} can be written as a finite Neumann expansion:

$$A_{ff}^{-1} = \sum_{i=0}^{n_f} L_{ff}^i.$$

Ideal restriction in Chapter 8 is then approximated using a kth order Neumann approximation for some $k \ll n_f$:

$$R = \begin{pmatrix} -A_{cf}\sum_{i=0}^{k} L_{ff}^i & I \end{pmatrix}. \tag{9.14}$$

For matrices with triangular structure, the kth order Neumann approximation is exact for entries within path distance k of the diagonal. In fact, this is equivalent to eliminating the contribution of F-point error within distance k to the coarse-grid right-hand side. For example, the Neumann expansion for $k = 0$ is given by $R = \begin{pmatrix} -A_{cf} & I \end{pmatrix}$. If A_{ff} is lower triangular, this is exactly eliminating the contribution of error at F-points distance one away from a given C-point:

$$RA = \begin{pmatrix} -A_{cf}(I - L_{ff}) + A_{cf} & I \end{pmatrix} = \begin{pmatrix} A_{cf}L_{ff} & I \end{pmatrix}.$$

Noting that L_{ff} is strictly lower triangular, $A_{cf}L_{ff}$ equals zero within the sparsity pattern of A_{cf}. Thus, for C-point C_i, this is equivalent to eliminating the contribution of F-points within distance one of C_i to the coarse-grid right-hand side at index i. For lower triangular matrices, this generalizes to any k, as well as the filtered Neumann approximations used in practice in Chapter 8. That is, Neumann approximations of ideal restriction are equivalent to AIR as proposed here in the case of lower triangular matrices.

A Neumann approximation to ideal restriction can also be used for a general (not necessarily lower triangular) matrix, but, in that case, even the diagonal of a truncated Neumann expansion is not necessarily exact. That is, when a matrix is not lower triangular in some ordering, a truncated Neumann approximation to A_{ff}^{-1} is likely to be much less accurate, and does not necessarily eliminate the contribution of F-point error to the coarse-grid right-hand side. Here, we recognize this as an important function of the restriction operator and have developed a general approach to approximate ideal restriction that eliminates the contribution of F-point error to the coarse-grid right-hand side. To demonstrate this, consider two model problems: (i) the steady-state transport equation considered in Chapter 8, and (ii) a streamline upwind Petrov-Galerkin (SUPG) discretization of an advection-dominated recirculating flow. Steady-state transport is scaled to be triangular in some ordering, while recirculating flow has small symmetric diffusion components, and advection is not triangular due to the recirculating velocity field. Table 9.1 gives a relative measure of how well we approximate the ideal operator using distance-one and -two Neumann and AIR for each problem. As expected, for the transport equation, AIR and Neumann perform identically. However, in the recirculating case, Neumann is hardly a more accurate approximation to ideal restriction than just letting $Z = \mathbf{0}$, while AIR retains a similar accuracy of approximation to the triangular case. Note that convergence factors of the resulting solvers are relatively consistent with the accuracy of approximating R_{ideal}.

	Problem	Neumann$_1$	Neumann$_2$	AIR$_1$	AIR$_2$
$\frac{\|Z+A_{cf}A_{ff}^{-1}\|_F}{\|A_{cf}A_{ff}^{-1}\|_F}$	Transport	0.46	0.11	0.46	0.11
	Recirculating	0.99	0.99	0.46	0.17

Table 9.1: Relative Frobenius error in approximating the F-block of ideal restriction using distance-one and -two Neumann and AIR approximations, for steady-state transport and a recirculating flow. Each problem has approximately 6000 DOFs, and classical AMG coarsening is used to form a CF-splitting.

Although AIR is a generalization of the Neumann approximation, the latter is an important contribution conceptually, as it provides insight into how AMGir is able to achieve strong convergence factors on difficult problems and high-order finite elements. In the triangular setting, it is relatively well understood how good convergence is obtained with AMGir (see Chapter 8.2). Despite the block analysis in Section 9.1.5, a good understanding of when and why AIR performs well is ongoing work. The equivalence is also interesting from an implementation perspective, as the two methods are quite different in that regard. In terms of floating point operations, the Neumann approach is cheaper in setup than AIR. However, the dense local solves are more amenable to a parallel implementation, as these can be done locally.

9.2.3 Filtering and lumping

A simple but effective technique for complexity reduction was used in Chapter 8 that consisted of eliminating relatively small entries from each row in the matrix, on every level in the hierarchy. For the steady-state transport equation considered there, it was found that entries could be eliminated rather greedily without causing a significant degradation in convergence; for example, for all test problems, entries were eliminated in row i that were smaller than $0.001 \cdot \max_j |a_{ij}|$. It is known that such an approach is typically not effective on diffusive matrices, prompting research into more advanced techniques for reducing the number of matrix nonzeros [13, 54, 176]. Here, we use a similar technique to the elimination used previously, but instead of actually eliminating entries, we add them to the diagonal in order to preserve the row sum. The concept of collapsing entries to the diagonal is one of the original pieces of AMG [15, 145], and proves to be a more robust technique than elimination when diffusion is introduced.

As an example, we test elimination and lumping on the discontinuous Galerkin (DG) discretization of advection-diffusion-reaction considered in Section 9.3.2. Table 9.2 shows the average convergence factor (CF) and so-called work-unit-per-digit-of-accuracy, denoted WPD, for various complexity reduction strategies. The WPD is introduced in detail with numerical results in Section 9.3; for now, consider it as a linear measure of time to solution, that is, dropping from 20 WPD to 10 is a 2× speedup in the solver.

A lumping tolerance of 0.001 has proven to be an effective choice for all problems we have tested. Solver complexity is typically reduced 10–50%, and convergence factors increase a small amount or none.

κ	None		Lumping						Elimination					
θ_D	0		0.001		0.01		0.1		0.001		0.01		0.1	
	WPD	CF	WPD	CF	WPD	CF	WPD	CF	WPD	CF	WPD	CF	WPD	CF
10^{-10}	12.2	0.37	9.5	0.38	9.1	0.38	11.3	0.49	9.6	0.38	9.1	0.38	12.7	0.53
10^{-7}	12.1	0.37	9.3	0.37	8.5	0.36	11.8	0.51	9.7	0.38	8.8	0.37	13.2	0.55
10^{-4}	20.1	0.51	17.7	0.51	18.0	0.56	114.7	0.93	17.9	0.52	27.3	0.68	267.9	0.97
10^{-1}	38.0	0.56	32.8	0.57	42.1	0.71	DNC		94.6	0.82	223.7	0.94	DNC	
10	45.0	0.61	38.4	0.62	48.9	0.74	DNC		105.7	0.84	179.6	0.92	DNC	

Table 9.2: Results for various complexity-reduction techniques applied to a DG discretization of advection-diffusion-reaction with diffusion coefficient κ and elimination/lumping tolerance θ_D. For all cases tested, lumping entries in row i smaller than $0.001 \cdot \max_j |a_{ij}|$ to the diagonal results in convergence factors approximately equal to those achieved without lumping/elimination, while decreasing the time to solution by $10-25\%$. DNC denotes that the iterations did not converge.

For all AIR results presented, all matrices in the hierarchy are lumped with tolerance 0.001. Although this is not a fundamental part of the solver, it provides a nice reduction in complexity.

9.3 AIR applied to advection-diffusion-reaction

The model problem considered is the advection-diffusion-reaction equation,

$$-\nabla \cdot \kappa \nabla u + \beta(x,y) \cdot \nabla u + \sigma u = f \quad \text{in } \Omega, \tag{9.15}$$

defined over a convex domain, Ω, with Lipschitz continuous boundary, Γ. This allows us to focus on a broad class of PDEs and the effects of nonsymmetry on the solver. Specifically, we take two cases of (9.15). First, we consider a divergence-free recirculating flow with Dirichlet boundary conditions, discretized using an SUPG method. This problem comes up fairly often, and is a good initial test; however, in the hyperbolic limit of no diffusion, the recirculating flow problem is not well-posed. In this sense, the recirculating flow can be thought of as a diffusion problem with added advection. In order to consider the hyperbolic limit (and most nonsymmetric case), Section 9.3.2 then considers an upwind discontinuous Galerkin (DGu) discretization of an advection-diffusion-reaction equation with Dirichlet inflow boundaries, Neumann outflow boundaries, and a velocity field β with no closed curves or stationary points [6, 171]. In the hyperbolic limit of no diffusion, this problem reduces to the steady-state transport equation, and can be thought of as steady-state transport with added diffusion. Further details on the discretization and results can be found in Section

9.3.1. Discretizations are generated using the Dolfin finite element package [103].

All results presented here use AMG V-cycles as a preconditioner for GMRES. Note that in Chapter 8, no Krylov acceleration was used because Krylov acceleration did not improve convergence in the strictly advective case, but increased wall-clock times due to additional floating-point operations and memory requirements. Because the error-propagation operator of AMGir was shown to be nilpotent in the strictly advective case, it makes sense that GMRES would not offer significant improvement in convergence. Here, we introduce diffusion to the problem, the error-propagation operator is no longer nilpotent, and Krylov acceleration tends to improve overall performance.

Coarsening is done using a classical AMG CF-splitting, with no second pass [145]. A second pass is not used because it is not well suited for parallel environments (for example, it is not used in Hypre [74]), and an algorithm amenable to parallelization is one of the important features of AIR. The lack of a second pass in coarsening to adjust C-points and F-points is accounted for through modified interpolation routines [42] that are used here. Strong connections for coarsening and the interpolation and restriction neighborhoods are determined using a classical strength of connection (SOC) based on a hard minimum:

$$\mathcal{N}_i = \Big\{ j \mid i \neq j, -a_{ij} \geq \theta \max_{k \neq i} |a_{ik}| \Big\},$$

for some tolerance θ, where $\mathcal{N}_i$ is the neighborhood of strong connections to node i. For degree-two neighborhoods in AIR, we only consider F-F-C connections, not F-C-C connections. This is consistent with the goal of approximating $A_{cf}A_{ff}^{-1}$ in R_{ideal}, which does not contain F-C-C connections. For coarsening, we use $\theta_C = 0.4$. It is interesting to note that tests indicate AIR and classical interpolation methods perform well with similar values of θ. Thus, strong connections for degree-one interpolation and restriction use $\theta_1 = 0.1$, and strong connections for degree-two interpolation and restriction use $\theta_2 = 0.2$. The larger θ for degree-two operators is meant to limit fill-in of the sparsity pattern.

All results based on AIR use one iteration of F-F-C Jacobi relaxation following coarse-grid correction, corresponding to two iterations of F-relaxation followed by one iteration of C-relaxation. Previously, in AMGir and in theory, we only discuss using F-relaxation. In fact, for almost all problems, F-relaxation is sufficient and adding one iteration of C-relaxation does not improve convergence. However, for a few cases, typically either higher-order finite elements and/or diffusion-dominated problems, one iteration of C-relaxation is necessary for good convergence. Thus, in order to demonstrate the method as robust with minimal parameter tuning, we use F-F-C relaxation for all results. Classical AMG does not perform well with just F-relaxation, requiring a global relaxation scheme for optimal results. To ensure the algorithms are similar from a comparative perspective, we run classical AMG methods with one pre- and post-relaxation sweep of weighted Jacobi.

In the results shown below, we denote the methods used to build restriction and interpolation, R_{build} and P_{build}, respectively, as a pair $(R_{\text{build}}, P_{\text{build}})$. AIR$_1$ refers to degree-one AIR and AIR$_2$ to degree-two AIR, and likewise for AMG$_1$ and AMG$_2$, specifically referring to modified classical interpolation (Eq. (4.4) in [42]) and Extended+i interpolation (Eqs. (4.10–4.11) in [43]), respectively. One-point interpolation introduced in Chapter 8.3, where each F-point is interpolated by value from its strongest C neighbor, is denoted 1P, and letting $R := P^T$ for a Galerkin coarse grid is denoted P^T. All problems are solved to a 10^{-12} relative residual tolerance with zero right-hand side and random initial guess. The approximate spatial mesh size is denoted h.

9.3.1 SUPG and Recirculating flow

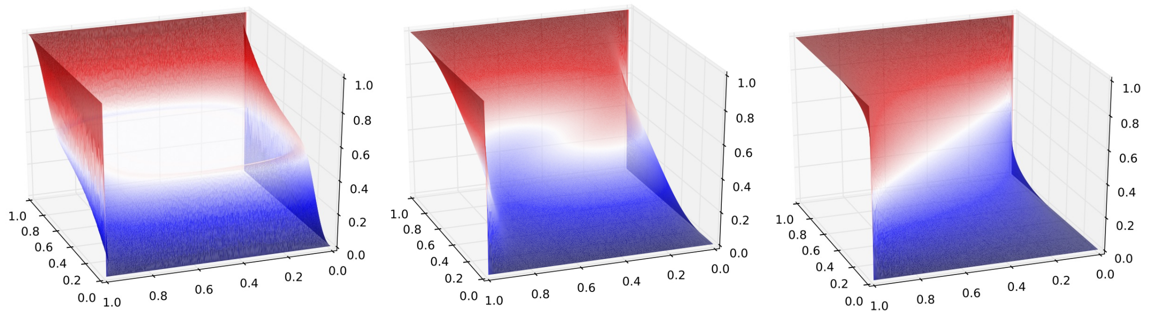
One of the most popular discretizations for flow-problems is a stabilized upwind discretization, in particular, the *streamline upwind Petrov-Galerkin* (SUPG) finite element discretization, where artificial numerical diffusion is added in the direction of the velocity field for stabilization purposes [32]. Our first test problem is a two-dimensional recirculating flow discretized with an SUPG discretization on a random, triangular, unstructured mesh. The continuous problem on domain Ω with boundary Γ, diffusion coefficient κ, and velocity field β is given as

$$\begin{aligned} -\nabla \cdot \kappa \nabla u + \beta \cdot \nabla u = f \qquad & \text{in } \Omega, \\ u = 0 \qquad & \text{on } \Gamma_0 \\ u = 1 \qquad & \text{on } \Gamma_1, \end{aligned}$$

where $\Omega = (0,1) \times (0,1)$, $\Gamma_1 = \{(x,y) \mid x = 1, y \in [0,1] \text{ or } y = 1, x \in [0,1]\}$, and $\Gamma_0 \cup \Gamma_1 = \Gamma$. The velocity field is a divergence-free recirculating flow given by

$$\beta(x,y) = \Big(x(1-x)(2y-1), -(2x-1)(1-y)y\Big).$$

The solution for varying levels of diffusion, κ, is shown in Figure 9.1.



(a) $\kappa = 10^{-4}$ (b) $\kappa = h = 0.005$ (c) $\kappa = 1$

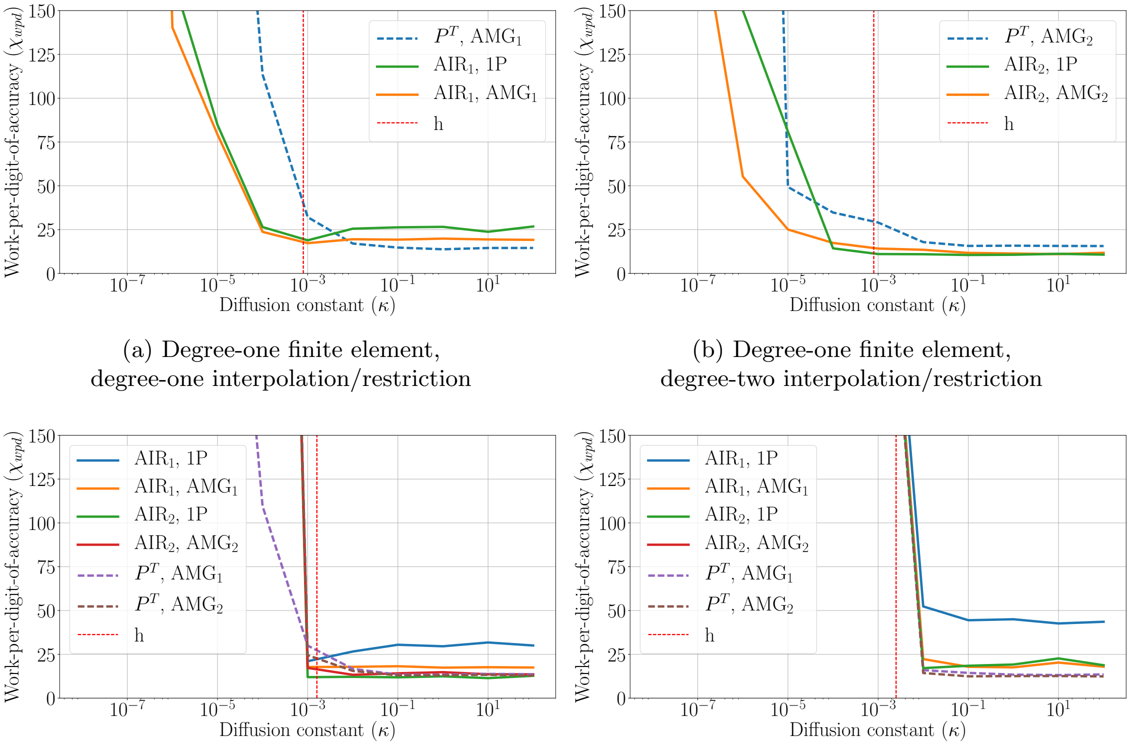
Figure 9.1: Solution of SUPG discretization of recirculating flow with varying diffusion coefficients, $\kappa \in \{10^{-4}, h, 1\}$, representing the advection-dominated, equal advection and diffusion, and diffusion-dominated cases, respectively.

Here we show the results of AIR and classical AMG applied to the recirculating flow. In this case, as $\kappa \to 0$, the problem is not well-posed and, in particular, the matrix becomes singular. Because of this, results are not expected to be good for $\kappa \approx 0$. As an example, Table 9.3 shows the approximate condition number of the matrices A and A_{ff} for $\kappa \in [10^{-10}, 100]$ and $h = \frac{1}{70}$. A pure diffusion discretization should have an approximate condition number of $\frac{1}{h^2}$, while a pure advection discretization should have a condition number of approximately $\frac{1}{h}$. For $\kappa = 1$, the conditioning of the SUPG discretization is close to the expected $\frac{1}{h^2}$, but as $\kappa \to 0$, $\text{cond}(A) \to \infty$ as opposed to $\frac{1}{h}$, indicating a singular matrix.

κ	10^{-10}	10^{-7}	10^{-5}	0.001	0.01	0.1	1
cond(A)	$7.1 \cdot 10^9$	$4.5 \cdot 10^9$	$1.23 \cdot 10^8$	$1.23 \cdot 10^6$	123,300	12,332	2,279
cond(A_{ff})	197,562	194,147	73,183	2,962	331	36	6.7

Table 9.3: Condition number of A and A_{ff} as a function of diffusion coefficient κ for degree-one finite elements on an unstructured mesh, with $h \approx \frac{1}{70}$ and 6300 DOFs (computed using NumPy [182]).

AIR performs well for reasonable values of κ. For linear finite elements (Figure 9.2), AIR slightly outperforms AMG for all κ in terms of WPD and, moreover, is able to effectively solve the problem for diffusion coefficients 1–2 orders of magnitude smaller than AMG, corresponding to matrix condition numbers likely two orders of magnitude larger (Table 9.3). Although results demonstrate AIR as a robust solver and an improvement over existing methods, we are not able to explore the highly nonsymmetric setting for which AIR is designed because the problem is not well-posed. This leads us to consider a time-dependent recirculating flow, where the linear system associated with pure advection is well posed (Section 9.3.1.1), followed by a different variation of steady-state advection-diffusion-reaction that is well-posed in the hyperbolic limit (Section 9.3.2).



(a) Degree-one finite element, degree-one interpolation/restriction

(b) Degree-one finite element, degree-two interpolation/restriction

(c) Degree-two finite elements

(d) Degree-three finite elements

Figure 9.2: WPD for AIR and classical AMG applied to an SUPG discretization of a recirculating flow on an unstructured mesh, using degree 1–3 finite elements. Spatial resolution is given by $h = \frac{1}{1250}, \frac{1}{625}, \frac{1}{400}$, respectively, leading to $\approx 2 \cdot 10^6$ DOFs for each problem. Classical AMG ($R := P^T$) results are shown in a dotted line, and variations in AIR in solid lines. As a reference, the typical convergence factor for χ_{wpd} of 20–25 WUs is $\rho \approx 0.5$.

9.3.1.1 Time-dependent recirculating flow

Thus far we have only considered the steady-state advection-diffusion equation which, in the case of a recirculating flow, is not well-posed for the purely advective case. However, an alternative approach is to consider the time-dependent advection-diffusion equation:

$$u_t - \nabla \cdot \kappa \nabla u + \beta \cdot \nabla u = f \qquad \text{in } \Omega, \tag{9.16}$$

with spatial boundary conditions as before and some initial condition, $u = u_0$ at time $t = 0$. As an example, consider using a first-order implicit backward Euler discretization in time and SUPG in space. Let $\mathcal{S}$ denote the discrete matrix associated with an SUPG spatial discretization of (9.16) and $u^{(i)}$ denote the solution at the ith time step with step size δt. Each time step then consists of solving the linear system

$$(I + \delta t \mathcal{S})u^{(i+1)} = u^{(i)} + \delta t f. \tag{9.17}$$

Table 9.4 shows the average convergence factor of AIR as applied to (9.17) for various parameter choices. Here, AIR proves to be an effective solver for implicit time-stepping of a recirculating flow, even with large time steps, pure advection, and higher-order finite elements. In practice, considerations must be taken for an appropriate combination of spatial and temporal discretization (e.g., CFL condition, h vs. δt, etc.); nevertheless, results indicate that AIR is a fast and robust solver for discretizations using implicit time-stepping. Interestingly with AIR, time-stepping in the advection-dominated regime is actually faster in all cases than the diffusion-dominated regime, sometimes significantly. It is worth noting that using solution $x^{(i)}$ as the initial guess to solve for $x^{(i+1)}$ gives an initial residual of $O(10^{-7})$, but the average convergence factor is the same with a random initial guess.

9.3.2 Upwind-DG and advection-diffusion-reaction

The second case of advection-diffusion-reaction that we will consider is a more general flow problem, written in conservative form [6, 171]. Let $\Gamma = \partial\Omega$ be the boundary of our domain, with inflow boundary

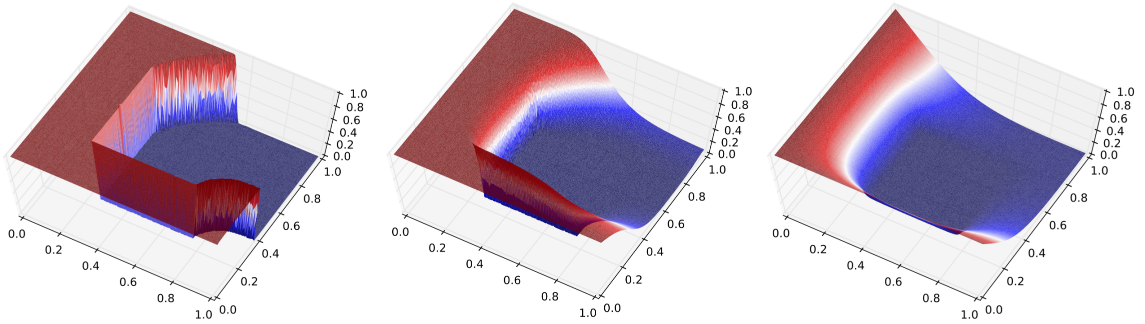
	Degree-one elements						Degree-two elements					
κ	0			1			0			1		
δt	dx^2	dx	$\sqrt{dx}$	dx^2	dx	$\sqrt{dx}$	dx^2	dx	$\sqrt{dx}$	dx^2	dx	$\sqrt{dx}$
CF	10^{-8}	0.01	0.29	0.1	0.38	0.49	10^{-8}	0.01	0.27	0.14	0.36	0.45
WPD	0.75	3.0	12.5	5.8	13.1	17.4	0.74	3.3	11.0	6.2	11.6	14.8

Table 9.4: Convergence factor (CF) of AIR applied to (9.17) with various choices of time step, diffusion coefficient, and initial guess, for $dx = \frac{1}{1000}$ and about $1.25 \cdot 10^6$ DOFs. For the advection-dominated case, even with a relatively large time step of $\delta t = \sqrt{dx}$, AIR achieves accuracy on the order of floating point precision in one iteration or, equivalently, for the computational cost of approximately five matrix-vector multiplies.

$\Gamma_{in} = \{x \in \Gamma \mid \beta(x) \cdot \mathbf{n}(x) < 0\}$ and outflow boundary $\Gamma_{out} = \{x \in \Gamma \mid \beta(x) \cdot \mathbf{n}(x) \geq 0\}$. Then consider

$$\begin{aligned} \nabla \cdot \sigma(u) + \gamma u &= f \qquad \text{in } \Omega \\ u &= g_D \qquad \text{on } \Gamma_{in} \\ -\kappa \nabla u \cdot \mathbf{n} &= g_N \qquad \text{on } \Gamma_{out}, \end{aligned}$$

where $\sigma := -\kappa \nabla u + \beta u$ is the physical flux. We make the additional assumptions that $\beta(x)$ has no closed curves and that $|\beta(x)| \neq 0$ for all $x \in \Omega$ to ensure that the hyperbolic limit of steady-state transport for $\kappa = 0$ is well-posed. In the spirit of steady-state transport, γ represents the *total cross section* and is taken to be piecewise constant over the domain, varying by multiple orders of magnitude, and representing the thickness of the background material. Specifically, we let our domain be $\Omega = (0, 1) \times (0, 1)$, with $\Gamma_D = \Gamma^-$



(a) $\kappa = 0$ (b) $\kappa = h = 0.005$ (c) $\kappa = 1$

Figure 9.3: Solution of DGu discretization of the specified advection-diffusion-reaction equation for varying diffusion coefficients, $\kappa \in \{0, h, 1\}$, representing the purely advective, equal advection and diffusion, and diffusion-dominated cases, respectively.

being the south and west boundaries and $\Gamma_N = \Gamma^+$ the north and east boundaries. Then, let $g_N(x,y) = 0$, $g_D(x,y) = 1$, and

$$\gamma(x,y) = \begin{cases} 10^4 & x,y \in (0.25, 0.75) \\ 10^{-4} & \text{otherwise} \end{cases}, \qquad \beta(x,y) = \Big(y^2, \cos(\pi x/2)^2\Big).$$

Such choices correspond to a curved velocity field, facing straight north at $y = 0$ and straight east at $x = 1$, with a total cross section that is thick ($\gamma \gg 1$) in a block in the center of the domain and thin ($\gamma \ll 1$) outside of this block. An upwind discontinuous Galerkin (DGu) formulation is used to discretize (Eq. 3.4 in [6]) on an unstructured mesh, and the solution for varying levels of diffusion is shown in Figure 9.3.

9.3.2.1 DG block structure

One of the unique features of a DG discretization is the inherent block structure associated with it, where each element comprises a set or block of DOFs in the matrix. In Chapter 8, the resulting linear system was scaled by the block-diagonal inverse, $A\mathbf{x} = \mathbf{b} \mapsto D_B^{-1}A\mathbf{x} = D_B^{-1}\mathbf{b}$, where D_B is the block-diagonal of A, in order to maintain a lower-triangular structure for the Neumann series approximation to A_{ff}^{-1}. In adding diffusion (that is, symmetric components to the matrix) and using generalized AIR that does not depend on a triangular structure to the matrix, it is not obvious that scaling by the block-diagonal inverse, or generally utilizing the block structure, is important. However, using the block structure significantly improves convergence of AIR for all advection-dominated problems. One possible explanation is the effect of block-diagonal scaling on the condition number of the matrix: [1]

κ	0	10^{-6}	10^{-4}	0.001	0.01	0.1	1	10	100
cond(A)	1164172	1074536	187700	92619	37836	7422	1132	413	684
cond($D_B^{-1}A$)	86.7	86.7	85.2	77.9	125	228	296	359	619

Table 9.5: Condition number of A and $D_B^{-1}A$ as a function of diffusion coefficient κ for degree-one finite elements on an unstructured mesh, with $h \approx \frac{1}{25}$ and 3030 DOFs (computed using NumPy [182]).

For advection-dominated problems, scaling by the block-diagonal inverse maps a near-singular matrix

[1] Due to the significant difference in conditioning of A and $D_B^{-1}A$, a detailed analysis connecting the finite element theory and linear algebra in this regard is likely in order, but outside the scope of this paper.

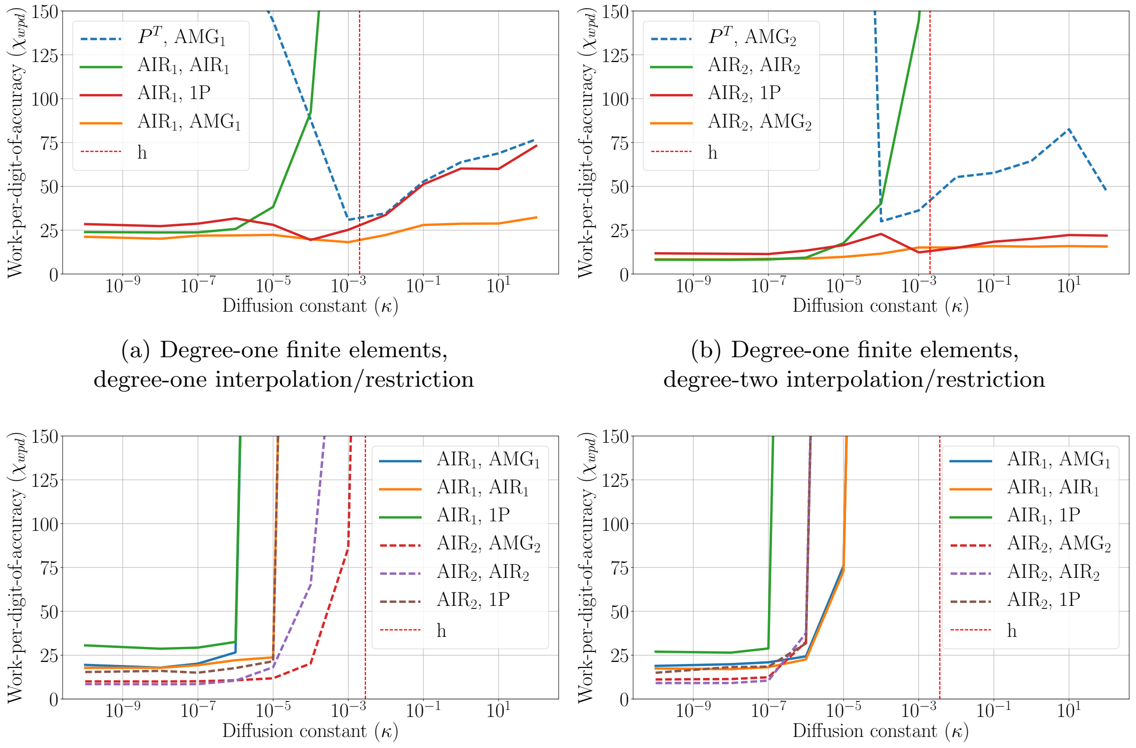
to one conditioned about as we would expect for a pure advection problem ($\frac{1}{h}$). Such improvement of conditioning of the linear system is good for iterative solvers in general. However, the importance to AIR is specifically that if $D_B^{-1}A$ is well-conditioned, then we should be able to pick F-points such that the resulting submatrix of F-F connections is also well-conditioned. Then (i) we should be able to form a good approximation to $-A_{cf}A_{ff}^{-1}$ in ideal restriction, and (ii) F-relaxation should converge well. The effect of scaling by D_B^{-1} on relaxation can be seen in Table 9.6, which shows the average convergence factor of Jacobi relaxation on the larger matrices, A and $D_B^{-1}A$, and the F-F submatrices, A_{ff} and $(D_B^{-1}A)_{ff}$. In the advection-dominated case, Jacobi relaxation does not converge on the submatrix A_{ff}, and diverges as applied to A. After scaling by D_B^{-1}, the pure advection case achieves convergence factors of ≈ 0.1 on the F-F block.

For diffusion-dominated problems, scaling by the block-diagonal inverse does not improve conditioning significantly, but convergence of Jacobi relaxation improves by a factor of about two. Scaling by D_B^{-1} in the diffusion-dominated case also maps a near-symmetric matrix to be nonsymmetric, which does not agree with classical AMG theory based on symmetric matrices. Perhaps due to these conflicting effects, tests confirm that for most diffusion-dominated problems, convergence of AMG and AIR are approximately the same when applied to A and $D_B^{-1}A$. As seen in the following sections, AMG and AIR both struggle with the diffusion-dominated case for higher-order finite elements and in three dimensions. However, it is well known that AMG can solve discretizations of scalar elliptic problems, including high-order DG discretizations using a different block approach [129]; here we are interested in developing a solver for nonsymmetric, advection-dominated problems, and include results on the diffusion-dominated case for completeness.

κ	0	10^{-8}	10^{-7}	10^{-6}	10^{-5}	10^{-4}	0.001	0.01	0.1	1	10	100
CF(A)	1.58	1.58	1.58	1.57	1.54	1.28	0.99	0.97	0.94	0.94	0.94	0.93
CF($D_B^{-1}A$)	0.95	0.95	0.95	0.95	0.94	0.91	0.89	0.88	0.89	0.89	0.88	0.88
CF(A_{ff})	1.00	1.00	1.00	1.00	0.98	0.94	0.83	0.65	0.67	0.70	0.71	0.73
CF($(D_B^{-1}A)_{ff}$)	0.12	0.33	0.38	0.47	0.47	0.51	0.52	0.49	0.54	0.50	0.50	0.50

Table 9.6: Average convergence factor of 50 iterations of Jacobi relaxation on A, A_{ff}, $D_B^{-1}A$, and $(D_B^{-1}A)_{ff}$. Degree-one finite elements are used on an unstructured mesh, with $h = \frac{1}{500}$, approximately $2 \cdot 10^6$ DOFs, and $\kappa \in [0, 100]$.

Block-matrix structure can be handled in a number of ways. Three natural approaches are (i) treating the matrix as is, without considering block structure, (ii) scaling the system by the block-diagonal inverse, and (iii) treating the entire AMG hierarchy nodally, that is, computing the SOC, CF-splitting, and transfer operators by block. If the block structure is not accounted for, no combinations of AIR and AMG converge in the advection-dominated case, which is the focus of this paper. Treating the system nodally leads to similar convergence factors to those obtained on the block-diagonally scaled matrix. However, wall-clock times are typically higher as well because a block neighborhood in AIR is larger than a scalar neighborhood, and solving the dense linear system for each row of R is more expensive. The block-diagonal-inverse scaling leads to the best WPD and wall-clock times for all problems and is used for results presented here.



(a) Degree-one finite elements, degree-one interpolation/restriction

(b) Degree-one finite elements, degree-two interpolation/restriction

(c) Degree-two finite elements

(d) Degree-three finite elements

Figure 9.4: WPD for AIR and classical AMG applied to a DGu discretization of advection-diffusion-reaction on an unstructured mesh and degree 1–3 finite elements. The spatial resolution for high-order finite element matrices was chosen so that all matrices have approximately the same number of degrees of freedom as the linear element matrix, that is, $h = \frac{1}{500}, \frac{1}{350}, \frac{1}{275}$, respectively, leading to $\approx 2 \cdot 10^6$ DOFs for each problem Classical AMG ($R := P^T$) did not converge for any κ for second- and third-order finite elements. As a reference, the typical convergence factor for χ_{wpd} of 20–25 WUs is $\rho \approx 0.5$.

9.3.2.2 AIR and convergence as a function of κ

Figure 9.4 shows WPD of AIR and classical AMG applied to the block-diagonally scaled system for $\kappa \in [10^{-10}, 100]$. For linear finite elements in Figures 9.4a and 9.4b, classical AMG performs reasonably well *only* for the diffusion-dominated case ($\kappa > h$), and approximating ideal operators for both R and P works well *only* for the advection-dominated case ($\kappa < h$). But, combinations of AIR and AMG such as (AIR$_1$,AMG$_1$) perform consistently well across the entire spectrum of diffusivity. In fact, even in the diffusion-dominated case for this problem, using AIR for restriction performs better than $R := P^T$.[2]

AIR also performs well with higher-order elements in the advection-dominated case. Figures 9.4c and 9.4d show the results of variations in AIR applied to degree-two and degree-three finite elements. For advection-dominated problems, the WPD is approximately equal to that for degree-one elements. Note that the actual convergence factors are larger for higher-order elements, but the CCs are lower, resulting in an approximately equal WPD.

9.3.2.3 Three-dimensional advection-diffusion-reaction

Moving from two-dimensional problems to three-dimensional problems can be difficult for solvers, so this section extends the DGu discretization of advection-diffusion-reaction to three dimensions. Here, we choose two sample variations in AIR, (AIR$_1$,AMG$_1$) and (AIR$_2$,1P), and show the WPD and convergence factor for $\kappa \in [10^{-10}, 100]$ and degree 1–3 finite elements in Figure 9.5. The key result here is that for advection-dominated problems, AIR performs equally well in three dimensions as in two, even for third-order finite elements.[3]

[2] Interestingly, classical AMG performs well for small problem sizes, but convergence degrades substantially as problem size increases. For example, for $h = \frac{1}{50}$, corresponding to about 20,000 DOFs, classical AMG achieves a WPD of less than 50 across the entire range of κ. However, this steadily grows as $h \to 0$ and, for the problems considered in Figure 9.4, classical AMG methods did not converge for $\kappa < 10^{-6}$.

[3] In three dimensions, classical AMG either required a WPD $\gg 500$ or did not converge for all κ, so results are not presented. Similar to higher-order finite elements in two-dimensions, all methods struggle with the diffusion-dominated case in 3d; however, it is well known that existing AMG methods are able to solve standard discretizations of three-dimensional diffusion.

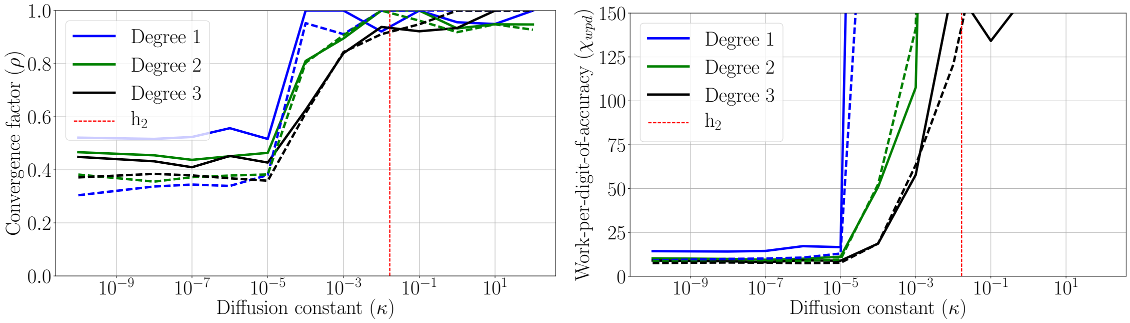


Figure 9.5: WPD and convergence factors for (AIR$_1$,AMG$_1$), solid line, and (AIR$_2$,1P), dotted line, applied to a DGu discretization of 3d advection-diffusion-reaction on an unstructured mesh, with approximately $2 \cdot 10^6$ DOFs. Mesh size, h, is given by $\{\frac{1}{85}, \frac{1}{60}, \frac{1}{45}\}$ for degree one, two and three finite elements (h is shown for degree-two).

Bibliography

[1] M L Adams and E W Larsen. Fast iterative methods for discrete-ordinates particle transport calculations. Progress in nuclear energy, 40(1):3–159, 2002.

[2] M P Adams, M L Adams, and W D Hawkins. Provably optimal parallel transport sweeps on regular grids. In International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering, pages 2535–2553, Idaho, 2013. Texas A and M University, College Station, United States.

[3] F L Alvarado and R Schreiber. Optimal parallel solution of sparse triangular systems. SIAM Journal on Scientific Computing, 14(2):446–460, 1993.

[4] S Amarala and J W L Wan. Multigrid Methods for Systems of Hyperbolic Conservation Laws. Multiscale Modeling & Simulation, 11(2):586–614, April 2013.

[5] Edward Anderson and Youcef Saad. Solving sparse triangular linear systems on parallel computers. International Journal of High Speed Computing, 1(01):73–95, 1989.

[6] Blanca Ayuso and L D Marini. Discontinuous Galerkin Methods for Advection-Diffusion-Reaction Problems. SIAM Journal on Numerical Analysis, 47(2):1391–1420, 2009.

[7] T S Bailey and R D Falgout. Analysis of massively parallel discrete-ordinates transport sweep algorithms with collisions. In International Conference on Mathematics, Computational Methods, and Reactor Physics, pages 1751–1765, Saratoga Springs, NY, 2009. Lawrence Livermore National Laboratory, Livermore, United States.

[8] A H Baker, T V Kolev, and U M Yang. Improving algebraic multigrid interpolation operators for linear elasticity problems. Numerical Linear Algebra with Applications, 17(2-3):495–517, 2010.

[9] Allison H Baker, R D Falgout, Tzanio V Kolev, and Ulrike Meier Yang. Scaling Hypre's Multigrid Solvers to 100,000 Cores. In High-Performance Scientific Computing, pages 261–279. Springer London, London, 2012.

[10] N R Bayramov and J K Kraus. Multigrid methods for convection–diffusion problems discretized by a monotone scheme. Comput. Methods Appl. Mech. Engrg., 317:723–745, April 2017.

[11] W. N. Bell, L. N. Olson, and J. B. Schroder. PyAMG: Algebraic multigrid solvers in Python v3.0, 2015. URL http://www.pyamg.org. Release 3.0.

[12] D Bertaccini, G H Golub, S S Capizzano, and C T Possio. Preconditioned HSS methods for the solution of non-Hermitian positive definite linear systems and applications to the discrete convection-diffusion equation. Numerische Mathematik, 99(3):441–484, December 2004.

[13] A Bienz, R D Falgout, W Gropp, and L N Olson. Reducing Parallel Communication in Algebraic Multigrid through Sparsification. SIAM Journal on Scientific Computing, 38(5):S332–S357, 2016.

[14] A Brandt. General highly accurate algebraic coarsening. Electronic transactions on numerical analysis, 10:1–20, 2000.

[15] A Brandt, S F McCormick, and J Huge. Algebraic multigrid (amg) for sparse matrix equations. Sparsity and its Applications, 257, 1985.

[16] A Brandt, J J Brannick, K Kahl, and I Livshits. Bootstrap AMG. SIAM Journal on Scientific Computing, 33(2):612–632, January 2011.

[17] A Brandt, J J Brannick, K Kahl, and Irene Livshits. Algebraic distance for anisotropic diffusion problems: multilevel results. Electronic transactions on numerical analysis, 44:472–496, 2015.

[18] Achi Brandt and Oren E Livne. Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics, Revised Edition. SIAM, 2011.

[19] J J Brannick and R D Falgout. Compatible relaxation and coarsening in algebraic multigrid. SIAM Journal on Scientific Computing, 32(3):1393–1416, 2010.

[20] J J Brannick and L T Zikatanov. Algebraic Multigrid Methods Based on Compatible Relaxation and Energy Minimization. In Domain Decomposition Methods in Science and Engineering XVI, pages 15–26. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[21] J J Brannick, M Brezina, S P MacLachlan, T A Manteuffel, S F McCormick, and J Ruge. An energy-based AMG coarsening strategy. Numerical Linear Algebra with Applications, 13(2-3):133–148, March 2006.

[22] J J Brannick, A Frommer, and K Kahl. Adaptive reduction-based multigrid for nearly singular and highly disordered physical systems. Electronic transactions on numerical analysis, 37:276–295, 2010.

[23] James Brannick, Yao Chen, and Ludmil Zikatanov. An algebraic multilevel method for anisotropic elliptic equations based on subgraph matching. Numerical Linear Algebra with Applications, 19(2): 279–295, 2012. ISSN 1099-1506. doi: 10.1002/nla.1804.

[24] James Brannick, Fei Cao, Karsten Kahl, Rob Falgout, and Xiaozhe Hu. Optimal interpolation and compatible relaxation in classical algebraic multigrid. arXiv preprint arXiv:1703.10240, 2017.

[25] M Brezina, R D Falgout, S P MacLachlan, T A Manteuffel, S F McCormick, and R Ruge. Adaptive Smoothed Aggregation (αSA). SIAM Journal on Scientific Computing, 25(6), June 2004.

[26] M Brezina, R D Falgout, S P MacLachlan, and T A Manteuffel. Adaptive smoothed aggregation (α SA) multigrid. SIAM review, 2005.

[27] M Brezina, T A Manteuffel, S F McCormick, J W Ruge, and G Sanders. Towards Adaptive Smoothed Aggregation (αSA) for Nonsymmetric Problems. SIAM Journal on Scientific Computing, 32(1):14–39, January 2010.

[28] Marian Brezina, Petr Vaněk, and Panayot S. Vassilevski. An improved convergence analysis of smoothed aggregation algebraic multigrid. Numerical Linear Algebra with Applications, 19(3):441–469, 2012. ISSN 1099-1506. doi: 10.1002/nla.775.

[29] F Brezzi, L D Marini, and E Süli. Discontinuous Galerkin methods for first-order hyperbolic problems. Mathematical models and methods in applied sciences, 14(12):1893–1903, 2004.

[30] N Brilliantov, J Schmidt, and F Spahn. Geysers of Enceladus: Quantitative analysis of qualitative models. Planetary and Space Science, 56(12):1596–1606, November 2008.

[31] N. V. Brilliantov and J. Schmidt. Aggregation kinetics in a flow: The role of particle-wall collisions. European Physical Journal Special Topics, 171:15–20, April 2009. doi: 10.1140/epjst/e2009-01006-X.

[32] A N Brooks and T JR Hughes. Streamline Upwind Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations. Comput. Methods Appl. Mech. Engrg., 32(1-3):199–259, 1982.

[33] L M Carvalho, L Giraud, and P Le Tallec. Algebraic two-level preconditioners for the Schur complement method. SIAM Journal on Scientific Computing, 22(6):1987–2005, 2001.

[34] Tony F. Chan and Petr Vanek. Detection of Strong Coupling in Algebraic Multigrid Solvers, pages 11–23. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000. ISBN 978-3-642-58312-4. doi: 10.1007/978-3-642-58312-4_2.

[35] Meng-Huo Chen and Anne Greenbaum. Analysis of an aggregation-based algebraic two-grid method for a rotated anisotropic diffusion problem. Numerical Linear Algebra with Applications, 22(4):681–701, 2015. ISSN 1099-1506. doi: 10.1002/nla.1980. nla.1980.

[36] T Clees. AMG Strategies for PDE Systems with Applications in Industrial Semiconductor Simulation. PhD thesis, 2005.

[37] G Colomer, R Borrell, F X Trias, and I Rodríguez. Parallel algorithms for Sn transport sweeps on unstructured meshes. Journal of Computational Physics, 232(1):118–135, January 2013.

[38] J E P Connerney. Magnetic fields of the outer planets. Journal of Geophysical Research, 98(E10): 18659–18679, October 1993.

[39] P D'Ambra and P S Vassilevski. Adaptive AMG with coarsening based on compatible weighted matching. Computing and Visualization in Science, 16(2), April 2013.

[40] John M. A. Danby. Fundamentals of Celestial Mechanics. Willmann-Bell, Inc., 2nd edition, 1988.

[41] M Davio. Kronecker Products and Shuffle Algebra. Ieee Transactions on Computers, 30(2):116–125, 1981.

[42] H De Sterck, U M Yang, and Jeffrey J Heys. Reducing Complexity in Parallel Algebraic Multigrid Preconditioners. SIAM Journal on Matrix Analysis and Applications, 27(4):1019–1039, January 2006.

[43] H De Sterck, R D Falgout, J W Nolting, and U M Yang. Distance-two interpolation for parallel algebraic multigrid. Numerical Linear Algebra with Applications, 15(2-3):115–139, 2008.

[44] Mehdi Dehghan and Masoud Hajarian. The general coupled matrix equations over generalized bisymmetric matrices. Linear Algebra and its Applications, 432(6):1531–1552, March 2010.

[45] F Deutsch. The angle between subspaces of a Hilbert space. NATO ASI Series C Mathematical and Physical . . . , (454):107–130, 1995.

[46] D.A. Di Pietro and A. Ern. Mathematical Aspects of Discontinuous Galerkin Methods. Mathématiques et Applications. Springer Berlin Heidelberg, 2011. ISBN 9783642229800.

[47] R. P. Di Sisto and M. Zanardi. Surface ages of mid-size saturnian satellites. Icarus, 264:90–101, January 2016. doi: 10.1016/j.icarus.2015.09.012.

[48] Jie Ding, Yanjun Liu, and Feng Ding. Iterative solutions to matrix equations of the form $A_i X B_i = F_i$. Computers and Mathematics with Applications, 59(11):3500–3507, June 2010.

[49] V Dobrev, T Kolev, and N A Petersson. Two-level convergence theory for parallel time integration with multigrid. SIAM Journal on Scientific Computing, 2016.

[50] Y Dong, T W Hill, and S Y Ye. Characteristics of ice grains in the Enceladus plume from Cassini observations. Journal of Geophysical Research: Space Physics, 120(2):915–937, February 2015.

[51] M. K. Dougherty, K. K. Khurana, F. M. Neubauer, C. T. Russell, J. Saur, J. S. Leisner, and M. E. Burton. Identification of a Dynamic Atmosphere at Enceladus with the Cassini Magnetometer. Science, 311:1406–1409, March 2006. doi: 10.1126/science.1120985.

[52] Howard C Elman, David J Silvester, and Andrew J Wathen. Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics. Oxford University Press (UK), 2014.

[53] V Faber, T A Manteuffel, and S V Parter. On the theory of equivalent operators and application to the numerical solution of uniformly elliptic partial differential equations. Advances in applied mathematics, 11(2):109–163, 1990.

[54] R D Falgout and J B Schroder. Non-Galerkin coarse grids for algebraic multigrid. SIAM Journal on Scientific Computing, 36(3):C309–C334, 2014.

[55] R D Falgout and P S Vassilevski. On Generalizing the Algebraic Multigrid Framework. SIAM Journal on Numerical Analysis, 42(4):1669–1693, January 2004.

[56] R D Falgout and U M Yang. hypre: A library of high performance preconditioners. European Conference on Parallel Processing, 2331 LNCS(PART 3):632–641, 2002.

[57] R D Falgout, P S Vassilevski, and L T Zikatanov. On two-grid convergence estimates. Numerical Linear Algebra with Applications, 12(5-6):471–494, 2005.

[58] R D Falgout, S Friedhoff, Tz V Kolev, S P MacLachlan, and J B Schroder. Parallel Time Integration with Multigrid. SIAM Journal on Scientific Computing, 36(6):C635–C661, January 2014.

[59] RD Falgout, T Manteuffel, B O'Neill, and JB Schroder. Multigrid Reduction in Time for Nonlinear Parabolic Problems. Numerical Linear Algebra with Applications, (in review).

[60] Hartmut Foerster, K Stüben, and Ulrich Trottenberg. Non-standard multigrid techniques using checkered relaxation and intermediate grids. Academic Press, New York, pages 285–300, 1981.

[61] Martin J Gander and S Vandewalle. Analysis of the Parareal Time-Parallel Time-Integration Method. SIAM Journal on Scientific Computing, 29(2):556–578, January 2007.

[62] P Gao, P Kopparla, X Zhang, and A P Ingersoll. Aggregate particles in the plumes of Enceladus. Icarus, 2016.

[63] Michael W. Gee, Jonathan J. Hu, and Raymond S. Tuminaro. A new smoothed aggregation multigrid method for anisotropic problems. Numerical Linear Algebra with Applications, 16(1):19–37, 2009. ISSN 1099-1506. doi: 10.1002/nla.593.

[64] J D Goguen, B J Buratti, R H Brown, R N Clark, P D Nicholson, M M Hedman, R R Howell, C Sotin, D P Cruikshank, K H Baines, K J Lawrence, J R Spencer, and D G Blackburn. The temperature and width of an active fissure on Enceladus measured with Cassini VIMS during the 14 April 2012 South Pole flyover. Icarus, 226(1):1128–1137, September 2013.

[65] H Guillard and P Vaněk. An aggregation multigrid solver for convection-diffusion problems on unstructured meshes. Technical report, 1998.

[66] C J Hansen, L W Esposito, A I F Stewart, J Colwell, A Hendrix, W Pryor, D Shemansky, and R A West. Enceladus' Water Vapor Plume. Science, 311(5):1422–1425, March 2006.

[67] C J Hansen, L W Esposito, A I F Stewart, B Meinke, B Wallis, J E Colwell, A R Hendrix, K W Larsen, W Pryor, and F Tian. Water vapour jets inside the plume of gas leaving Enceladus. Nature, 456(7):477–479, November 2008.

[68] C J Hansen, D E Shemansky, L W Esposito, A I F Stewart, B R Lewis, J E Colwell, A R Hendrix, R A West, J H Jr Waite, B Teolis, and B A Magee. The composition and structure of the Enceladus plume. Geophysical Research Letters, 38(1):L11202, June 2011.

[69] W Hawkins. Efficient massively parallel transport sweeps. In Transactions of the American Nuclear Society, pages 477–481. Texas A and M University, College Station, United States, December 2012.

[70] M M Hedman, P D Nicholson, M R Showalter, R H Brown, B J Buratti, and R N Clark. Spectral Observations of the Enceladus Plume with Cassini-Vims. The Astrophysical Journal, 693(2):1749–1762, March 2009.

[71] M M Hedman, C M Gosmeyer, P D Nicholson, C Sotin, R H Brown, R N Clark, K H Baines, B J Buratti, and M R Showalter. An observed correlation between plume activity and tidal stresses on Enceladus. Nature, 500(7):182–184, August 2013.

[72] P Helfenstein and C C Porco. Enceladus' Geysers: Relation to Geological Features. The Astronomical Journal, 150(3):96, September 2015.

[73] A R Hendrix, C J Hansen, and Greg M Holsclaw. The ultraviolet reflectance of Enceladus: Implications for surface composition. Icarus, 206(2):608–617, April 2010.

[74] V E Henson and U M Yang. BoomerAMG: A parallel algebraic multigrid solver and preconditioner. Applied Numerical Mathematics, 41(1):155–177, April 2002.

[75] M Hochbruck and G Starke. Preconditioned Krylov subspace methods for Lyapunov matrix equations. SIAM Journal on Matrix Analysis and Applications, 16(1):156–171, 1995.

[76] M Horányi. Charged dust dynamics in the solar system. Annual Review of Astronomy and Astrophysics, 34(1):383–418, 1996.

[77] M. Horányi, J. A. Burns, M. M. Hedman, G. H. Jones, and S. Kempf. Diffuse Rings. In Dougherty, M. K., Esposito, L. W., & Krimigis, S. M., editor, Saturn from Cassini-Huygens, pages 511–536. Springer, 2009. doi: 10.1007/978-1-4020-9217-6_16.

[78] C J A Howett, J R Spencer, J C Pearl, and M Segura. High heat flow from Enceladus' south polar region measured using 10-600 cm-1 Cassini/CIRS data. Journal of Geophysical Research, 116(E): E03003, March 2011.

[79] Hsiang-Wen Hsu, F Postberg, Yasuhito Sekine, Takazo Shibuya, S Kempf, M Horányi, Antal Juhász, N Altobelli, Katsuhiko Suzuki, Yuka Masaki, Tatsu Kuwatani, Shogo Tachibana, Sin-iti Sirono, Georg Moragas-Klostermeyer, and R Srama. Ongoing hydrothermal activities within Enceladus. Nature, 519 (7):207–210, March 2015.

[80] Xiaozhe Hu, Panayot S Vassilevski, and Jinchao Xu. A two-grid sa-amg convergence bound that improves when increasing the polynomial degree. Numerical Linear Algebra with Applications, 23(4): 746–771, 2016.

[81] T A Hurford, P Helfenstein, G V Hoppa, R Greenberg, and B G Bills. Eruptions arising from tidally controlled periodic openings of rifts on Enceladus. Nature, 447(7142):292–294, May 2007.

[82] T A Hurford, P Helfenstein, and J N Spitale. Tidal control of jet eruptions on Enceladus as observed by Cassini ISS between 2005 and 2007. Icarus, 220(2):896–903, August 2012.

[83] L. Iess, D. J. Stevenson, M. Parisi, D. Hemingway, R. A. Jacobson, J. I. Lunine, F. Nimmo, J. W. Armstrong, S. W. Asmar, M. Ducci, and P. Tortora. The Gravity Field and Interior Structure of Enceladus. Science, 344:78–80, April 2014. doi: 10.1126/science.1250551.

[84] T Imken, B Sherwood, J Elliott, A Frick, K McCoy, D Oh, P Kahn, A Karapetian, R Polit-Casillas, M Cable, J I Lunine, S Kempf, B S Southworth, S Tucker, and J Hunter Waite. Sylph-A SmallSat Probe Concept Engineered to Answer Europa's Big Question. In Conference on Small Satellites. AIAA/USU, 2016.

[85] A P Ingersoll and S P Ewald. Total particulate mass in Enceladus plumes and mass of Saturn's E ring inferred from Cassini ISS images. Icarus, 216(2):492–506, December 2011.

[86] A P Ingersoll and S P Ewald. Decadal timescale variability of the Enceladus plumes inferred from Cassini images. Icarus, 282(C):260–275, January 2017.

[87] A. P. Ingersoll, C. C. Porco, P. Helfenstein, R. A. West, and Cassini ISS Team. Models of the Enceladus Plumes. In AAS/Division for Planetary Sciences Meeting Abstracts #38, volume 38 of Bulletin of the American Astronomical Society, page 508, September 2006.

[88] A Janka, H Guillard, and P Vaněk. Convergence of algebraic multigrid based on smoothed aggregation ii: Extension to a petrov-galerkin method. Technical report, INRIA, 1999.

[89] R. Jaumann, R. N. Clark, F. Nimmo, A. R. Hendrix, B. J. Buratti, T. Denk, J. M. Moore, P. M. Schenk, S. J. Ostro, and R. Srama. Icy Satellites: Geological Evolution and Surface Processes, pages 637–681. 2009. doi: 10.1007/978-1-4020-9217-6_20.

[90] D Kamowitz and S V Parter. On MGR[ν] Multigrid Methods. SIAM Journal on Numerical Analysis, 24(2):366–381, 1987.

[91] S Kempf, U Beckmann, and J Schmidt. How the Enceladus dust plume feeds Saturn's E ring. Icarus, 206(2):446–457, April 2010.

[92] S. Kempf, N. Altobelli, C. Briois, T. Cassidy, E. Grün, M. Horanyi, F. Postberg, J. Schmidt, S. Shasharina, R. Srama, and Z. Sternovsky. Compositional Mapping of Europa's Surface with a Dust Mass Spectrometer. LPI Contributions, 1774:4052, February 2014.

[93] S.W. Kieffer, X. Lu, C.M. Bethke, J.R. Spencer, S. Marshak, and A. Navrotsky. A clathrate reservoir hypothesis for Enceladus' south polar plume. Science, 314(5806):1764–1766, December 2006.

[94] H H Kim, J Xu, and L T Zikatanov. A multigrid method based on graph matching for convection–diffusion equations. Numerical Linear Algebra with Applications, 2003.

[95] A. V. Krivov, H. Krüger, E. Grün, K.-U. Thiessenhusen, and D. P. Hamilton. A tenuous dust ring of Jupiter formed by escaping ejecta from the Galilean satellites. Journal of Geophysical Research (Planets), 107:5002, January 2002. doi: 10.1029/2000JE001434.

[96] Harald Krüger, Douglas P Hamilton, Richard Moissl, and Eberhard Grün. Galileo in-situ dust measurements in Jupiter's gossamer rings. Icarus, 203(1):198–213, September 2009.

[97] L A Krukier, T S Martinova, B L Krukier, and O A Pichugina. Special iterative methods for solution of the steady Convection-Diffusion-Reaction equation with dominant convection. In International Conference On Computational Science, pages 1239–1248. Elsevier Masson SAS, 2015.

[98] P. Lesaint and P.A. Raviart. On a finite element method for solving the neutron transport equation. C. de Boor (Ed.), Mathematical Aspects of Finite Elements in Partial Differential Equations, pages 89–123, 1974.

[99] Ruipeng Li and Y Saad. GPU-accelerated preconditioned iterative linear solvers. The Journal of Supercomputing, 63(2):443–466, 2013.

[100] J Liesen and Z Strakos. GMRES Convergence Analysis for a Convection-Diffusion Model Problem. SIAM Journal on Scientific Computing, 26(6):1989–2009, January 2005.

[101] W Liu, A Li, J Hogg, I S Duff, and B Vinter. A Synchronization-Free Algorithm for Parallel Sparse Triangular Solves. European Conference on Parallel Processing, 9833:617–630, 2016.

[102] O E Livne. Coarsening by compatible relaxation. Numerical Linear Algebra with Applications, 2004.

[103] Anders Logg and Garth N Wells. Dolfin: Automated finite element computing. ACM Transactions on Mathematical Software (TOMS), 37(2):20, 2010.

[104] J Lottes. Towards Robust Algebraic Multigrid Methods for Nonsymmetric Problems. Springer Theses. Springer International Publishing, Cham, 2017.

[105] S P MacLachlan, T A Manteuffel, and S F McCormick. Adaptive reduction-based AMG. Numerical Linear Algebra with Applications, 13(8):599–620, 2006.

[106] J Mandel. On block diagonal and Schur complement preconditioning. Numerische Mathematik, 58(1): 79–93, 1990.

[107] J Mandel, M Brezina, and P Vaněk. Energy Optimization of Algebraic Multigrid Bases. Computing, 62(3):205–228, June 1999.

[108] T A Manteuffel, L N Olson, J B Schroder, and B S Southworth. A root-node based algebraic multigrid method. SIAM Journal on Scientific Computing, (accepted), 2017.

[109] T A Manteuffel, S F McCormick, S Münzenmaier, J W Ruge, and B S Southworth. Reduction-based Algebraic Multigrid for Upwind Discretizations. SIAM Journal on Scientific Computing, submitted.

[110] T A Manteuffel, J W Ruge, and B S Southworth. Algebraic Multigrid Based on Local Approximate Ideal Restriction (LAIR) for Nonsymmetric Linear Systems. SIAM Journal on Scientific Computing, submitted.

[111] S F McCormick. An Algebraic Interpretation of Multigrid Methods. SIAM Journal on Numerical Analysis, 19(3):548–560, 1982.

[112] P Meier, H Kriegel, U Motschmann, J Schmidt, F Spahn, T W Hill, Y Dong, and G H Jones. A model of the spatial and size distribution of Enceladus' dust plume. Planetary and Space Science, 104: 216–233, December 2014.

[113] P Meier, U Motschmann, J Schmidt, and F Spahn. Modeling the total dust production of enceladus from stochastic charge equilibrium and simulations. Planetary and Space Science, 119:208–221, 2015.

[114] C Mense and R Nabben. On algebraic multi-level methods for non-symmetric systems – Comparison results. Linear Algebra and its Applications, 429(10):2567–2588, November 2008.

[115] S Míka and P Vaněk. Acceleration of convergence of a two-level algebraic algorithm by aggregation in smoothing process. 1992.

[116] J E Morel and J S Warsa. An S_n Spatial Discretization Scheme for Tetrahedral Meshes. Nuclear science and engineering, 151(2):157–166, October 2005.

[117] J E Morel and J S Warsa. Spatial Finite-Element Lumping Techniques for the Quadrilateral Mesh S_n Equations in X-Y Geometry. Nuclear science and engineering, 156(3):325–342, July 2007.

[118] M F Murphy, G H Golub, and A J Wathen. A note on preconditioning for indefinite linear systems. SIAM Journal on Scientific Computing, 21(6):1969–1972, 2000.

[119] A. L. Nahm and S. A. Kattenhorn. A unified nomenclature for tectonic structures on the surface of Enceladus. Icarus, 258:67–81, September 2015. doi: 10.1016/j.icarus.2015.06.009.

[120] Artem Napov and Yvan Notay. Algebraic analysis of aggregation-based multigrid. Numerical Linear Algebra with Applications, 18(3):539–564, 2011. ISSN 1099-1506. doi: 10.1002/nla.741.

[121] NASA. The SPICE Toolkit.

[122] F Nimmo, C Porco, and C Mitchell. Tidally Modulated Eruptions on Enceladus: Cassini ISS Observations and Models. The Astronomical Journal, 148(3):46, September 2014.

[123] Y Notay. A robust algebraic multilevel preconditioner for non-symmetric M-matrices. Numerical Linear Algebra with Applications, 7(5):243–267, 2000.

[124] Y Notay. Algebraic analysis of two-grid methods: The nonsymmetric case. Numerical Linear Algebra with Applications, 17(1):73–96, January 2010.

[125] Y Notay. An aggregation-based algebraic multigrid method. Electronic transactions on numerical analysis, 37:123–146, 2010.

[126] Y Notay. Aggregation-based algebraic multigrid for convection-diffusion equations. SIAM Journal on Scientific Computing, 34(4):A2288–A2316, 2012.

[127] L N Olson and J B Schroder. Smoothed aggregation for Helmholtz problems. Numerical Linear Algebra with Applications, 17:361–386, 2010.

[128] L N Olson and J B Schroder. A new perspective on strength measures in algebraic multigrid. Numerical Linear Algebra with Applications, 17(4):713–733, 2010.

[129] L N Olson and J B Schroder. Smoothed aggregation multigrid solvers for high-order discontinuous Galerkin methods for elliptic problems. Journal of Computational Physics, 230(1):6959–6976, August 2011.

[130] L N Olson, J B Schroder, and R S Tuminaro. A General Interpolation Strategy for Algebraic Multigrid Using Energy Minimization. SIAM Journal on Scientific Computing, 33(2):966–991, April 2011.

[131] C W Oosterlee, F J Gaspar, T Washio, and R Wienands. Multigrid Line Smoothers for Higher Order Upwind Discretizations of Convection-Dominated Problems. Journal of Computational Physics, 139 (2):274–307, January 1998.

[132] K D Pang, C C Voge, J W Rhoads, and J M Ajello. The E-Ring of Saturn and Satellite Enceladus. Journal of Geophysical Research, 89(NB11):9459–9470, 1984.

[133] H Park, D A Knoll, R M Rauenzahn, C K Newman, J D Densmore, and A B Wollaber. An Efficient and Time Accurate, Moment-Based Scale-Bridging Algorithm for Thermal Radiative Transfer Problems. SIAM Journal on Scientific Computing, 35(5):S18–S41, January 2013.

[134] C C Porco, P Helfenstein, P C Thomas, A P Ingersoll, J Wisdom, R A West, G Neukum, T Denk, R Wagner, T Roatsch, S W Kieffer, E Turtle, A McEwen, T V Johnson, J Rathbun, J Veverka, D Wilson, J Perry, J N Spitale, A Brahic, J A Burns, A Delgenio, L Dones, C D Murray, and S Squyres. Cassini observes the active South Pole of Enceladus. Science, 311(5766):1393–1401, 2006.

[135] C C Porco, D DiNino, and F Nimmo. How the Geysers, Tidal Stresses, and Thermal Emission across the South Polar Terrain of Enceladus are Related. The Astronomical Journal, 148(3):45, September 2014.

[136] C C Porco, L Dones, and C Mitchell. Could It Be Snowing Microbes on Enceladus? Assessing Conditions in Its Plume and Implications for Future Missions. Astrobiology, pages ast.2017.1665–26, August 2017.

[137] F Postberg, S Kempf, J Schmidt, N Brilliantov, A Beinsen, B Abel, U Buck, and R Srama. Sodium salts in E-ring ice grains from an ocean below the surface of Enceladus. Nature, 459(7):1098–1101, June 2009.

[138] F Postberg, J Schmidt, J K Hillier, S Kempf, and R Srama. A salt-water reservoir as the source of a compositionally stratified plume on Enceladus. Nature, 474(7):620–622, June 2011.

[139] W.H. Press, W.T. Vetterling, S.A. Teukolsky, and B.P. Flannery. Numerical Recipes in C++. Cambridge University Press, 3 edition, 2007.

[140] Lynnae C Quick, Olivier S Barnouin, Louise M Prockter, and G Wesley Patterson. Constraints on the detection of cryovolcanic plumes on Europa. Planetary and Space Science, 86:1–9, September 2013.

[141] J C Ragusa, J L Guermond, and G Kanschat. A robust SN-DG-approximation for radiation transport in optically thick and diffusive regimes. Journal of Computational Physics, 231(4):1947–1962, February 2012.

[142] W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. Tech. Rep. LA-UR-73-479, Los Alamos Scientific Laboratory, Los Alamos, NM, 1973.

[143] M Ries, U Trottenberg, and G Winter. A note on MGR methods. Linear Algebra and its Applications, 49:1–26, 1983.

[144] L. Roth, J. Saur, K.D. Retherford, D.F. Strobel, P.D. Feldman, M.A. McGrath, and F. Nimmo. Transient water vapor at Europa's south pole. Science, 343(6167):171–174, January 2014.

[145] J Ruge and K Stüben. Algebraic multigrid. Multigrid methods, 3(13):73–130, 1987.

[146] Y Saad and M Sosonkina. Distributed Schur complement techniques for general sparse linear systems. SIAM Journal on Scientific Computing, 21(4):1337–1356, 1999.

[147] M Sala and R S Tuminaro. A New Petrov–Galerkin Smoothed Aggregation Preconditioner for Nonsymmetric Linear Systems. SIAM Journal on Scientific Computing, 31(1):143–166, January 2008.

[148] J Saur, N Schilling, F M Neubauer, D F Strobel, S Simon, M K Dougherty, Christopher T Russell, and Robert T Pappalardo. Evidence for temporal variability of Enceladus' gas jets: Modeling of Cassini observations. Geophysical Research Letters, 35(20):1406–5, October 2008.

[149] S Schaffer. A semicoarsening multigrid method for elliptic partial differential equations with highly discontinuous and anisotropic coefficients. SIAM Journal on Scientific Computing, 20(1):228–242, 1998.

[150] Paul Schenk, D P Hamilton, R E Johnson, W B McKinnon, Chris Paranicas, J Schmidt, and M R Showalter. Plasma, plumes and rings: Saturn system dynamics as recorded in global color patterns on its midsize icy satellites. Icarus, 211(1):740–757, January 2011.

[151] J Schmidt, N Brilliantov, F Spahn, and S Kempf. Slow dust in Enceladus' plume from condensation and wall collisions in tiger stripe fractures. Nature, 451(7179):685–688, February 2008.

[152] J B Schroder. Smoothed aggregation solvers for anisotropic diffusion. Numerical Linear Algebra with Applications, 19(2):296–312, January 2012.

[153] F Scipioni, P Schenk, F Tosi, E D'Aversa, R N Clark, J Ph Combe, and C M Dalle Ore. Deciphering sub-micron ice particles on Enceladus surface. Icarus, 290:183–200, July 2017.

[154] M Sea d and A Klar. Efficient preconditioning of linear systems arising from the discretization of radiative transfer equation. In Challenges in scientific computing—CISC 2002, pages 211–236. Springer, Berlin, Berlin, Heidelberg, 2003.

[155] R. Shapiro and D. Schulze-Makuch. The Search for Alien Life in Our Solar System: Strategies and Priorities. Astrobiology, 9:335–343, May 2009. doi: 10.1089/ast.2008.0281.

[156] D E Shemansky, Y L Yung, X Liu, J Yoshii, Candice J Hansen, Amanda R Hendrix, and Larry W. Esposito. A new understanding of the europa atmosphere and limits on geophysical activity. ApJ, 797 (2):84, December 2014.

[157] S Simon, J Saur, H Kriegel, F M Neubauer, U Motschmann, and M K Dougherty. Influence of negatively charged plume grains and hemisphere coupling currents on the structure of Enceladus' Alfvén wings: Analytical modeling of Cassini magnetometer observations. Journal of Geophysical Research, 116(A): 4221, April 2011.

[158] V Simoncini and V Druskin. Convergence Analysis of Projection Methods for the Numerical Solution of Large Lyapunov Equations. SIAM Journal on Numerical Analysis, 47(2):828–843, January 2009.

[159] H T Smith, R E Johnson, M E Perry, D G Mitchell, R L McNutt, and D T Young. Enceladus plume variability and the neutral gas densities in Saturn's magnetosphere. Journal of Geophysical Research, 115(A10):n/a–n/a, October 2010.

[160] P Sonneveld and M B van Gijzen. IDR(s): A Family of Simple and Fast Algorithms for Solving Large Nonsymmetric Systems of Linear Equations. SIAM Journal on Scientific Computing, 31(2):1035–1062, January 2009.

[161] B S Southworth, S Kempf, and J Schmidt. Modeling Europa's dust plumes. Geophysical Research Letters, 42(2), December 2015.

[162] B S Southworth, S Kempf, J Schmidt, F Postberg, T Economou, and G Moragas-Klostermeyer. CDA encounters the Enceladus plume: Evidence for large particles. in preparation, 2017.

[163] B S Southworth, J J Brannick, S P MacLachlan, and J B Schroder. The Role of Energy Minimization in Algebraic Multigrid Interpolation. Technical report, Lawrence Livermore National Laboratory, in preparation.

[164] F. Spahn, N. Albers, M. Hörning, S. Kempf, A. V. Krivov, M. Makuch, J. Schmidt, M. Seiß, and M. Sremčević. E ring dust sources: Implications from Cassini's dust measurements. Planet. Space Sci., 54:1024–1032, August 2006. doi: 10.1016/j.pss.2006.05.022.

[165] F Spahn, J Schmidt, N Albers, M Horning, M Makuch, M Seiss, S Kempf, R Srama, V Dikarev, S Helfert, G Moragas-Klostermeyer, A V Krivov, M Sremčević, A J Tuzzolino, T Economou, and E Grün. Cassini dust measurements at Enceladus and implications for the origin of the E ring. Science, 311(5766):1416–1418, 2006.

[166] W B Sparks, B E Schmidt, M A McGrath, K P Hand, J R Spencer, M Cracraft, and S E Deustua. Active Cryovolcanism on Europa? The Astrophysical Journal Letters, 839(2), April 2017.

[167] J R Spencer, J C Pearl, M Segura, F M Flasar, A Mamoutkine, P Romani, B J Buratti, A R Hendrix, L J Spilker, and R M C Lopes. Cassini encounters Enceladus: Background and the discovery of a south polar hot spot. Science, 311(5766):1401–1405, 2006.

[168] J N Spitale and C C Porco. Association of the jets of Enceladus with the warmest regions on its south-polar fractures. Nature, 449(7):695–697, October 2007.

[169] J N Spitale and B S Southworth. Short-term varaibility in the enceladus plume. submitted, 2017.

[170] J N Spitale, T A Hurford, A R Rhoden, E E Berkson, and S S Platts. Curtain eruptions from Enceladus' south-polar terrain. Nature, 521(7550):57–60, May 2015.

[171] Shuyu Sun and Mary F Wheeler. Symmetric and Nonsymmetric Discontinuous Galerkin Methods for Reactive Transport in Porous Media. SIAM Journal on Numerical Analysis, 43(1):195–219, January 2005.

[172] Daniel B Szyld. The many proofs of an identity on the norm of oblique projections. Numerical Algorithms, 42(3-4):309–323, October 2006.

[173] The HDF Group. Hierarchical data format version 5, 2000-2010. URL http://www.hdfgroup.org/HDF5.

[174] P C Thomas, R Tajeddine, M S Tiscareno, J A Burns, J Joseph, T J Loredo, P Helfenstein, and C C Porco. Enceladus's measured physical libration requires a global subsurface ocean. Icarus, 264(C): 37–47, January 2016.

[175] F. Tian, A. I. F. Stewart, O. B. Toon, K. W. Larsen, and L. W. Esposito. Monte Carlo simulations of the water vapor plumes on Enceladus. Icarus, 188:154–161, May 2007. doi: 10.1016/j.icarus.2006.11.010.

[176] E Treister and I Yavneh. Non-Galerkin Multigrid Based on Sparsified Smoothed Aggregation. SIAM Journal on Scientific Computing, 37(1):A30–A54, January 2015.

[177] P Vaněk, J Mandel, and M Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. Computing, 56(3):179–196, 1996.

[178] P Vaněk, M Brezina, and J Mandel. Convergence of algebraic multigrid based on smoothed aggregation. Numerische Mathematik, 2001.

[179] P S Vassilevski. Multilevel Block Factorization Preconditioners. Matrix-based Analysis and Algorithms for Solving Finite Element Equations. Springer Science & Business Media, October 2008.

[180] P S Vassilevski. Lecture notes on multigrid methods. Lawrence Livermore National Laboratory, 2010.

[181] J. H. Waite, M. R. Combi, W.-H. Ip, T. E. Cravens, R. L. McNutt, W. Kasprzak, R. Yelle, J. Luhmann, H. Niemann, D. Gell, B. Magee, G. Fletcher, J. Lunine, and W.-L. Tseng. Cassini Ion and Neutral Mass Spectrometer: Enceladus Plume Composition and Structure. Science, 311:1419–1422, March 2006. doi: 10.1126/science.1121290.

[182] Stéfan van der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. Computing in Science & Engineering, 13(2):22–30, 2011.

[183] W L Wan, T F Chan, and B Smith. An energy-minimizing interpolation for robust multigrid methods. SIAM Journal on Scientific Computing, 21(4):1632–1649, 1999.

[184] T A Wiesner, R S Tuminaro, W A Wall, and M W Gee. Multigrid transfers for nonsymmetric systems based on Schur complements and Galerkin projections. Numerical Linear Algebra with Applications, 21(3):415–438, June 2013.

[185] C R Wu and H C Elman. Analysis and Comparison of Geometric and Algebraic Multigrid for Convection-Diffusion Equations. SIAM Journal on Scientific Computing, 28(6):2208–2228, January 2006.

[186] Li Xie, Yanjun Liu, and Huizhong Yang. Gradient based and least squares based iterative algorithms for matrix equations $AXB + CXTD = F$. Applied Mathematics and Computation, 217(5):2191–2199, November 2010.

[187] I Yavneh and M Weinzierl. Nonsymmetric Black Box multigrid with coarsening by three. Numerical Linear Algebra with Applications, 19(2):194–209, January 2012.

[188] I Yavneh, C H Venner, and A Brandt. Fast multigrid solution of the advection problem with closed characteristics. SIAM Journal on Scientific Computing, 19(1):111–125, 1998.

[189] L T Zikatanov. Two-sided bounds on the convergence rate of two-level methods. Numerical Linear Algebra with Applications, 15(5):439–454, 2008.