Robust Stability Analysis for Large-Scale Power Systems

by

Richard Yi Zhang

B.E. (Hons), University of Canterbury, New Zealand (2009)
S.M., Massachusetts Institute of Technology (2012)

Submitted to the Department of Electrical Engineering & Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2017

© Massachusetts Institute of Technology 2017. All rights reserved.

Author .....
Department of Electrical Engineering & Computer Science
October 31, 2016

Certified by .....
Jacob K. White
Cecil H. Green Professor of Electrical Engineering & Computer Science
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering & Computer Science
Chair, Department Committee on Graduate Theses

Robust Stability Analysis for Large-Scale Power Systems

by

Richard Yi Zhang

Submitted to the Department of Electrical Engineering & Computer Science
on October 31, 2016, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Innovations in electric power systems, such as renewable energy, demand-side participation, and electric vehicles, are all expected to increase variability and uncertainty, making stability verification more challenging. This thesis extends the technique of robust stability analysis to large-scale electric power systems under uncertainty. In the first part of this thesis, we examine the use of the technique to solve real problems faced by grid operators. We present two case studies: small-signal stability for distributed renewables on the IEEE 118-bus test system, and large-signal stability for a microgrid system. In each case study, we show that robust stability analysis can be used to compute stability margins for entire collections of uncertain scenarios.

In the second part of this thesis, we develop scalable algorithms to solve robust stability analysis problems on large-scale power systems. We use preconditioned iterative methods to solve the Newton direction computation in the interior-point method, in order to avoid the $O(n^6)$ time complexity associated with a dense-matrix approach. The per-iteration costs of the iterative methods are reduced to $O(n^3)$ through a hierarchical block-diagonal-plus-low-rank structure in the data matrices. We provide evidence that the methods converge to an ϵ-accurate solution in $O(1/\sqrt{\epsilon})$ iterations, and characterize two broad classes of problems for which the enhanced convergence is guaranteed.

Thesis Supervisor: Jacob K. White
Title: Cecil H. Green Professor of Electrical Engineering & Computer Science

Acknowledgments

My greatest appreciation goes to my advisor, Prof. Jacob White for his guidance and mentorship, for providing me the fertile environment and the endless encouragement to develop and to intellectually mature. Prof. White personifies the insatiable drive to learn new things and to solve interesting problems. I am truly fortunate to have had the opportunity to work with him.

My utmost gratitude also goes towards Prof. John G. Kassakian, my former advisor, and a committee member for this thesis. Working with Prof. Kassakian on the MIT Future of the Electric Grid study was an inspiring experience that set me on the path of my studies and research today. Prof. Kassakian is a charismatic leader and a masterful educator. I remain indebted to him for all his tireless support through the years.

I am deeply grateful to my committee members Prof. Konstantin Turitsyn and Dr. Eugene Litvinov, for their thought provoking questions and insightful remarks. Prof. Turitsyn provided me an invaluable sounding board for mathematical insights and potential impact, while Dr. Litvinov kept me grounded to the real problems at hand, to never lose sight of the bigger picture.

This thesis owes a large part to my collaborators, Jorge Elizondo Martinez, Mike Po-Hsu Huang, and Al-Thaddeus Avestruz, and my mentors at ISO New England, Kevin Feng Ma, Xiao-Chuan Luo, and Slava Maslennikov. The inspiration to apply modern control theory to power systems applications came largely out of extensive discussions with Jorge, Mike, and Al. The potential implications for the grid operator and the power systems engineer were greatly refined through conversations with Kevin, Xiao-Chuan, and Slava.

I owe a big "thank you" to all my friends in CPG, LEES, EECS, the GSC, the Energy Club, and the MIT Kiwi community, for making my MIT experience so unforgettable. I offer my most heartfelt gratitude to my closest friends Alex Guo, Wardah Inam, Ahmed Sheraz Malik, Leila Pirhaji, Jason Gao, László Miklós Lovász, David Jenicek, Daniel Day, Samantha Gunter, and Kendall Nowocin, all of whom are nothing short of family for me.

I dedicate this thesis to my parents, for their relentless support and unconditional love all throughout my life. I owe to them all that I am and all that I have accomplished.

Contents

Chapter 1

Introduction

The U.S. electric power grid is remarkably reliable, despite its massive size and complexity. The grid serves more than 143 million residential, commercial, and industrial customers through more than 6 million miles of transmission and distribution lines owned by more than 3,000 highly diverse investor-owned, government-owned, and cooperative enterprises. Yet power interruptions occur just 1-2 times a year, with all incidents combined to last 30 seconds to 5 minutes in urban centers [1].

Much of the reliability is achieved through a *preventative* paradigm. Potential major issues are forecast far in advance, and large safety factors are built into every aspect of system planning, design and operations. Such an approach is inherently pessimistic—not all issues can be predicted—but the pessimism has generally proved to be manageable in practice. Decades of operating experience, on the same systems and using the same equipment, have allowed grid operators to streamline the cost-reliability trade-off.

Today, the preventative paradigm is being actively challenged by the proliferation of new technologies, like renewable energy, electric vehicles, and demand-side participation. Grid operations are evolving towards a *reactive* paradigm, in which potential issues—big and small—are identified as they arise, and remedial actions are devised and enacted in real-time. Much of this transition is borne out of the need to accommodate for increasing variability and uncertainty: the output of wind and solar generators changes considerably over time and is imperfectly predictable; electric vehicles, demand response, and energy efficiency efforts can all greatly increase load variability, potentially boosting demand during select hours of the year. Under a preventative paradigm, holding the grid operator to the same reliability standard would cause electricity costs to escalate to unreasonable levels.

An essential requisite for grid operations under a reactive paradigm is the ability to verify stability under uncertainty. Existing stability techniques are deterministic; their sampling-based approach to uncertainty introduces a level of subjectivity that naturally lends to operations under a preventative paradigm. Instead, we examine robust stability analysis techniques, which explicitly treat uncertainty, and are able to make conclusive guarantees.

1.1 Stability under Uncertainty

A power system is said to be *stable* if, following a disturbance, it is able to recover to an acceptable equilibrium [2]:

- (Rotor angle stability) The generators connected to the system will remain in synchronization;
- (Voltage stability) The system voltages will recover to within a tolerance band of their nominal magnitudes; and
- (Frequency stability) The system frequency will remain within a tight bound of the nominal 60 Hz value.

Stability is always prioritized before economic considerations, because an unstable system has the potential to "run-away" or "run-down", possibly escalating into cascading outages and eventual shutdown of a major portion of the system.

Existing techniques for stability analysis (which we review in Chapter 2) are inherently deterministic. When presented with a large, possibly infinite, number of uncertain scenarios, it becomes necessary to select a subset of representative scenarios—to *sample*—using a mixture of experience, engineering intuition, and statistical arguments. Every power system is subject to an infinite number of load-generation profiles throughout a given year. However, for stability analysis, only characteristic profiles can be considered, typically the peak and light load conditions in summer and winter, perhaps with certain large generators out-of-service or important transmission lines congested.

Such a sampling-based approach to uncertainty is inconclusive and subjective by its very nature, but decades of operating experience have given engineers the deep intuition required to interpret the results meaningfully. Unfortunately, the introduction of renewables, electric vehicles, and demand-side participation erodes away much of this intuition. Stability issues are becoming more commonplace in areas with high penetrations of renewable energy like Texas [3–5] and Ireland, despite the extensive stability studies done on these systems [6–8].

At any rate, the credibility of stability analysis results still rests on the accuracy and faithfulness of the models considered. Historically, these models were often inaccurate. After the catastrophic WECC blackout of August 10, 1996, it was discovered that the recorded observations could not be recreated in simulation using the same models previously used to perform stability analysis; everything from the steady-state equilibrium to the dynamic behavior showed serious discrepancies with what was actually observed [9]. Today, high fidelity measurement units have been used to calibrate generation and transmission models, possibly in real-time [10], but load and renewable models remain over-simplified and inaccurate [3].

1.2 Robust Stability Analysis

Robust stability analysis is a set of techniques used to conclusively verify, or *certify*, the stability of models subject to uncertainty. A prominent example is the *quadratic*

stability test. Consider the linear parameter varying (LPV) model

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = A(\delta(t))x(t), \qquad \delta(t) \in \Delta, \tag{1.1}$$

in which the uncertain variable δ can take on any waveform that satisfies $\delta(t) \in \Delta$ for all time t. By construction, the LPV system has a single equilibrium at the origin $x = 0$. If we can find a symmetric positive definite coefficient matrix P that satisfies the convex constraint

$$A(\delta)^T P + PA(\delta) \text{ is negative definite for all } \delta \in \Delta, \tag{1.2}$$

then, by Lyapunov's Theorem, the LPV is *stable under uncertainty* or *robustly* stable: the solution $x(t)$ is guaranteed to converge to this equilibrium, starting from any initial condition $x(0)$, and over every time-varying parameter $\delta(t) \in \Delta$. The search for P—known as the quadratic stability certificate—can often be solved in polynomial time using an interior-point method.

The uncertain variable δ can be used to capture everything from modeling error, uncertain operating conditions, and nonlinearities. In all cases, the existence of a stability certificate guarantees stability under uncertainty. The conclusiveness of the approach explains its widespread use in diverse applications ranging from aerospace, automotive, defense, manufacturing, to other industries [11–14]. Building on top of this framework, controller tuning and synthesis can be performed by optimizing over all of the instances that are certifiably stable.

Robust stability tests are conservative by their nature, meaning that they can fail to certify a robustly stable model as being so. The conservatism arises in three components:

1. (Sufficiency) Robust stability tests are usually sufficient but not necessary for robust stability. For example, the existence of a quadratic stability certificate—a symmetric positive definite P satisfying (1.2)—is proof for robust stability, but the model may still be robustly stable even if such a P does not exist. Many explicit examples exist; Boyd *et al.* give a 2×2 case in [15, p.73].

2. (Reformulation) Robust stability tests are often posed in forms that cannot be directly verified. For example, the quadratic stability condition (1.2) involves an infinite number of constraints; it is impossible—except under very special circumstances—to verify whether a given P simulataneously satisfies all of these constraints. In some cases, the robust stability test can be reformulated into a finite number of constraints. These reformulations are usually sufficient conditions: they guarantee the stability test to hold true whenever a feasible point is found for the reformulation, but the stability test may hold true even if a feasible point does not exist.

3. (Uncertainty model) The uncertain model may capture more uncertainty that needed, and the additional uncertainty makes it more difficult for it to remain robustly stable. For example, the LPV model in (1.1) allows the uncertain

parameters $\delta(t)$ to vary arbitrarily quickly with time, but such rapid transitions in $\delta(t)$ may not be physically realizable.

The simple quadratic stability test described above has a reputation for being conservative [16–18]. More sophisticated tests, ranging from parameter-dependent Lyapunov functions to matrix sum-of-squares, offer better trade-offs between the three sources of conservertism, but at the cost of significantly increased computation. (The interested reader is referred to [12, 19] for surveys.)

Unfortunately, all robust stability tests scale poorly with the size of the models considered. Power system models are often considerably larger than those found in other applications, and from a practical perspective, robust stability analysis is computationally intractable. Even the simple quadratic stability test in (1.2) has a time complexity of $O(n^6)$ for a state-space model with n state variables, meaning that ten-fold increase in the number of state variables results in a million-fold increase in the amount of time required to verify its stability. Suppose it took just 1 second to solve (1.2) on an airplane model containing $n = 30$ state variables. Then it would take 24 hours to solve the same problems on a reduced-order model of a power system with $n = 200$ state variables, and 240 days to solve on a more realistic model with $n = 500$.

1.3 Main Contributions

In this thesis, we will focus our attention on a specific LPV model structure known as the polytopic LDI (linear differential inclusion)

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = Mx(t), \qquad M \in \mathrm{conv}\,\{M_1, \ldots, M_m\}. \tag{1.3}$$

As before, we will denote the number of state variables in (1.3) as n. It is well-known that the quadratic stability test in (1.2) can be reformulated into the problem of finding a symmetric positive definite matrix P satisfying

$$M_i^T P + P M_i \text{ is negative definite for all } i \in \{1, \ldots, m\}, \tag{1.4}$$

and solved using an interior-point method in $O(n^6 + mn^5)$ time and $O(n^4)$ storage. This resulting convex feasibility problem is often referred to as the *Lyapunov inequalities* problem.

In the first part of this thesis, we use the polytopic LDI framework to analyze the stability of power systems under uncertainty. Two case studies are considered. In the first case study, we examine the small-signal stability of an IEEE 118-bus model under the uncertainty of distributed renewables. More specifically, we retire 30% of the conventional generation, and replace their output by PV inverters installed at each of the 118 buses in the network. The PV output is left to be uncertain, and we use local optimization and quadratic stability tests to show that sampling-based statistical analysis can be remarkably misleading in characterizing the stability of a system under high dimensional uncertainty.

In our second case study, we use the same techniques to verify the stability of a microgrid subject to large-signal intermittency from a PV panel. We develop a suitable LPV model and derive large-signal stability margins for which stability would be guaranteed. In the process, we show that eigenvalue analysis—widely used to study stability under small-signal intermittency—is overly optimistic when the intermittency becomes large-signal.

In both case studies, the polytopic LDI framework is found to be a surprisingly effective tool for robust stability, in spite of its conservative reputation. The primary bottleneck is the computational power required to solve (1.4). This finding motivates us to investigate theoretical and computational methods to scale the same theory to larger, more realistically-sized problems.

In the second part of this thesis, we develop mixed first-second order methods to solve the Lyapunov inequalities problem (1.4) with worst-case complexity of $O(m^2n^4+(1/\sqrt{\epsilon})mn^3)$ time and $O(m^2n^2\log n)$ storage, where the accuracy tolerance ϵ is a measure of the difficulty of the feasibility problem. In practice, the algorithm has an average complexity closer to $O(n^4+(1/\sqrt{\epsilon})\sqrt{m}n^3)$ time and $O(n^2\log n)$ storage. Our method is driven by two important insights. First, all first-order method solution of (1.4) requires the solution of a system of equations with the following coefficient matrix

$$\mathbf{H}=\sum_{i=1}^{m}(M_i\otimes I+I\otimes M_i)^T(M_i\otimes I+I\otimes M_i) \tag{1.5}$$

at every iteration. When the data matrices $M_1,\ldots,M_m$ arise from linearizations of time-domain power system models, we show that they display a hierarchical structure that is then inherited by $\mathbf{H}$. In turn, the matrix can be factored in $O(n^4)$ time and the inverse applied in $O(\sqrt{m}n^3)$ time, thereby also reducing the per-iteration cost of first-order methods to $O(\sqrt{m}n^3)$.

Second, we show that first-order methods can be optimally accelerated using a Krylov subspace method when their update equations are linear, and that this motivates the use of a Krylov-accelerated first order method for the solution of the Newton subproblems associated with any interior-point method. We prove that both GMRES-accelerated ADMM and conjugate-gradients-accelerated projected gradient descent are consistently able to solve the j-th Newton step in $O(1/\sqrt{\epsilon_j})$ iterations, where ϵ_j is the duality gap at this step. Amortized over all Newton steps, these methods are first-order methods that converge at the accelerated error rate of $O(1/k^2)$ at the k-th iteration.

1.4 Thesis Outline

In Chapter 2, we review classical power system models, and stability analysis techniques based on steady-state analysis, time-domain simulations, and eigenvalue analysis.

In Chapter 3, we review the theory of robust stability analysis, and apply it to two case studies.

In Chapter 4, we review interior-point methods and first-order methods, and suggest the use of Krylov subspace acceleration to accelerate the convergence of first-order methods. We prove an improved convergence rate for the accelerated version of a simple projected gradient descent algorithm.

In Chapter 5, we describe the hierarchical solver used to factorization and solve (1.5).

In Chapter 6, we prove a result on the convergence rate of the Krylov subspace accelerated version of ADMM.

Finally, in Chapter 7, we summarize our findings and discuss directions for future work.

1.5 Notation

Vector spaces. We use $\mathbb{R}^n$ and $\mathbb{C}^n$ to denote the space of size-n column vectors with real and complex coefficients, respectively. We use $\mathbb{S}^n$, $\mathbb{S}^n_+$, $\mathbb{S}^n_{++}$ to denote the space of $n \times n$ real symmetric matrices, positive semidefinite matrices (real symmetric with positive eigenvalues), and positive definite matrices (real symmetric with nonnegative eigenvalues), respectively.

Matrix inequalities. We use $X \succ Y$ to mean that the matrix $X - Y$ is positive definite, and $X \succeq Y$ to mean that the matrix $X - Y$ is positive semidefinite .

Concatenation. Row and column concatenation are denoted using the comma and the semicolon, respectively, as in

$$[a, b] = \begin{bmatrix} a & b \end{bmatrix}, \qquad [a; b] = \begin{bmatrix} a \\ b \end{bmatrix}.$$

Direct sum. Given two matrices A and B, their direct sum is written

$$A \oplus B = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}.$$

Vectorization & Matricization. Given the $m \times n$ matrix X, we define its vectorization as the familiar "column-stacking" operator

$$\operatorname{vec} X = [X_{1,1}, \ldots, X_{m,1}, X_{1,2}, \ldots, X_{m,2}, \ldots, X_{1,n}, \ldots, X_{m,n}]^T.$$

and we define $X = \operatorname{mat} x$ as the inverse operator.

Kronecker product. The Kronecker product is defined given the $n \times q$ matrix A and the $m \times p$ matrix B as the $nm \times pq$ matrix as

$$A \otimes B = \begin{bmatrix} A_{1,1}B & \cdots & A_{1,q} \\ \vdots & \ddots & \vdots \\ A_{n,1}B & \cdots & A_{n,q}B \end{bmatrix}.$$

Chapter 2

Power System Stability Analysis

The electric power system consists of *generation* units where primary energy—from fossil or nuclear fuels, or from wind, solar, geothermal, or hydro energy—is converted into electric power, high-voltage *transmission* networks that transport the bulk power to low-voltage local *distribution* networks, and consumers or *loads* where power is used; this division is illustrated in Fig. 2-1. System operators—sometimes affiliated with a particular utility or sometimes independent and responsible for multiple utility areas—manage the flow of electricity in a power system. Operators *commit* generators and transmission lines to be available on specific days, and *dispatch* instructions in real-time in order for supply and demand to be balanced at the lowest cost.

It is the grid operator's job to guarantee power system stability under uncertainty. Historically, the dominant concern was *transient stability* [2]—the ability of a system to recover to safe operating conditions following a large disturbance, such as the loss of an important transmission line or the short-circuiting of a major generator. The industry is standardized around the $N-1$ criterion, stating that the power system should remain stable following the loss of any one component. Reflecting the needs of the grid operator, the classical tools for stability analysis are primarily based on recreating transient events within numerical simulations.

2.1 Power System Models

The emphasis on simulation-based transient stability analysis has given rise to a particular modeling philosophy. From decades of operating experience, engineers have found the largest destabilizing forces in a power system to have *electromechanical* origins. As a consequence, generators are always modeled in great detail, while the loads and the network itself are only coarsely modeled. It is generally argued that electromagnetic phenomena like propagation delay, standing waves, and harmonics, are relatively insignificant for stability.

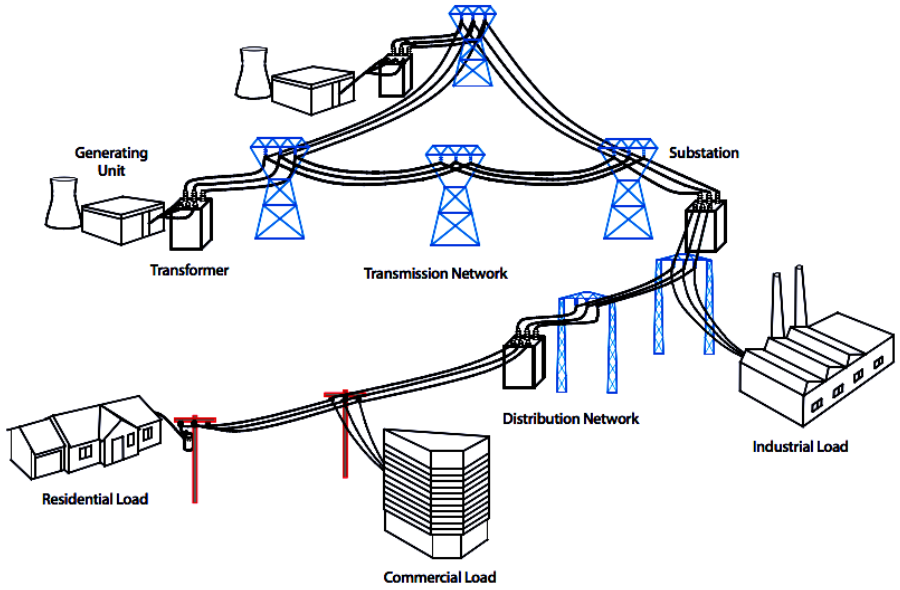


Figure 2-1: The power system and its form major divisions: generation, transmission, distribution and loads. Source: [1].

2.1.1 Mathematical Framework

Mathematically, power systems are *flow networks*, i.e. undirected graphs labeled with potential variables at its nodes and flow variables at its edges. In a power system, the vertices are known as *buses*, the edges as *branches*, the potential variables as *voltages*, and the flow variables as *currents*.

The voltages and currents in a power system are defined in a number of ways[1]. The *instantaneous* description assigns a time-dependent voltage variable to each system node, and a time-dependent current variable to each system branch. Power systems are three-phase, meaning that power is carried along three lines, with each phase at a nominal 120 degree shift from the other two phases. Conventional labels for these three phases are "a", "b", and "c" respectively. Hence, in an n-bus, m-branch power system, the instantaneous voltages $v_{abc}(t) \in \mathbb{R}^{3n}$ and instantaneous currents $i_{abc}(t) \in \mathbb{R}^{3m}$ may be written

$$v_{abc}(t) = \begin{bmatrix} v_a(t) \\ v_b(t) \\ v_c(t) \end{bmatrix}, \qquad i_{abc}(t) = \begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \end{bmatrix}.$$

In a real power system, these instantaneous voltages and currents can be directly measured by field workers, using voltage and current probes and a multimeter or oscilloscope.

However, the instantaneous description is cumbersome for stability analysis. Classical techniques analyze systems that converge towards a fixed, unchanging steady-

[1] A more detailed exposition can be found in most power systems textbooks, e.g. [20].

state, but power systems are designed to oscillate even at equilibrium. Instead, the *phasor* description addresses this issue by defining the instantaneous values as the projections of rotating complex phasors $\hat{v}_{abc}(t) \in \mathbb{C}^{3n}$ and $\hat{i}_{abc}(t) \in \mathbb{C}^{3m}$

$$v_{abc}(t) = \mathrm{Re}\{\hat{v}_{abc}(t)e^{j\omega_0 t}\}, \qquad i_{abc}(t) = \mathrm{Re}\{\hat{i}_{abc}(t)e^{j\omega_0 t}\}. \tag{2.1}$$

The constant $\omega_0 = 2\pi \cdot 60$ is the nominal system angular frequency. In a stable power system, the phasor quantities will converge towards constant values while their instantaneous counterparts may continue to oscillate. No information is gained or lost in converting instantaneous quantities to the phasor counterparts. However, to measure phasors directly requires an accurate, synchronized time reference. An important recent development is the phasor measurement unit (PMU), which obtains a synchronized time reference using satellite GPS.

Finally, three-phase power systems are always designed and operated to be balanced under normal conditions; they tend to be well-approximated as balanced even when an imbalance does occur.

Assumption 1 (Three-phase quantities are balanced)**.** Three-phase voltages and currents are sinusoidal and balanced: given the quantities for one phases, those same quantities are replicated and shifted by ± 120 degrees in the other two phases. Three-phase admittances and impedances are also balanced: each resistor, inductor and capacitor connected to one phase is perfectly replicated at the other phases in either a Y- or Δ- connection.

Assuming balance, the three-phase system may be reduced to an equivalent single-phase system, known as the *positive sequence* description. Rewriting each three-phase phasor triple as shifted versions of a single phasor,

$$\begin{bmatrix}\hat{v}_a(t)\\ \hat{v}_b(t)\\ \hat{v}_c(t)\end{bmatrix} = \begin{bmatrix}\hat{v}(t)\\ e^{+2\pi j/3}\hat{v}(t)\\ e^{-2\pi j/3}\hat{v}(t)\end{bmatrix} = \mathbf{U}_n\hat{v}(t), \qquad \begin{bmatrix}\hat{i}_a(t)\\ \hat{i}_b(t)\\ \hat{i}_c(t)\end{bmatrix} = \begin{bmatrix}\hat{i}(t)\\ e^{+2\pi j/3}\hat{i}(t)\\ e^{-2\pi j/3}\hat{i}(t)\end{bmatrix} = \mathbf{U}_m\hat{i}(t), \tag{2.2}$$

the number of variables has also been reduced by a factor of three. Mathematically, this is a simple projective model order reduction via the linear basis $\mathrm{blkdiag}(\mathbf{U}_n, \mathbf{U}_m)$. The reduction (2.2) is only an approximation for imbalanced systems—where Assumption 1 fails to hold—and the imbalance information is irreversibly lost.

2.1.2 Network Equations

Transmission systems are usually modeled using a simple admittance model. In system with q buses, this is the algebraic equation

$$\begin{bmatrix}Y_{11} & \cdots & Y_{1q}\\ \vdots & \ddots & \vdots\\ Y_{q1} & \cdots & Y_{qq}\end{bmatrix}\begin{bmatrix}v_1(t)\\ \vdots\\ v_q(t)\end{bmatrix} = \begin{bmatrix}i_1^{\mathrm{bus}}(t)\\ \vdots\\ i_q^{\mathrm{bus}}(t)\end{bmatrix},$$

where for each k-th bus, $v_k(t) \in \mathbb{C}$ is the voltage phasor at the k-th bus, $i_k^{\text{bus}}(t) \in \mathbb{C}$ is the nodal current injection phasor, and the governing sparse matrix has the same sparsity pattern as the underlying system

$$Y_{i,j} = \begin{cases} \text{nonzero} & \text{bus } i \text{ connects to bus } j \\ 0 & \text{otherwise.} \end{cases}$$

The matrix is usually (but not always) complex symmetric, i.e. $Y_{i,j} = Y_{j,i}$.

The model is written more succinctly as the matrix equation

$$\mathbf{Y} v(t) = i_{\text{bus}}(t), \tag{2.3}$$

and the admittance $\mathbf{Y}$ matrix can be assembled directly using sparse matrix techniques (see e.g. [21, Sec.3]). But to understand the simplifying assumptions associated with the model, let us derive (2.3) from first principles. We begin by assuming the transmission network to be linear, thereby neglecting a number of nonlinear branch effects, including transformer and series-connected reactor saturation, series-connected grid-level power electronics devices (i.e. "FACTS"), and a variety of thermal effects.

Assumption 2 (Network is linear)**.** The transmission system is a linear network, i.e. comprised only of resistors, inductors and capacitors.

Converting transmission lines and transformers into simple Pi models and applying the standard technique of modified nodal analysis from circuit analysis [22], we obtain a system of equations of the following form

$$\begin{bmatrix} \mathbf{C}_{abc} & 0 \\ 0 & \mathbf{L}_{abc} \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} v_{abc}(t) \\ i_{abc}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{G}_{abc} & \mathbf{F}_{abc} \\ -\mathbf{F}_{abc}^T & \mathbf{R}_{abc} \end{bmatrix} \begin{bmatrix} v_{abc}(t) \\ i_{abc}(t) \end{bmatrix} = \begin{bmatrix} r_{abc}(t) \\ 0 \end{bmatrix}, \tag{2.4}$$

in which $r_{abc}(t)$ denotes nodal current injections, the capacitance matrix $\mathbf{C}_{abc}$ and conductance matrix $\mathbf{G}_{abc}$ are in graph Laplacian form, the inductance matrix $\mathbf{L}_{abc}$ and resistance matrix $\mathbf{R}_{abc}$ are diagonal, and $\mathbf{F}_{abc}$ is a directed incidence matrix for the resistor-inductor branches in the system. The model (2.4) rewritten for the phasor quantities, by applying the product rule to (2.1),

$$\begin{bmatrix} \mathbf{C}_{abc} & 0 \\ 0 & \mathbf{L}_{abc} \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \hat{v}_{abc}(t) \\ \hat{i}_{abc}(t) \end{bmatrix} + \begin{bmatrix} j\omega_0 \mathbf{C}_{abc} + \mathbf{G}_{abc} & \mathbf{F}_{abc} \\ -\mathbf{F}_{abc}^T & j\omega_0 \mathbf{L}_{abc} + \mathbf{R}_{abc} \end{bmatrix} \begin{bmatrix} \hat{v}_{abc}(t) \\ \hat{i}_{abc}(t) \end{bmatrix} = \begin{bmatrix} \hat{r}_{abc}(t) \\ 0 \end{bmatrix}. \tag{2.5}$$

Next, we assume that the three-phase network is balanced (Assumption 1). The assumption allows us to reduce the network model (2.5) by a factor of three in size, to the *positive-sequence model*

$$\begin{bmatrix} \mathbf{C} & 0 \\ 0 & \mathbf{L} \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \hat{v}(t) \\ \hat{i}(t) \end{bmatrix} + \begin{bmatrix} j\omega_0 \mathbf{C} + \mathbf{G} & \mathbf{F} \\ -\mathbf{F}^* & j\omega_0 \mathbf{L} + \mathbf{R} \end{bmatrix} \begin{bmatrix} \hat{v}(t) \\ \hat{i}(t) \end{bmatrix} = \begin{bmatrix} \hat{r}(t) \\ 0 \end{bmatrix}, \tag{2.6}$$

where $\mathbf{C} = \mathbf{U}_n^* \mathbf{C}_{abc} \mathbf{U}_n$, $\mathbf{L} = \mathbf{U}_m^* \mathbf{L}_{abc} \mathbf{U}_m$, and similarly for $\mathbf{F}, \mathbf{G}, \mathbf{R}$. The positive sequence phasors $\hat{v}(t), \hat{i}(t)$ and the basis matrices $\mathbf{U}_n, \mathbf{U}_m$ were defined earlier in

(2.2).

Finally, to emphasize electromechanical interactions over electromagnetic ones, the network dynamics are assumed to be negligible.

Assumption 3 (Network experiences no transients)**.** The network is in permanent 60 Hz sinusoidal steady-state. If a perturbation causes this to be affected, then a new steady-state is instantly established, without going through the electromagnetic transients in between.

Assumption 3 artificially sets $\frac{\mathrm{d}}{\mathrm{d}t}\hat{v}(t)$ and $\frac{\mathrm{d}}{\mathrm{d}t}\hat{i}(t)$ to zero in (2.6), thereby yielding the algebraic relation

$$\begin{bmatrix} j\omega_0\mathbf{C}+\mathbf{G} & \mathbf{F} \\ -\mathbf{F}^* & j\omega_0\mathbf{L}+\mathbf{R} \end{bmatrix} \begin{bmatrix} \hat{v}(t) \\ \hat{i}(t) \end{bmatrix} + \begin{bmatrix} \hat{r}(t) \\ 0 \end{bmatrix} = 0,$$

Eliminating the branch current variable $\hat{i}(t)$ using Gaussian elimination yields an admittance relationship

$$\left[(j\omega_0\mathbf{C}+\mathbf{G}) + \mathbf{F}(j\omega_0\mathbf{L}+\mathbf{R})^{-1}\mathbf{F}^*\right]\hat{v}(t) = \mathbf{Y}\hat{v}(t) = \hat{r}(t),$$

as desired.

2.1.3 Generator equations

Generators are modeled as voltage-driven current sources: given a voltage phasor $v_{\text{in}}(t) \in \mathbb{C}$, the model responds with a current phasor $i_{\text{out}}(t) \in \mathbb{C}$. The most important aspect of the model is the electromechanical coupling, which is modeled using the Lagrangian

$$\mathscr{L}(q,i,\theta,\omega) = \frac{1}{2}i^T L(\theta) i + \frac{1}{2}M\omega^2, \qquad L(\theta) \triangleq e^{J\theta}L_0 e^{-J\theta}$$

where $\theta \in [0, 2\pi)$ is the machine rotor angle, $i \in \mathbb{R}^p$ is the current injection phasor, and $\omega = \mathrm{d}\theta/\mathrm{d}t$ and $q = \int_0^t i(t)\mathrm{d}t$ are their respective dual variables. The skew-symmetric matrix J is used to generates a rotation $e^{J\theta}$ upon the current injection phasor, corresponding with the physical rotation of the machine rotor. Applying the Euler-Lagrange equations, and introducing forcing terms u, φ and damping terms R, D yields

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\hat{\varphi} + J\hat{\varphi} + R\hat{i} &= e^{-J\theta}u \\ M\frac{\mathrm{d}\omega}{\mathrm{d}t} + D\omega &= \hat{i}^T J\hat{\varphi} + \tau_m \\ \hat{\varphi} &= L_0\hat{i} \end{aligned} \tag{2.7}$$

in the machine inertial frame[2], with $\hat{i} = e^{-J\theta}i$ and $\hat{\varphi} = e^{-J\theta}\varphi$.

[2]Assuming that the damping matrix R commutes with J, i.e. $RJ = JR$.

One of the most common generator models is the round-rotor model "GENROU", originally developed for the Siemens software PSS/E. In its simplest form, GENROU is an implementation of (2.7) with[3]

$$L_0 = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix}, \qquad J = \begin{bmatrix} 0 & -\mathrm{diag}(1,0,0) \\ \mathrm{diag}(1,0,0) & 0 \end{bmatrix},$$
$$u = \begin{bmatrix} \mathrm{Re}(v_{\mathrm{in}}) & v_{fd} & 0 & \mathrm{Im}(v_{\mathrm{in}}) & 0 & 0 \end{bmatrix}^T$$
$$i = \begin{bmatrix} \mathrm{Re}(i_{\mathrm{out}}) & i_{d2} & i_{d3} & \mathrm{Im}(i_{\mathrm{out}}) & i_{q2} & i_{q3} \end{bmatrix}^T$$

and L_d, L_q are 3×3 dense symmetric positive definite matrices. The calculations needed to populate the matrices L_d, L_q, R, and constants M, D using field measurements can be found in most standard textbooks [23, 24], and also in IEEE standard 115. Other generator models differ from GENROU primarily in the number of state variable considered. The salient-rotor "GENSAL" is essentially the same as GENROU, but with one fewer state variable.

The generator model is completed by incorporating an exciter model, and optionally a governor and stabilizer model. The machine exciter is a control loop that actuates the field excitation (the element v_{fd} in u) in order to maintain the bus voltage magnitude $|v_k|$ to a given setpoint. The machine governor model is a control loop that maintains a constant machine speed ω, by actuating mechanical torque τ_m. The stabilizer is a control loop that senses the machine speed ω and actuates a countersignal to the exciter in order to dampen potential oscillations. Standardized models for each of these components are described in the IEEE standards 421 and 1100.

2.1.4 Load Equations

Loads are most commonly modeled using the ZIP model. Given a voltage phasor $v_{\mathrm{in}}(t) \in \mathbb{C}$, the ZIP model responds with a current phasor $i_{\mathrm{out}}(t) \in \mathbb{C}$ via the algebraic relation

$$i_{\mathrm{out}}(t) = \left[Y + \frac{I}{|v_{\mathrm{in}}(t)|} + \frac{S}{|v_{\mathrm{in}}(t)|^2} \right] v_{\mathrm{in}}(t), \tag{2.8}$$

and the model parameters are the constant admittance $Y \in \mathbb{C}$, the constant current phasor $I \in \mathbb{C}$, and the constant complex power $S \in \mathbb{C}$. Let us emphasize some important facts about the model:

1. The output is linear with respect to its parameters Y, I, S;
2. The output is nonlinear with respect to its input, except when $I = S = 0$;
3. The output is invariant with respect to the input phase. Rotating the input voltage phasor by θ will simply rotate the output current current phasor θ.

The ZIP model is commonly specified using a total complex power consumption and a "ZIP ratio", such as 10%-20%-70%. This simply means that, at nominal voltage,

[3] GENROU further incorporates the two simplifications described in Section 5.1 of Kundur [23], as well as a mechanism to model the effects of magnetic saturation.

the total complex power is distributed as 10% over the admittance-like portion of the load, 20% over the constant-current-like portion, and 70% over the constant-power-like portion.

2.1.5 Steady-State Model

The steady-state model for a power system is commonly known as the *powerflow equations*. In a system with q buses, we begin by taking a *nominal* voltage magnitude and complex power production / consumption pair at each of its buses

$$\{V_k,\ P_k + jQ_k\} \qquad k \in \{1, \ldots, q\}.$$

These can be considered as the "ideal", or "set-point" quantities for our power system. In practice, their values are dispatched by the operator while considering regulatory limits, actual load consumption, and dispatched generation production. The generators and loads in a power system attempt to achieve these nominal values, while subject to the physical limitations of the network.

Writing $v_k, i_k \in \mathbb{C}$ as the *actual* steady-state voltage and current injection phasors at the k-th bus, three standard constraints are used to model a single lumped generator or load at steady-state:

- The PV constraint, used to model generators. The actual voltage magnitude and real power produced are constrained to match the nominal, as in

$$\mathrm{Re}\{i_k^* v_k\} = P_k, \qquad v_k^* v_k = V_k^2, \qquad \forall k \in \mathcal{PV}. \tag{2.9}$$

  This behavior closely matches the control action of a typical generator at steady-state, which outputs a fixed amount of real power, and whatever reactive power necessary to maintain a constant voltage magnitude at the generator bus.

- The PQ constraint, used to model loads (and generators subject to reactive power limits). The real and reactive power produced or consumed are constrained to match the nominal, as in

$$\mathrm{Re}\{i_k^* v_k\} = P_k, \qquad \mathrm{Im}\{i_k^* v_k\} = Q_k, \qquad \forall k \in \mathcal{PQ}. \tag{2.10}$$

- The slack constraint, used to model the generator providing regulation services. The actual voltage magnitude is enforced to match the nominal, and the phase angle is set to be zero, as in

$$\mathrm{Re}\{v_k\} = V_k, \qquad \mathrm{Im}\{v_k\} = 0, \qquad \forall k \in \mathcal{S}. \tag{2.11}$$

  This way, the generator outputs whatever real and reactive power necessary to make system-wise production equal to consumption plus losses.

Imposing one of these models at each bus, and enforcing the network model $\mathbf{Y}v = i$ completes the model. After some minor algebraic manipulation, these can be posed

as a system of $2n$ real quadratic equations over $2n$ real variables, and solved using Newton's method.

2.1.6 Time-Domain Model

Time-domain models of the power system are constructed by interconnecting generator and load models—one model per bus—to the network model. Due to the inductor-like nature of real generators and loads, these models are voltage-driven current-sources. At the k-th bus, we use the nonlinear function pair $f_k(\cdot,\cdot)$ and $g_k(\cdot,\cdot)$ to implement a nonlinear state-space model

$$\frac{\mathrm{d}}{\mathrm{d}t}x_k(t) = f_k(x_k(t), v_k(t)), \qquad i_k(t) = g_k(x_k(t), v_k(t)), \tag{2.12}$$

with state variables $x_k(t) \in \mathbb{R}^{n_k}$. The model takes the k-th voltage phasor $v_k(t) \in \mathbb{C}$ as the input of the model, and returns the k-th nodal current injection phasor $i_k(t) \in \mathbb{C}$ as the output.

The steady-state model, described in the previous subsection, is used to initialize the time-domain models. At time $t = 0$, the voltage and current injection phasors at the k-th bus $v_k(0)$, $i_k(0)$ are assumed to take on their steady-state values, and an initialization procedure—specific to the exact generation / load model—is used to select the initial values for the state variable $x_k(0)$ in order to satisfy $f_k(x_k(0), v_k(0)) = 0$ and $g_k(x_k(0), v_k(0)) = i_k(0)$.

Finally, enforcing the relationship between voltage and current injection phasors via the network model (2.3) yields the complete time-domain model

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1(t) \\ \vdots \\ x_q(t) \end{bmatrix} = \begin{bmatrix} f_1(x_1(t), v_1(t)) \\ \vdots \\ f_q(x_q(t), v_q(t)) \end{bmatrix}, \qquad \begin{bmatrix} Y_{11} & \cdots & Y_{1q} \\ \vdots & \ddots & \vdots \\ Y_{q1} & \cdots & Y_{qq} \end{bmatrix} \begin{bmatrix} v_1(t) \\ \vdots \\ v_q(t) \end{bmatrix} = \begin{bmatrix} g_1(x_1(t), v_1(t)) \\ \vdots \\ g_q(x_q(t), v_q(t)) \end{bmatrix},$$

which is written more succinctly as

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = f(x(t), u(t)), \qquad \mathbf{Y}u(t) = g(x(t), u(t)). \tag{2.13}$$

Note that communication between the different buses is entirely facilitated through the admittance matrix; if this matrix were diagonal, then the system model would decouple into q independent models.

2.2 Classical Stability Analysis

Power system engineers typically classify stability analysis techniques into three categories:

- Voltage stability (Steady-state analysis). Given a certain steady-state profile,

verify that all voltage constraints are satisfied. Proportionally increase the steady-state load, in order to compute the margin to collapse.

- Transient stability (Simulation-based analysis). Simulate the loss of generation or a fault on a transmission line, and verify that the system is able to recover to an acceptable steady-state.

- Small-signal Stability (Eigenvalue-based analysis). Given a certain steady-state profile, linearize the time-domain equations, and verify that the eigenvalues of the Jacobian are sufficiently stable.

These classical analyses are performed using the deterministic model derived in the previous section. Each analysis is repeated over numerous different load / generation profiles, as well as different loss-of-generation or loss-of-transmission contingency scenarios. In essence, the engineer construct a large, high-dimensional uncertainty set based on engineering intuition and operating experience, and then attempts to validate the stability of the uncertainty set by sampling individual individual scenarios.

Let us formalize the idea using parameterization, by introducing the parameter variable δ. We restrict δ to lie within an uncertainty set Δ, defined to encompass a range of load / generation profiles, and the enumerate all of the important contingency scenarios. In principle, we can define Δ in a way to include both time-varying uncertainty (e.g. the output of a solar panel) and time-invariant uncertainty (e.g. the inertia of a particular machine). Parameterizing the deterministic model in the previous section over δ yields an uncertain model

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = f(x(t), v(t), \delta(t)), \qquad \mathbf{Y}(\delta(t))\, v(t) = g_\delta(x(t), v(t), \delta(t)), \qquad \delta \in \Delta. \quad (2.14)$$

Let us proceed to examine stability analysis for the uncertain model (2.14).

2.2.1 Steady-State Analysis

In order to establish whether an acceptable steady-state exists for every uncertain parameter choice $\delta \in \Delta$, we implicitly define steady-state functions $x_0(\delta)$, $v_0(\delta)$ to satisfy

$$f_\delta(x_0(\delta), v_0(\delta)) = 0, \qquad g_\delta(x_0(\delta), v_0(\delta)) = \mathbf{Y}(\delta) v_0(\delta).$$

We can then perform steady-state analysis by examining the individual elements of $x_0(\delta)$ and $v_0(\delta)$. For example, we may attempt to enumerate the elements in $v_0(\delta)$, and verify that each satisfies the appropriate constraints.

The main difficult is in the evaluation of the implicitly defined steady-state functions $x_0(\delta)$ and $v_0(\delta)$. If the uncertain conditions they describe are well-conditioned, then they may simply be implemented via the basic Newton powerflow procedure described in Section 2.1.5. However, as the system approach collapse, the Newton system becomes singular, and Newton's method begins to fail. Sophisticated techniques have been developed to address this issue. Continuation power flow schemes are homotopy methods, designed to evaluate a collapsing condition by slowly ramping

up from a well-conditioned initial point. More recently, techniques based on analytic continuation and

Ultimately, the existing methods are based on sampling elements from Δ, which is inconclusive by its very nature. There have been many attempts to capture the image of these mappings as convex sets

$$\mathcal{X}_0 \supseteq \{x_0(\delta) : \delta \in \Delta\}, \qquad \mathcal{V}_0 \supseteq \{x_0(\delta) : \delta \in \Delta\}.$$

If a tight description could be constructed, then it would be possible to exhaustively and conclusive establish the feasibility of each element, i.e. to valid that every element satisfies a set of constraints. The engineer would then be able to conclusively establish voltage stability under uncertainty.

2.2.2 Simulation-based Analysis

To establish whether the system can converge to an acceptable steady-state, we select a choice of $\delta(t)$ and integrate the differential-algebraic equations (2.14) using a time-stepping rule, starting from $t = 0$ and ending at some termination time T. If we observe, at any time $t \in [0, T]$, that the state variables are diverging away from the intended steady-state (or failing to satisfy constraints), then we may terminate the simulation and mark that choice of $\delta(t)$ as being unstable. If the system does not become unstable for a sufficiently large T, then we may mark this particular choice of $\delta(t)$ as stable.

The simulation-based approach is inherently inconclusive, and suffers from three characteristic issues. First, the simulations must always end at a finite time horizon, so we can never be sure whether the system may eventually become unstable at some distant time in the future. Second, it is heavily dependent on the accuracy and faithfulness of the models. Third, the time-stepping itself introduces error to the simulation; it is possible for a numerical instabilities to cause a stable scenario to become unstable, or more worryingly, to dampen an unstable scenario enough to make it stable.

Today, the grid operator manages these three issues by increasing the computational power used to perform the simulations. The termination time T is extended far into the future, highly detailed models are incorporated, and conservative parameters are chosen for the time-stepping rule. Unfortunately, these modifications low the simulations, limiting them to off-line analysis.

Direct stability methods have been proposed to speed up and potentially replace simulation-based analysis. These methods work by analytically constructing a Lyapunov function, and using it to compute the region-of-attraction for a particular stable equilibrium, or at least a conservative, inner approximation of the region-of-attraction. Given a transient stability scenario and an associated stable equilibrium, we may simulate the scenario, and terminate as soon as the system falls within the region-of-attraction [25]. If our estimation of the region-of-attraction is not too conservative, then very little simulation is actually required to determine stability. If our estimation of the region-of-attraction were exact, then it may also be used to

determine instability: the system is immediately unstable as soon as it exits the region-of-attraction.

The core idea of direct methods dates back to the 1940s [26,27]; the main difficulty persisting to this day is constructing an estimation of the region-of-attraction that is large enough to be useful. Early techniques, such as the "nearest unstable equilibrium method", proved to be far too conservative [28, 29]. More recent techniques found success by computing the region-of-attraction of a simplified model, and using this to inform properties of the original system [25, 28, 30]. All of these techniques are based on analytical arguments, and cannot be easily extended to arbitrary nonlinear models.

2.2.3 Eigenvalue-based Analysis

Small-signal stability analysis (also known as eigenvalue analysis) makes stability predictions under the assumption of small-signal disturbance. We state this loosely for the purpose of exposition.

Assumption 4 (Small-signal disturbance)**.** Decompose the state variable $x(t)$, the algebraic variable $v(t)$ and the parameter variable $\delta(t)$ into a constant bias and a time-dependent perturbation, as in

$$x(t) = x_0 + \Delta x(t), \qquad v(t) = v_0 + \Delta v(t), \qquad \delta(t) = \delta_0 + \Delta\delta(t).$$

Then the time-dependent perturbations $\Delta x(t)$, $\Delta v(t)$, and $\Delta\delta(t)$, are sufficiently small as to be considered negligible.

Under this assumption, the nonlinear stability of a system within a small-signal neighborhood of a given operating point $\theta = \{x_0, v_0, \delta_0\}$ can be analyzed by examining the stability of its linearization. Defining suitable Jacobian matrices, the model (2.14) can be put into the form

$$\frac{\mathrm{d}x}{\mathrm{d}t} = A(\theta)x + B(\theta)v, \qquad \mathbf{Y}(\theta)v = A(\theta)x + B(\theta)v,$$

and reduced to the state-space model

$$\frac{\mathrm{d}x}{\mathrm{d}t} = M(\theta)x, \qquad M(\theta) \triangleq A(\theta) - B(\theta)[D(\theta) - \mathbf{Y}(\theta)]^{-1}C(\theta).$$

If this state-space model is stable, then we may conclude that the nonlinear model is small-signal stable, i.e. stable when confined within a neighborhood of the operating point. Given a collection of operating points $\theta \in \Theta$, possibly constructed using the steady-state procedure described earlier, we may repeat this analysis for each possible operating point in order to establish the small-signal stability of the model under uncertainty.

In turn, the stability of each linearized state-space model can be analyzed by examining its eigenvalues. Labeling the k-th eigenvalue of $M(\theta)$, two common metrics

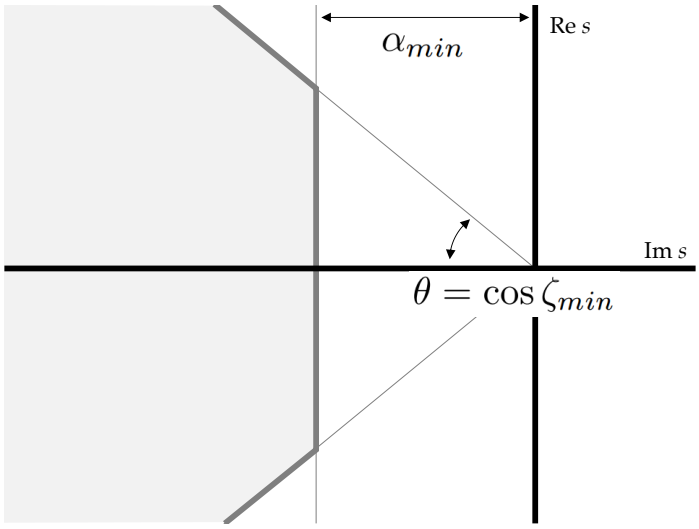


Figure 2-2: Small-signal stability on the complex plane. Every eigenvalue $\lambda_k = -\alpha_k + j\omega_k$ placed in the shaded region will satisfy the decay rate constraint $\alpha_k \geq \alpha_{min}$ and the decay ratio constraint $\alpha_k/|\lambda_k| \geq \zeta_{min}$.

of stability are the *damping ratio*,

$$\zeta \triangleq \min_k \{-\mathrm{Re}\,\lambda_k/|\lambda_k|\}, \tag{2.15}$$

and the *decay rate*,

$$\alpha \triangleq \min_k \{-\mathrm{Re}\,\lambda_k\}. \tag{2.16}$$

A power system is said to be *small-signal stable* (about its equilibrium x_0) if its decay rate and damping ratio satisfy certain thresholds:

$$\zeta \geq \zeta_{lim} \qquad \alpha \geq \alpha_{lim}. \tag{2.17}$$

From a control theory perspective, these stability criteria form a trapezoidal envelope on the complex plane, as shown in Fig. 2-2. In order for the linearized system to be deemed "sufficiently stable", all of its eigenvalues must lie within this envelope. For large power networks, minimum damping ratios are usually specified to exceed 3% to 5%. Low-frequency oscillations may be required to have damping ratios as high as 15% [13]. Minimum decay rate ares are typically 0.05 to 0.1 per-second.

Chapter 3

Extending Robust Stability Analysis to Power Systems

In this chapter, we review the theory of robust stability analysis, and apply these techniques to two power systems case studies. Our objective is to investigate the possible applications of robust stability for the power systems application, and to summarize the overarching implications for the grid operator. At the same time, we wish to expose the underlying mathematical structure of robust stability analysis.

In the first case study, we examine the impact of generating 30% of the power in the IEEE 118 bus test network with 118 distributed renewable sources. High penetrations of distributed renewables can dramatically increase uncertainty in the transmission system, making small-signal stability verification far more challenging. We show that multipoint local optimization can find less stable scenarios that are easily missed by sampling. In addition, we show that robust stability analysis is computationally tractable, but as yet, only by linearizing and dimension-reducing the parametric variation.

In our second case study, we give an illustrative example of a microgrid that is guaranteed to be stable under small-signal intermittency, and show that it can be made unstable when the intermittency becomes large-signal. The classic approach of small-signal stability analysis may lead to overly optimistic conclusions, because it implicitly assumes that the intermittency is small-signal in nature. Instead, robust stability analysis can be used to provide large-signal stability guarantees that overcome this limitation. We compute large-signal stability margins, and show that the small- and large-signal stability margins are related by the maximum allowable slew-rate of the intermittency.

3.1 Stability Certificates for LPV Models

Most of modern control theory is developed for models placed in linear parameter varying (LPV) form[1]

$$\dot{x}(t) = A(\delta(t))x(t), \qquad \delta(t) \in \Delta, \tag{3.1}$$

in which $A(\cdot)$ is a continuous matrix-valued function, $\delta(t)$ is a list of time-varying parameters, and Δ is an uncertainty set. LPV models are essentially LTI models placed under time-varying uncertainty: its coefficient matrix is not precisely known and may possibly vary with time. The LPV form is naturally suited for modeling uncertain systems, as well as capturing the inherent imperfections in deterministic systems, such as modeling errors, imprecise physical measurements, and model order-reduction. As we will show in Section 3.2, linearized uncertainty can also be used to encompass the effects of deterministic but nonlinear dynamics.

By construction, the LPV system has a single equilibrium at the origin $x = 0$. We say that the LPV is *robustly stable* if, starting from any initial condition $x(0)$, the solution $x(t)$ converges to this equilibrium *for every valid choice of* $\delta(t)$. Clearly, classical stability analysis techniques can never be use to establish robust stability. The uncertain time-varying parameter $\delta(t)$ can take on any of a continuous range of values at any point in time for all time, and it is impossible to simulate even a representative set of instances, let alone an exhaustive one. Whereas an eigenvalue-based analysis can, to an extent, analyze entire sets of time-varying parameters, it additionally requires the time-varying component to be "small-signal" relative to the time-invariant offset.

Instead, robust stability can be verified using a certification-based approach. These methods reformulate robust stability analysis into a convex feasibility problem; any feasible point is a numerical proof of robust stability known as a *stability certificate*. The simplest of such methods is the quadratic stability test.

Definition 5 (Quadratic Stability)**.** The LPV (3.1) is said to be *quadratically stable* if it there exists a *quadratic stability certificate*, i.e. a matrix $P \succ 0$ that satisfies

$$A(\delta)^T P + P A(\delta) \prec 0 \text{ for all } \delta \in \Delta. \tag{3.2}$$

Theorem 6. *Every quadratically stable LPV is also robustly stable.*

The theorem easily established by using $V(x) = x^T P x$ as a Lyapunov function. For the sake of exposition, however, let us give a more intuitive argument using a simple change-of-variables.

Proof. Given a quadratic stability certificate P, let U be any invertible matrix satisfying $U^T U = P$. (For example, we may pick U to be the upper-triangular Cholesky factorization of P.) We claim that the *transformed* state variable $y(t) = Ux(t)$ has a squared Euclidean norm $\|y(t)\|^2 = y_1^2(t) + \cdots + y_n^2(t)$ that strictly dissipates with

[1]Slight modifications of the LPV description are known in a number of other names, such as the linear differential inclusion (LDI), and the linear time-varying (LVT) model.

time, i.e. decays monotonically to zero, irrespective of the choice of $\delta(t)$. To see this, note that its rate-of-change satisfies the following

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\|y(t)\|^2 = \dot{y}(t)^T y(t) + y(t)^T \dot{y}(t) &= y(t)^T \left[U^{-T} A(\delta(t))^T U^T + U A(\delta(t)) U^{-1}\right] y(t) \\ &= \left[U^{-1} y(t)\right]^T \left[A(\delta(t))^T P + P A(\delta(t))\right] \left[U^{-1} y(t)\right] \\ &\leq \alpha \|y(t)\|^2 \text{ for all } \delta(t)\end{aligned}$$

where the constant α is negative (i.e. $\alpha < 0$) by virtue of the negative definiteness in (3.2). Integrating this relation yields $\|y(t)\|^2 \leq \|y(0)\|^2 e^{\alpha t}$, so the transformed state variable $y(t)$ must converge to the zero vector as $t \to \infty$. Since our original state variable can be recovered $x(t) = U^{-1} y(t)$, it too must converge to the zero vector. □

Quadratic stability is a simple but relatively conservative test for robust stability. It is only a sufficient condition: an LPV can be robustly stable without being quadratically stable (see e.g. [15, p.73]). Moreover, our notion of robust stability itself can be overly pessimistic when the waveforms taken on by the parameters $\delta(t)$ are restricted in some sense, e.g. they are time-invariant or only slowly varying. These shortcomings of quadratic stability are significant motivation for more sophisticated stability tests [17, 19], but the associated feasibility problems also become more difficult to solve.

The search for a suitable quadratic stability certificate is a convex feasibility problem subject to an infinite number of linear matrix inequality (LMI) constraints. Under certain restrictions on the matrix-valued function $A(\cdot)$ and the uncertainty set Δ, however, this search can be reformulated into a finite number of constraints, and solved using standard techniques. In the remainder of this section, we will review two prominent cases where this is possible.

3.1.1 The linear fractional representation (LFR)

If the individual elements of the matrix-valued function $A(\delta)$ can be expressed as *rational functions* of the uncertain parameters $\delta \in \mathbb{R}^m$, then there exists a choice of the coefficient matrix $M = \left[\begin{smallmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{smallmatrix}\right]$ and a vector of positive integers $r = [r_1, \ldots, r_m]^T$ such that

$$A(\delta) = M_{11} + M_{12} D (I - M_{22} D)^{-1} M_{21} \text{ where } D = \mathrm{diag}(\delta_1 I_{r_1}, \ldots, \delta_m I_{r_m}).$$

The coefficient matrix M and the vector r combine to form a *linear fractional representation* (LFR) for the matrix-valued function $A(\delta)$.

If the uncertainty set is given as the unit hypercube $\Delta = [-1, 1]^d$, then the quadratic stability condition (3.2) can be *relaxed* by introducing the multiplier variables S, G, as in

$$\begin{bmatrix} M_{11}^T P + P M_{11} + M_{21}^T S M_{21} & P M_{12} + M_{21}^T S M_{22} + M_{21}^T G \\ M_{12}^T P + M_{22}^T S M_{21} - G M_{21} & M_{22}^T S M_{22} - S + M_{22}^T G - G M_{22} \end{bmatrix} \prec 0, \tag{3.3}$$

subject to the structure conditions

$$S = S^T \succ 0, \quad G = -G^T, \quad S, G \text{ block-diagonal with block sizes } r_1, \ldots, r_m. \quad (3.4)$$

We refer the reader to [31] for a derivation, and [19,32] for a more general exposition of the underlying proof techniques. This reformulation is a generalization of the classic quadratic stability test for diagonally norm-bound LDIs [15, p.64], and is closely associated with structured singular value analysis (μ-analysis) [11]. Note that (3.3)-(3.4) is only a sufficient condition for quadratic stability, which by itself is already a relatively conservative test for robust stability.

The vast majority of physical models can be written in terms of rational matrix-valued functions $A(\delta)$; those that cannot can still be approximated to high precision using one, e.g. with Padé approximants. Hence, at least in principle, the LFR framework should be widely applicable to all applications. The weakness of the approach, however, is that it requires explicit expressions for the matrix-valued function $A(\cdot)$. It cannot directly accommodate data-driven models, nor models whose $A(\cdot)$ function is given as a black box.

The LFR framework is also one of the few tractable formulations for high-dimensional uncertainty sets and / or highly nonlinear models. The computational bottleneck is the LMI constraint (3.3), which is of the same size as the coefficient matrix M. The size of this matrix scales linearly with the dimension of the uncertainty set, when the maximum order of the element-wise rational functions (i.e. the "degree" of nonlinearity) is fixed. The incremental cost of each additional dimension of uncertainty is modest, and the LFR approach is often used to analyze systems subject to tens or even hundreds of dimensions of uncertainty.

Computational effort can be reduced by using a minimal realization of $A(\delta)$, i.e. an LFR that minimizes the size of its coefficient matrix M. Given a rational matrix-valued function, computing a minimal realization or a lower-order approximation is a well-studied problem, closely related to the theory of minimal realizations and model order reduction for state-space models; there are several MATLAB toolboxes available to perform this task automatically [33, 34].

3.1.2 The polytopic representation

Suppose that the matrix uncertainty set $A(\Delta)$ were a *polytope*, meaning that it can be written as the convex hull of a finite number of vertices

$$A(\Delta) = \text{conv}\{M_1, M_2, \ldots, M_m\},$$

then the semi-infinite quadratic Lyapunov LMIs (3.2) is equivalent to the Lyapunov inequalities

$$M_i^T P + P M_i \prec 0 \qquad \forall i \in \{1, \ldots, m\}. \quad (3.5)$$

The reformulation is often known as a *vertex-based test*, and LPVs with such a structure are known as polytopic LPVs, polytopic LDIs or matrix polytopes. The power of the polytopic representation lies in its *exactness*: the constraints (3.5) alongside

$P \succ 0$ are both necessary and sufficient conditions for quadratic stability.

The uncertainty set Δ is often provided as a polytope in practice. For example, it is common for each uncertain parameter were specified within a fixed interval, then the corresponding uncertainty set $\Delta = \{\delta : \underline{\delta}_i \leq \delta_i \leq \overline{\delta}_i\}$ is a hypercube, and hence also a polytope. If additionally, the matrix-valued function $A(\cdot)$ were affine, meaning that it can be written as the sum

$$A(\delta) = A_0 + \delta_1 A_1 + \delta_2 A_2 + \cdots + \delta_m A_m, \tag{3.6}$$

then the matrix uncertainty set $A(\Delta)$ must also be a polytope.

If $A(\cdot)$ is nonlinear, then the image set $A(\Delta)$ is not usually a polytope. Nevertheless, the vertex-based test can be used to perform stability analysis in a heuristic manner. One approach is to simply linearize the parameter dependence about an expansion point $\delta = \hat{\delta}$, as in

$$\tilde{A}_0 \triangleq A(\hat{\delta}), \qquad \tilde{A}_i \triangleq \left.\frac{\partial A}{\partial \delta_i}\right|_{\delta=\hat{\delta}}.$$

Then, defining the linear approximation $\tilde{A}(\delta) = \tilde{A}_0 + \sum_i \delta_i \tilde{A}_i$, the vertex-based stability test (3.5) can be applied to the polytope $\tilde{A}(\Delta)$. One advantage of this approach is that it works even when $A(\cdot)$ is provided as a black-box function, since the partial derivatives may be computed using finite difference. A significant drawback, however, is that it makes no guarantees in either direction—it may mark a stable system as unstable, and an unstable stable as stable.

Alternatively, we may sample a finite number of elements from the continuous set, $\Delta_0 \subset \Delta$, and to apply the polytopic stability test the vertex-based stability test (3.5) to the convex hull $\mathrm{conv}A(\Delta_0)$. For example, when Δ is provided as a hypercube $\Delta = [0,1]^d$, the samples may be selected from a uniform grid, or sampled from a uniform distribution. The resulting stability test is a necessary condition: $\mathrm{conv}A(\Delta_0)$ must be quadratically stable in order for $A(\Delta)$ to be quadratically stable.

Unfortunately, most polytopes have an exponential number of vertices for their given dimensionality. To give an illustration, the d-dimensional hypercube $[-1,1]^d$ is a polytope with 2^d vertices, and every additional dimension of uncertainty would *double* the number of constraints considered. As a consequence, the practical use of the vertex-based test is typically limited to models with a small number (e.g. less than 10) dimensions of uncertainty.

3.2 Certifying Nonlinear Models

Much of the motivation for LPV robust stability analysis comes from the fact that they can be used to certify the stability of nonlinear systems. Consider the nonlinear state-space model

$$\dot{x}(t) = f(x(t)), \qquad x(t) \in \mathcal{X}. \tag{3.7}$$

Let us assume without loss of generality that the origin is an equilibrium, i.e. $f(0) = 0$ and $0 \in \mathcal{X}$. Then we say that the nonlinear model is *globally stable* if every solution satisfying $x(t) \in \mathcal{X}$ for all time would also converge to this equilibrium[2]. Global stability of the nonlinear model (3.7) can be certified by analyzing the stability of a related LPV model.

3.2.1 Quasi-LPV

In the *quasi-LPV* approach, we define a matrix-valued function $A(\cdot)$ such that

$$A(x)x = f(x) \text{ holds for all } x \in \mathcal{X}.$$

Then every trajectory $x(t)$ satisfying (3.7) is also a trajectory of the LPV model

$$\dot{x}(t) = A(\delta(t))x(t) \qquad \delta(t) \in \mathcal{X}, \tag{3.8}$$

since enforcing the equality constraint $\delta(t) = x(t)$ would make the two models equivalent. Accordingly, the nonlinear model (3.7) is globally stable whenever the associated LPV (3.8) is stable; a stability certificate for the LPV is also a global stability certificate for the nonlinear model.

The quasi-LPV approach is conservative, as there are many trajectories of the LPV that are not trajectories of the original nonlinear system. Put in another way, a stable nonlinear system may produce an unstable quasi-LPV description. Conservatism can be reduced in two ways.

First, any nonlinear system may admit a number of LPV descriptions, some of which can be less conservative than others. In essence, we would like the size of the image $A(\mathcal{X}) \triangleq \{A(\delta) : \delta \in \mathcal{X}\}$ to be as small as possible. Quite a lot is known about how to make these models less conservative. By exploiting domain expertise and problem structure, it is possible to construct a description that is less conservative. Constructing the most accurate LPV representation of a nonlinear system remains an open problem and the topic of active research [35, 36].

Second, conservatism may be reduced by reducing the size of the uncertainty set. For example, if the rate of change for each $x(t)$ can be bounded *a priori*, then we may consider the rate-limited LPV model

$$\dot{x}(t) = A(\delta(t))x(t) \qquad \delta(t) \in \mathcal{X} \qquad \dot{\delta}(t) \in \mathcal{V},$$

with much fewer trajectories than before. Rate-limited quadratic stability certificates are not too much more difficult

In all cases, the quasi-LPV approach is inherently intrusive, since it requires explicit expressions for the nonlinear function $f(\cdot)$. Its effective use requires considerable domain expertise.

[2]Not all solutions starting from an initial point inside $\mathcal{X}$ are guaranteed to remain inside $\mathcal{X}$ for all time. Our notion of stability applies only to those solutions that do remain inside $\mathcal{X}$.

3.2.2 Global Linearization

Where only a black-box description of $f(\cdot)$ is available, we may adopt the *global linearization* approach, by defining a space of Jacobian matrices $\nabla f(\mathcal{X}) \triangleq \{\nabla f(x) : x \in \mathcal{X}\}$ and considering a special LPV known as a linear differential inclusion (LDI)

$$\dot{\xi}(t) = A(t)\xi(t), \qquad A(t) \in \text{conv}\nabla f(\mathcal{X}). \tag{3.9}$$

The Jacobian can be computed numerically using finite differences. Again, it can be shown that every trajectory of the nonlinear system (3.7) is also a trajectory of the LDI [15, p.55], so a stability certificate for the LDI is also a global stability certificate for the nonlinear model.

Global linearization is conservative for the same reason that quasi-LPV is conservative. Whereas quasi-LPV can be made less conservative by reformulation, there's nothing we can do about global linearization. Whenever $f(\cdot)$ is highly nonlinear, then the space of Jacobian matrices is "large".

3.2.3 Local Linearization

Finally, we may also use LPV techniques to certify the local stability of the nonlinear model (3.7). Given a known trajectory $\tilde{x}(t) \in \mathcal{X}$ that satisfies the equation (3.7), we say that the the nonlinear model is *locally stable about* $\tilde{x}(t)$ if every trajectory satisfying $x(t) \approx \tilde{x}(t)$ converges onto $\tilde{x}(t)$, i.e. $x(t) \to \tilde{x}(t)$ with $t \to \infty$. For example, the simplest trajectory is to fix the state variable at the equilibrium, $\tilde{x}(t) = 0$. Making a first-order expansion yields the linear time-varying model

$$\dot{\xi}(t) \approx A(\tilde{x}(t))\xi(t) \tag{3.10}$$

in which we have defined the Jacobian matrix-valued function $A(\delta) \triangleq \frac{\partial f}{\partial x}\big|_{x=\delta}$ and the deviation term $\xi(t) = x(t) - \tilde{x}(t)$. Certifying (3.10) to be stable using the same LPV techniques discussed above also certifies the nonlinear model to be locally stable about $\tilde{x}(t)$.

Local stability certification tends to be considerably less conservative than global stability certification (i.e. it gets closer to being necessarily and sufficient). On the other hand, its guarantees should be interpreted with care, since they are valid only within a local neighborhood of the intended trajectory. The local stability approach forms the basis for a class of nonlinear controller design techniques known as gain scheduling [37, 38].

3.3 Case Study: Robust Small-Signal Stability

Our first case study is an example of the *robust small-signal stability* problem: certifying the stability of the LPV model

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = A(\delta(t))x(t), \qquad \delta(t) \in \Delta, \qquad \dot{\delta}(t) = 0.$$

in which δ is high-dimensional and fixed-but-uncertain. We use the quadratic stability test to analyze robust small-signal stability, and a branch-and-bound strategy to reduce conservatism. Our results find this combined approach to be surprisingly effective, at least for this particular problem.

3.3.1 Motivation

High penetrations of renewable energy resources will introduce unprecedented uncertainty to the power system, making stability analysis considerably more challenging. Previous studies have generally concurred that–assuming that sufficient inertia remains on the system–high penetrations of renewables generation would not significantly worsen small-signal stability in the *average-case scenario* [39,40]. Considerably less is known in the *worst-case scenario*, whether it is possible for renewables to significantly impact stability, and how might such issues manifest.

One popular approach is to apply statistical, or "Monte Carlo" techniques, to the study of small-signal stability in the presence of uncertainty [41,42]. By sampling the stability of selected or random scenarios, and by assuming a particular underlying distribution, a prediction interval can establish that stability is *expected* for, say, 99% of all possible scenarios. However, such predictions can often mislead if the impact of significant outliers are not adequately considered.

In this section, we present a case study based on the IEEE 118-bus test system, in which 30% of the power is generated by distributed renewable generation added to each of the 118 buses, in order to highlight the challenge presented by outliers, and the importance of a certification approach. First, three conventional generators (located at buses 10, 25 and 89) are retired from the system. Then, the displaced generation capacity (around 1,277 MW) is compensated by installing renewable generation at each of 118 buses throughout the system. In the "base case" scenario, the amount of distributed generation allocated to each bus is proportional to the size of the existing load, in order to reflect the fact that larger load centers tend to accrue more renewables.

Statistical analysis shows the system to be stable and unassuming on average, with the boundary to instability located more than 6 standard deviations away. But using local optimization, we were able to find 100 unstable or nearly unstable scenarios, suggesting that outlier scenarios may be far more common than first appeared. Finally, a stability certificate for a low-dimensional, linearized version of our system model is computed, bounding the worst-case instability by a figure that is not too much worse than the unstable scenarios found via local optimization.

3.3.2 System Description

The IEEE 118-bus model is a classic test case, containing 118 buses, 186 lines, and 54 generators. Descriptions of the system are widely available, e.g. from [43]. The system contains a large number of generators, but only 17 of which are actively generating more than 10 MW power. To simplify the analysis, the remaining 37 generators are taken out of service, but their respective buses are left intact. Each time power flow

is solved throughout the system, reactive power limits at each generator are enforced to prevent the obvious instability caused by sinking too much reactive power into a generator.

The base power for each machine is computed by taking the $P_{\max}$, $Q_{\max}$ and $Q_{\min}$ figures quoted in the power flow case file, and assuming that the capability of each machine is to produce up to 1.0 per-unit real power, 0.8 per-unit reactive power in over-excitation, and 0.6 per-unit in under-excitation. These are typical figures for large, transmission-level synchronous machines [44].

The dynamical model for each generator is constructed from a standard round-rotor generator model (GENROU), a standard DC exciter model (DC1A [45, Sec. 5.1]), alongside a suitably designed voltage compensator [45, Sec. 4]. Governors are generally considered to be too insensitive to initiate small-signal events, so are not modeled for this study. Identical per-unit parameters are rescaled to different machine base powers.

Loads are modeled as a mixture of 70% constant-impedance and 30% constant-current, with negligible dynamics. The constant-impedance portion models lights, heaters and appliances, as well as the various transformers and lines in the conduction path, while the constant-current portion models induction motors, which are widespread for industrial loads.

The renewable generation at each bus is modeled using ZIP models as 90% real power injections and 10% direct-axis current injections. The vast majority of renewable resources interface with the power system through power electronics, which have near-instantaneous dynamics that can be neglected for the purposes of a transmission-level simulation. In the presence of power-point tracking mechanisms, these renewables will act as constant real power injections; without power tracking, they will behave like direct-axis current sources. Since there is no mandate in the U.S. for small, distribution-level renewables to provide reactive support, we simply assume that their reactive power contributions are negligible.

3.3.3 LPV Formulation

The objective of our study is to simulate the uncertainty associated with renewables generation, and to quantify its impact on system-wide small-signal stability. To this end, we define the uncertain set as an 118-dimensional hypercube $\Delta = [0, 2]^{118}$. Each i-th parameter variable δ_i is assumed to have a fixed-but-unknown value, acting as a "multiplier" for the production at the corresponding bus. For example, a value of $\delta_5 = 1.2$ would set the renewable generation at bus 5 to output 20% more power than its nominal amount in the "base case", whereas a value of $\delta_{20} = 0$ would shut off the renewable generation at bus 20 altogether.

Our LPV model is constructed using the "parameterized linearization" approach. We begin with the standard differential algebraic model of the power system described in the previous section,

$$\dot{x}(t) = f(x(t), y(t), \delta), \qquad 0 = g(x(t), y(t), \delta), \tag{3.11}$$

Table 3.1: Decay rate sample statistics (units of $10^{-2}/s$)

Sample size	Mean	Median	Mode	Std. Dev.	Min	Max
60	3.05	3.13	3.87	0.433	1.81	3.86
600	2.99	3.02	4.28	0.421	1.43	4.28
6000	2.98	3.00	4.32	0.439	0.900	4.32
60,000	2.99	3.00	4.63	0.437	0.746	4.63
360,000	2.99	3.00	4.64	0.437	0.584	4.64

Table 3.2: Decay rate prediction intervals (units of $10^{-2}/s$)

Sample size	Prediction Confidence			
	99%	99.9%	99.99%	99.999%
60	[1.89, 4.21]	[1.54, 4.56]	[1.23, 4.87]	[0.94, 5.16]
600	[1.90, 4.08]	[1.60, 4.38]	[1.34, 4.64]	[1.11, 4.87]
6000	[1.85, 4.11]	[1.54, 4.42]	[1.27, 4.69]	[1.04, 4.92]
60,000	[1.86, 4.12]	[1.55, 4.43]	[1.29, 4.69]	[1.06, 4.92]
360,000	[1.86, 4.12]	[1.55, 4.43]	[1.29, 4.69]	[1.06, 4.92]

in which $x(t), y(t)$ are state and algebraic variables, and f, g are nonlinear but smooth, differentiable functions, and δ is the vector of fixed-but-unknown parameters. Using an underlying power flow model, we define the functions $x_0(\delta)$ and $y_0(\delta)$ to parameterize a set of equilibrium points satisfying

$$f(x_0(\delta), y_0(\delta)) = 0, \qquad g(x_0(\delta), y_0(\delta)) = 0$$

for every $\delta \in \Delta$. Linearizing the nonlinear system about the each equilibrium $x_0(\delta)$, $y_0(\delta)$ yields the descriptor space

$$\dot{\xi}(t) = A(\delta)\xi(t) + B(\delta)\eta(t), \qquad 0 = C(\delta)\xi(t) + D(\delta)\eta(t), \tag{3.12}$$

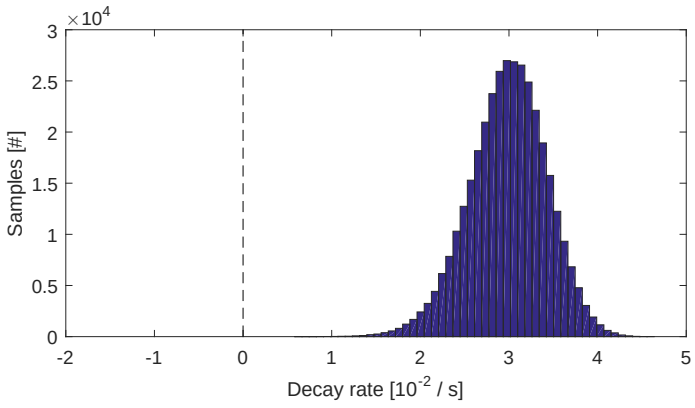
where $\xi(t) = x(t) - x_0(\delta)$ and $\eta(t) = y(t) - y_0(\delta)$, and the matrices $A(\delta), B(\delta), C(\delta), D(\delta)$ are the Jacobians of the functions f and g evaluated at $(x_0(\delta), y_0(\delta), \delta)$. The matrix $D(\delta)$ is nonsingular except in cases of voltage collapse; eliminating the variable $\eta(t)$ yields the LPV model

$$\dot{\xi} = M(\delta)\xi, \qquad \delta \in [0, 2]^{118},$$

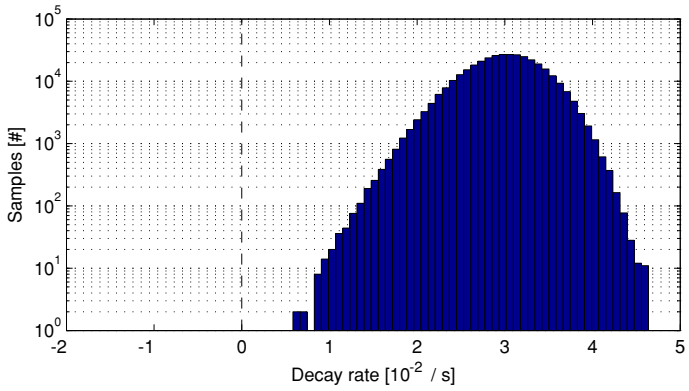
where $M(\delta) \triangleq A(\delta) - C(\delta)D(\delta)^{-1}B(\delta)$.

3.3.4 Statistical Analysis

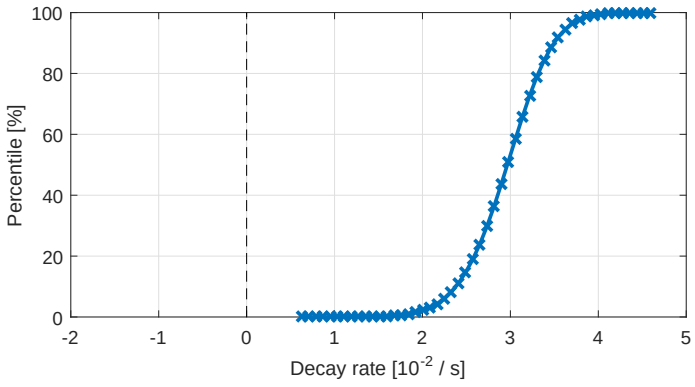
In comparing the relative stability of different scenarios, it is often helpful to quantify the stability of the system with a number. For this case study, we use the *decay rate*, in units of "fraction reduction per second", which refers to the exponential damping rate for the lightest-damped eigenmode of a linear system. Specifically, given the LTI



(a)



(b)



(c)

Figure 3-1: Distribution of decay rate over 360,000 samples: (a) histogram; (b) histogram in log scale; (c) cumulative probability distribution.

system $\dot{x} = Mx$, the damping rate, α, is defined as

$$\alpha(M) = -\max_i \mathrm{Re}\lambda_i\{M\}. \tag{3.13}$$

With a positive decay rate, i.e. $\alpha(M) > 0$, the LTI system will decay exponentially towards steady-state as $\sim \exp(-t\alpha(M))$. For a nonpositive decay rate, i.e. $\alpha(M) \leq 0$, the LTI system is unstable.

Our objective is to analyze the worst-case decay rate over all possible choices of the uncertain parameter δ. Defining the set of all possible decay rates as

$$\mathscr{D} = \{\alpha(M(\delta)) : \delta \in \Delta\} \subset \mathbb{R}, \tag{3.14}$$

the worst-case decay rate is the *minimum* of this set, i.e. $\min \mathscr{D}$. Unfortunately, solving this minimization is, in general, nonconvex and NP-hard.

However, sampling from $\mathscr{D}$ is relatively straightforward. With enough samples, quantitative statements about stability can be made using statistical analysis, and a prediction interval can be made for the worst-case decay rate. Table 3.1 shows the results obtained by sampling δ from an 118-dimensional uniform distribution. The histogram / cumulative distribution function for the largest sample size is shown in Fig. 3-1. Assuming an underlying normal distribution, prediction intervals are computed for each sample size and shown in Table 3.2.

The results suggest that an "average" uncertain scenario is relatively unassuming, admitting a decay rate of ∼3% per second, corresponding to a damping ratio of around 1-2%. On average, a high penetration of distributed renewables does not appear to significantly impact system stability, at least within the modeling assumptions contained in this paper. This result concurs with studies performed on real power systems [39, 46].

All five intervals predict around a 1 in 100,000 chance for a scenario to admit a decay rate being below 1% per second. However, examining the statistics closer suggests that the assumption of a normal distribution may be overly optimistic. There is considerable skew and kurtosis (i.e. "long-tailed-ness") in the distribution, and the mode differs considerably from the mean and median for all sample sizes in Table 3.1. Scenarios with decay rates below 1% per second are found in practice within just 6000 samples. While the distribution may appear to be normal at first glance, the results show that outlier cases are far more common, and this can lead to large errors when making predictions.

3.3.5 Unstable Scenarios via Local Optimization

While the minimization problem for the worst-case decay rate

$$\min \mathscr{D} = \min_{\delta \in \Delta} \alpha(M(\delta)) \tag{3.15}$$

is generally intractable, finding locally optimal solutions is easy: a strictly feasible initial point can be incrementally improved, e.g. using a trust-region quasi-Newton's

method, until no further progress can be made. Since the problem is highly nonconvex, we would expect numerous local minima to be found, so the procedure should be repeated with different, randomized initial points.

Any unstable or insufficiently stable scenario found through local optimization immediately tells us that not all uncertain scenarios are acceptable; the worst-case is at least as bad the one found through optimization. The corresponding choice of δ can be thought of as a "suboptimal but good enough" solution to the optimization problem (3.15), because it allowed us to draw a definitive conclusion. The use of local optimization to look for "suboptimal but good enough" solutions to highly nonconvex problems is ubiquitous in applications ranging from controller synthesis [47] to machine learning.

However, failure to find an unstable scenario does not mean that one does not exist. We can only increase our chances of catching the worst-case by restarting the search at different initial points. And even if the worst-case scenario were found, it would be indistinguishable from a locally least-stable scenario. While local optimization is an effective heuristic for stability analysis, it cannot be relied upon for stability guarantees.

Statistical approaches can never be conclusive about *worst-case* behavior, but in this case study, local optimization shows that the statistic approach is surprisingly misleading. After 100 runs of local optimization performed using `fmincon` in MATLAB, 100 locally least-stable solutions are found. These solutions are visualized in Fig. 3-2, as the 0%, 25%, 50%, 75% and 100% quantiles for the δ values allocated to each system bus. The solutions span a wide combined range, but many of them are closely gathered towards a median "bad-case scenario". As shown in Fig. 3-3, all 100 scenarios are considerably less stable than those sampled in the previous section, deviating a remarkable 6-7 standard deviations from the mean.

It is important to validate that the instabilities found correspond to real, physical phenomena, and are not simply a manifestation of the optimizer exploiting modeling errors. We provide an illustration for the most unstable of the 100 solutions, which has an eigenvalue pair at $\lambda = 0.002 \pm 3.9512j$. The instability would manifest as a 0.6 Hz oscillation that grows in magnitude at a rate of 0.2% per second, or about 12% per minute. Computing the participation factors [48] reveals that only machine rotor speeds and rotor angles participate in the unstable modes. The most affected machines at bus 25 and bus 111, which are located at opposite extremes of the network. These are all tell-tale signs of interarea oscillation; indeed the suspicion is confirmed using time-domain simulation, as shown in Fig. 3-4.

3.3.6 Stability Guarantees via Stability Certification

In the previous subsection, local optimization was successful in catching several unstable or nearly-unstable outlier scenarios. But both local optimization and statistical analysis will inevitably fall short in making conclusive predictions about the true worst-case scenario. Both methods leave us wondering whether there is a significantly less stable scenario that is simply overlooked.

Instead, we may quadratic stability certificates to produce lower-bounds on the

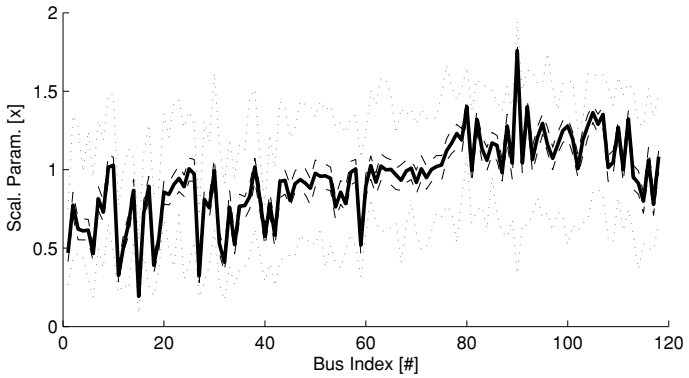


Figure 3-2: Solution distribution for 100 runs of local optimization. Black, solid: median; black, dashed: 25% and 75% quantiles; gray, dotted: 0% and 100% quantiles.

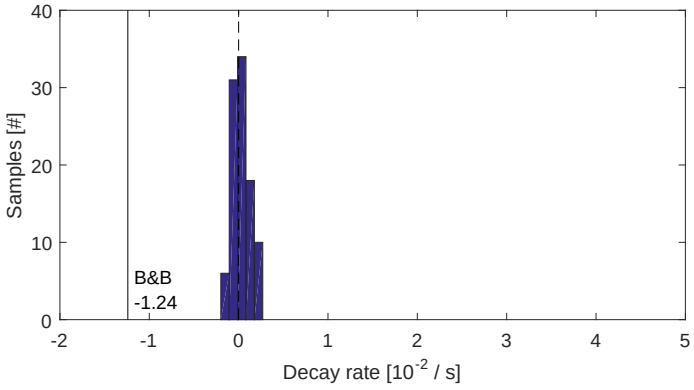
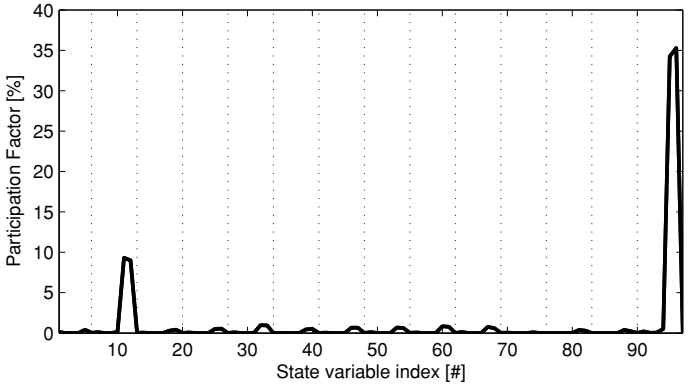
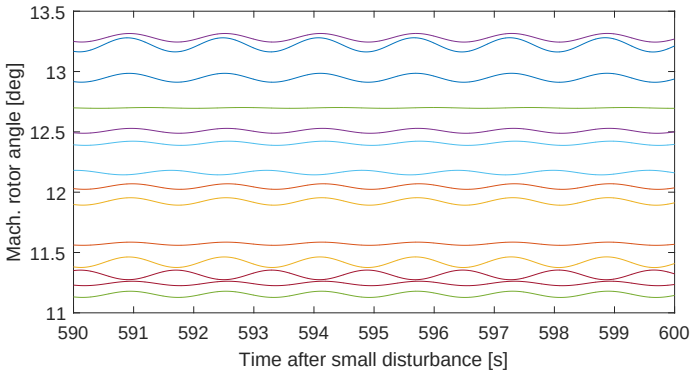


Figure 3-3: Histogram of damping rates for 100 locally least-stable scenarios found via local optimization. The least stable scenario has a damping rate of $-0.002s^{-1}$, and branch-and-bound on the order-reduced, linearized approximation predicts $\alpha \geq -0.0124s^{-1}$ in the neighborhood of these scenarios.



(a)



(b)

Figure 3-4: Data from an example unstable scenario: (a) Participation factors of the generator state variables, revealing the most affected machines as being at buses 25 and 111; (b) Time domain simulation of the rotor angles after a small disturbance. The oscillations are undamped and grow slowly in magnitude.

worst-case decay rate. If this lower-bound is positive and sufficiently large, then we may conclude our stability analysis with the knowledge that the worst-case scenario is guaranteed to be sufficiently stable.

Proposition 7. *Chose $\lambda \in \mathbb{R}$ to make the identity-shifted LPV*

$$\dot{\xi}(t) = [M(\delta(t)) + \lambda I]\xi(t) \qquad \delta(t) \in \Delta \tag{3.16}$$

quadratically stable (Definition 5). Then λ is a strict lower-bound on the worst-case decay rate, i.e. $\lambda < \min_{\delta\in\Delta} \alpha(M(\delta))$.

Proof. We claim that the worst-case decay rate of this identity-shifted model must be strictly positive, i.e. $\min_{\delta\in\Delta} \alpha(M(\delta) + \lambda I) > 0$. Suppose that this were not the case, i.e. there exist some $\delta^\star$ such that $\alpha(M(\delta^\star) + \lambda I) \leq 0$. Then selecting the parameter to be time-invariant $\delta(t) = \delta^\star$ would make the LPV (3.16) unstable, thereby contradicting the premise that (3.16) is quadratically stable. Next, applying the algebraic identity $\alpha(A + \lambda I) = \alpha(A) - \lambda$ yields

$$\min_{\delta\in\Delta} \alpha(M(\delta) + \lambda I) > 0 \qquad \Longleftrightarrow \qquad \min_{\delta\in\Delta} \alpha(M(\delta)) > \lambda,$$

so we find that λ serves as a lower-bound on the worst-case decay rate of the original model. □

Maximization over all valid choices of λ yields classic LMI-based decay rate lower-bound [15, pp.66-67]

$$\alpha_{\text{lb}}(M, \Delta) \triangleq \sup_{\lambda\in\mathbb{R}} \{\lambda : (3.16) \text{ is quadratically stable}\}. \tag{3.17}$$

The conservatism of this bound may be reduced by partitioning the uncertainty set. More specifically, let us partition $\Delta = \Delta_1 \cup \Delta_2 \cup \cdots \cup \Delta_p$. Then it is possible to show that

$$\alpha_{\text{lb}}(M, \Delta) \leq \min_{i\in\{1,\ldots,p\}} \alpha_{\text{lb}}(M, \Delta_i) \leq \min_{\delta\in\Delta} \alpha(M(\delta)). \tag{3.18}$$

As the partitions are made smaller, the second inequality in (3.18) approaches an equality, and the corresponding lower-bound approaches the true global minimum [49]. This sort of partitioning may be incorporated into a branch-and-bound framework to yield a global optimization algorithm, as was done in [50, 51].

Evaluating the lower-bound (3.17) requires us to solve a quasi-convex optimization problem using the bisection method. Each iteration fixes the value of λ and attempts to compute a quadratic stability certificate for (3.16). In order to perform the quadratic stability test using standard techniques, the individual elements of the matrix-valued function $M(\delta)$ should be given as rational functions of δ. Unfortunately in our case, $M(\delta)$ does not even admit a closed-form solution, since a part of its evaluation requires solving the power flow equations using Newton's method.

Instead, viewing $M(\delta)$ as a black-box model, we can *approximate* $M(\delta)$ using a polynomial matrix-valued function $\tilde{M}(\delta)$, e.g. by collocation at a finite number of

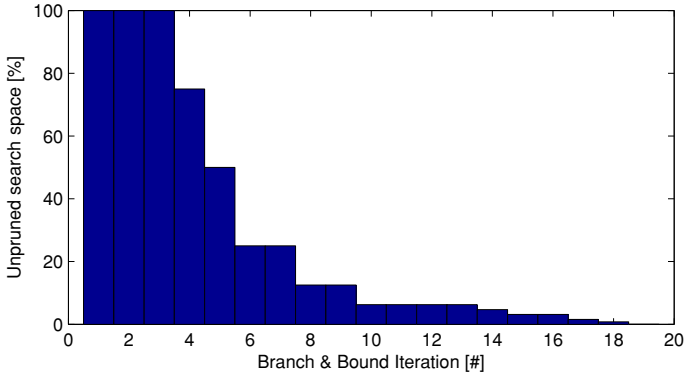


Figure 3-5: Progress of branch & bound as applied to the linearized model. Each iteration took between 2,000-4,000 s on a 16-core Intel Xeon E5 CPU.

points $\{x_1, \ldots, x_p\} = \Delta_c \subset \Delta$. Such an approach is well-known and widely adopted. For this case study, we chose $\tilde{M}(\delta)$ to be an affine function

$$\tilde{M}(\delta) = \tilde{M}_0 + \delta_1 \tilde{M}_1 + \cdots + \delta_{118} \tilde{M}_{118},$$

and select the collocation points Δ_c to lie within a neighborhood of the worst locally-unstable scenario found earlier via local optimization. In effect, we are constructing the first-order expansion for $M(\delta)$ about this locally-unstable scenario. The physical intuition is to replace the a.c. power flow equations with the "d.c. power flow" analog; this approach is sometimes used to formulate optimal power flow [52] and unit commitment [53] problems as linear and mixed integer programs.

The resulting affine matrix-valued function is a function of 118 uncertain parameters, but most of these are redundant in describing the underlying uncertainty. Applying order reduction, we find that just 7 dimensions of uncertainty (i.e. principal components) are needed to capture the behavior of the LPV model to an approximation error of below 1%. The vertex-based formulation in Section 3.1.2 is used to compute quadratic stability certificates for this reduced model, and a branch-and-bound scheme similar to the one in [50] is used to refine the conservatism of the bound.

The final result is shown alongside the decay rates of the unstable scenarios found via local optimization in Fig. 3-3. The certificate predicted a decay rate lower-bound of $-0.0124/s$, suggesting that the worst-case scenario should not be too much worse than the outlier scenarios already found using local optimization. Despite the use of a reduced-order approximation, each stability certificate still required a significant amount of computational effort. Branch-and-bound converged in 19 iterations (shown in Fig. 3-5), but each iteration required the solution of a conic optimization problem with $97^2 = 9409$ primal decision variables and $2^7 \cdot 97^2 \approx 1.2 \times 10^6$ dual decision variables. This took around an hour each time, even when using application-specific, custom-tailored codes on expensive hardware, and the combined running time for all 19 iterations was around a full day.

3.4 Case Study: Robust Large-Signal Stability

Our second case study is an example of the *robust large-signal stability* problem: certifying the stability of the LPV model

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = A(\delta(t))x(t), \qquad \delta(t) \in \Delta,$$

in which $\delta(t)$ is time-varying and may vary arbitrarily quickly. We use quadratic stability as a proxy for robust large-signal stability. While the approach has a reputation for being conservative, our results find it to be surprisingly effective.

3.4.1 Motivation

The uptake of high penetrations of renewable energy raises concerns that their second-to-second variability in power output, which we refer to as *intermittency*, may cause the power system to become unstable [39, 42, 54]. A common approach to certifying stability is via small-signal stability analysis, in which a detailed, nonlinear model of the system is linearized, and the locations of its eigenvalues used to guarantee stability. But the approach hinges on two implied assumptions: that the models are sufficiently accurate to capture the desired modes of instability, and that the intermittency itself is small-signal with respect to the greater power system.

Both assumptions hold up well in the context of large transmission networks with low to moderate penetrations of renewable energy. Synchronous machines models are mature, and although not always precise, their limitations and predictive powers are well-quantified and well-understood from decades of experience. Also, renewable intermittency really does appear as small-signal in a transmission network, due to geographical diversification, which tends to "average out" the magnitude and rate-of-change associated with network-wide intermittency [55, 56].

Unfortunately, neither assumptions are particularly realistic for smaller systems like distribution networks and microgrids. Here, the generator, solar panel, and load models are often novel and nonstandard, and the localized renewable penetration can often be high enough to subject large-signal intermittency to the rest of the system. Insisting on the use of small-signal stability analysis can lead to overly optimistic conclusions.

In this section, we illustrate the limitations of small-signal stability analysis for large-signal intermittency, and the use of a computational approach to Lyapunov functions to overcome these limitations. We examine a simple microgrid whose operating point is kept (approximately) constant using a fast-acting energy storage unit. We compute stability margins for the example system based on eigenvalue analysis, and show that the system can nonetheless be destabilized under large-signal intermittency from a solar panel. Then, we derive stability margins using Lyapunov analysis that remain valid regardless of the nature of the solar panel output. Finally, we show that the small- and large-signal stability margins are related by the maximum allowable slew-rate of the solar panel output.

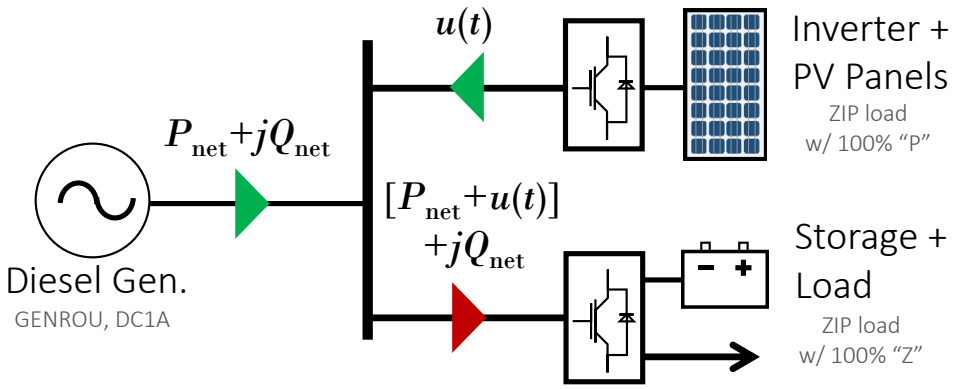


Figure 3-6: Schematic of the microgrid under study at nominal 1 p.u. bus voltage. $P_{\text{gen}} = 1$ p.u., $Q_{\text{gen}} = -0.3$ p.u.

3.4.2 System Description

Our goal in this case study is to compare the small- and large-signal stability certification approaches on a microgrid that is simple enough for the corresponding results to be intuitive. To this end, let us consider a system containing just three distinct elements: a diesel generator, a solar panel connected to the system via an inverter, and a load, buffered by a rapid-responding form of energy storage, such as a battery bank or a flywheel. The energy storage is used to provide regulation against load variations, and against the possibly intermittent output of the solar panel, so that the diesel generator may approximately operate in steady-state. The arrangement is shown in Fig. 3-6. Under ideal conditions, i.e. with nominal bus voltage magnitude, the net load seen by the diesel generator would remain perfectly constant, irrespective of the solar panel output.

We use algebraic impedance-current-power (ZIP) models to represent the load-storage combination and the solar panel inverter, without any associated dynamics. Since both elements would be realized using power electronics, it is reasonable to assume that their behavior would be sufficiently fast as to be considered instantaneous, at least from the perspective of the diesel generator. The ZIP ratio for the load-storage combination is set to 100% impedance, to reflect the voltage droop characteristic typically used to enhance small-signal voltage stability. The inverter ZIP ratio is set to 100% power, to capture the presence of a maximum power-point tracking controller. Small-scale solar panel inverters are very rarely configured to provide reactive power support, so we assume that only active power is produced.

We use standard models to represent the diesel generator. More specifically, the classic round-rotor model GENROU is used to model the magnetic circuit and swing behavior, and the IEEE DC1A exciter model and a suitably tuned voltage compensator are used to model the automatic voltage regulator. Governors and prime-movers are too slow and too insensitive to participate in the instabilities of interest in this case-study, so are not modeled; the machine rotor speed variable is simply initiated at its nominal values and left uncontrolled. The system is configured so that the net load seen by the diesel generator is $P_{\text{gen}} = 1.0$ p.u. and $Q_{\text{gen}} = -0.3$ p.u. (at nominal

conditions of 1 p.u. voltage and frequency).

3.4.3 LPV Formulation

At time t, let the real vector $x(t) \in \mathbb{R}^n$ contain the generator state variables, let the complex scalar $v(t) \in \mathbb{C}$ describe the voltage phasor at the interconnecting bus, and let the real scalar $0 \leq u(t) \leq u_{\max}$ describe the solar panel inverter active power output in per-unit. We use a differential-algebraic model of the microgrid, cast into nonlinear descriptor state-space form

$$E\dot{y}(t) = f(y(t), u(t)), \tag{3.19}$$

in which E is the singular matrix

$$E = \begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix},$$

the vector $y(t) = [x(t); v(t)]$ is the generalized state variable.

We use y_{nom} to refer to the system's nominal operating point. At the nominal system conditions of 1 p.u. voltage and frequency, we assume that the regulation from the energy storage system is sufficiently fast and accurate as to *decouple* the operating point from the inverter output $u(t)$. Mathematically, the energy storage system is modeled to allow the point y_{nom} to solve the steady-state conditions

$$0 = f(y_{\text{nom}}, u(t)),$$

for every choice of $u(t)$. Hence, every choice of $y(t) = y_{\text{nom}}$ and $u(t)$ yields a valid solution for (3.19). Or physically, starting perfectly at the operating point, $y(0) = y_{\text{nom}}$, the system would remain fixed at the equilibrium, with $y(t) = y_{\text{nom}}$, irrespective of the solar panel output $u(t)$.

We develop two LPV models in this case study. Our first model, named the *nominal operating point* model, is obtained by linearizing about all trajectories about the system's nominal operating point. This is done by setting the uncertain parameter as the solar panel output, $\delta(t) \triangleq u(t)$, and defining the matrix-valued function

$$M_{\text{nom}}(u(t)) \triangleq \left.\frac{\partial f}{\partial x}\right|_{y=y_{\text{nom}}, u=u(t)}$$

to yield the LPV model

$$\dot{\xi} = M_{\text{nom}}(\delta(t))\xi, \qquad 0 \leq \delta(t) \leq u_{\max}, \tag{3.20}$$

where $\xi = y(t) - y_{\text{nom}}$. It is worth noting that the matrix-valued function $M_{\text{nom}}(\delta(t))$ is *affine* by construction, meaning that it can be written

$$M_{\text{nom}}(\delta(t)) = A_0 + \delta(t)A_1,$$

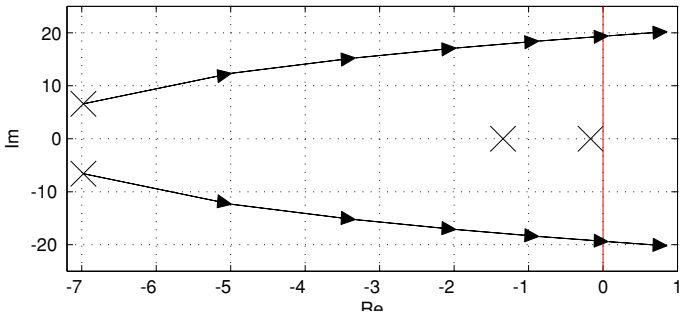


Figure 3-7: Root locus plot for the 4 least stable poles of the system. The markers show $u_0 = 0$, and each arrowhead increments u_0 by 2. The poles reach the imaginary axis at $u_0 = 9.8347$.

where $A_0 = M_{\text{nom}}(0)$ and $A_1 = M_{\text{nom}}(1)$. The physical explanation is simply that the admittance of a ZIP load is, by definition, linear with respect to its designated impedance, current, and power values (controlled by u), and nonlinear only with respect to the bus voltage (controlled by y).

Unfortunately, stability guarantees developed for the nominal operating point model are only valid when the system operating point is within a small-signal neighborhood of the nominal. Our second LPV model, named the *uncertain operating point* model, performs global linearlization over an entire uncertainty set of operating points. To do this, we define the uncertainty parameter to incorporate the 5 elements of y that affects the Jacobian $\frac{\partial f}{\partial x}$:

- The generator direct- and quadrature-axis flux variables, with default values of $\varphi_d = 0.862$ p.u. and $\varphi_q = -0.507$ p.u.
- The generator rotor speed and rotor angle, with default values of $\omega_{\text{rot}} = 1$ p.u. and $\delta_{\text{rot}} = 30.4$ degrees.
- The interconnecting bus voltage magnitude, with default value of $|v| = 1$ p.u.

Setting $\delta = [\varphi_d, \varphi_q, \omega_{\text{rot}}, \delta_{\text{rot}}, |v|, u]^T$, defining $M(\delta(t)) \triangleq \frac{\partial f}{\partial x}$ and the uncertainty set $\mathcal{Y}_0$ to enforce the bounds $\varphi_d \in [0.77, 0.94]$, $\varphi_q \in [-0.56, -0.45]$, $\omega_{\text{rot}} \in [0.95, 1.05]$, $\delta_{\text{rot}} \in [25, 35]$, and $|v| \in [0.95, 1.05]$ produces our LPV model

$$\dot{\xi} = M(\delta(t))\xi, \qquad \delta(t) \in \mathcal{Y}_0 \times [0, u_{\max}]. \tag{3.21}$$

3.4.4 Eigenvalue analysis

We begin by examining the small-signal stability of the system at the nominal operating point over a range of solar outputs. In other words, let us fix $y_0 = y_{\text{nom}}$ and sweep the value of u_0 upwards from zero. Plotting the trajectory of the least-stable eigenvalues of $E\dot{y} = f(y, u)$ at $y = y_0$ and $u = u_0$ yields the root locus plot shown in Fig. 3-7. The eigenvalues remain in the left-half of the complex plane for all $0 \leq u_0 < 9.8347$. Hence, we conclude that, subject to the assumption of small-signal disturbance and

perfect regulation from the storage unit, the system can accommodate for up to $9.8347/(9.8347+1) = 90.770\% \approx 90.8\%$ solar penetration.

Next, we recompute the small-signal stability margin while allowing the operating point to (slowly) vary within an uncertainty set, $\mathcal{Y}_0$, containing the nominal operating point. We sweep the value of u_0 upwards from zero, while validating that that system $E\dot{y} = f(y,u)$ at $y = y_0$ and $u = u_0$ has eigenvalues in the left-half plane for every $y_0 \in \mathcal{Y}_0$. This eigenvalue check over all infinite choices of the operating point $y_0 \in \mathcal{Y}_0$ is approximated by sampling over a 5-dimensional, uniformly spaced Cartesian grid, with 7 points per side for a total of 16,807 samples. We find that the system remains stable under uncertainty for all $0 \leq u_0 < 3.8402$ p.u. This corresponds to a solar penetration of $3.8402/(3.8402+1) = 79.34\% \approx 79.3\%$ solar penetration, which is considerably reduced from the figure in the previous subsection.

3.4.5 Limitations of Eigenvalue Analysis

The central weakness of an eigenvalue-based analysis, however, is its reliance on the small-signal disturbance assumption, which often causes its predictions to be too optimistic for practical use. Let us examine an example choice of $u(t)$ that satisfies the stability margin computed in the previous subsection, but actually causes the system to be unstable. To keep the example relatively simple, we will fix the system initial state $y(0)$ within a small-signal neighborhood of its nominal operating point, y_{nom}.

Consider switching the solar output $u(t)$ between the values of 0 and 9 p.u. with 50% duty cycle and a period of $T = 0.21$ second,

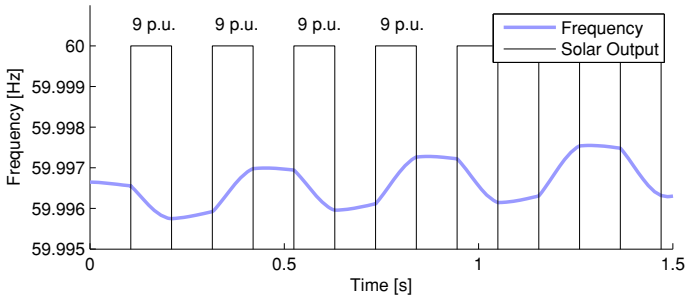
$$u(t) = \begin{cases} 0 & kT \leq t < \left(k+\frac{1}{2}\right)T, \\ 9 & \left(k+\frac{1}{2}\right)T \leq t < (k+1)T, \end{cases}$$

for all $k \in \{1, 2, \ldots\}$, as shown in Fig. 3-8a. The small-signal dynamics of this system are governed by the nominal operating point LPV model

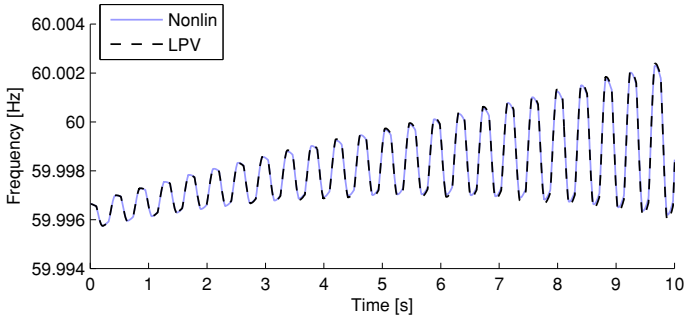
$$E\dot{\xi}(t) = M_{\text{nom}}(u(t))\xi(t) + w(t). \tag{3.22}$$

Essentially, the LPV model switches between the two linear time-invariant (LTI) models, $E\dot{\xi}(t) = M_{\text{nom}}(0)\xi(t)$ and $E\dot{\xi}(t) = M_{\text{nom}}(9)\xi(t)$, at the prescribed period and duty cycle. Each instantaneous value of $u(t)$ satisfies the stability margin computed in the previous subsection, so the eigenvalues of each LTI model reside strictly in the open left-half of the complex plane. We may be tempted to conclude that the switched system is also small-signal stable.

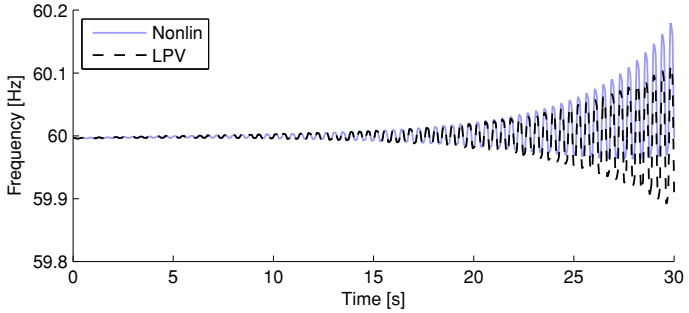
Yet the small-signal disturbance assumption is violated, since the time-varying component of $u(t)$ is much too large to be considered negligible. And as shown in Fig. 3-8b and Fig. 3-8c, the switched system is not stable. Starting with an initial point $y(0)$ that is a small-signal perturbation (with relative Euclidean norm of 10^{-6}) away from y_{nom}, we discover a nonlinear "mode" that accumulates and destabilizes the system. This mode of instability persists into the original nonlinear model in (3.19);



(a)



(b)



(c)

Figure 3-8: An unstable choice of $0 \leq u(t) \leq 9.8$: (a) The choice of $u(t)$, switching between 0 and 9 p.u. with a period of 0.21 s, and the resulting system response; (b) The unstable response over the first 10 seconds; and (c) Over the first 30 seconds.

we see an excellent agreement between the LPV model and the fully nonlinear model in the small-signal regime[3].

Note that we are able to induce this instability despite a number of simplifications and idealizations intended to stabilize the model:

- Perfect regulation from the energy storage unit, which decouples the nominal operating point from the instantaneous solar panel output;
- The initial condition is confined within a small-signal neighborhood of the nominal operating point;
- The large-signal behavior is confined to the solar panel output, and not the generalized state variables.

In practice, these idealizations are unlikely to hold, and this would only further exacerbate potential instabilities. For example, without perfect regulation from the energy storage unit, we may expect rapid, switching behavior to occur in both $u(t)$ as well as $y(t)$. This would further invalidate the small-signal disturbance assumption, and suggests that the more conservative stability margin in the previous subsection may still be too optimistic for practical use.

3.4.6 Lyapunov Analysis

Lyapunov analysis overcomes many of the limitations of eigenvalue analysis for nonlinear systems. It certifies nonlinear stability in the presence of large-signal intermittency, by demonstrating the existence of a Lypuanov function. Historically, the analytical difficulties in deriving the Lyapunov function have prevented the approach from seeing widespread use. Instead, an optimization approach can be used to shift the analytical burden to a computational one. By formulating the Lyapunov function candidate as a generic polynomial function, its polynomial coefficients may be obtained by solving a convex optimization problem.

First, we attempt to compute a quadratic Lypuanov function of the form $V(y) = y^T E^T P y$ for the nominal operating point LPV model (3.20). If it exists, then the original nonlinear model is guaranteed to be stable for arbitrarily large changes in $u(t) \in [0, u_{\max}]$, so long as $y(t)$ remains within a small-signal neighborhood of y_{nom}. The candidate $V(y)$ is a Lyapunov function if and only if its coefficient matrix P satisfies the LMIs

$$E^T P = P^T E \succeq 0, \tag{3.23}$$

$$M_{\text{nom}}(\delta)^T P + P^T M_{\text{nom}}(\delta) \prec 0, \tag{3.24}$$

for all $\delta \in \{0, u_{\max}\}$. We use the YALMIP parser [57] to set up these three constraints, and the MOSEK interior-point solver [58] to find a feasible point.

[3]The same simulation was repeated using two different integrators (ode23t and ode15s) and over a wide range of error tolerances, to ensure that the instability is not caused by numerical errors.

Repeating the above procedure with incrementing $u_{\max}$, we find that the nominal operating point is large-signal stable for up to $u_{\max} = 5.9713$ p.u. Hence, we conclude that, assuming perfect regulation from the energy storage unit, and that the system is initialized within a small-signal neighborhood of the nominal operating point, the system can accommodate for up to $5.9713/(5.9713 + 1) = 85.655\% \approx 85.7\%$ solar penetration, regardless of the nature of the solar output.

Next, we repeat the same approach for the uncertain operating point LPV model. The resulting stability guarantee is valid for arbitrarily large changes in $u(t) \in [0, u_{\max}]$, so long as $y(t)$ is constrained to lie within the uncertainty set $\mathcal{Y}_0$ as defined earlier. Again, $V(y)$ is a Lyapunov function if and only if P satisfies

$$E^T P = P^T E \succeq 0, \tag{3.25}$$

$$M(\delta)^T P + P^T M(\delta) \prec 0, \tag{3.26}$$

for all $\delta \in \Delta$. The matrix-valued function $M(\delta)$ is not affine, nor rational, so the semi-infinite constraint over δ cannot be easily reformulated or relaxed into a finite number of constraints. Instead, we take a gridding approach, similar to that of [59], in which we enforce the semi-infinite LMI constraint over a finite number of grid points $\Delta_0 \subset \Delta$. The associated feasibility problem is an optimistic underestimate, but asymptotically approaches exactness as the number of grid points is increased. In other words, this procedure will overestimate the stability margin compared to the true, underlying value, but the overestimation gap approaches zero as the number of samples are increased. We again use YALMIP and MOSEK to solve the problem.

Incrementing $u_{\max}$ and performing this procedure with 15625 samples collected along a six-dimensional grid, we find that the nominal operating point is large-signal stable under uncertainty for up to $u_{\max} = 2.9727$ p.u. We make the optimistic conclusion that the same system can accommodate for up to $2.9727/(2.9727 + 1) = 74.828\% \approx 74.8\%$ solar penetration, when the generalized state variable vector $y(t)$ is restricted to lie within the uncertainty set $\mathcal{Y}_0$.

3.4.7 Slew-rate Analysis

Small-signal stability analysis and Lyapunov analysis have produced two different sets of stability margins. In this subsection, we will show that the stability margins are related by the maximum rate-of-change in the solar output, i.e. the slew-rate of the intermittency. The more optimistic stability margins computed in Section 3.4.4 are applicable at the zero-slew limit, when the solar output $u(t)$ changes slowly enough to be considered constant by the rest of the system. The more pessimistic stability margins computed in Section 3.4.6 are applicable at the infinite-slew limit, when the solar output $u(t)$ is allowed to change arbitrarily quickly.

To make the connection between the two stability margins, we perform a *slew-rate limited* Lyapunov stability analysis using a parameter-dependent Lyapunov function. We will attempt to construct a quadratic, parameter-affine, Lyapunov function $V(y, \delta) = y^T E^T (P_0 + \delta P_1) y$ for the nominal operating point LPV model (3.20) subject to the slew-rate limit $|\dot{\delta}(t)| \leq w_{\max}$. If it exists, then the original nonlinear model

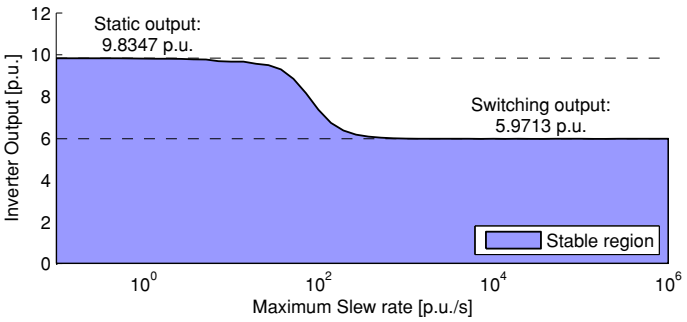


Figure 3-9: Solar output stability margin vs output slew-rate.

is stable for all $\delta(t)$ subject to the same conditions, so long as $y(t)$ remains within a small-signal neighborhood of y_{nom}. A sufficient condition for $V(y, \theta)$ to be a Lyapunov function is if its coefficient matrices P_0, P_1 satisfy

$$E^T P(\delta) = P(\delta)^T E \succeq 0, \tag{3.27}$$

$$wE^T P_1 + M_0(\delta)^T P(\delta) + P^T(\delta) M_0(\delta) \prec 0, \tag{3.28}$$

$$G^T P_1 + P_1^T G \succeq 0, \tag{3.29}$$

for all combinations of $\delta \in \{0, u_{\max}\}$, $w \in \{-w_{\max}, w_{\max}\}$.

Performing this Lyapunov function test while sweeping the values of $u_{\max}$ and $w_{\max}$ yields the plot shown in Fig. 3-8d. We see that for small slew-rate limits, the small-signal intermittency assumption remains valid, and that the stability margin of $0 \leq u(t) \leq 9.8347$ p.u. concurs with our previous results in Section 3.4.4. But as the slew-rate is allowed to increase, the system transitions into a regime of large-signal intermittency. In the limit of arbitrarily fast slew-rates, the size of the stability margin is reduced to $0 \leq u(t) \leq 5.9713$ p.u., which is the same result as that for classic Lyapunov analysis.

3.5 Conclusions

In this chapter, we used the quadratic stability test to analyze the robust small-signal stability of a transmission system with distributed renewables, and the robust large-signal stability of a microgrid system connected to an intermittent solar panel. More specifically, we used vertex-based quadratic stability tests, written

$$\text{find } P \succ 0 \text{ such that } M_i^T P + P M_i \prec 0 \text{ holds for all } i \in \{1, \ldots, m\}, \tag{3.30}$$

which is the necessary and sufficient condition for the quadratic stability of the polytopic LPV model

$$\frac{\mathrm{d}}{\mathrm{d}t} x(t) = M(t)x(t), \qquad M(t) \in \text{conv}\{M_1, \ldots, M_m\}.$$

In order to apply the technique to nonlinear problems, we use the linearization and grid-based sampling heurstics described in Section 3.1.2.

Quadratic stability has a reputation for being conservative and inflexible, but in our results, we found that it worked surprisingly well. Combining the quadratic stability test with a bisection-based partitioning strategy allowed a remarkably high-dimensional uncertainty set to be established as being robustly small-signal stable in just 19 branching steps. The gridding-based quadratic stability test was able to compute useful stability margins for a microgrid system under large-signal intermittency.

Conservatism aside, the primary bottleneck for the approach is the solution of the linear matrix inequality feasibility problem (3.30). In the transmission problem, each instance of the problem took around an hour to solve, and so the 19 iterations of the branch-and-bound algorithm took a full day to solve. Further progress in theoretical and computational methods are needed to scale theory to realistic-sized problems, which may contain tens of thousands of buses and thousands of generators. This is an important area of future research.

Chapter 4

Algorithms for Large-Scale Lyapunov Inequalities

In Chapter 3, the vertex-based quadratic stability test was found to be surprisingly effective for robust stability analysis on power systems. The key computation for the stability test is a linear matrix inequality (LMI) problem, which can be solved using an interior-point method. Unfortunately, the associated $O(n^6)$ time complexity restricts the approach to small, artificial models, containing no more than 150 state variables.

This chaper is motivated by the desire to extend the vertex-based quadratic stability test to large-scale problems. In applications like image processing and machine learning, the poor scalability of interior-point methods is commonly overcome using of first-order methods. When the objectives are smooth (or can be decomposed into a smooth component), then *accelerated* first-order methods can converge at the rate of $O(1/k^2)$ objective error at the k-th iteration. Unfortunately, our application has a nonsmooth objective, and first-order methods converge at the much slower rate of $O(1/k)$.

In this chapter, we investigate accelerating the converge of first-order methods using Krylov subspace methods like conjugate gradients (CG) and GMRES. We show that when first-order methods are used to solve the Newton subproblem of interior-point methods, that Krylov subspace acceleration allows the overall method to achieve an error rate of $O(1/k^2)$, matching the fastest first-order methods for smooth optimization.

4.1 Introduction

Given m square matrices $M_1, \ldots, M_m$, each of dimension $n \times n$, the linear matrix inequality (LMI) feasibility problem

$$Y \succ 0, \qquad M_i Y + Y M_i^T \prec 0 \text{ for all } i \in \{1, \ldots, m\}, \tag{4.1}$$

lies the heart of classical robust controls. The matrix Y—when it exists—is used as a stability certificate, a Lypuanov function, and a starting point for controller

synthesis [12,15]. Its eigendecomposition can be used to highlight the dominant modes of the system, and the associated orthogonal projectors are often used for nonlinear model order reduction [60]. If Y fails to exist, then the matrices $X_1, \ldots, X_m$ satisfying the Lagrangian dual,

$$\sum_{i=1}^{m} \operatorname{tr} X_i = 1, \quad \sum_{i=1}^{m} (M_i^T X_i + X_i M_i) \succeq 0, \quad X_i \succeq 0 \text{ for all } i \in \{1, \ldots, m\}, \tag{4.2}$$

are guaranteed to exist; these serve as proof for the inexistence of Y, and are known as the infeasibility certificate for (4.1).

The linear matrix inequality (4.1) is a conic feasibility problem. The standard approach is to embed it into a (slightly larger) primal-dual linear conic optimization pair

$$\begin{aligned} &\text{minimize } \mathbf{b}^T y \text{ s.t. } \mathbf{A}^T x = \mathbf{c},\ x \in \mathcal{K}, \\ &\text{maximize } \mathbf{c}^T y \text{ s.t. } \mathbf{A} y + s = \mathbf{b},\ s \in \mathcal{K}, \end{aligned} \tag{4.3}$$

posed over the self-dual cone of positive semidefinite matrices $\mathcal{K}$, and to apply a feasible interior-point method. It is a famous result that no more than $O(\sqrt{mn})$ interior-point iterations are required to solve (4.3) to machine precision; in practice, no more than 50 iterations are required [61, 62].

The main computation at each interior-point iteration is the solution of the Newton search direction

$$\begin{bmatrix} 0 & \mathbf{A}^T & 0 \\ \mathbf{A} & 0 & I \\ 0 & I & \mathbf{D} \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_c \end{bmatrix}, \tag{4.4}$$

with the positive definite scaling matrix $\mathbf{D}$, and the residuals r_p, r_d, r_c changing on every iteration. The equation is commonly solved by forming its Schur complement,

$$(\mathbf{A}^T \mathbf{D} \mathbf{A})(\Delta y) = r_p - \mathbf{A}^T [r_c - \mathbf{D} r_d], \tag{4.5}$$

and back-substituting $\Delta s = r_d - \mathbf{A}\Delta y$, $\Delta x = r_c - \mathbf{D}\Delta s$. When $\mathcal{K}$ is chosen to be the positive definite cone, the scaling matrix $\mathbf{D}$ is fully dense. Despite any sparsity structure originally present in the data matrix $\mathbf{A}$, the system of equations (4.5) is also fully-dense. All popular interior-point solvers for semidefinite programs solve the dense system by explicitly forming and factoring the dense Hessian matrix in (4.5). Given that y represents a matrix variable with n^2 degrees of freedom, the cubic complexity of dense Cholesky factorization results in $O(n^6)$ time per interior-point iteration.

4.1.1 First-Order Methods

The $O(n^6)$ time complexity of interior-point methods restricts the use of Lyapunov inequalities—and the classical techniques of robust control associated with Lyapunov inequalities—to medium-scale problems, with no more than $n \leq 150$ state variables.

This chapter is motivated by the desire to extend these techniques to large-scale power systems. A typical model of a realistic-sized power system may have on the order of a thousand state variables (i.e. $n \approx 1000$), far outside the capabilities of modern solvers. Suppose it took just 1 minute to solve the $n = 100$ problem using SeDuMi or MOSEK; then it would take 694 days to solve the $n = 1000$ problem using the same software.

Similar scaling issues with the interior-point solution of large-scale semidefinite programs also frequently arise in signal processing, machine learning, and related fields. For example, the nuclear-norm regularized optimization problems, which include matrix completion [63] and sparse principal component analysis [64], are semidefinite programs where the value of n can easily reach thousands. Semidefinite relaxations of graph theoretic problems, such as the Lovasz θ-function and MAX-CUT [65], also suffer from similar scaling issues.

A popular and widely successful approach to large-scale semidefinite programs is to adopt a suitable first-order method. By avoiding the dense second-order information, first-order methods have very low per-iteration costs, that can often be custom-tailored to the problem structure of a specific application. In the case of the Lyapunov inequalities (4.1), we show in Section 4.3 that the per-iteration cost can be reduced to $O(n^3)$, after an initial factorization step of $O(n^4)$, by exploiting a certain hierarchical structure.

Unfortunately, first order methods converge significantly slower than interior-point methods. The standard first-order algorithms (which we review in Section 4.3) are only able to converge to an ϵ-accurate solution in $O(1/\epsilon)$ iterations, for an error rate of $O(1/k)$ at the k-th iteration. In practice, this means that only low-accuracy solutions can be obtained over thousands of iterations.

4.1.2 Main Result

Instead, we investigate the ability to accelerate the convergence of first-order methods by reusing the information collected in previous iterations. When the optimization problem at hand is a quadratic minimization subject to equality constraints, the information collected in previous iterates can be optimally used to accelerate converge, using a Krylov subspace method like CG or GMRES.

First-order methods developed for (4.1) cannot be accelerated using a Krylov subspace method. However, these same methods become compatible when they are applied to the Newton subproblem associated with the interior-point solution of (4.1). In this Chapter, we show that, under the assumption of strong complementary slackness and low-rank solutions, the preconditioned conjugate gradients (PCG) solution of the Schur complement equation (4.5) with $(\mathbf{A}^T\mathbf{A})^{-1}$ as a preconditioner is able to converge in $O(\kappa_D^{1/4})$ iterations, for a square-root factor improvement over the basic first-order method. Given that κ_D scales with the inverse-square of the duality gap ϵ, this implies that an interior-point method based around the method would converge to an ϵ-accurate solution in $O(1/\sqrt{\epsilon})$. This is an entire square-root factor better than the standard first-order methods. Later, in Chapter 6, we prove a similar fourth-root result for the GMRES-accelerated version of ADMM.

4.1.3 Notation

Throughout the chapter, we use the integer n to refer to the size of the matrix variable, Y, and the integer m to refer to the number of Lyapunov inequalities in our original problem (4.1).

In order to avoid burdensome notation, we use $\mathrm{vec}\,(X)$ and $\otimes$ to denote the regular, or *nonsymmetric*, vectorization operator and the Kronecker product, previously defined in Chapter 1. The former is defined to satisfy the inner product identity $\mathrm{tr}\,X^TY = (\mathrm{vec}\,X)^T(\mathrm{vec}\,Y)$, while the latter is defined to satisfy the Kronecker identity,

$$(A \otimes B)\mathrm{vec}\,X = \mathrm{vec}\,(BXA^T).$$

All of our results remain valid for the *symmetric* vectorization operator (see [66, 67]), which captures the degrees of freedom in a symmetric $n \times n$ matrix

$$\mathrm{svec}\,X = [X_{1,1}, \sqrt{2}X_{2,1}, \ldots, \sqrt{2}X_{n,1}, X_{2,2}, \sqrt{2}X_{3,2}, \ldots, \sqrt{2}X_{n,2}, \ldots]^T,$$

while also satisfying $\mathrm{tr}\,XY = (\mathrm{svec}\,X)^T(\mathrm{svec}\,Y)$, so long as we substitute the symmetric Kronecker operator, which is implicitly defined to satisfy a symmetric Kronecker identity

$$(A \otimes_s B)\mathrm{svec}\,X = \frac{1}{2}\mathrm{svec}\,(BXA^T + AXB^T)$$

for both symmetric and nonsymmetric $n \times n$ matrices A, B. The only difference in our bounds is a factor-of-two reduction in the number of degrees of freedom, from n^2 in $\mathrm{vec}\,X$ to $n(n+1)/2$ in $\mathrm{svec}\,X$.

4.2 Interior-Point Formulation

4.2.1 Preprocessing

We begin with two preprocessing steps to filter out trivial problem instances, and to simplify the remaining Lyapunov inequality problem. Our first test filters out problems that are "obviously feasible", by attempting to present $Y = I$ as a solution to the Lyapunov inequalities (4.1). This test is implemented by forming the Hermitian matrices $-(M_i + M_i^T)$ and verifying their positivity using the Cholesky factorization. If the positivity test passes for all M_i, then the choice of $Y = I$ is feasible, and we may terminate immediately.

Our second test is designed to filter out problems that are "obviously infeasible", based on the notion of Hurwitz stability.

Definition 8. The matrix M is said to be *Hurwitz stable* if all of its eigenvalues lie in the open left-half plane, i.e. the condition $\mathrm{Re}\lambda(M) < 0$ is satisfied.

Proposition 9. *Let Y satisfy $M_iY + YM_i^T \prec 0$ for all $i \in \{1, \ldots, m\}$. Then $Y \succ 0$ if and only if each of the matrices $M_1, \ldots, M_m$ is Hurwitz stable.*

Proof. We recall a classic result from control theory: given any $S \succ 0$, the Lyapunov equation $MP + PM^T + S = 0$ solves with a unique solution $P \succ 0$ if and only if M is Hurwitz stable. The "if" part follows by independently applying this statement to each Lyapunov equation $M_i P_i + P_i M_i^T + S_i = 0$, while the "only if" part follows by noting that all of these solutions coincide with Y, i.e. we have $Y = P_1 = \cdots = P_m$. $\square$

Accordingly, we may proceed to verify whether each of the matrices $M_1, \ldots, M_m$ is Hurwitz stable. This test is implemented by performing a nonsymmetric eigendecomposition on each M_i. If we find a choice of M_i that is not Hurwitz stable, then the Lyapunov inequality is infeasible, and we may terminate immediately.

After validating that all m matrices $M_1, \ldots, M_m$ are Hurwitz, the positivity constraint $Y \succ 0$ may be dropped. Solving the reduced LMI problem

$$M_i Y + Y M_i^T \prec 0 \text{ for all } i \in \{1, \ldots, m\}$$

would always produce a solution $Y \succ 0$ in view of Proposition 9. This seemingly minor detail plays a surprising role in the remainder of this chapter; we will return to the point in the computational results.

4.2.2 Feasible Optimization Formulation

After the preprocessing step, the Lyapunov inequalities feasibility problem (4.1) has been reduced to the primal-dual feasibility pair

$$\text{find } Y \text{ such that } \mathcal{A}(Y) \prec 0, \tag{P}$$
$$\text{find } X \succeq 0,\ X \neq 0 \text{ such that } \mathcal{A}^T(X) = 0, \tag{D}$$

rewritten in standard form as

$$\text{find } y \text{ such that } \mathbf{A}y + s = 0,\ s \in \mathcal{K}, \tag{P}$$
$$\text{find } x \neq 0 \text{ such that } \mathbf{A}^T x = 0,\ x \in \mathcal{K}^*, \tag{D}$$

over the self-dual cone $\mathcal{K} = \mathbb{S}^n_+ \times \cdots \times \mathbb{S}^n_+$, the direct sum of m semidefinite cones of order n. For future reference, we will write $N = mn$ as the order of the cone $\mathcal{K}$.

The linear matrix-valued function $\mathcal{A}$ encodes the problem data. It is a map from the space of $n \times n$ matrices to the space of block-diagonal matrices with m blocks of size $n \times n$ matrices:

$$\mathcal{A} : Y \mapsto (M_1 Y + Y M_1^T) \oplus (M_2 Y + Y M_2^T) \oplus \cdots \oplus (M_m Y + Y M_m^T). \tag{4.6}$$

Its adjoint is the matrix-valued function

$$\mathcal{A}^T : X_1 \oplus X_2 \oplus \cdots \oplus X_m \mapsto \sum_{i=1}^{m} (M_i^T X_i + X_i M_i). \tag{4.7}$$

The vectorized version of $\mathcal{A}$ is the $mn^2 \times n^2$ matrix

$$\mathbf{A} = \begin{bmatrix} M_1 \otimes I + I \otimes M_1 \\ M_2 \otimes I + I \otimes M_2 \\ \vdots \\ M_m \otimes I + I \otimes M_m \end{bmatrix}. \tag{4.8}$$

All classical interior-point methods begin with a strictly feasible primal-dual initial point, but such a point does not even exist for the conic feasibility (P)-(D) pair. Following a standard technique often known as the "Big-M" method (see e.g. [62]), we embed the feasibility problems within slightly larger optimization problems for which strictly feasible initial points are easy to find

$$\text{maximize } -y_0 \text{ such that } \mathcal{A}(Y) \preceq y_0 I,\ -\operatorname{tr}\mathcal{A}(Y) \leq 1, \tag{P$'$}$$
$$\text{minimize } x_0 \text{ such that } \mathcal{A}^T(X - x_0 I) = 0, \operatorname{tr} X = 1, X \succeq 0, x_0 \geq 0, \tag{D$'$}$$

rewritten in standard form as

$$\max \left\{ \begin{bmatrix} -1 \\ 0 \end{bmatrix}^T \begin{bmatrix} y_0 \\ y \end{bmatrix} : \begin{bmatrix} 0 & -\mathbf{1}^T\mathbf{A} \\ -\mathbf{1} & \mathbf{A} \end{bmatrix} \begin{bmatrix} y_0 \\ y \end{bmatrix} + \begin{bmatrix} s_0 \\ s \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \{s_0, s\} \in \mathbb{R}_+ \times \mathcal{K} \right\}, \tag{P$'$}$$
$$\min \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}^T \begin{bmatrix} x_0 \\ x \end{bmatrix} : \begin{bmatrix} 0 & -\mathbf{1}^T \\ -\mathbf{A}^T\mathbf{1} & \mathbf{A}^T \end{bmatrix} \begin{bmatrix} x_0 \\ x \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \{x_0, x\} \in \mathbb{R}_+ \times \mathcal{K} \right\}, \tag{D$'$}$$

using the $mn^2 \times 1$ identity vector $\mathbf{1} = [\operatorname{vec} I; \ldots; \operatorname{vec} I]$. An example strictly-feasible primal-dual pair is $y_0 = N$, $y = 0$, $x_0 = 1$, and $x = \frac{1}{N}\mathbf{1}$.

Since Slater's condition is satisfied, strong duality holds, and the primal and dual objective coincide. Feasibility of the original problems (P)-(D) is determined by the sign of the optimal primal-dual objective: if it is strictly positive, then the primal solution for (P$'$) yields a feasible point $Y^\star$ for (P); if it is zero, then the dual solution (D$'$) is an infeasibility certificate $X_1^\star, \ldots, X_m^\star$ for (D). Note that the objective cannot be negative, since the zero vector is already a feasible point for the primal problem (P$'$).

The problems (P$'$)-(D$'$) are closely related to the homogenous self-dual embedding of Ye et al., which is the technique used by SeDuMi and MOSEK to transform the feasibility problem (P)-(D) into an optimization problem with feasible points. Although we will not repeat the derivation here, we note that (P$'$)-(D$'$) are actually equivalent to the self-dual embedding of (P)-(D) with the same initial points.

4.2.3 Interpretation of accuracy

The previous subsection uses an optimization problem to solve an underlying feasibility problem. Accordingly, the solution accuracy required to find a feasible point can be viewed as a measure of difficulty. For example, an "obviously feasible" problem may only require a coarse duality gap in its corresponding optimization problem, one

that is easily achievable using a simple first-order method. On the other hand, a "borderline feasible" problem may require such a small duality gap in the optimization problem that it is only be solvable using an interior-point method.

In this subsection, we establish a link between the accuracy of the optimization problem and the difficulty of the feasibility problem. We begin by converting the primal optimization problem (P′) into an eigenvalue optimization problem

$$\text{maximize } \lambda_{\min}(S) \text{ subject to } \operatorname{tr} S \leq 1, \quad S \in \mathscr{A},$$

using an epigraph argument to eliminate the variable y_0. Here, $\mathscr{A}$ is the range of the matrix-valued map $\mathcal{A}$. Since $S = 0$ is always a valid solution, the actual optimal point $S^\star$ will always be (at least) positive semidefinite. Enforcing this constraint to yields

$$\text{maximize } \frac{\lambda_{\min}(S)}{\operatorname{tr} S} \text{ subject to } S \in \mathscr{A},\ S \succeq 0,$$

whose optimal objective coincides with that of (P′). In order to make sense of the solution, let us define the *optimal condition number over the feasible set* as the following

$$\kappa_F \triangleq \min_S \left\{ \lambda_{\max}(S)/\lambda_{\min} : S \in \mathscr{A},\ S \succeq 0 \right\}.$$

Then the optimal objective for (P′) is bound

$$\frac{1}{N\kappa_F} \leq p^\star \leq \frac{1}{\kappa_F}$$

via the trace inequality $\lambda_{\max}(S) \leq \operatorname{tr} S \leq N\lambda_{\max}(S)$. Hence, we see that the feasibility problem is "hard" if the optimal condition over the feasible set is large, and "easy" if it is small. In particular, the feasibility problem is the easiest if there exists a choice of Y satisfying $\mathcal{A}(Y) = I$, because it would set $\kappa_F = 1$.

In all cases, assuming that the original Lyapunov inequalities problem is indeed feasible (i.e. κ_F is finite), then we can expect to start finding a feasible iterate Y satisfying

$$\mathcal{A}(Y) \prec 0, \qquad \operatorname{cond}(\mathcal{A}(Y)) \leq \kappa_F. \tag{4.9}$$

when the absolute duality gap ϵ drops below $1/\kappa_F$. Conversely, since the value of κ_F is unknown in practice, the current duality gap ϵ serves as a lower-bound $\kappa_F \geq 1/(N\epsilon)$.

4.3 First order methods

In the literature, first-order methods are generally classified into gradient methods and proximal methods. Methods in the former class are based on approximating a nonlinear objective using local gradient or subgradient information [68, Sec.1.3], and work best for smooth objective functions with Lipschitz continuous gradient functions. Those in the latter class are based on proximal operators, which can be interpreted as the solution to a trust-region subproblem [69, Ch.3], and tend to work better when the proximal operators can be efficiently evaluated, possibly in closed form. While having

very different motivations and derivations, the two classes are remarkably similar in practice, both in the actual operations carried out at each iteration as well as the convergence rate of the iterates.

In this section, we develop a projected gradient method and a proximal point method for the feasible optimization formulation (P′), posed as a maximum-eigenvalue minimization

$$\text{minimize } \lambda_{\max}(Z) \text{ subject to } Z \in \mathcal{H}, \tag{4.10}$$

over the space of matrices,

$$\mathcal{H} \triangleq \{Z \in \mathbb{S}^N : \mathcal{A}(Y) = Z, -\operatorname{tr} Z \leq 1\}. \tag{4.11}$$

Our first-order methods closely follow the style of the most popular algorithms for large-scale machine learning and image processing (specifically, NESTA [70] / FISTA [71] and ADMM [72]), converging at the error rate of $O(1/k)$ at the k-th iteration. At each iteration, both methods perform the following two operations: 1) an eigendecomposition of the matrix Z, a size-mn block-diagonal matrix of size-n blocks, for a cost of $O(mn^3)$; and 2) a projection onto the space of matrices $\mathcal{H}$ in (4.11), which can be evaluated using an active-set algorithm.

Algorithm 10 ($\text{proj}_{\mathcal{H}}$)**. Input:** Any symmetric $N \times N$ matrix Z

Output: A choice of Y to satisfy $\mathcal{A}(Y) = \text{proj}_{\mathcal{H}}(Z) \triangleq \arg\min\{\|S - Z\|_F^2 : S \in \mathcal{H}\}$.

1. (Initial projection) Compute $y = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T \text{vec}\, Z$, and write $Y = \text{mat}\, y$.
2. (Active set) If $-\operatorname{tr} \mathcal{A}(Y) \leq 1$, return Y. Otherwise, go to Step 3.
3. (Rank-1 update) Compute $u = \mathbf{A}^T\mathbf{1}$, $v = (\mathbf{A}^T\mathbf{A})^{-1}u$, and $\Delta y = v(u^Tv)^{-1}(1 + u^Tz)$. Return $Y' = \text{mat}\,(y + \Delta y)$.

The projection onto $\mathcal{H}$ at each iteration is the computational bottleneck for both methods. Each call requires solving the same size-n^2 system of equations

$$(\mathbf{A}^T\mathbf{A})\Delta y = \left[\sum_{i=1}^{m}(M_i \otimes I + I \otimes M_i)^T(M_i \otimes I + I \otimes M_i)\right]\Delta y = f, \tag{4.12}$$

for one or two new right-hand sides. The coefficient matrix is dense whenever the data matrices $M_1, \ldots, M_m$ are dense, so a solution via Cholesky factorization requires $O(n^6)$ factorization time, and each subsequent right-hand side requires $O(n^4)$ time to solve.

Alternatively, the first-order method may still converge when (4.12) is solved approximately, e.g. using an iterative method like conjugate gradients (CG). Unfortunately, most convergence theorems do not hold, and we often observe dramatically slowed convergence in practice [72, 73]. In particular, Devolder, Glindeur & Nesterov [74] showed that with inexact oracles, fast gradient methods (which underpins

one of our first-order methods) must necessarily suffer from error accumulation; they may converge slower than naive gradient methods.

Of course, it is also possible to develop a first-order method for (4.10) without solving (4.12). For example, reformulating in terms of the matrix variable Y yields

$$\text{minimize } \lambda_{\max}(\mathcal{A}(Y)) \text{ subject to } -\operatorname{tr}\mathcal{A}(Y) \leq 1. \tag{4.13}$$

First-order methods developed for (4.13) require only matrix-vector products with $\mathbf{A}$ and $\mathbf{A}^T$, and also converge at the $O(1/k)$ error rate. Unfortunately, the actual convergence rate of these methods becomes heavily influenced by the condition number of the matrix $\mathbf{A}$. Many first-order methods developed for problems of the form (4.13) assume a well-conditioned $\mathbf{A}$, but this is too strong of an assumption for problems arising from stability analysis.

In Chapter 5 of this thesis, we show that if the data matrices $M_1, \ldots, M_m$ arise as the linearization of power system models, then the system of equations (4.12) factored in $O(n^4)$, and each right-hand side is solved in $O(n^3)$ time. The key insight is a *hierarchical* structure in each data matrix, due to the property of bounded tree-width in power systems. At least for the power systems application, we may proceed by assuming the existence of an efficient oracle for matrix-vector products with $(\mathbf{A}^T\mathbf{A})^{-1}$.

4.3.1 Projected Gradient Method

The max-eig objective in (4.10) is neither smooth nor strongly convex. A common approach is to minimize a smooth approximation $\Phi_\mu(Z) \approx \lambda_{\max}(Z)$ developed using Nesterov's smoothing method [75]. Given the $N \times N$ real symmetric matrix Z, let us compute the eigenvalue decomposition $Z = \sum_i \lambda_i v_i v_i^T$, and define the smoothed max-eig objective [76] as

$$\Phi_\mu(Z) \triangleq \mu \log \sum_{i=1}^{N} e^{\lambda_i/\mu},$$

with the smoothing parameter $\mu > 0$. The smoothed max-eig objective bounds the original max-eig objective from above and from below

$$\Phi_\mu(S) - \mu \log N \leq \lambda_{\max}(S) \leq \Phi_\mu(S),$$

and its gradient function

$$\nabla\Phi_\mu(Z) = \left(\sum_{i=1}^{N} e^{\lambda_i/\mu}\right)^{-1} \sum_{i=1}^{N} e^{\lambda_i/\mu} v_i v_i^T$$

is Lipschitz continuous with respect to the Frobenius norm

$$\|\nabla\Phi_\mu(X) - \nabla\Phi_\mu(Y)\|_F \leq \frac{1}{\mu}\|X - Y\|_F.$$

Indeed, the smoothed version already finds a wide range of applications in large-scale optimization. The same smoothing technique is also widely popular for ℓ_1-norm and nuclear-norm objectives, particularly in LASSO and matrix completion problems [69, 70].

Replacing the nonsmooth max-eig objective with its smooth approximation

$$\text{minimize } \Phi_\mu(Z) \text{ subject to } Z \in \mathcal{H}, \tag{4.14}$$

we may apply an *accelerated* projected gradient descent method [68,70,71,75,77]. For example, FISTA [71] is defined by the sequence starting given $\theta_1 = 1$ and any Z_0 and $U = Z_0$, for $k = 1, 2, \ldots$

$$\begin{aligned} Z_k &= \text{proj}_{\mathcal{H}}(U - \mu \nabla \Phi_\mu(U)), \\ \theta_{k+1} &= \frac{1 + \sqrt{1 + 4\theta_k^2}}{2}, \\ U &= Z_k + \frac{\theta_k - 1}{\theta_{k+1}}(Z_k - Z_{k-1}), \end{aligned} \tag{4.15}$$

with step-size μ coinciding with the inverse of the gradient Lipschitz constant. The resulting sequence satisfies a famous convergence theorem.

Theorem 11 (FISTA [71, Thm.4.4])**.** *The sequence in (4.15) converges*

$$\Phi_\mu(Z_{k+1}) - \Phi_\mu(Z^\star) \leq \frac{2\|Z_0 - Z^\star\|_F^2}{\mu(k+1)^2} \qquad \forall Z^\star \in \mathcal{Z}^\star$$

where $\mathcal{Z}^\star$ is the solution set for the problem (4.14).

Let us use Theorem 11 to estimate the number of iterations needed to obtain an ϵ-accurate solution to the original problem (4.10). We set $\mu = \epsilon/(2 \log N)$, in order to allow $\Phi_\mu(Z^\star)$ to be within $\epsilon/2$ of the true optimum. Then, to achieve the error estimate $\Phi_\mu(Z) - \Phi_\mu(Z^\star) \leq \epsilon/2$, Theorem 11 estimates k iterations, where

$$k \leq \frac{\sqrt{8}}{\epsilon} \|Z_0 - Z^\star\|_F \sqrt{\log N}.$$

The $O(1/\epsilon)$ iterations needed to converge to an ϵ-accurate solution implies an error rate of $O(1/k)$ at the k-th iteration. Note that the bound is relatively sharp, since value of $\|Z^\star\|_F$ will never be very large once ϵ becomes small, since its largest eigenvalues are bound $\lambda_{\max}(Z^\star) \leq \Phi_\mu(Z^\star) \leq \epsilon/2$ from above, while its trace is bound $\operatorname{tr} Z^\star \geq -1$ from below, by virtue of $Z^\star \in \mathcal{H}$.

4.3.2 Proximal-Point Method

We begin by noting that the proximal operator associated with $\lambda_{\max}$ can be evaluated efficiently. Given the $N \times N$ matrix Z, let us compute its eigendecomposition $Z =$

$\sum_i \lambda_i v_i v_i^T = V\Lambda V^T$. Then we have, via a change-of-basis

$$\begin{aligned}\operatorname{prox}_{\mu\lambda_{\max}}(S) &\triangleq \arg\min_S \left\{ \frac{1}{2}\|S - Z\|_F^2 + \mu\lambda_{\max}(S) \right\} \\ &= V \arg\min_D \left\{ \frac{1}{2}\|D - \Lambda\|_F^2 + \mu\lambda_{\max}(D) \right\} V^T \\ &= V\operatorname{diag}(d^\star)V^T,\end{aligned}$$

in which $d^\star$ evaluates the proximal operator for the *maximum function*,

$$d^\star = \arg\min_d \left\{ \frac{1}{2}\|d - \lambda\|^2 + \mu \max_i d_i \right\}. \tag{4.16}$$

This latter optimization (4.16) can be efficiently solved using the bisection formula in [69, Sec.6.4.1].

Accordingly, (4.10) may be put into a composite function form

$$\text{minimize } \lambda_{\max}(Z) + I_{\mathcal{H}}(-Z) \tag{4.17}$$

in which $I_{\mathcal{H}}$ is the indicator function for $\mathcal{H}$

$$g(Z) = \begin{cases} 0 & S \in \mathcal{H}, \\ +\infty & \text{otherwise}, \end{cases}$$

and solved using an operator-splitting method. In this section, we use ADMM (see [72] and the references therein), which is one of the most popular algorithms. The method converts (4.17) into the consensus problem

$$\text{minimize } \lambda_{\max}(Z) + I_{\mathcal{H}}(S) \text{ subject to } Z + S = 0, \tag{4.18}$$

constructs an augmented Lagrangian with parameter t

$$\mathscr{L}_t(Z, S, X) = \lambda_{\max}(Z) + I_{\mathcal{H}}(S) + \operatorname{tr} X(Z + S) + \frac{t}{2}\|Z + S\|_F^2,$$

and performs the alternating direction minimization:

$$\begin{aligned} Z_{k+1} &= \arg\min_Z \mathscr{L}_t(Z, S_k, X_k) \\ S_{k+1} &= \arg\min_S \mathscr{L}_t(Z_{k+1}, S, X_k) \\ X_{k+1} &= X_k + t(Z_{k+1} + S_{k+1}) \end{aligned}$$

Redefining the dual variable $X_k \leftarrow (1/t)X_k$ reveals the following sequence, after some

manipulations

$$\begin{aligned} Z_{k+1} &= \operatorname{prox}_{\lambda_{\max}/t}(-S_k - X_k), \\ S_{k+1} &= \operatorname{proj}_{\mathcal{H}}(-Z_{k+1} + X_k), \\ X_{k+1} &= X_k + (Z_{k+1} + S_{k+1}), \end{aligned}$$

The sequence is guaranteed to converge for any fixed step-size t, but its exact choice will greatly influence the speed of convergence in practice.

Unfortunately, it is difficult to bound the convergence rate of ADMM. Neither objective functions in (4.18) is smooth nor strongly convex, so most of the existing bounds do not apply (see [78] and the references therein). We only know that the sequence converges with objective error $O(1/k)$ in an ergodic sense [79], and this suggest that an ϵ-accurate solution can be obtained in $O(1/\epsilon)$ iterations. In practice, the ADMM method performs comparably to the projected gradient descent method derived above.

4.4 Krylov Subspace Acceleration

Iterative methods can often be accelerated by reusing past information. For example, it is well-known that the basic gradient method for convex optimization

$$x_{k+1} = x_k + \alpha \Delta x_k, \tag{4.19}$$

cannot minimize the objective error faster than $O(1/k)$ at the k-th iteration. But by reusing the last search direction, in a scheme often described intuitively as adding "momentum",

$$x_{k+1} = x_k + \alpha \Delta x_k + \beta \Delta x_{k-1}. \tag{4.20}$$

it is possible to achieve an objective error of $O(1/k^2)$. Indeed, this is the underlying principle behind Nesterov acceleration [80], as well as other accelerated schemes like heavy-ball [81]. While both methods share same search oracle and the same objective function, the accelerated method (4.20) is able to converge an entire order-of-magnitude faster by reusing past information.

This idea of accelerating convergence by reusing past search directions in (4.20) can be generalized. Consider the affine search space

$$\mathcal{X}_k \triangleq x_0 + \operatorname{span}(\Delta x_0, \Delta x_1, \ldots, \Delta x_k) \tag{4.21}$$

in which Δx_k is the current search direction as computed by an oracle, and each Δx_0, Δx_1,..., Δx_k is a previous search direction. Combined, $\mathcal{X}_k$ contains all of the information gathered up to the k-th iteration. Ideally, we would select the optimal element within the entire search space $x^\star_{k+1} \in \mathcal{X}_k$ at the k-th iteration. An iterative method based on this principle is not only accelerated, but also necessarily *optimal*: fixing the objective function and search oracle, no iterative method can produce a sequence that converges more quickly.

Unfortunately, the problem of optimizing over $\mathcal{X}_k$ is intractable in all but two special cases. The first is the unconstrained minimization of a quadratic objective $q(x)$ using its gradient function as the search oracle. The conjugate gradients (CG) algorithm solves the minimization problem of $q(x)$ over $\mathcal{X}_k$ at each iteration

$$x_{k+1} = \arg\min\{q(x) : x \in \mathcal{X}_k\}, \qquad \Delta x_{k+1} = \nabla q(x_{k+1}),$$

and can be viewed as an optimally accelerated version of gradient descent. The second is the acceleration of the *linear* fixed-point iterations $x_{k+1} = T(x_k)$. The GMRES algorithm forces the sequence to converge in the Euclidean norm via

$$x_{k+1} = \arg\min\{\|x - T(x)\| : x \in \mathcal{X}_k\}, \qquad \Delta x_{k+1} = T(x_{k+1}) - x_{k+1},$$

and can be viewed as an optimal version of relaxation. These two methods work because, under their respective special conditions, the search oracle is *affine*, and the space $\mathcal{X}_k$ reduces to a *Krylov subspace*. The corresponding optimization problems over Krylov subspaces can be efficiently solved using the Lanczos or the Arnoldi algorithms. The actual implementation details of CG and GMRES are standard, and we refer the reader to classic texts [82,83]. We only emphasize that the dominant cost per-iteration is the evaluation of the search oracles: ∇q in CG and T in GMRES.

This section is motivated by the optimality of CG and GMRES. Neither algorithms can be directly applied to our standard form optimization problems (P′)-(D′), nor to the first-order methods derived in the previous section. However, they are indeed applicable for the Newton subproblem, which is a simple equality-constrained quadratic program, but also the most computationally expensive part of interior-point methods. Since every interior-point method converges in around 50 iterations, the ability to solve the Newton subproblem efficiently immediately translates into the same thing for our original problem.

In this section, we show that when an efficient matrix-vector product is available for $(\mathbf{A}^T\mathbf{A})^{-1}$, that an interior-point method based on a Krylov subspace solution of the Newton subproblem can, under certain circumstances, converge to an ϵ-accurate solution in $O(1/\sqrt{\epsilon})$ iterations, for an error rate of $O(1/k^2)$ at the k-th iteration. This is an order-of-magnitude acceleration over the first-order methods in the previous section.

4.4.1 The Newton Subproblem

The dominant cost of each interior-point iteration is the computation of the *Newton equations*

$$\begin{bmatrix} 0 & \mathbf{A}^T & 0 \\ \mathbf{A} & 0 & I \\ 0 & I & \mathbf{D} \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_c \end{bmatrix}, \tag{4.22}$$

which can be viewed as the Karush–Kuhn–Tucker conditions for an equality-constrained quadratic program which we name the *Newton subproblem*

$$\begin{aligned} &\text{minimize } \frac{1}{2}\Delta s^T \mathbf{D} \Delta s - r_c^T \Delta s - r_p^T \Delta y, \\ &\text{subject to } \mathbf{A}\Delta y + \Delta s = r_d. \end{aligned} \tag{4.23}$$

The Newton equations (4.22) are usually solved via the Schur complement equation

$$(\mathbf{A}^T \mathbf{D} \mathbf{A})(\Delta y) = r_p - \mathbf{A}^T [r_c - \mathbf{D} r_d], \tag{4.24}$$

and back-substituting $\Delta s = r_d - \mathbf{A}\Delta y$, $\Delta x = r_c - \mathbf{D}\Delta s$. These can be viewed as the normal equation to the unconstrained problem

$$\text{minimize } \frac{1}{2}\Delta (r_d - \mathbf{A}\Delta y)^T \mathbf{D} (r_d - \mathbf{A}\Delta y) + (\mathbf{A}^T r_c - r_p)^T \Delta y, \tag{4.25}$$

and can also be obtained by eliminating the variable $\Delta s = r_d - \mathbf{A}\Delta y$ in (4.23). The matrix $\mathbf{A}^T \mathbf{D} \mathbf{A}$ in (4.24) is commonly known as the Hessian matrix.

The dense positive-definite scaling matrix $\mathbf{D}$ is provided[1] in a convenient Kronecker product form

$$\mathbf{D} = (W_1 \otimes W_1) \oplus (W_2 \otimes W_2) \oplus \cdots \oplus (W_m \otimes W_m), \tag{4.26}$$

motivating solution using a first-order method. Gradient evaluations require matrix-vector products with $\mathbf{D}$, and these can be efficiently performed in $O(mn^3)$ time via the Kronecker identity $(W_i \otimes W_i)\text{vec}\, X_i = \text{vec}\,(W_i X_i W_i)$ for each $i \in \{1, \ldots, m\}$. Also, as we will show in Chapter 6, the proximal operator for $\frac{1}{2}s^T \mathbf{D} s$ can also be performed in $O(mn^3)$ time.

When a first-order method is used to solve either the constrained problem (4.25) or the unconstrained problem (4.23), the number of iterations to converge to δ objective error is bound $O(\sqrt{\kappa_D} \log \delta^{-1})$, where κ_D is the condition number

$$\kappa_D = \frac{\lambda_{\max}(\mathbf{D})}{\lambda_{\min}(\mathbf{D})}. \tag{4.27}$$

It is a folklore theorem from the study of interior-point and barrier methods that, close to a solution, this condition number scales $\kappa_D \in \Theta(N^2/\epsilon^2)$, where N is the order of the convex cone and ϵ is the current duality gap; see [84,85] for the barrier method, and [66,86,87] for the more general statement for primal-dual interior-point methods. Hence, the Newton subproblem of an ϵ-accurate Newton step will require $O(1/\epsilon)$ first-order iterations to solve in the worst case, which is the same as the first-order methods derived in the previous sections.

[1] Assuming that the primal-scaling, dual-scaling, or primal-dual Nesterov-Todd (NT) scaling is used for the underlying interior-point method

4.4.2 PCG-Schur

We begin by revisiting an old idea—using CG to solve the unconstrained problem (4.25). Within the framework of Krylov subspace acceleration, we may say that CG is used to optimally accelerate the convergence of the gradient method. This approach can be traced all the way back to Karmakar's original paper [88]. In exact arithmetic, CG solves an k-dimensional problem in k iterations, and this can be used to derive an interior-point method with worst-case time complexity of $O(n^{5.5}m^{1.5})$, and an average complexity closer to $O(n^4m^{1.5})$ [89].

Unfortunately, the naive CG approach is not robust in practice, and an interior-point method based on CG is only able to compute low-accuracy solutions. Instead, a CG-based interior-point method is considerably enhanced by preconditioning. In the context of semidefinite programming programming problems, diagonal or block-diagonal preconditioners have been empirically shown to be highly effective for low-accuracy problems. Incomplete factorization preconditioners, which had been highly successful in the context of linear programming, have been less effective for semidefinite programming, due to the density of the scaling matrices. More recent preconditioners are based on eigenvalue deflation of the scaling matrices [90, 91]; these are highly effective, but very computationally expensive to construct and apply.

Motivated by the first-order methods in the previous section, let us consider using the matrix $\mathbf{P}^{-1} \triangleq (\mathbf{A}^T\mathbf{A})^{-1}$ to precondition the Hessian $\mathbf{H} \triangleq \mathbf{A}^T\mathbf{D}\mathbf{A}$. In other words, we use the *preconditioned* conjugate gradients (PCG) algorithm to solve the preconditioned problem

$$(\mathbf{P}^{-1/2}\mathbf{H}\mathbf{P}^{-1/2})\Delta y = \mathbf{P}^{-1/2}r. \tag{4.28}$$

Each PCG iteration incurs a single matrix-vector product with $\mathbf{P}^{-1}$ and $\mathbf{H}$, for the same complexity as a single iteration of either of the two first order methods derived in the previous section.

However, we will prove in this section that the PCG solution of the Schur complement equation, which we name PCG-Schur, converges at a much faster rate. More specifically, under certain assumptions upon the original semidefinite program, we show that the method solves (4.28) in $O(\kappa_D^{1/4})$ iterations, where κ_D was previously defined in (4.27). Given that $\kappa_D \in O(1/\epsilon^2)$, this implies that an interior-point method based around PCG-Schur converges to an ϵ-accurate solution in $O(1/\sqrt{\epsilon})$ iterations, for an error rate of $O(1/k^2)$ at the k-th iteration.

Assumption 12. Let $y_0^\star$, $Y^\star$and $x_0^\star$, $X_1^\star, \ldots, X_m^\star$ be the solution to (P′)-(P′), and write $s_0^\star = 1 + \operatorname{tr}\mathcal{A}(Y^\star)$, and $S_i^\star = y_0^\star I - M_i Y^\star - Y^\star M_i^T$ as the optimal slack variables. Define $X^\star = x_0^\star \oplus X_1^\star \oplus \cdots \oplus X_m^\star$ and define $S^\star = s_0^\star \oplus S_1^\star \oplus \cdots \oplus S_m^\star$. Then we assume:

- (Strong complementary slackness)

$$\operatorname{rank}(X^\star) + \operatorname{rank}(S^\star) = N + 1.$$

- (Low-rank dual solution) There exists an absolute constant c such that

$$\operatorname{rank}(X_i) \leq c \qquad \forall i \in \{1, \ldots, m\}.$$

Theorem 13. *Consider solving a problem satisfying the conditions in Assumption 12 using a primal-scaled, dual-scaled, or NT-scaled interior-point method. Let* $\mathbf{D}$ *be the scaling matrix at an interior-point iteration with a sufficiently small duality gap. Then conjugate gradients with preconditioner* $(\mathbf{A}^T\mathbf{A})^{-1}$ *solves* $(\mathbf{A}^T\mathbf{D}\mathbf{A})x = b$ *to* δ*-residual in* $O(m + \kappa_D^{1/4} \log \delta^{-1})$ *iterations.*

For further reference, we define the matrix $\mathbf{Q} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1/2}$ as the orthogonal matrix spanning the range of $\mathbf{A}$. We prove the theorem by bounding the distribution of eigenvalues in $\mathbf{Q}^T\mathbf{D}\mathbf{Q}$, and heuristically solving an eigenvalue approximation problem.

Theorem 14 ([92, Lem.6.28])**.** *Given the* $n \times n$ *symmetric positive definite system of equations* $Hx = g$*, symmetric positive definite preconditioner* P*, and the initial point* $x^{(0)}$*, conjugate gradients generates (in exact arithmetic) an iterate* $x^{(k)}$ *at the* k*-th iteration satisfying*

$$\frac{\|x^{(k)} - x^\star\|_G}{\|x^{(0)} - x^\star\|_G} \leq \min_{p_k} \max_{i=1,\ldots,n} |p_k(\lambda_i)|,$$

where $\lambda_1, \ldots, \lambda_n$ *are the eigenvalues of* $G \triangleq P^{-1/2}HP^{-1/2}$*,* $x^\star \triangleq H^{-1}g$ *is the exact solution,* $\|x\|_G \triangleq \sqrt{x^TGx}$ *is the* G*-norm, and the minimum is taken over all polynomials* p_k *of degree* k *or less with* $p_k(0) = 1$*.*

Only in very rare cases is an explicit closed-form solution known, but any heuristic choice of polynomial $p(\cdot)$ will provide a valid upper-bound. We also state a famous closed-form solution attributed to Chebyshev.

Theorem 15. *Let* $\mathcal{I}$ *denote the interval* $[c-a, c+a]$ *on the real line. Then assuming that* $+1 \notin \mathcal{I}$*, the polynomial approximation problem has closed-form solution*

$$\min_{\substack{p \in \mathbb{P}_k \\ p(1)=1}} \max_{z \in \mathcal{I}} |p(z)| = \frac{1}{|T_k(\frac{1-c}{a})|} \leq 2\left(\frac{\sqrt{\kappa_I} - 1}{\sqrt{\kappa_I} + 1}\right)^k, \tag{4.29}$$

where $T_k(z)$ *is the degree-*k *Chebyshev polynomial of the first kind, and* $\kappa_I = (|1-c| + a)/(|1-c| - a)$ *is the condition number for the interval. The minimum is attained by the Chebyshev polynomial* $p^\star(z) = T_k(\frac{z-c}{a})/|T_k(\frac{1-c}{a})|$*.*

Proof. See e.g. [93]. □

Let us now bound the distribution of eigenvalues in $\mathbf{D}$.

Lemma 16. *Except for* $O(m)$ *outliers, the eigenvalues of* $\mathbf{D}$ *are distributed over an interval with condition number* $O(\sqrt{\kappa_D})$*.*

Proof. Let us define $\mu = \epsilon/N$ as the central path parameter of the interior-point method, and use c as the maximum rank of each X_i as defined in Assumption 12. Repeating the same analysis as in [84, Lem.3.1,3.2,Thm.3.1,3.3] for the slack and primal variables, we see that each W_i has $n-c$ small eigenvalues of $\Theta(1)$ and c large eigenvalues of $\Theta(\mu^{-1})$. Accordingly, each $W_i \otimes W_i$ has c^2 large eigenvalues of $\Theta(\mu^{-2})$, and the remaining eigenvalues are spread within an interval of $\Omega(1)$ and $O(\mu^{-1})$, for a condition number of $O(\mu^{-1})$. Repeating this for each $W_1, \ldots, W_m$, we find that $\mathbf{D}$ has condition number of $\Theta(\mu^{-2})$ due to the presence of mc^2 outliers; its remaining eigenvalues lie within an interval of condition number $O(\mu^{-1}) = O(\sqrt{\kappa_D})$. □

Applying Theorem 15 to this outliers-plus-interval structure then yields the second square-root factor.

Proof of Theorem 13. Applying the Cauchy interlacing eigenvalues theorem to our previous Lemma shows that the spectrum of $\mathbf{Q}^T\mathbf{D}\mathbf{Q}$ is comprised of $O(m)$ large outliers and an interval of condition number $O(\sqrt{\kappa_D})$. Now, consider solving the polynomial approximation problem in Theorem 14, using the first $O(m)$ zeros to cancel the outliers, and the remaining zeros spread over the $O(\sqrt{\kappa_D})$-conditioned interval, according to Theorem 15. The approximation error arises entirely due to the interval, and hence the polynomial achieves linear convergence $O(\kappa_D^{1/4} \log \delta^{-1})$ after the initial $O(m)$ iterations. The optimal polynomial used by CG must converge at least as quickly as this polynomial. □

4.4.3 ADMM-GMRES

Alternatively, let us attempt to use ADMM to solve the original Newton subproblem. Introducing the augmented Lagrangian

$$\mathscr{L}_t(s, y, x) = \frac{1}{2} s^T \mathbf{D} s - r_c^T s - r_p^T y + x^T(\mathbf{A}y + s - r_d) + \frac{t}{2}\|\mathbf{A}y + s = r_d\|^2$$

with step-size t, we again perform the alternating minimization in ADMM to yields the steps

$$\begin{aligned}
s_{k+1} &= \arg\min_s \frac{1}{2} s^T \mathbf{D} s - r_c^T c + \frac{t}{2}\|s + \mathbf{A}y_k - r_d + x_k\|^2, \\
y_{k+1} &= \arg\min_z -r_p^T y + \frac{t}{2}\|s_{k+1} + \mathbf{A}y - r_d + x_k\|^2, \\
x_{k+1} &= x_k + (s_{k+1} + \mathbf{A}y_k - r_d),
\end{aligned}$$

which reduces to the update equations

$$\begin{bmatrix} s_{k+1} \\ y_{k+1} \\ x_{k+1} \end{bmatrix} = \begin{bmatrix} t^{-1}D + I & 0 & 0 \\ \mathbf{A}^T & \mathbf{A}^T\mathbf{A} & 0 \\ I & \mathbf{A} & -I \end{bmatrix}^{-1} \left(\begin{bmatrix} 0 & -\mathbf{A} & -I \\ 0 & 0 & -\mathbf{A}^T \\ 0 & 0 & -I \end{bmatrix} + \begin{bmatrix} t^{-1}r_c \\ t^{-1}r_p \\ r_d \end{bmatrix} \right),$$

which we can write as

$$w_{k+1} = G(t)w_k + u(t) \tag{4.30}$$

on the iterates $w_k = [s_k; y_k; x_k]$. This sequence can be forced to converge (in the Euclidean norm) using GMRES. Essentially, this entails tasking the GMRES algorithm with the fixed-point equation

$$[I - G(t)]w = u(t). \tag{4.31}$$

The algorithm performs a single matrix-vector product with the matrix $[I - G(t)]$ at each iteration, and this has the same cost as a single iteration of ADMM. In Chapter 6, we prove that, with the parameter choice $t = \sqrt{\lambda_{\max}(\mathbf{D})\lambda_{\min}(\mathbf{D})}$, the GMRES-accelerated version of the ADMM method in (4.31) is often able to produce an iterate satisfying $\|[I - G(t)]w_k - u(t)\| \leq \epsilon\|u(t)\|$ in

$$O(\kappa_D^{1/4} \log \epsilon^{-1}) \text{ iterations.}$$

Again, considering that $\kappa_D \in O(1/\epsilon^2)$, the fourth-root iteration bound implies that an interior-point method based around ADMM-GMRES also converges to an ϵ-accurate solution in $O(1/\sqrt{\epsilon})$ iterations, for an error rate of $O(1/k^2)$ at the k-th iteration.

4.5 Computational Results

In order to test our results in a realistic environment, we develop a simple interior-point method in which PCG-Schur is used to compute its Newton search directions, and use the resulting method to solve Lyapunov inequalities posed on the IEEE 118-bus example presented earlier in Chapter 3. More specifically, the $n \times n$ data matrices $M_1, \ldots, M_m$ have sizes

$$n \in \{34, 41, 48, 55, 62, 69, 76, 83, 90, 97\}.$$

The number of state variables n is modified by taking some of the generator models offline. For each n, $m = 20$ data matrices are generated by randomizing the load patterns in the system network. For each pair of n, m, 30 trials are performed. This means we solve a total of 300 randomized problems, of varying n, all posed on the same IEEE 118-bus system.

4.5.1 An Example Barrier Method

Our barrier method is essentially identical to the fixed-step method described in [94]. Beginning with the primal problem (P′), we replace the inequality constraints by scaled self-concordant logarithmic barrier functions

$$\varphi(t, y) = ty_0 - \log\det\left[y_0 I - \mathcal{A}(Y)\right]_+ - \log\left[1 + \operatorname{tr}\mathcal{A}(Y)\right]_+, \tag{4.32}$$

where $y = [y_0; \operatorname{vec} Y]$ and $[\cdot]_+$ denotes restriction onto the positive semidefinite cone. We then proceed to compute, using Newton's method, a sequence of approximate solutions $y^\star(t) = \arg\min \varphi(t, y)$ that approach the true solution as $t \to \infty$.

Algorithm 17 (Barrier method)**. Input:** Data matrices $\{M_1, \ldots, M_m\}$; absolute duality gap tolerance ϵ_{gap}; algorithm parameters $\alpha \in (0, 0.5)$, $\beta \in (0, 1)$, $\epsilon_{\text{cen}} > 0$, and $\gamma > 0$.
Output: Lyapunov function Y or ϵ_{gap}-accurate infeasibility certificate $X_1, \ldots, X_m$.
Initialization: Set the initial point $y^{(0)} = [1; 0]$, $t = N$, $k = 0$, and select an outer-loop duality gap reduction η such that $\eta - 1 - \log \eta = \gamma/(N+1)$.

1. (Duality gap reduction) Set $t \leftarrow \eta t$.

2. (Analytic centering) Solve $\arg\min_y \varphi(t, y)$

   (a) (Newton subproblem) Compute the Newton direction Δy and the Newton decrement λ

   $$\Delta y = [\nabla_y^2 \varphi(y^{(k)})]^{-1} [\nabla_y \varphi(t, y^{(k)})],$$
   $$\lambda = (\Delta y)^T [\nabla_y \varphi(t, y^{(k)})].$$

   (b) (Backtracking line search) Increment $\ell = 0, 1, 2, \ldots$ and stop on the first value that satisfies the sufficient decrement condition

   $$\varphi(t, y^{(k)} - \beta^\ell \Delta y) \leq \varphi(t, y^{(k)}) - \alpha \beta^\ell \lambda.$$

   Accept the step by setting $y^{(k+1)} = y^{(k)} - \beta^\ell \Delta y$ and incrementing $k \leftarrow k+1$.

   (c) (Early termination) Decompose $y^{(k)} = [y_0; \operatorname{vec} Y]$. If $y_0 < 0$, then exit, and return Y as a feasible point for the original problem.

   (d) (Centering check) If the Newton increment is below tolerance, i.e. $\lambda \leq \epsilon_{\text{cen}}$, then go to Step 3. Otherwise, go to Step 2a).

3. (Dual variables) Update the dual variable $x = [x_0; \operatorname{vec} X]$ using the newly computed primal variable $y^{(k)} = [y_0; \operatorname{vec} Y]$ via

   $$x_0 = \frac{1}{t}[1 + \operatorname{tr} \mathcal{A}(Y)]^{-1}, \qquad X = \frac{1}{t}[1 + \operatorname{tr} \mathcal{A}(Y)]^{-1}.$$

4. (Duality gap check) If $(N+1)/t < \epsilon_{\text{gap}}$, then terminate and output the current value of X as an infeasibility certificate for the original problem. Otherwise, go to Step 1.

In a practical implementation, the outer-loop duality gap reduction η is picked directly. The implicit definition γ is only useful in order to prove the classic iteration bound. Typical values for the algorithm parameters are $\alpha = 0.01$ and $\beta = 0.5$, $\epsilon_{\text{cen}} = 10^{-3}$, and $\eta = 100$.

Proposition 18 ([94]). *Algorithm 17 terminates within* $O(\sqrt{mn}\log(1/\epsilon_{\text{gap}}))$ *Newton iterations.*

At each iteration, the Newton subproblem in Step 2a) is solved using the PCG-Schur algorithm described in Section 4.4.2, to a relative residual value between 10^{-9} and 10^{-12}.

4.5.2 Schur-PCG solution in $O(\kappa_D^{1/4})$ iterations

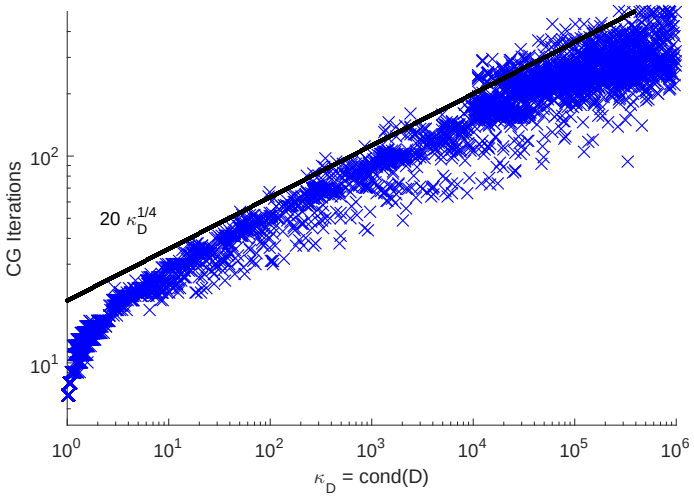
The fourth-root result for the PCG-Schur algorithm (Theorem 13) is the key step that proves the $O(1/\sqrt{\epsilon})$ iteration bound. However, recall that the result was contingent on the dual Lyapunov inequality problem having low-rank solutions (Assumption 12). In this subsection, we show that the Assumption holds up well, at least for the IEEE 118-bus problems that we have considered.

Figure 4-1 shows the number of PCG iterations with the $(\mathbf{A}^T\mathbf{A})^{-1}$ preconditioner to solve each Newton system to machine precision plotted against κ_D, for all ~ 50 Newton iterations of the 270 test problems described above. An "exact" preconditioner is used in Fig. 4-1a, meaning that the matrix $(\mathbf{A}^T\mathbf{A})$ is explicitly formed and factorized using dense Cholesky factorization. The expected $O(\kappa_D^{1/4})$ trend can be readily observed. The hierarchical preconditioner from Chapter 5 is used in Fig. 4-1b. The factorization is constructed with two levels of hierarchy, and compressed using an aggressive tolerance of 10^{-3}. Nevertheless, the $O(\kappa_D^{1/4})$ trend is preserved, albeit with a few more iterations than the exact case.

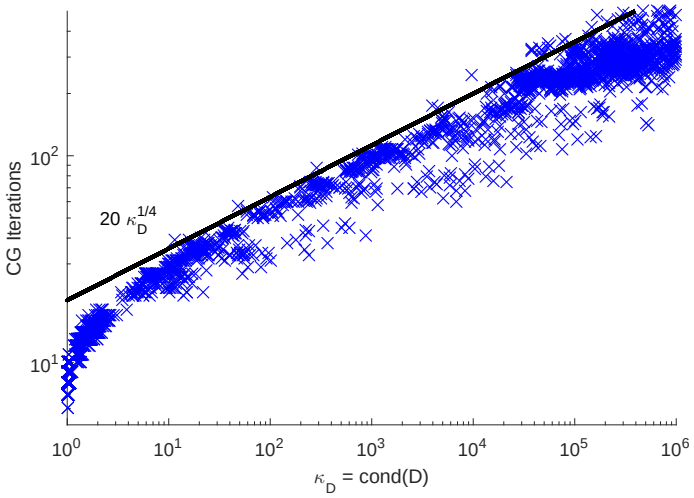
4.5.3 Overall error rate of $O(1/k^2)$

Given that the PCG-Schur algorithm converges in $O(\kappa_D^{1/4})$ iterations, and that the condition number κ_D itself scales $O(1/\epsilon^2)$ with respect to the objective error, we would expect convergence to an ϵ-accurate iterate in $O(1/\sqrt{\epsilon})$ PCG iterations, or equivalently, an error rate of $O(1/k^2)$ at the k-th PCG iteration.

Figure 4-2 shows the accumulated inner PCG iterations over all outer interior-point iterations, plotted against the (absolute value of the) primal objective. In this problem, the primal objective is also an upper-bound on the objective error, since the dual problem is always feasible with objective zero. Again, an "exact" preconditioner based on Cholesky factorization is used in Fig. 4-2a, and an "approximate" preconditioner based on the hierarchical decomposition in Chapter 5 is used in Figure 4-2b. Note that the barrier method does not monotonously decrease the primal objective, so the relationship between the primal objective and k has the shape of a staircase, rather than a straight line. Nevertheless, an $O(1/k^2)$ error rate is observed in both figures.

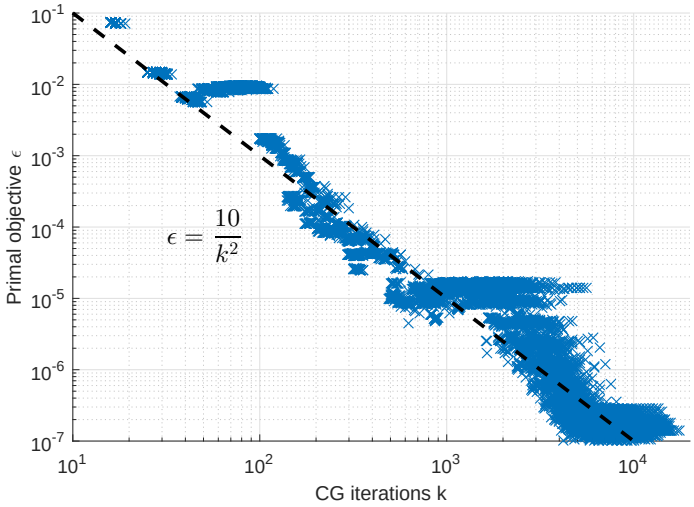


(a)

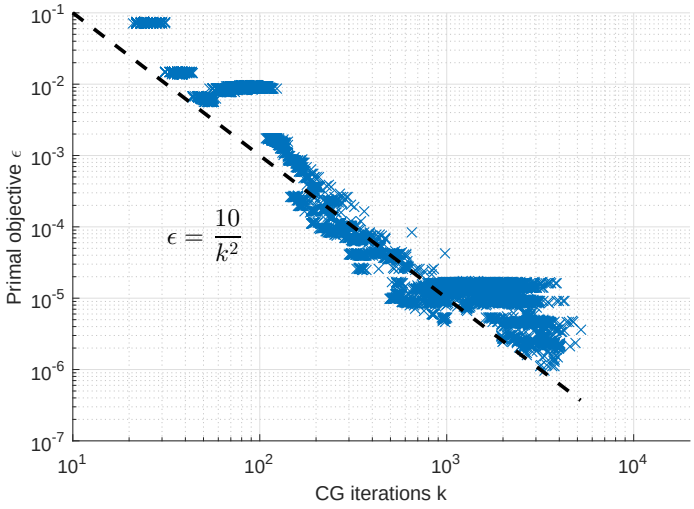


(b)

Figure 4-1: Solving the Newton system $(\mathbf{A}^T\mathbf{D}\mathbf{A})\Delta y = r$ using PCG with the preconditioner $(\mathbf{A}^T\mathbf{A})^{-1}$: (a) "exact" preconditioner via Cholesky factorization; (b) "approximate" preconditioner via hierarchical decomposition.



(a)



(b)

Figure 4-2: Overall error rate of $O(1/k^2)$ for the PCG-Schur interior-point method over all PCG iterations: (a) "exact" preconditioner via Cholesky factorization; (b) "approximate" preconditioner via hierarchical decomposition.

Chapter 5

A Hierarchical Direct Solver for Lyapunov Least-Squares

In Chapter 4, we saw that first-order methods for the Lyapunov inequalities problem solved a particular matrix least squares problem for one or two new right-hand sides at each iteration. Naively solving the least squares problem using Cholesky factorization would incur $O(n^6)$ factorization time and $O(n^4)$ time per right-hand side. A first-order method implemented in this manner has the same time complexity figure as an interior-point method, but takes many more iterations to converge.

In this chapter, we show that when the data matrices are provided as the linearizations of time-domain power system models, that the time complexities can be reduced to $O(n^4)$ factorization and $O(n^3)$ per right-hand side. The insight is to exploit an underlying hierarchical structure, which arises due to the bounded tree-width property of power system networks.

5.1 Introduction

Given the $n \times n$ matrices $M_1, \ldots, M_m$, the Lyapunov inequalities problem,

$$\text{find } P \succ 0 \text{ such that } M_i P + P M_i^T \prec 0 \text{ for all } i \in \{1, \ldots, m\}, \tag{5.1}$$

is a semidefinite feasibility problem fundamental to the field of robust control. When a first-order method is used to solve (5.1), the computational bottleneck is the $n^2 \times n^2$ symmetric positive definite linear system system of equations

$$\mathbf{H}x = \left[\sum_{i=1}^{m} (M_i \otimes I + I \otimes M_i)^T (M_i \otimes I + I \otimes M_i) \right] x = f, \tag{5.2}$$

which must be solved at each iteration with different right-hand sides. Since (5.2) can be viewed as the normal equations for the matrix least-squares

$$\text{minimize} \quad -\operatorname{tr} F^T X + \frac{1}{2}\sum_{i=1}^{m} \|M_i X + X M_i^T\|_F^2 \tag{5.3}$$

with $x = \operatorname{vec} X$ and $f = \operatorname{vec} F$, we we will refer to (5.2) as the *Lyapunov least-squares* problem.

It is common to solve (5.2) using a direct method like Cholesky factorization. After computing and storing the Cholesky factor in an initial factorization step, all subsequent solves for different right-hand sides are made cheap by reusing the stored factorization. Unfortunately, when the data matrices $M_1, \ldots, M_m$ are fully dense, the initial factorization step also has a time complexity of $O(n^6)$. A first-order method implemented this way can never be too much faster than interior-point methods.

To avoid the prohibitive factorization step, an iterative method like conjugate gradients (CG) can be used to solve (5.2). Exploiting the Kronecker structure in the coefficient matrix, the matrix-vector product with $\mathbf{H}$ at each CG iteration costs just $O(n^3 m)$ time. Most implementations restrict the maximum number of CG iterations (per first-order method iteration) to a fixed constant, noting that (5.2) does not have to be solved exactly in order for the underlying first-order method to converge. Unfortunately, the resulting convergence rate is often dramatically slowed when compared to a method implemented with an exact solution.

5.1.1 Main Result

Our $n \times n$ data matrices $M_1, \ldots, M_m$ arise as linearized time-domain models of the power system. We show in Section 5.2 that the data matrices are *hierarchical* whenever the power system has a bounded tree-width. Loosely speaking, it means that each of these matrices can be permuted into a block-diagonal-plus-low-rank form, as in

$$P^T M Q = \begin{bmatrix} A_1 & & \\ & \ddots & \\ & & A_p \end{bmatrix} + \begin{bmatrix} L_1 \\ \vdots \\ L_p \end{bmatrix} K^{-1} \begin{bmatrix} R_1 & \cdots & R_p \end{bmatrix},$$

and the blocks $A_1, \ldots, A_p$ can be recursively permuted and decomposed in the same manner. We show that this hierarchical property is inherited by the Hessian matrix $\mathbf{H}$ in (5.2) in Section 5.3.

Hierarchical matrices are well-known in the study of partial differential equations. It is known that an implicit, hierarchical representation of the matrix inverse may be constructed, and matrix-vector products with the matrix inverse can be performed at a cost significantly less expensive than a naive dense matrix-vector product [95–100].

Based on these ideas, we develop a hierarchical algorithm in Section 5.5 that will factorize the matrix $\mathbf{H}$ in $O(n^4 m^2 + n^3 m^3)$ time and $O(m^2 n^2 \log n)$ storage, and apply the inverse in $O(mn^3 + m^2 n^2 \log n)$ time. Alternatively, the algorithm factors and solves an ϵ-approximation of $\mathbf{H}$ with $O(\sqrt{m} n^4 + mn^3)$ factorization time and

$O(\sqrt{m}n^3)$ per application of the inverse.

5.1.2 Notation

Throughout the chapter, we use the integer n to refer to the size of the matrix variable, P, and the integer m to refer to the number of Lyapunov inequalities in our original problem (5.1).

The number of columns in a matrix A is denoted $\text{ncols}(A)$. Given the $n \times n$ matrix B and a subset of indices $\mathcal{I}, \mathcal{J} \subseteq \{1, \ldots, n\}$, we use the notation $B[\mathcal{I}, \mathcal{J}]$ to refer to the submatrix generated by restricting B to the rows indicated by $\mathcal{I}$ and columns indicated by $\mathcal{J}$.

5.2 Hierarchy in Power System Matrices

As a consequence of the power system application, each of our data matrices is provided in a Schur complement form

$$M = A - B(D - G)^{-1}C,$$

and each of the constituent matrices itself inherits a specific structure. More specifically, for a given block partitioning, the matrices A, B, C, D are *block-diagonal*, and the matrix G has a block-sparsity pattern corresponding to the graph of the underlying power system.

In this section, we will show that if the underlying power system has a graph with *bounded tree-width*, then M is hierarchical.

Definition 19 (Separability)**.** The $m \times n$ rectangular matrix M with $m \geq n$ is said to be $f(n)$-separable into p parts, or simply *separable*, if:

1. It can be written in the form

$$\begin{aligned} M &= \begin{bmatrix} Q_0 & Q_1 & \cdots & Q_p \end{bmatrix} \begin{bmatrix} M_0 & 0 & & 0 \\ 0 & M_1 & & 0 \\ & & \ddots & \\ 0 & 0 & & M_p \end{bmatrix} \begin{bmatrix} P_0 & P_1 & \cdots & P_p \end{bmatrix}^T - LK^{-1}R \\ &= Q\hat{M}P^T - LK^{-1}R. \end{aligned} \tag{5.4}$$

2. The matrix K and the zeroth subblock satisfy

$$\text{ncols}(K) \leq f(n), \qquad \text{ncols}(A_0) \leq f(n).$$

3. The subblocks $i \in \{1, \ldots, p\}$ satisfy

$$M_i \text{ has full column rank and } \text{ncols}(M_i) \leq n/p.$$

Definition 20 (Hierarchy)**.** The $N \times n$ rectangular matrix A with $N \geq n$ is said to be $f(n)$-hierarchical into p parts, or simply *hierarchical*, if:

1. $n \leq f(n)$; OR
2. A is $f(n)$-separable into p parts and each subblock $\{A_1,\ldots,A_p\}$ is itself $f(n)$-hierarchical into p parts;

In fact, the assumption of a small, bounded tree-width is always true for real life power systems. While not usually characterized in this manner, the implications have been known for a long time. More specifically, power system sparse matrices are well-known to have extremely efficient LU factorizations, with fill-in factors never exceeding more than 10-20 times. It is a folklore theorem amongst graph theorists that small fill-in factors imply an underlying graph with small tree-widths [101–103]. Recently, the tree-widths for classic test cases have been explicitly computed [104]. It was found that most standard IEEE test cases have tree-widths of no more than 10, and even the large-scale 3000-bus Polish system test cases have tree-widths of no more than 24.

5.2.1 Time-Domain Models of the Power System

Electric power systems are networks in the graph theoretical sense; their vertices and edges are respectively known as *buses* and *branches*, and each vertex label is a time-dependent complexity quantity $v_i(t) \in \mathbb{C}$ known as the *bus voltage phasor*.

The classic time-domain model for q-bus system is a set of differential algebraic equations (DAE)

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = f(x(t), v(t)), \qquad Y_{\text{bus}}v(t) = g(x(t), v(t)), \tag{5.5}$$

in which f, g are composed from q independent state-space models, and communication between the independent models is entirely facilitated by an algebraic equation relating the voltage phasors, as in

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1(t) \\ \vdots \\ x_q(t) \end{bmatrix} = \begin{bmatrix} f_1(x_1(t), v_1(t)) \\ \vdots \\ f_q(x_q(t), v_q(t)) \end{bmatrix}, \qquad \begin{bmatrix} Y_{11} & \cdots & Y_{1q} \\ \vdots & \ddots & \vdots \\ Y_{q1} & \cdots & Y_{qq} \end{bmatrix}\begin{bmatrix} v_1(t) \\ \vdots \\ v_q(t) \end{bmatrix} = \begin{bmatrix} g_1(x_1(t), v_1(t)) \\ \vdots \\ g_q(x_q(t), v_q(t)) \end{bmatrix}. \tag{5.6}$$

By construction, the bus admittance matrix $Y_{\text{bus}} \in \mathbb{C}^{q\times q}$ has the same nonzero structure as the graph of the underlying power system, meaning that

$$Y_{ij} = \begin{cases} \text{nonzero} & \text{bus } i \text{ connects to bus } j \\ 0 & \text{otherwise} \end{cases}$$

The matrix is usually (but not always) complex symmetric in practice, i.e. $Y_{\text{bus}} = Y_{\text{bus}}^T$.

Each of our data matrices arises by linearizing (5.6) around a specific operating point, and reducing to the canonical state-space form $\frac{\mathrm{d}}{\mathrm{d}t}x(t) = Mx(t)$. To do this, we evaluate the component-wise Jacobians

$$A_i = \frac{\partial f_i}{\partial x_i}, \qquad B_i = \begin{bmatrix} \dfrac{\partial f_i}{\partial(\operatorname{Re} v_i)} & \dfrac{\partial f_i}{\partial(\operatorname{Im} v_i)} \end{bmatrix},$$
$$C_i = \begin{bmatrix} \dfrac{\partial(\operatorname{Re} g_i)}{\partial x_i} \\ \dfrac{\partial(\operatorname{Im} g_i)}{\partial x_i} \end{bmatrix}, \qquad D_i = \begin{bmatrix} \dfrac{\partial(\operatorname{Re} g_i)}{\partial(\operatorname{Re} v_i)} & \dfrac{\partial(\operatorname{Re} g_i)}{\partial(\operatorname{Im} v_i)} \\ \dfrac{\partial(\operatorname{Im} g_i)}{\partial(\operatorname{Re} v_i)} & \dfrac{\partial(\operatorname{Im} g_i)}{\partial(\operatorname{Im} v_i)} \end{bmatrix},$$

and define a real embedding for each $Y_{i,j}$ and $v_i(t)$

$$G_{i,j} = \begin{bmatrix} \operatorname{Re} Y_{i,j} & -\operatorname{Im} Y_{i,j} \\ \operatorname{Im} Y_{i,j} & \operatorname{Re} Y_{i,j} \end{bmatrix}, \qquad y_i(t) = \begin{bmatrix} \operatorname{Re} v_i(t) \\ \operatorname{Im} v_i(t) \end{bmatrix}.$$

Putting the matrices together yields a real, linearized version of (5.6)

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1(t) \\ \vdots \\ x_q(t) \end{bmatrix} = \begin{bmatrix} A_1 & & 0 \\ & \ddots & \\ 0 & & A_q \end{bmatrix}\begin{bmatrix} x_1(t) \\ \vdots \\ x_q(t) \end{bmatrix} + \begin{bmatrix} B_1 & & 0 \\ & \ddots & \\ 0 & & B_q \end{bmatrix}\begin{bmatrix} y_1(t) \\ \vdots \\ y_q(t) \end{bmatrix}, \tag{5.7}$$
$$\begin{bmatrix} G_{11} & \cdots & G_{1q} \\ \vdots & \ddots & \vdots \\ G_{qq} & \cdots & G_{qq} \end{bmatrix}\begin{bmatrix} y_1(t) \\ \vdots \\ y_q(t) \end{bmatrix} = \begin{bmatrix} C_1 & & 0 \\ & \ddots & \\ 0 & & C_q \end{bmatrix}\begin{bmatrix} x_1(t) \\ \vdots \\ x_q(t) \end{bmatrix} + \begin{bmatrix} D_1 & & 0 \\ & \ddots & \\ 0 & & D_q \end{bmatrix}\begin{bmatrix} y_1(t) \\ \vdots \\ y_q(t) \end{bmatrix}.$$

Eliminating the algebraic variable $y_1(t)$ reduces (5.7) to the canonical form $\frac{\mathrm{d}}{\mathrm{d}t}x(t) = Mx(t)$ where

$$M = \begin{bmatrix} A_1 & & 0 \\ & \ddots & \\ 0 & & A_q \end{bmatrix} - \begin{bmatrix} B_1 & & 0 \\ & \ddots & \\ 0 & & B_q \end{bmatrix}\begin{bmatrix} D_1 - G_{11} & \cdots & -G_{1q} \\ \vdots & \ddots & \vdots \\ -G_{qq} & \cdots & D_q - G_{qq} \end{bmatrix}^{-1}\begin{bmatrix} C_1 & & 0 \\ & \ddots & \\ 0 & & C_q \end{bmatrix} \tag{5.8}$$
$$= A - B(D-G)^{-1}C.$$

As illustrated in (5.8), the matrices A, B, C, D are block-diagonal, and the block sparsity pattern of G coincides with the graph of the underlying power system. Let us state these two ideas more rigorously. We define the block index partitions $\mathcal{I}_1, \ldots, \mathcal{I}_q$ as the location of each block $x_1(t), \ldots, x_q(t)$ within the vector concatenation $x(t)$, and $\mathcal{J}_1, \ldots, \mathcal{J}_q$ as the location of each block $y_1(t), \ldots, y_q(t)$ within the vector concatenation $y(t)$.

Definition 21 (Block-diagonal)**.** Given a partitioning of the indices $\{1, \ldots, n\} = \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_q$ and $\{1, \ldots, \nu\} = \mathcal{J}_1 \cup \cdots \cup \mathcal{J}_q$ each into q possibly empty parts. We say

that the matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times\nu}$, $C \in \mathbb{R}^{\nu\times n}$, $D \in \mathbb{R}^{\nu\times\nu}$ are *block-diagonal* if all off-diagonal blocks are zero, i.e. if $A[\mathcal{I}_i, \mathcal{I}_j] = 0$, $B[\mathcal{I}_i, \mathcal{J}_j] = 0$, $C[\mathcal{J}_i, \mathcal{I}_j] = 0$ and $D[\mathcal{J}_i, \mathcal{J}_j] = 0$ for all $i \neq j$ with $\mathcal{I}_i$ and $\mathcal{J}_j$ both nonempty.

Definition 22 (Block-sparsity graph)**.** Given a partitioning of the indices $\{1, \ldots, n\} = \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_q$ into q nonempty parts. Let $\mathcal{G} = \{V, E\}$ be an undirected graph defined on q vertices. We say that $\mathcal{G}$ is a *block-sparsity graph* for the matrix $G \in \mathbb{R}^{n\times n}$ if the matrix G has zero blocks for every vertex pair in $\mathcal{G}$ not adjacent to each other, i.e. if $G[\mathcal{I}_i, \mathcal{I}_j] = 0$ and $G[\mathcal{I}_j, \mathcal{I}_i] = 0$ for all i, j such that $v_i, v_j \in V$ and $\{v_i, v_j\} \notin E$.

5.2.2 Bounded Tree-width & Nested Dissection

A central insight throughout this chapter is that power systems admit graphs with small tree-widths. Loosely speaking, these are graphs that can be recursively bisected, each time by removing no more than a small constant number of vertices. A precise definition is technical, and we refer the reader to e.g. [102, 105] for a more thorough exposition.

Definition 23. An α-vertex separator of $W \subseteq V$ in $\mathcal{G} = \{V, E\}$ is a set $S \subseteq V$ of vertices such that every connected component of the graph $\mathcal{G}' = \mathcal{G}[V - S]$, obtained by removing S from $\mathcal{G}$, contains at most $\alpha \cdot |W|$ vertices of W.

Lemma 24 ([102, Lem.6])**.** *Let $\mathcal{G} = \{V, E\}$ be a graph with tree-width $\leq \tau$. Let $W \subseteq V$. Then there exists a $\frac{1}{2}$-vertex separator of W in $\mathcal{G}$ of size at most $\tau + 1$.*

The classical application of vertex separators is in the nested dissection solution of sparse symmetric positive definite systems [106]. Let $D = D^T$ be any $n \times n$ symmetric positive-definite matrix with $\mathcal{G}$ as its sparsity graph. The complexity of solving the system $Dx = b$ depends entirely on the number of nonzeros in the Cholesky factorization $U^T U = D$.

Suppose that $\mathcal{G}$ has bounded tree-width. Then by there exists a vertex separator $S \subset V$ that divides $V \backslash S$ into two disconnected partitions, X and Y. Defining Π as the reordering $\{X, Y, S\} \mapsto V$ reveals a familiar "arrow" structure,

$$\Pi^T D \Pi = \begin{bmatrix} D_X & 0 & D_{XS} \\ 0 & D_Y & D_{YS} \\ D_{XS}^T & D_{YS}^T & D_S \end{bmatrix}$$

where the block D_S is "small", and the blocks D_X and D_Y are each no more than half the size of D. Reordered this way, the matrix can be factored

$$\Pi^T D \Pi = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ D_{XS}^T D_X^{-1} & D_{YS}^T D_Y^{-1} & I \end{bmatrix} \begin{bmatrix} D_X & 0 & 0 \\ 0 & D_Y & 0 \\ 0 & 0 & K \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ D_{XS}^T D_X^{-1} & D_{YS}^T D_Y^{-1} & I \end{bmatrix}^T, \tag{5.9}$$

where $K = D_S - D_{XS}^T D_X^{-1} D_{XS} - D_{YS}^T D_Y^{-1} D_{YS}$. Put in another way, the factorization of D contains the factorization of two of its submatrices, each of half its size, plus a

further $O(n)$ nonzeros. Since each of these subblocks D_X and D_Y can be recursively treated in the same manner (i.e. the "nested" part of nested dissection), the combined factorization of D can be shown to contain no more than $O(n \log n)$ nonzeros. In fact, this figure is within $O(\log n)$ of the best possible.

Nested dissection is closely related to the idea of hierarchical matrices. Loosely speaking, if a nested dissection ordering exists for the matrix D, then the inverse D^{-1} is hierarchical. To show this, let use invert the factorized form in (5.9)

$$\begin{aligned}\Pi^T D^{-1} \Pi &= \begin{bmatrix} I & 0 & -D_X^{-1} D_{XS} \\ 0 & I & -D_Y^{-1} D_{YS} \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} D_X^{-1} & 0 & 0 \\ 0 & D_Y^{-1} & 0 \\ 0 & 0 & K^{-1} \end{bmatrix} \begin{bmatrix} I & 0 & -D_X^{-1} D_{XS} \\ 0 & I & -D_Y^{-1} D_{YS} \\ 0 & 0 & I \end{bmatrix}^T \\ &= \begin{bmatrix} D_X^{-1} & 0 & 0 \\ 0 & D_Y^{-1} & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} -D_X^{-1} D_{XS} \\ -D_Y^{-1} D_{YS} \\ I \end{bmatrix} K^{-1} \begin{bmatrix} -D_X^{-1} D_{XS} \\ -D_Y^{-1} D_{YS} \\ I \end{bmatrix}^T,\end{aligned}$$

to reveal the familiar block-diagonal-plus-low-rank structure, suggesting that D^{-1} is separable. Repeating the same argument for each subblock D_X and D_Y, we find that D^{-1} is also hierarchical.

The fact that every nested dissection matrix admits a hierarchical inverse is a folklore theorem well-known in the numerical solution of partial differential equations [107]. In many cases, D is the finite difference / finite element discretization of a differential operator, and D^{-1} is a discretization of the corresponding integral operator. While D^{-1} is fully dense, its hierarchy allows a data-sparse representation to be constructed, and matrix-vector products with D^{-1} may be performed at just $O(n)$ or $O(n \log n)$ complexity.

In order for the twin ideas of nested dissection and hierarchical matrices to be extended to nonsymmetric and symmetric indefinite matrices, we must have the strong factorizability property.

Definition 25 (Strongly block-factorizable)**.** Given a partitioning of the indices $\{1, \ldots, n\} = \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_q$ into q nonempty parts. We say that $G \in \mathbb{R}^{n \times n}$ is strongly block-factorizable if every symmetric block-permutation of G is block-factorizable.

5.2.3 Hierarchy of the data matrix

Let us now formalize the intuition in the previous subsection, and extend it to the block-matrix case for $M = A - B(D - G)^{-1}C$.

Theorem 26. *Given a partitioning of $\{1, \ldots, n\} = \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_q$ into q equally-sized parts, and a partitioning of $\{1, \ldots, \nu\} = \mathcal{J}_1 \cup \cdots \cup \mathcal{J}_q$ into q equally-sized parts. Let $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times \nu}$, $C \in \mathbb{R}^{\nu \times n}$, and $D \in \mathbb{R}^{\nu \times \nu}$ be block-diagonal, let $G \in \mathbb{R}^{\nu \times n}$ be such that $D - G$ is strongly block-factorizable, and let $\mathcal{G}$ be a block-sparsity graph for G with tree-width $\leq \tau$. Then the matrix $M = A - B(D - G)^{-1}C$ is $\gamma(\tau + 1)$-hierarchical into 2 parts, where $\gamma = \max_{1 \leq i \leq q} \{|\mathcal{I}_i|, |\mathcal{J}_i|\}$.*

Let us fix some notation before proceeding with the proof. We identify the q vertices in $\mathcal{G}$ with the q index partitions $\mathcal{I}_1, \ldots, \mathcal{I}_q$ and the q index partitions $\mathcal{J}_1, \ldots, \mathcal{J}_q$. Given a set of vertices $X \subseteq V$, we use the notation $\mathcal{I}_X \triangleq \bigcup_{i \in X} \mathcal{I}_i$ and $\mathcal{J}_X \triangleq \bigcup_{i \in X} \mathcal{J}_i$ to refer to the union of the corresponding index partitions. Given two sets of vertices, $X, Y \subseteq V$, we use the notations $A_X \triangleq A[\mathcal{I}_X, \mathcal{I}_X]$ and $A_{XY} \triangleq A[\mathcal{I}_X, \mathcal{I}_Y]$ to refer to the corresponding submatrices of A, and similarly for the submatrices $B_{XY} \triangleq B[\mathcal{I}_X, \mathcal{J}_Y]$, $C_{XY} \triangleq C[\mathcal{J}_X, \mathcal{I}_Y]$, $D_{XY} \triangleq D[\mathcal{J}_X, \mathcal{J}_Y]$, and $G_{XY} \triangleq G[\mathcal{J}_X, \mathcal{J}_Y]$.

Proof. We will show that every

$$M_W \triangleq A_W - B_W (D_W - G_W)^{-1} C_W$$

is hierarchical. We begin by showing that M_W is separable. Again, the key step is to use Lemma 24 to select the vertex separator $S \subset W$ and two disconnected partitions X, Y for $W \backslash S$. Define Π as the reordering $\{\mathcal{I}_S, \mathcal{I}_X, \mathcal{I}_Y\} \mapsto \mathcal{I}_W$ and Φ as the reordering $\{\mathcal{J}_S, \mathcal{J}_X, \mathcal{J}_Y\} \mapsto \mathcal{J}_W$. Then each submatrix is given

$$A_W = \Pi \begin{bmatrix} A_S & 0 & 0 \\ 0 & A_X & 0 \\ 0 & 0 & A_Y \end{bmatrix} \Pi^T, \qquad B_W = \Pi \begin{bmatrix} B_S & 0 & 0 \\ 0 & B_X & 0 \\ 0 & 0 & B_Y \end{bmatrix} \Phi^T,$$

$$C_W = \Phi \begin{bmatrix} C_S & 0 & 0 \\ 0 & C_X & 0 \\ 0 & 0 & C_Y \end{bmatrix} \Pi^T, \quad D_W - G_W = \Phi \begin{bmatrix} D_S - G_S & -G_{SX} & -G_{XY} \\ -G_{XS} & D_X - G_X & 0 \\ -G_{YS} & 0 & D_Y - G_Y \end{bmatrix} \Phi^T.$$

Applying the Sherman–Morrison–Woodbury formula to $(D_W - G_W)^{-1}$ reveals a block-diagonal-plus-low-rank decomposition

$$\begin{aligned} M_W = \Pi \begin{bmatrix} A_S & 0 & 0 \\ 0 & M_X & 0 \\ 0 & 0 & M_Y \end{bmatrix} \Pi^T \\ + \Pi \begin{bmatrix} B_S \\ B_X (D_X - G_X)^{-1} G_{XS} \\ B_Y (D_Y - G_Y)^{-1} G_{YS} \end{bmatrix} K^{-1} \begin{bmatrix} C_S^T \\ C_X^T (D_X - G_X)^{-T} G_{SX}^T \\ C_Y^T (D_Y - G_Y)^{-T} G_{SY}^T \end{bmatrix}^T \Pi^T \end{aligned}$$

where

$$K = G_S - D_S + G_{SX} (D_X - G_X)^{-1} G_{XS} + G_{SY} (D_Y - G_Y)^{-1} G_{YS}.$$

The block A_S is at most size $|\mathcal{I}_S| \leq \gamma(\tau + 1)$, the matrix K is at most size $|\mathcal{J}_S| \leq \gamma(\tau + 1)$, and the blocks M_X and M_Y are at most size $\max\{|\mathcal{I}_X|, |\mathcal{I}_Y|\} \leq |\mathcal{I}_W|/2$. Hence, M_W is $\gamma(\tau + 1)$-separable into two parts. Inductively repeating the same argument for the blocks M_X and M_Y, we find that M_W is actually $(\tau+1)$-hierarchical into two parts. □

Remark 27. We would like our matrix $D - G$ to be strongly block-factorizable. One case where this is definitively true is when Y_{bus} is complex symmetric, $\operatorname{Re} Y_{\text{bus}}$ is

symmetric positive definite, and $D = 0$, since it would make G quasi-definite [108]. Physically, this is a power system containing only generators, simple branches (i.e. transmission lines and simple transformers), and constant-impedance loads. For more general power systems with directional branches and constant-current / constant-power loads, however, strong factorizability is only an assumption.

5.3 Hierarchy in the Hessian Matrix

Now, consider a set of m matrices $M_1, \ldots, M_m$, each $n \times n$ and of the form $M_j = A_j - B_j(D_j - G_j)^{-1}C_j$, with A_j, B_j, C_j, and D_j block-diagonal, $D_j - G_j$ strongly block-factorizable, and all G_j sharing a common block-sparsity graph $\mathcal{G}$ on q vertices, with tree-width τ. By Theorem 26 in the previous section, each of these matrices is $\gamma(\tau+1)$-hierarchical into 2 parts. In this section, we will show that the $mN \times N$ rectangular matrix

$$\mathbf{A} = \begin{bmatrix} M_1 \otimes I + I \otimes M_1 \\ \vdots \\ M_m \otimes I + I \otimes M_m \end{bmatrix}. \tag{5.10}$$

is $O(m\sqrt{N})$-hierarchical, which immediately implies the same statement for $\mathbf{H} = \mathbf{A}^T\mathbf{A}$ via the expansion

$$\begin{aligned} \mathbf{A}^T\mathbf{A} &= (Q\hat{\mathbf{A}}P^T - BD^{-1}C)^T(Q\hat{\mathbf{A}}P^T - BD^{-1}C) \\ &= P(\hat{\mathbf{A}}^T\hat{\mathbf{A}})P^T - \begin{bmatrix}(\hat{\mathbf{A}}Q)^TB & C^T\end{bmatrix}\begin{bmatrix}B^TB & D^T \\ D & 0\end{bmatrix}^{-1}\begin{bmatrix}(\hat{\mathbf{A}}Q)^TB & C^T\end{bmatrix}^T, \end{aligned} \tag{5.11}$$

and recursive application of this expansion to each subblock of $\hat{\mathbf{A}}^T\hat{\mathbf{A}}$.

5.3.1 Shared hierarchy and compression

By the conditions stated at the start of this section, each of our data matrices can be written in the form

$$M_j = \Pi\hat{M}_j\Pi^T - L_jK_j^{-1}R_j$$

using the same permutation matrices Π, and sharing the same block divisions in $\hat{A}_1, \ldots, \hat{A}_m$. Then, vertically stacking these matrices,

$$M = \begin{bmatrix} M_1 \\ \vdots \\ M_m \end{bmatrix} = \begin{bmatrix} \Pi & & 0 \\ & \ddots & \\ 0 & & \Pi \end{bmatrix}\begin{bmatrix} \hat{M}_1 \\ \vdots \\ \hat{M}_m \end{bmatrix}\Pi^T - \begin{bmatrix} L_1 & & 0 \\ & \ddots & \\ 0 & & L_m \end{bmatrix}\begin{bmatrix} K_1 & & 0 \\ & \ddots & \\ 0 & & K_m \end{bmatrix}^{-1}\begin{bmatrix} R_1 \\ \vdots \\ R_m \end{bmatrix}$$

we find that M is $O(m)$-hierarchical. The $O(m)$ factor here greatly inflates the complexity of hierarchical algorithms designed to invert the normal matrix M^TM.

Instead, we may construct an ϵ-accurate $O(1)$-hierarchical approximation of $\mathbf{M}$.

Consider performing a singular value decomposition on the perturbation term, and partitioning the singular values into a larger set Σ_a and a smaller set Σ_b, as in

$$\begin{bmatrix} L_1 K_1^{-1} R_1^T \\ \vdots \\ L_m K_m^{-1} R_m^T \end{bmatrix} = \begin{bmatrix} U_1 \\ \vdots \\ U_m \end{bmatrix} \Sigma V^T = \begin{bmatrix} U_{1,a} & U_{1,b} \\ \vdots & \vdots \\ U_{m,a} & U_{m,b} \end{bmatrix} \begin{bmatrix} \Sigma_a & 0 \\ 0 & \Sigma_b \end{bmatrix} \begin{bmatrix} V_a^T \\ V_b^T \end{bmatrix}. \tag{5.12}$$

Truncating the smaller singular values yields an approximation

$$\tilde{M} = \begin{bmatrix} \Pi & & 0 \\ & \ddots & \\ 0 & & \Pi \end{bmatrix} \begin{bmatrix} \hat{M}_1 \\ \vdots \\ \hat{M}_m \end{bmatrix} \Pi^T - \begin{bmatrix} U_{1,a} \\ \vdots \\ U_{m,a} \end{bmatrix} \Sigma_a \begin{bmatrix} V_a^T \end{bmatrix},$$

satisfying the error bounds

$$\|M - \tilde{M}\| = \|\Sigma_b\|.$$

The matrix $\tilde{M}$ is γr-hierarchical, where $r = \text{rank}(\Sigma_a) \in \{1, \ldots, m(\tau+1)\}$ is just the number of singular values remaining in Σ_a. It is always possible to pick $r \in O(1)$, for some sufficiently large choice of ϵ, in order to make the resulting approximation is $O(1)$-hierarchical. In practice, we often found that $r \in O(1)$ even when ϵ is set to zero, due to a level of low-rank redundancy in the matrices $R_1, \ldots, R_m$.

The singular value decomposition in (5.12) can be efficiently performed in $O(m^2\tau^2 n + m^3\tau^3)$ time and $O(m^2\tau n)$ memory by exploiting the low-rank structure of each submatrix $B_j D_j^{-1} C_j$.

Algorithm 28 (Low-rank SVD)**. Input:** matrices $\{L_1, \ldots, L_m\}$, $\{K_1, \ldots, K_m\}$, and $\{R_1, \ldots, R_m\}$.
Output: the singular value decomposition $\{U_1, \ldots, U_m\}$, Σ, and V, satisfying (5.12).

1. Perform m size $n \times (\tau+1)$ QR decompositions for each $L_j K_j^{-1} = Q_j \hat{L}_j$.

2. Perform a single size $n \times m(\tau+1)$ QR decomposition of $[R_1; \ldots; R_m] = \hat{R} P^T$.

3. Compute the size-$m(\tau+1)$ singular value decomposition

$$\begin{bmatrix} \hat{L}_1 & & \\ & \ddots & \\ & & \hat{L}_m \end{bmatrix} \hat{R} = \Phi \Sigma \Psi^T$$

   and partition the rows $\Phi = [\Phi_1; \ldots; \Phi_m]$ into m blocks of $\tau+1$.

4. Form $U_i = Q_i \Phi_i$ and $V = P\Psi$ and return.

5.3.2 Hierarchy of the matrix $M \otimes I + I \otimes M$

After the singular value decomposition step in the previous subsection, our a set of m hierarchical data matrices $M_1, \ldots, M_m$ now satisfy

$$M_j = \Pi \hat{M}_j \Pi^T - B_j D^{-1} C,$$

where the matrices C, D are now commonly to all m matrices.

Let us consider the hierarchy of the $N \times N$ matrix $\mathbf{A}_j = M_j \otimes I + I \otimes M_j$,

$$\begin{aligned} \mathbf{A}_j =& (\Pi \otimes \Pi)^T (\hat{M}_j \otimes I + I \otimes \hat{M}_j)(\Pi \otimes \Pi) \\ & - \begin{bmatrix} B_j \otimes I & I \otimes B_j \end{bmatrix} \begin{bmatrix} D \otimes I & 0 \\ 0 & I \otimes D \end{bmatrix}^{-1} \begin{bmatrix} C \otimes I \\ I \otimes C \end{bmatrix} \\ =& (\Pi \otimes \Pi)^T (\hat{M}_j \otimes I + I \otimes \hat{M}_j)(\Pi \otimes \Pi) - \mathbf{B}_j \mathbf{D}^{-1} \mathbf{C}. \end{aligned} \tag{5.13}$$

The matrix $\hat{M}_j \otimes I + I \otimes \hat{M}_j$ is *not* block-diagonal, but as we will show below, there always exists a permutation matrix Ψ to make it block-diagonal. The resulting block-diagonal-plus-low-rank decomposition is

$$\mathbf{A}_j = \mathbf{P} \hat{\mathbf{A}}_j \mathbf{P}^T - \mathbf{B}_j \mathbf{D}^{-1} \mathbf{C} \tag{5.14}$$

where $\mathbf{P} = (\Pi \otimes \Pi)\Psi$ and $\hat{\mathbf{A}}_j = \Psi^T (\hat{M}_j \otimes I + I \otimes \hat{M}_j)\Psi$. Accordingly, we have shown that $\mathbf{A}$, as defined at the start of the section in (5.10), is hierarchical.

To illustrate the choice of the permutation Ψ, let us consider a simple block-diagonal matrix $D = \mathrm{diag}(D_1, D_2)$. Applying the Kronecker identity $(A \otimes B)\mathrm{vec}\, X = \mathrm{vec}\,(BXA^T)$ yields

$$(D \otimes I + I \otimes D)\mathrm{vec}\, X = \mathrm{vec} \begin{bmatrix} D_1 X_{11} + X_{11} D_1^T & D_1 X_{12} + X_{12} D_2^T \\ D_2 X_{21} + X_{p1} D_1^T & D_2 X_{22} + X_{22} D_2^T \end{bmatrix}$$

with the appropriate block-divisions. If we define Ψ as the *block-vectorization* permutation

$$\mathrm{vec} \left(\begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \right) \mapsto \begin{bmatrix} \mathrm{vec}\, X_{11} \\ \mathrm{vec}\, X_{21} \\ \mathrm{vec}\, X_{12} \\ \mathrm{vec}\, X_{22} \end{bmatrix},$$

then the permuted matrix $\Psi^T (D \otimes I + I \otimes D)\Psi$ is block-diagonal. It is easy to extend these arguments to the p-block case.

Lemma 29. *Given the $n_1 \times n_1$, α-separable matrix A into p_1 parts, and the $n_2 \times n_2$, β-separable matrix B into p_1 parts. Define the product $N = n_1 n_2$ and the aspect ratio $r = \max\{\frac{n_1}{n_2}, \frac{n_2}{n_1}\}$. Then the $N \times N$ matrix $\mathbf{M} = A \otimes I_{n_2} + I_{n_1} \otimes B$ is $(\alpha + \beta)\sqrt{rN}$-separable into $p_1 p_2$ parts.*

Proof. By definition, we have $A = P_1 \hat{A} Q_1^T - L_1 K_1^{-1} R_1^T$ and $B = P_2 \hat{B} Q_2^T - L_2 K_2^{-1} R_2^T$ with block diagonal $\hat{A} = \bigoplus_{i=0}^{p_1} A_i$ and $\hat{B} = \bigoplus_{j=0}^{p_2} B_j$. Then some simple algebraic

manipulations yields

$$\mathbf{M} = \mathbf{P}\hat{\mathbf{M}}\mathbf{Q}^T + \mathbf{L}\mathbf{K}^{-1}\mathbf{R}$$

where, letting Φ be a suitable block-vectorization permutation, we have

$$\begin{aligned}
\mathbf{P} &= (P_1 \otimes P_2)\Phi, & \mathbf{Q} &= (Q_1 \otimes Q_2)\Phi, \\
\hat{\mathbf{M}} &= \Phi^T(\hat{A} \otimes I_{n_2} + I_{n_1} \otimes \hat{B})\Phi, & \mathbf{L} &= \begin{bmatrix} L_1 \otimes I_{n_2} & I_{n_1} \otimes L_2 \end{bmatrix}, \\
\mathbf{K} &= \begin{bmatrix} K_1 \otimes I_{n_2} & 0 \\ 0 & I_{n_1} \otimes K_2 \end{bmatrix}, & \mathbf{R} &= \begin{bmatrix} R_1 \otimes I_{n_2} \\ I_{n_1} \otimes R_2 \end{bmatrix}.
\end{aligned}$$

The matrix $\hat{\mathbf{M}} = \bigoplus_{i=0}^{p_1} \bigoplus_{j=0}^{p_2} (A_i \otimes I + I \otimes B_j)$ is block-diagonal with $(p_1+1)(p_2+1)$ blocks. Grouping the blocks associated with the zeroth blocks reduces this into p_1p_2+1 subblocks

$$\begin{aligned}
\hat{\mathbf{M}} &= \mathbf{M}_0 \oplus \left(\bigoplus_{i=1}^{p_1} \bigoplus_{j=1}^{p_2} \mathbf{M}_{i,j} \right) \text{ where} \\
\mathbf{M}_0 &= \left(\bigoplus_{i=0}^{p_1} A_i \otimes I + I \otimes B_0 \right) \oplus \left(\bigoplus_{j=1}^{p_2} A_0 \otimes I + I \otimes B_j \right) \text{ and} \\
\mathbf{M}_{ij} &= A_i \otimes I + I \otimes B_j.
\end{aligned}$$

Both the blocks $\mathbf{M}_0$ and $\mathbf{D}$ are at most size $\beta n_1 + \alpha n_2$, while each subblock $\mathbf{M}_{i,j}$ is at most size $(d_1d_2)/(p_1p_2)$. Since $\max\{n_1, n_2\} \leq \sqrt{rN}$, we have $\beta n_1 + \alpha n_2 \leq (\alpha+\beta)\sqrt{rN}$. □

An equivalent hierarchy statement is more difficult, because the aspect ratio $r = \max\{\frac{n_1}{n_2}, \frac{n_2}{n_1}\}$ tends to become worse with each additional level of hierarchy. To illustrate, suppose that A were an α-hierarchical $n \times n$ matrix with p parts. Then by definition, there exists the decomposition

$$A = Q \begin{bmatrix} A_0 & 0 & & 0 \\ 0 & A_1 & & 0 \\ & & \ddots & \\ 0 & 0 & & A_p \end{bmatrix} P^T - LK^{-1}R,$$

where A_0 is as large as α, and each of the subblocks $A_1, \ldots, A_p$ can be as large as n/p. Since the size of these subblocks must add up to the size of the original matrix

A, this also implies a lower-bound

$$\begin{aligned}\text{ncols}(A_1) &= \text{ncols}(A) - \sum_{i=2}^{p} \text{ncols}(A_i) - \text{ncols}(A_0) \\ &\geq n\left[1 - (p-1)\frac{1}{p}\right] - \alpha \\ &= \frac{n}{p} - \alpha.\end{aligned}$$

The gap between and the upper- and lower-bound is a constant α, but the block sizes have become smaller.

At the next level, each i-th submatrix A_i is itself α-hierarchical into p parts

$$A_i = Q_i \begin{bmatrix} A_{0,i} & 0 & & 0 \\ 0 & A_{1,i} & & 0 \\ & & \ddots & \\ 0 & 0 & & A_{p,i} \end{bmatrix} P_i^T - L_i K_i^{-1} R_i \qquad i \in \{1, \ldots, p\},$$

where $A_{0,i}$ is again as large as α, and each $A_{1,i}, \ldots, A_{p,i}$ is as large as $\text{ncols}(A_1)/p$. Using the same logic as before we find that

$$\begin{aligned}\text{ncols}(A_{1,1}) &= \text{ncols}(A_1) - \sum_{i=2}^{p} \text{ncols}(A_{i,1}) - \text{ncols}(A_{0,1}) \\ &\geq \text{ncols}(A_1)\left[1 - (p-1)\frac{1}{p}\right] - \alpha \\ &= \frac{\text{ncols}(A_1)}{p} - \alpha = \frac{n}{p^2} - \frac{\alpha}{p} - \alpha.\end{aligned}$$

Hence, while the blocks have gotten another factor of p smaller, the upper-lower-bound gap has gotten slightly bigger. Extending these statements by induction to the k-th level yields an expression for the upper-lower-bound gap at the k-th level.

Lemma 30. *The size of the k-th level hierarchical subblock of A is bound*

$$\frac{d}{p^k} - c\alpha \leq \text{ncols}(A_{1,\ldots,1}) \leq \frac{d}{p^k}, \tag{5.15}$$

where $c = 1/(1 - 1/p)$.

Hence, the ratio between the upper- and lower-bounds bounds the aspect ratio encountered by a hierarchical expansion of $A \otimes I + I \otimes A$ at the k-th level

$$r_k \leq \frac{n}{n - cp^k\alpha},$$

is a strictly increasing function of k. Bounding the maximum ratio yields to our main result for this section.

Theorem 31. *Given the α-hierarchical $n \times n$ matrix M with into p parts. Let $p \geq 2$, and $\alpha\sqrt{2} \leq n$. Then the $N \times N$ matrix $\mathbf{M} = M \otimes I + I \otimes M$ is $\alpha\sqrt{8N}$-hierarchical into p^2 parts.*

Proof. Suppose we chose the maximum number of levels ℓ to satisfy $cp^\ell\alpha \leq n/2$. Then our aspect ratios are bounded $r_k \leq 2$ for all $k \in \{1, \ldots, \ell\}$, and we may recursively apply Lemma 29 to establish the second clause in Definition 20. Let us show that the ℓ-th level have sizes smaller than $\alpha\sqrt{8N}$. First, solving for equality yields

$$\ell = \log_p\left(\frac{n}{2c\alpha}\right) = \frac{1}{\log p}\log\left(\frac{n}{2c\alpha}\right).$$

Next, the ℓ-th level has matrix size at most $N/p^{2\ell} = (2c\alpha)^2$. Since $c = 1/(1-1/p) \leq 2$ for all $p \geq 2$, we have $\mathrm{ncols}(A_{1,\ldots,1}) \leq 4\alpha^2 \leq \alpha\sqrt{8N}$ as desired. □

5.4 Direct Solvers for Hierarchical Matrices

Finally, we describe direct methods for solving the system of equations

$$Ax = f, \tag{5.16}$$

when A is a square, invertible matrix that is $f(n)$-separable with parameters $\{p, N\}$. By definition, there exists a choice of the matrices Q, P, L, K, R and block-diagonal $\hat{A}$ such that $A = Q\hat{A}P^T - LK^{-1}R$. Viewing A as the Schur complement of an enlarged matrix, we may rewrite (5.16) into the following

$$\begin{bmatrix} K & RP \\ Q^TL & \hat{A} \end{bmatrix}\begin{bmatrix} y \\ P^Tx \end{bmatrix} = \begin{bmatrix} 0 \\ Q^Tf \end{bmatrix}, \tag{5.17}$$

which has the familiar "arrow" structure, due to the block-diagonal structure of $\hat{A}$,

$$\left[\begin{array}{cc|ccc} D & RP_0 & RP_1 & \cdots & RP_p \\ Q_0^TL & A_0 & 0 & \cdots & 0 \\ \hline Q_1^TL & 0 & A_1 & & \\ \vdots & \vdots & & \ddots & \\ Q_p^TL & 0 & & & A_p \end{array}\right]\left[\begin{array}{c} y \\ P_0^Tx \\ \hline P_1^Tx \\ \vdots \\ P_p^Tx \end{array}\right] = \left[\begin{array}{c} 0 \\ Q_0^Tf \\ \hline Q_1^Tf \\ \vdots \\ Q_p^Tf \end{array}\right].$$

In effect, we have written the inverse of the fully-dense matrix A as a submatrix of the inverse of a larger but sparser matrix in (5.17).

Proposition 32. *Given the $f(n)$-separable matrix A, the matrix in (5.17) is invertible whenever A is invertible.*

Proof. Let us denote the matrix in (5.17) as B. Performing block elimination, we havedet$(B) = \det(K)\det(\hat{A} - Q^TLK^{-1}RP) = \det(K)\det(A)$. The matrix K is invertible by construction, so $\det(M) = 0$ if and only if $\det(A) = 0$. □

Defining $\tilde{P} = \begin{bmatrix} 0 & P \end{bmatrix}$ and $\tilde{Q} = \begin{bmatrix} 0 & Q \end{bmatrix}$ as zero-padded versions of P, Q, the above statement implies $A^{-1} = \tilde{P} B^{-1} \tilde{Q}^T$.

5.4.1 Explicit Matrix Scheme

If each block matrix within $\hat{A}$ is also hierarchical, then the same expansion in (5.17) can be recursively applied to each subblock in (5.17), thereby converting the dense system of equations (5.16) into a larger, and ultimately, sparse system of equations. Furthermore, this larger sparse matrix can be factored into *sparse* triangular factors. Our proof is constructive, and produces an efficient algorithm.

Algorithm 33 (Explicit-matrix Factorization)**.** $\{\Phi, W, S^{-1}, T, \Psi\} = \mathbf{ExpFac}(A)$
Input: invertible $n \times n$ matrix A that is $f(n)$-hierarchical into p partitions according to (5.4);
Output: unit-diagonal triangular factors W, T, block-diagonal S^{-1}, and rectangular permutation matrix Φ satisfying $A^{-1} = \Phi^T (WST)^{-1} \Psi$.

1. (Subproblems) For each $i \in \{1, \ldots, p\}$, compute a factorized representation of the A_i^{-1} subblock via the recursive call $\{\Phi_i, W_i, S_i^{-1}, T_i, \Psi_i\} = \mathbf{ExpFac}(A_i)$.

2. (Master problem) Form the Schur complement

$$K_0 = K - \sum_{i=1}^{p} R P_i \Phi_i^T T_i^{-1} S_i^{-1} W_i^{-1} \Psi_i Q_i^T L, \qquad S_0 = \begin{bmatrix} K_0 & R P_0 \\ Q_0^T L & A_0 \end{bmatrix}.$$

3. (Output) Form the block-diagonal matrix S^{-1}, and the triangular factors W, T, via

$$\Phi = \begin{bmatrix} 0 & P_0 & P_1 & \cdots & P_p \end{bmatrix}^T \tag{5.18}$$

$$W = \begin{bmatrix} I & 0 & R P_1 T_1^{-1} S_1^{-1} & \cdots & R P_p T_p^{-1} S_p^{-1} \\ 0 & I & 0 & \cdots & 0 \\ 0 & 0 & W_1 & & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & & W_p \end{bmatrix} \tag{5.19}$$

$$S^{-1} = S_0^{-1} \oplus S_1^{-1} \oplus \cdots \oplus S_p^{-1} \tag{5.20}$$

$$T = \begin{bmatrix} I & & 0 & 0 & & 0 \\ 0 & & I & 0 & & 0 \\ S_1^{-1} W_1^{-1} Q_1^T L & & 0 & T_1 & & 0 \\ \vdots & & \vdots & & \ddots & \\ S_p^{-1} W_p^{-1} Q_p^T L & & 0 & 0 & & T_p \end{bmatrix} \tag{5.21}$$

$$\Psi = \begin{bmatrix} 0 & Q_0 & Q_1 & \cdots & Q_p \end{bmatrix}^T \tag{5.22}$$

Remark 34. The call $\{\Phi, W, S^{-1}, T, \Psi\} = \mathbf{ExpFac}(A)$ converts the smaller, dense problem $Ax = f$ into the larger sparse problem $(WST)(\Phi x + z) = (\Psi f)$ with $\Phi^T z = 0$,

in exactly the same process as previously outlined in (5.17). The resulting sparse triangular factors can be used to apply the inverse A^{-1} at a lower cost than the naive figure of $O(n^2)$.

Let us outline the steps for complexity analysis, given an arbitrary $n \times n$ matrix A that is $f(n)$-hierarchical into p parts. In the ensuing discussion, we denote $\mathbf{sol}_A(n)$ as the cost to apply the size-n hierarchical matrix inverse, and $\mathbf{nnz}_S(n)$, $\mathbf{nnz}_{WT}(n)$ as the number of nonzeros in each size-n factors S, W, T respectively, and $\mathbf{fac}_A(n)$ as as the cost to apply **ExpFac** to a size-n hierarchical matrix.

Since the factorization step calls upon the solution step, let us begin by estimating $\mathbf{sol}_A(n)$. Given explicit factors W, T, and S^{-1}, the cost of applying the inverse $A^{-1} = T^{-1}S^{-1}W^{-1}$ is the same order of magnitude as the number of nonzeros in the factorization, as in

$$\mathbf{sol}_A(n) \sim \mathbf{nnz}_S(n) + \mathbf{nnz}_{WT}(n).$$

Examining (5.19)-(5.21) and counting the number of nonzeros yields a recursive expression

$$\begin{aligned}\mathbf{nnz}_S(n) &\sim p\,\mathbf{nnz}_S(n/p) + f(n)^2,\\ \mathbf{nnz}_{WT}(n) &\sim p\,\mathbf{nnz}_{WT}(n/p) + nf(n),\end{aligned}$$

for each level of hierarchy in Algorithm 33. After ℓ levels of recursion, we have

$$\begin{aligned}\mathbf{nnz}_S(n) &\sim p^\ell \mathbf{nnz}_S(n/p^\ell) + \sum_{k=0}^{\ell-1} p^k f(n/p^k)^2,\\ \mathbf{nnz}_{WT}(n) &\sim p^\ell \mathbf{nnz}_{WT}(n/p^\ell) + n\sum_{k=0}^{\ell-1} f(n/p^k).\end{aligned}$$

The factors at the ℓ-th level are stored as dense matrices, with $\mathbf{nnz}_S(n/p^\ell) \sim (n/p^\ell)^2$ and similarly for W and T. Substituting yields the following expression

$$\mathbf{nnz}_S(n) \sim n^2/p^\ell + \sum_{k=0}^{\ell-1} p^k f(n/p^k)^2, \tag{5.23}$$

$$\mathbf{nnz}_{WT}(n) \sim n^2/p^\ell + n\sum_{k=0}^{\ell-1} f(n/p^k). \tag{5.24}$$

Accordingly, we see that the efficiency of the algorithm is driven in part by the depth of the hierarchical expansion—there is an optimal choice of ℓ that minimizes (5.23) and (5.24) for each given $f(\cdot)$.

Finally, let us consider the cost of calling **ExpFac**, which is divided between the formation of the Schur complement and its factorization. The former requires the explicit formation of the matrix L, the factorization and application of each $A_1^{-1}, \ldots, A_p^{-1}$ to this rectangular matrix, and the matrix-vector product of the resul-

tant with R. Combined, we have the recursive expression

$$\mathbf{fac}_A(n) \sim p\,\mathbf{fac}_A(n/p) + f(n)\,\left[n + p\,\mathbf{sol}_A(n/p) + f(n)\,n + f(n)^2\right], \tag{5.25}$$

which expands after ℓ levels of hierarchy to the following

$$\begin{aligned}\mathbf{fac}_A(n) &\sim p^\ell \mathbf{fac}_A(n/p^\ell) \\ &\quad + \sum_{k=0}^{\ell-1} p^k f(n)\left[n/p^k + p\,\mathbf{sol}_A(n/p^{k+1}) + f(n/p^k)\,n/p^k + f(n/p^k)^2\right].\end{aligned} \tag{5.26}$$

The matrices at the ℓ-th level are factored as dense matrices. Assuming $(n/p^\ell)^2$ work to form these matrices and $(n/p^\ell)^3$ work to factor them yields

$$\begin{aligned}\mathbf{fac}_A(n) &\sim n^3/p^{2\ell} \\ &\quad + \sum_{k=0}^{\ell-1} p^k f(n/p^k)\left[n/p^k + p\,\mathbf{sol}_A(n/p^{k+1}) + f(n/p^k)\,n/p^k + f(n/p^k)^2\right].\end{aligned} \tag{5.27}$$

Again, we see that the number of levels ℓ must be carefully chosen to minimize (5.27). In practice, the cost of forming these dense matrices may at times surpass the cost of the ensuing factorization, so for a given $f(n)$, the corresponding complexity analysis should begin at (5.26).

5.4.2 Implicit Matrix Scheme

At the same time, note that the enlarged "arrow" matrix in (5.17) is sparse, with a block-triangular factorization given in closed-form,

$$M = \left[\begin{array}{cc|ccc} K_0 & RP_0 & RP_1 & \cdots & RP_p \\ Q_0^T L & A_0 & 0 & \cdots & 0 \\ \hline 0 & 0 & A_1 & & \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & & & A_p \end{array}\right] \left[\begin{array}{cc|ccc} I & 0 & 0 & \cdots & 0 \\ 0 & I & 0 & \cdots & 0 \\ \hline A_1^{-1}Q_1^T L & 0 & I & & \\ \vdots & \vdots & & \ddots & \\ A_p^{-1}Q_p^T L & 0 & & & I \end{array}\right]$$

where $K_0 = K - \sum_{i=1}^p RP_i A_i^{-1} Q_i^T L$. If matrix-vector products with the side matrices L, R are available, then the triangular factorization may be applied without even forming them.

Algorithm 35 (Implicit-matrix Factorization)**.** $S^{-1} = \mathbf{ImpFac}(A)$
Input: invertible $n \times n$ matrix A that is $f(n)$-hierarchical into p partitions according to (5.4);
Output: factorized representation of A^{-1}.

1. (Subproblems) For each $i \in \{1, \ldots, p\}$, compute a factorized representation of the A_i^{-1} subblock via the recursive call $S_i^{-1} = \mathbf{ImpFac}(A_i)$.

2. (Master problem) Form the Schur complement

$$K_0 = K - \sum_{i=1}^{p} RP_i A_i^{-1} Q_i^T L, \qquad S_0 = \begin{bmatrix} K_0 & RP_0 \\ Q_0^T L & A_0 \end{bmatrix},$$

using $X = \mathbf{ImpSol}(A_i, S_i^{-1}, F)$ to implement each $X = A_i^{-1}F$. Compute inverse S_0^{-1}.

3. (Output) Form the block-diagonal matrix $S^{-1} = \mathrm{diag}(S_0^{-1}, S_1^{-1}, \ldots, S_p^{-1})$.

Algorithm 36 (Implicit-matrix Solution)**.** $x = \mathbf{ImpSol}(A, S^{-1}, f)$
Input: invertible $n \times n$ matrix A that is $f(n)$-hierarchical into p partitions according to (5.4); factorized representation $S^{-1} = \mathbf{ImpFac}(A)$; right-hand side f.
Output: the solution $x = A^{-1}f$.

1. Identify the subblocks $S^{-1} = \mathrm{diag}(S_0^{-1}, S_1^{-1}, \ldots, S_p^{-1})$ such that $S_i^{-1} = \mathbf{ImpFac}(A_i)$ holds for each $i \in \{1, \ldots, p\}$

2. Compute temporary variables

$$u_i \triangleq A_i^{-1} Q_i^T f \qquad \forall i \in \{1, \ldots, p\}, \tag{5.28}$$

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \triangleq \begin{bmatrix} K_0 & RP_0 \\ Q_0^T L & A_0 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{i=1}^{p} RP_i u \\ Q_0^T f_k \end{bmatrix}, \tag{5.29}$$

$$w_i \triangleq u_i - A_i^{-1} Q_i^T L v_1, \tag{5.30}$$

using $\mathbf{ImpSol}(A_i, S_i^{-1}, Q_i^T f)$ to implement each $A_i^{-1} Q_i^T f$ in (5.28) and (5.30), and S_0^{-1} to solve (5.29). Output the solution

$$x = P_0 v_2 + \sum_{i=1}^{p} P_i w_i.$$

Let us again outline the step for complexity analysis, given an arbitrary $n \times n$ matrix A that is $f(n)$-hierarchical into p parts. In the ensuing discussion, we denote $\mathbf{sol}_A(n)$ as the cost to call **ImpSol** with a size $n \times n$ matrix, and $\mathbf{mvp}_{LR}(n)$ the cost to apply either L or R, possibly in a matrix-implicit manner, and $\mathbf{fac}_A(n)$ as as the cost to call **ImpFac**.

The cost of solution at each level is two matrix-vector products with each $A_1^{-1}, \ldots, A_p^{-1}$, one matrix-vector product with each L and R, and a single matrix-vector product with the Schur complement inverse S_0^{-1}. Each level of hierarchy yields the following expression,

$$\mathbf{sol}_A(n) \sim 2p\mathbf{sol}_A(n/p) + \mathbf{mvp}_{LR}(n) + f(n)^2,$$

which expands after ℓ levels of hierarchy to

$$\mathbf{sol}_A(n) \sim (2p)^\ell \mathbf{sol}_A(n/p^\ell) + \sum_{k=0}^{\ell-1} (2p)^k \left[\mathbf{mvp}_{LR}(n/p^k) + f(n/p^k)^2\right].$$

Treating the ℓ-th level as dense yields the following estimation

$$\mathbf{sol}_A(n) \sim 2^\ell n^2/p^\ell + \sum_{k=0}^{\ell-1} (2p)^k \left[\mathbf{mvp}_{LR}(n/p^k) + f(n/p^k)^2\right]. \tag{5.31}$$

In particular, we see that we must have $p > 2$ for the implicit matrix algorithm to have a lower complexity than simple matrix-vector product with the dense matrix inverse.

The cost for the factorization is divided between the formation of the Schur complement and its factorization. The former requires the explicit formation of the matrix L, the factorization and application of each $A_1^{-1}, \ldots, A_p^{-1}$ to this rectangular matrix, and the matrix-vector product of the resultant with R. The latter is the factorization of an $f(n) \times f(n)$ symmetric indefinite matrix. This yields the order-of-magnitude expression

$$\mathbf{fac}_A(n) \sim p\,\mathbf{fac}_A(n/p) + f(n)\ \left[n + p\,\mathbf{sol}_A(n/p) + \mathbf{mvp}_{LR}(n) + f(n)^2\right],$$

which expands to

$$\begin{aligned}\mathbf{fac}_A(n) &\sim p^\ell \mathbf{fac}_A(n/p^\ell) \\ &\quad + \sum_{k=0}^{\ell-1} p^k f(n/p^k) \left[n/p^k + p\,\mathbf{sol}_A(n/p^{k+1}) + \mathbf{mvp}_{LR}(n/p^k) + f(n/p^k)^2\right]\end{aligned} \tag{5.32}$$

at the ℓ-th level.

Finally, the storage requirement is limited to the nonzeros of S^{-1}, which is given using the same analysis as the explicit-matrix version as

$$\mathbf{nnz}_S(n) \sim n^2/p^\ell + \sum_{k=0}^{\ell-1} p^k f(n/p^k)^2, \tag{5.33}$$

after ℓ levels of hierarchy.

5.5 A Direct Solver for Lyapunov Least Squares

We are finally ready to present our solver for the Lyapunov least squares problem. For the following discussion, we will define the matrix

$$\mathbf{H} \triangleq \sum_{i=1}^{m} (M_i \otimes I + I \otimes M_i)^T (M_i \otimes I + I \otimes M_i)$$

as the coefficient matrix for the Lyapunov least squares problem. Two versions of the algorithm are possible: an explicit-matrix version and an implicit-matrix version.

Algorithm 37 (Lyapunov Least Squares)**. Input:** Compression tolerance ϵ, size

$n \times n$ data matrices $M_1, \ldots, M_m$, each $O(1)$-hierarchical; and size-n^2 right-hand sides $r_1, \ldots, r_q$.
Output: size-n^2 solution vectors $x_1, \ldots, x_q$, each satisfying $\|\mathbf{H}x_k - r_k\| \leq \epsilon \|x_k\|$ for all $k \in \{1, \ldots, q\}$.

1. (Compression) For each $i = 1, \ldots, m$, define the truncation tolerance

$$\eta_i \triangleq \frac{\epsilon}{6 \min\{\|M_i\|, \epsilon\} m \log_2 n}.$$

   For each level of hierarchy in $M_i = P_i^T \hat{M} Q_i + L_i K_i^{-1} R_i$, use Algorithm 28 to compute the singular decomposition $L_i K_i^{-1} R_i = U_i \Sigma V^T$. Truncate all singular values smaller than η_i in order to form the approximation $\tilde{M}_i$.

2. (Construct decomposition) For each $i = 1, \ldots, m$ form the hierarchical representation of $\tilde{\mathbf{M}}_i = \tilde{M}_i \otimes I + I \otimes \tilde{M}_i$ using (5.13)-(5.14). Then, form the hierarchical representation of $\tilde{\mathbf{H}} = \sum_{i=1}^m \tilde{\mathbf{M}}_i^T \tilde{\mathbf{M}}_i$ using (5.11).

Explicit-matrix version:

3. (Factorization) Compute the factorization $\{W, S^{-1}, T\} = \mathbf{ExpFac}(\tilde{\mathbf{H}})$ using Algorithm 33.

4. (Solution) For each $j = 1, \ldots, q$, evaluate $x_j = T^{-1} S^{-1} W^{-1} r_j$.

Implicit-matrix version:

3. (Factorization) Compute the factorization $S^{-1} = \mathbf{ImpFac}(\tilde{\mathbf{H}})$ using Algorithm 35.

4. (Solution) For each $j = 1, \ldots, q$, evaluate $x_j = \mathbf{ImpSol}(\tilde{\mathbf{H}}, S^{-1}, r_j)$ using Algorithm 36.

Let us write $N = n^2$. The first two steps of Algorithm 37 are designed to construct a approximation $\tilde{\mathbf{H}}$ that is $O(m\sqrt{N})$-hierarchical into 4 parts, while satisfying the spectral approximation bound $\|\mathbf{H} - \tilde{\mathbf{H}}\| \leq \epsilon$.

Theorem 38. *The explicit method computes the factorization in $O(m^2 n^4 + m^3 n^3 \log^2 n)$ time, solves each right-hand side in $O(mn^3 + m^2 n^2 \log n)$ time, and uses $O(mn^3 + m^2 n^2 \log n)$ memory.*

Proof. Substituting $f(N) \in O(m\sqrt{N})$ and $\ell = \log_p N$ into (5.23) and (5.24) yields $\mathbf{nnz}_S(N) \in O(m^2 N \log N)$ and $\mathbf{nnz}_{WT}(N) \in O(mN^{1.5})$. The cost of applying the factorization is the same order as the number of nonzeros in the factorization, i.e. $\mathbf{sol}_A(N) \in O(mN^{1.5} + m^2 N \log N)$. Similarly substituting $f(N) \in O(m\sqrt{N})$ and $\mathbf{sol}_A(N) \in O(mN^{1.5} + m^2 N \log N)$ into (5.26) yields

$$\begin{aligned}\mathbf{fac}_A(N) &\sim p^\ell \mathbf{fac}_A(N/p^\ell) + \sum_{k=0}^{\ell-1} \left[m^2 N^2 / p^{0.5k+0.5} + m^3 N^{1.5} \log N \right], \\ &\sim N^3/p^{2\ell} + mN^2/p^\ell + m^2 N^2 + \ell m^3 N^{1.5} \log N,\end{aligned}$$

where we note that the cost of explicitly forming a size-(N/p^ℓ) block matrix is $\sim m(N/p^\ell)^2$ via the expansion

$$\begin{aligned}&\sum_{i=1}^{m}(M_i\otimes I + I\otimes M_i)^T(M_i\otimes I + I\otimes M_i),\\ =&\sum_{i=1}^{m} M_i^TM_i\otimes I + I\otimes M_i^TM_i + M_i^T\otimes M_i + M_i\otimes M_i^T,\end{aligned}$$

while the cost of explicitly factoring the same matrix is $\sim (N/p^\ell)^3$. Again, setting $\ell = \log_p N$ yields $\mathbf{fac}_A(N) \in O(m^2N^2 + m^3N^{1.5}\log^2 N)$ levels. Substituting $N = n^2$ completes the proof. □

By chosing a sufficient aggressive ϵ, it is possible for $\tilde{\mathbf{H}}$ to be $O(\sqrt{N})$-hierarchical, independent of m. Accordingly, the dependence on m in the factorization and solution steps is entirely removed.

Corollary 39. *Suppose that ϵ were chosen sufficiently large for the approximation $\tilde{\mathbf{H}}$ to be $O(\sqrt{N})$-hierarchical, independent of m. Then the explicit method computes the factorization in $O(n^4)$ time, solves each right-hand side in $O(n^3+n^2\log n)$ time, and uses $O(n^3+n^2\log n)$ memory.*

In order to analyze the complexity of the implicit method, we must note that the cost of each matrix-vector product with the low-rank basis is $\mathbf{mvp}_{LR}(N) \sim mN$.

Theorem 40. *The implicit method computes the factorization in $O(m^2n^4 + m^3n^3)$ time, solves each right-hand side in $O(mn^3)$ time, and uses $O(m^2n^2\log n)$ memory.*

Proof. Substituting $f(N) \in O(m\sqrt{N})$ and $\mathbf{mvp}_{LR} \in O(mN)$ into (5.31) yields the expression

$$\mathbf{sol}_A(N) \sim 2^\ell N^2/p^\ell + m^2N\sum_{k=0}^{\ell-1}2^k.$$

Since $p = 4$, setting $\ell = \log_2(m^{-1}N^{0.5})$ levels minimizes the exponent of this expression, yielding $\mathbf{sol}_A(N) \in O(mN^{1.5})$. Similarly expanding (5.32) yields

$$\begin{aligned}\mathbf{fac}_A(n) &\sim p^\ell\mathbf{fac}_A(N/p^\ell) + \sum_{k=0}^{\ell-1}\left[p^{-0.5}m^2N^2/p^k + m^3N^{1.5}/p^{0.5k}\right],\\ &\sim N/p^{2\ell} + mN^2/p^\ell + m^2N^2 + m^3N^{1.5}\end{aligned}$$

and when ℓ is set as above, we have $\mathbf{fac}_A(n) \in O(m^2N^2 + m^3N^{1.5})$. The number of nonzeros in S^{-1} is the same as the explicit matrix method. Finally, substituting $N = n^2$ completes the proof. □

Corollary 41. *Suppose that ϵ were chosen sufficiently large for the approximation $\tilde{\mathbf{H}}$ to be $O(\sqrt{N})$-hierarchical, independent of m. Then the explicit method computes the*

factorization in $O(\sqrt{m}n^4 + mn^3)$ *time, solves each right-hand side in* $O(\sqrt{m}n^3)$ *time, and uses* $O(n^2 \log n)$ *memory.*

Proof. Repeat the above proof with $f(N) \in O(\sqrt{N})$, $\mathbf{mvp}_{LR} \in O(mN)$, and $\ell = \log_2(m^{-0.5}N^{0.5})$. □

5.6 Computational Results

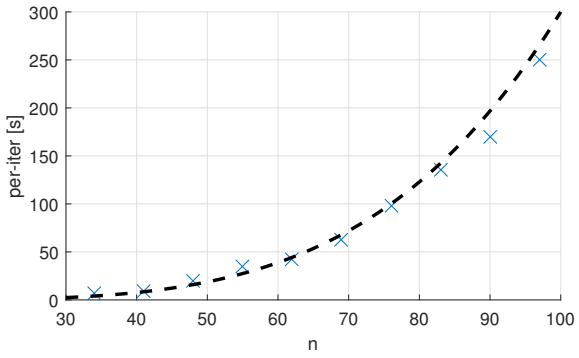
In order to benchmark the per-iteration time of realistic examples, we consider the Lyapunov inequalities on the IEEE 118-bus system as previously considered in Chapter 4. More specifically, we consider the $n \times n$ data matrices $M_1, \ldots, M_m$, with

$$n \in \{34, 41, 48, 55, 62, 69, 76, 83, 90, 97\}, \qquad m = 20.$$

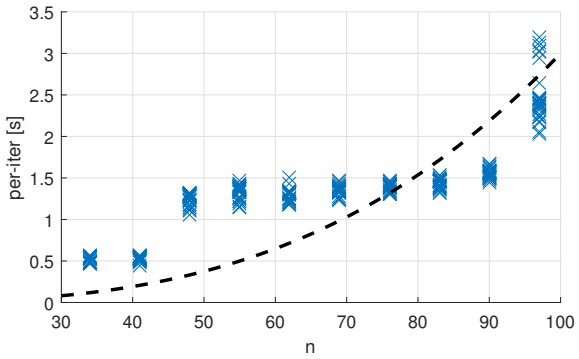
For each pair of n, m, 30 trials are performed.

Figure 5-1 shows the factorization and solution times using the implicit-matrix algorithm, with the $O(n^4)$ and $O(n^3)$ trends superimposed in the background. The results confirm the expected trends. They also exhibit a staircase-like shape, since the number of levels of hierarchy can only increase by one at a time. For $n \leq 41$, no hierarchy is used; for $n \in \{48, \ldots, 90\}$, one level of hierarchy is used, and for $n = 97$, two levels of hierarchy are used.

To show that the results extend to larger problems, we also consider a larger example on the IEEE 300-bus system, with $n = \{40, 178, 253, 375\}$ and $m = 3$. Figure 5-2 shows the factorization time using the implicit-matrix algorithm, plotted against the $O(n^4)$ trend. We see that the $n = 375$ case is factored in around 1300 seconds, or about 22 minutes.



(a)



(b)

Figure 5-1: The hierarchical solver for different values of n: (a) Factorization times, plotted against $O(n^4)$; (b) Solution time per right-hand side, plotted against $O(n^3)$.

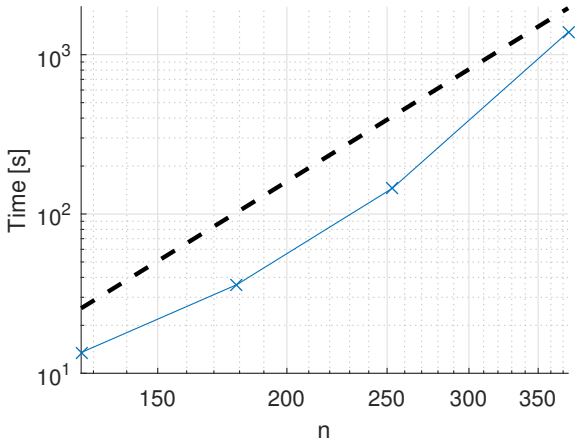


Figure 5-2: Per right-hand side times of the hierarchical solver for different values of n. The dotted curve plots $O(n^3)$.

Chapter 6

ADMM-GMRES Convergence in $O(\kappa^{1/4} \log \epsilon^{-1})$ Iterations

This chapter investigates the generalized minimum residual method (GMRES) in its ability to accelerate the convergence of the alternating direction method-of-multipliers (ADMM). We provide evidence that ADMM-GMRES can consistently converge to an ϵ-accurate solution for a κ-conditioned problem in $O(\kappa^{1/4} \log \epsilon^{-1})$ iterations, and characterize two broad classes of problems for which the enhanced convergence is guaranteed. At the same time, we construct a class of problems that forces ADMM-GMRES to converge at the same asymptotic rate as ADMM. To demonstrate the enhanced convergence rate in practice, the accelerated method is applied to the Newton direction computation for the interior-point solution of semidefinite programs in the SDPLIB test suite.

6.1 Introduction

The alternating direction method-of-multipliers (ADMM) solves problems of the form

$$\begin{array}{ll} \underset{x,z}{\text{minimize}} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array} \tag{6.1}$$

with variables $x \in \mathbb{R}^{n_x}$ and $z \in \mathbb{R}^{n_z}$ and constant data $A \in \mathbb{R}^{n_y \times n_x}$, $B \in \mathbb{R}^{n_y \times n_z}$, and $c \in \mathbb{R}^{n_y}$. Beginning with a choice of the quadratic-penalty / step-size parameter $\beta > 0$ and initial points $\{x^{(0)}, z^{(0)}, y^{(0)}\}$, the method generates iterates

$$\begin{aligned} \text{Local variable update: } x^{(k+1)} &= \arg\min_x f(x) + \frac{\beta}{2}\|Ax + Bz^{(k)} - c + \frac{1}{\beta}y^{(k)}\|^2, \\ \text{Global variable update: } z^{(k+1)} &= \arg\min_z g(x) + \frac{\beta}{2}\|Ax^{(k+1)} + Bz - c + \frac{1}{\beta}y^{(k)}\|^2, \\ \text{Multiplier update: } y^{(k+1)} &= y^{(k)} + \beta(Ax^{(k+1)} + Bz^{(k+1)} - c), \end{aligned}$$

that are guaranteed to converge under mild assumptions. The method finds a wide range of applications in statistics, machine learning, and related areas; cf. [72] for an extensive review.

Existing use of ADMM is mostly limited to applications where solutions of modest accuracy would be adequate. The reason is that, as a first-order method, it is subject to a fundamental trade-off between convergence speed and the smoothness of the underlying problem. A classic complexity result due to Nesterov [109, Thm. 2.1.13] asserts that, given constants $0 < m \leq \ell < \infty$, no first-order method can minimize every convex objective, with gradient Lipschitz constant ℓ and strong convexity parameter m, to ϵ-accuracy with an iteration bound better than

$$O(\sqrt{\kappa} \log \epsilon^{-1}) \text{ iterations}, \tag{6.2}$$

where the condition number is defined $\kappa = \ell/m$. ADMM is known to attain (6.2) with the right choice of β and under various regularizing assumptions [110–112], but even in these cases, convergence is usually not fast enough to be competitive for high-accuracy applications.

This chapter is motivated by a surprising observation. When ADMM is accelerated using the generalized minimum residual method (GMRES), ϵ-accurate solutions are *consistently* produced in just

$$O(\kappa^{\frac{1}{4}} \log \epsilon^{-1}) \text{ iterations}. \tag{6.3}$$

ADMM can take thousands of iterations to converge on examples where ADMM-GMRES converges in just tens of iterations. Our main results characterize two broad classes of problems, i.e. choices of objective functions and constraint matrices in (6.1), that are guaranteed to enjoy this enhanced convergence, both in theory and in practice. At the same time, we show in Section 6.6 that one can construct problems for which ADMM-GMRES will converge no faster than the estimate in (6.2), and so Nesterov's complexity bound is not violated. To demonstrate the effectiveness of ADMM-GMRES in applications, we use it to solve the Newton direction subproblems associated with the interior-point solution of large-scale semidefinite programs in Section 6.8.

6.1.1 ADMM for quadratics and a surprising observation

The general convergence properties of ADMM are most commonly analyzed using the theory of maximum monotone operators (cf. [72, Sec. 3.5] for a historical review). One may establish that ADMM converges to the solution from every initial point [72,113], with error that scales $O(1/k)$ or $O(1/k^2)$ at the k-th iteration [79,114], and error that scales $O(e^{-k})$ at the k-th iteration under strong convexity assumptions [115–118].

The local convergence properties are best understood by modeling the objectives as quadratics, and applying classic techniques from spectral analysis. In particular, existing parameter selection and convergence rate results have mostly been derived within this context, including the variations of ADMM that attains the bound in

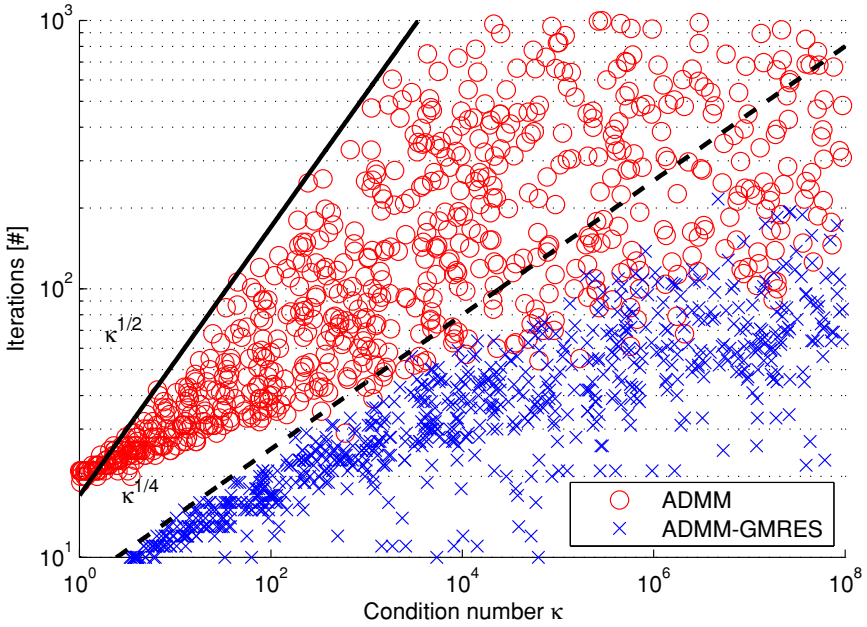


Figure 6-1: Given the same 1000 randomly-generated problems and using the same parameter choice $\beta = \sqrt{m\ell}$, ADMM (circles) converges in $O(\sqrt{\kappa})$ iterations while GMRES-accelerated ADMM (crosses) converges in $O(\kappa^{\frac{1}{4}})$ iterations. The problems have random dimensions $1 \leq n_x \leq 1000$, $1 \leq n_y \leq n_x$, $1 \leq n_z \leq n_y$. Primal-dual residual tolerance is $\epsilon = 10^{-6}$.

(6.2) [111, 119, 120].

In this chapter, we will restrict our attention to the quadratic-linear objectives,

$$f(x) = \frac{1}{2} xDx + p^T x, \qquad g(z) = q^T z, \tag{6.4}$$

alongside the following strong convexity assumption, which guarantees that the error will scale $O(e^{-k})$ at the k-th iteration [115].

Assumption 42 (Strong convexity)**.** The matrix D is symmetric positive definite, the matrix B has full column-rank, i.e. $B^T B$ is invertible, and the matrix A has full row-rank, i.e. AA^T is invertible.

Defining the associated strong convexity parameter m and the gradient Lipschitz constant ℓ respectively

$$m = \lambda_{\min}(\tilde{D}), \qquad \ell = \lambda_{\max}(\tilde{D}), \qquad \tilde{D} \triangleq (AD^{-1}A^T)^{-1}, \tag{6.5}$$

the complexity lower-bound in (6.2) is attained for every problem with the parameter choice $\beta = \sqrt{m\ell}$; the leading constant in the estimate can be further improved by introducing over-relaxation [110–112].

In the context of quadratic objectives, GMRES can be applied to accelerate ADMM in a largely plug-and-play manner, to yield a consistent and significant speed-

up over regular ADMM. Figure 6-1 makes this comparison for 1000 problems in which the A, B, D matrices are randomly generated by selecting random orthonormal bases and random singular values from a log-normal distribution.

6.1.2 Main results

For the objectives in (6.4) alongside Assumption 42, the unique solution is specified through the Karush–Kuhn–Tucker (KKT) conditions,

$$\begin{bmatrix} D & 0 & A^T \\ 0 & 0 & B^T \\ A & B & 0 \end{bmatrix} \begin{bmatrix} x^\star \\ y^\star \\ z^\star \end{bmatrix} = \begin{bmatrix} -p \\ -q \\ c \end{bmatrix} \qquad \Leftrightarrow \qquad Mu^\star = r. \tag{6.6}$$

Accordingly, ADMM reduces to linear fixed-point iterations, and GMRES convergence analysis reduces to a polynomial approximation problem over the eigenvalues of the corresponding iteration matrix. We will refer to the Euclidean norm of the KKT residual in dealing with notions of convergence.

Definition 43 (Residual convergence)**.** Given the initial and final iterates $u^{(0)} = [x^{(0)}; z^{(0)}; y^{(0)}]$ and $u^{(k)} = [x^{(k)}, z^{(k)}, y^{(k)}]$, we say ϵ residual convergence is achieved in k iterations if $\|Mu^{(k)} - r\| \leq \epsilon \|Mu^{(k)} - r\|$, where M and r are the KKT matrix and vector in (6.6).

We show in Section 6.4 that the additional square-root factor arises from a Chebyshev polynomial approximation applied to the purely-real eigenvalues of the ADMM iteration matrix. This is precisely the same mechanism that gives conjugate gradients the same square-root factor speed-up over gradient descent [121, Ch. 2-3]. However, our iteration matrix is non-normal, so our statement requires the normality qualifier in Assumption 58. The assumption is standard within this context and not particularly strong in practice; cf. Remark 59.

Theorem 44 (Dimension-based estimate)**.** *For any $A \in \mathbb{R}^{n_y \times n_x}$, $B \in \mathbb{R}^{n_y \times n_z}$, $c \in \mathbb{R}^{n_y}$, $p \in \mathbb{R}^{n_x}$, $q \in \mathbb{R}^{n_z}$, and $D \in \mathbb{R}^{n_x \times n_x}$ satisfying Assumption 42, define $\tilde{D} = (AD^{-1}A^T)^{-1}$, $m = \lambda_{\min}(\tilde{D})$, $\ell = \lambda_{\max}(\tilde{D})$, and*

$$k_0 = 2\max\{n_z, n_y - n_z\}, \qquad k_{\max} = n_y.$$

Then with the fixed choice of $\beta = \sqrt{m\ell}$, GMRES-accelerated ADMM solves (6.1) with f, g defined in (6.5) to ϵ residual convergence in

$$2 + \min\left\{k_{\max}, \quad k_0 + \left\lceil k_0 \kappa^{\frac{1}{4}} \log\kappa + \kappa^{\frac{1}{4}} \log(c_1 \kappa_P \kappa_X \epsilon^{-1}) \right\rceil\right\} \text{ iterations,}$$

where the condition number is $\kappa = \ell/m$, the matrix normality term κ_X is defined in Assumption 58, and the scalars c_1, κ_P are defined in Lemmas 55 & 56.

Proof. The proof is provided in Section 6.4. □

Remark 45. The scalars c_1 and κ_P are relatively benign bookingkeeping terms, with values that grow no faster than polynomial with respect to the conditioning of the data matrices A, B, D and the choice of the parameter β. Even very large values have relatively little theoretical impact: suppose $c_1\kappa_P = 10^{12}$ at the limits of double precision; then our iteration bound for an $\epsilon = 10^{-6}$ accurate solution is only increased by a multiplicative factor of 3.

Theorem 44 guarantees the enhanced convergence rate when $k_0 \ll k_{\max}$, i.e. whenever the matrix B is very thin or almost square, subject to B^TB being nonsingular. Such problems arise, e.g. during the active-set solution of quadratic programs, which is the subject of the convergence analysis in [111].

But Theorem 44 fails to explain the computational results seen in Figure 6-1. In almost every one of the 1000 problems, k_0 is on the same order as $k_{\max}$, and for a few select problems, they are equal. In Section 6.5, we explain this puzzling phenomenon by observing that the eigenvalues with nonzero imaginary parts, which we name *complicating eigenvalues*, are often much better conditioned than the purely-real ones. This observation allows us to derive our second iteration estimate.

Theorem 46 (Coherency-based estimate)**.** *Let $A, B, D, c, p, q, m, \ell, \kappa$, and β be the same as in Theorem 44. Define δ_{lb} as in*

$$1 - \delta_{\text{lb}} = \frac{1}{2}\left[\lambda_{\max}\{\Pi_B(+H)\Pi_B\} + \lambda_{\max}\{\Pi_B^{\perp}(-H)\Pi_B^{\perp}\}\right], \tag{6.7}$$

where the two projectors are $\Pi_B = B(B^TB)^{-1}B^T$ and $\Pi_B^{\perp} = I - \Pi_B$, and the symmetric indefinite matrix

$$H = \frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\left[2A(\beta^{-1}D + A^TA)^{-1}A^T - I\right] \tag{6.8}$$

satisfies $\|H\| = 1$ by definition. Then GMRES-accelerated ADMM solves (6.1) with f, g defined in (6.5) to ϵ residual convergence in

$$2 + \left\lceil \frac{(c_2 + \delta_{\text{lb}}^{-1})(\kappa^{\frac{1}{4}}+1)}{2c_2 + (\delta_{\text{lb}}\kappa^{\frac{1}{4}})^{-1}} \log(2c_1\kappa_P\kappa_X\epsilon^{-1}) \right\rceil \text{ iterations,}$$

where $c_2 = 1/[4\log(1+\sqrt{2})]$ is an absolute constant. The matrix normality term κ_X is defined in Assumption 58, and the scalars c_1, κ_P are defined in Lemmas 55 & 56.

Proof. The proof is provided in Section 6.5. □

Remark 47. Loosely speaking, Theorem 46 predicts convergence to an ϵ-accurate solution in $O(\delta_{\text{lb}}^{-1}\kappa^{\frac{1}{4}}\log\epsilon^{-1})$ iterations, or $O(\sqrt{\kappa}\log\epsilon^{-1})$ iterations if $\delta_{\text{lb}}^{-1} \notin O(\kappa^{\frac{1}{4}})$, e.g. if $\delta_{\text{lb}} = 0$.

Remark 48. The exact value of δ_{lb} is driven by the *mutual coherency* between the eigenspace of $\tilde{D}$ and the column-space of B, and by the decay of eigenvalues in

$\tilde{D} = (AD^{-1}A^T)^{-1}$. Consider substituting $\|H\| = 1$ into (6.7) and rewriting (6.8)

$$2\delta_{\text{lb}} \geq \|H\| - \min\left\{\|Q^THQ\|, \|P^THP\|\right\},$$
$$H = \frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} V[(\beta^{-1}\Lambda + I)^{-1} - (\beta\Lambda^{-1} + I)^{-1}]V^T,$$

where $V\Lambda^{-1}V^T = AD^{-1}A^T$ is the eigendecomposition for $\tilde{D}^{-1}$, and Q, P are any orthogonal matrices satisfying $QQ^T = \Pi_B$ and $PP^T = \Pi_B^{\perp}$. If V, Q and P are mutually incoherent (e.g. in the sense defined by Donoho & Huo [122]), then we should expect the gap between $\|H\|$ and $\min\{\|Q^THQ\|, \|P^THP\|\}$ to be nonzero in general. If the singular values of H also decay quickly, then this gap should be large, and δ_{lb} should be bounded away from zero.

For the the set of 1000 random problems shown in Figure 6-1, the sample mean for δ_{lb} is 0.3, and the sample minimum is 0.05. Hence, Theorem 46 sufficient to explain the enhanced convergence rate.

Finally, in Section 6.6, we show that there are problem constructions that force ADMM-GMRES to converge in $O(\sqrt{\kappa}\log\epsilon^{-1})$ iterations. In particular, we show that in the worst-case, the optimal polynomial that bounds the convergence of GMRES is precisely the polynomial associated with over-relaxed ADMM. In other words, ADMM-GMRES converges at the same rate as over-relaxed ADMM in the worst-case.

6.1.3 Application Example: Interior-point Newton direction for SDPs

The interior-point Newton direction problem for semidefinite programs (SDP) is a challenging environment where highly-accurate solutions are desired for dense, large-scale, and extremely ill-conditioned quadratic problems [89–91]. As progress is made and the duality gap is reduced, the system of equations becomes increasingly ill-conditioned. The condition numbers routinely grow as large as $\kappa \approx 10^8$, and the equations must be solved to a sufficiently high level of accuracy to guarantee quadratic converge in the underlying Newton's method.

The Newton direction problem provides a realistic scenario to benchmark the performance of GMRES-accelerated ADMM. In Section 6.8, we compare the performance of ADMM and GMRES-accelerated ADMM in their ability to recompute the Newton directions as generated by SeDuMi [123], a popular, open-source, MATLAB-based linear conic programming solver. More specifically, problems selected from the SDPLIB test suite [124] are pre-solved using SeDuMi, the Newton step problems at each interior-point step are exported, and ADMM and ADMM-GMRES are used to recompute the solution.

Our results show that ADMM-GMRES converges in $O(\kappa^{\frac{1}{4}}\log\epsilon^{-1})$ iterations over each of 1038 Newton problem considered. Many of these problems fit within the two characterizations described in this chapter, and the enhanced convergence rate is explained by Theorems 44 & 46. For many others, the enhanced convergence rate

is observed even when neither theorems are applicable. The result suggests further improvements to be made to the characterizations presented in this chapter.

6.1.4 Future work

The theoretical and empirical results in this chapter hint that ϵ-convergence to a κ-conditioned problem in $O(\kappa^{\frac{1}{4}} \log \epsilon^{-1})$ iterations may be the *average-case* behavior for ADMM-GMRES, at least in the context of the quadratic objectives of the forms in (6.4) and the associated regularity assumptions. An important next step is to make this observation more precise. One possible way to do this is to introduce a statistical framework, in order to study the probabilistic distribution of δ_{lb} in Theorem 46. As we had noted in Remark 48, the quantity is closely associated with the idea of mutual coherency, so it may be possible to adopt existing results from compressed sensing and related fields for this analysis.

6.2 Preliminaries

6.2.1 Definitions & Notation

Our notations are standard: upper-case Latin letters for matrices, and lower-case Latin and Greek letters for scalars and vectors. The set of real numbers is denoted $\mathbb{R}$, and the set of complex numbers is denoted $\mathbb{C}$. A complex number with zero imaginary component is said to be purely-real.

Given a matrix M, we use $\lambda_i(M)$ to refer to its i-th eigenvalue, and $\Lambda\{M\}$ to denote its set of eigenvalues, including multiplicities. If M is singular, then the notation $\Lambda_{nz}\{M\} \subseteq \Lambda\{M\}$ is used to refer to its nonzero eigenvalues. The spectral radius is the supremum of the eigenvalue moduli, and is denoted $\rho(M)$. If the eigenvalues are purely-real, then $\lambda_{\max}(M)$ refers to its most positive eigenvalue, and $\lambda_{\min}(M)$ its most negative eigenvalue. We will often refer to an eigenvalue with nonzero imaginary parts as a "complicating eigenvalue", for reasons made clear in Section 6.4.

Let $\|\cdot\|$ denote the l_2 vector norm, as well as the associated induced norm, also known as the spectral norm. We use $\sigma_i(M)$ to refer to the i-th largest singular value.

Finally, in describing the number of iterations to solve a κ-conditioned problem to ϵ-accuracy, we will often refer to an estimate of the form $O(\sqrt{\kappa} \log \epsilon^{-1})$ as simply $O(\sqrt{\kappa})$, with the implicit understanding that its relationship with ϵ is logarithmic.

6.2.2 ADMM as linear fixed-point iterations

Since the KKT conditions are linear, the ADMM update equations are also linear, and can be written in the form

$$u^{(k+1)} = G_{\text{AD}}(\beta) u^{(k)} + b(\beta), \tag{6.9}$$

upon the vector of local, global, and multiplier variables, $u^{(k)} = [x^{(k)}; z^{(k)}; y^{(k)}]$. Fixing the value of β further reduces (6.9) to linear fixed-point iterations, whose convergence

properties are entirely determined by the spectral properties of the iteration matrix, $G_{\mathrm{AD}}(\beta)$.

Following this framework, a number of previous authors have shown that, with a well-chosen value of β, ADMM is able to converge in $O(\sqrt{\kappa})$ iterations [111, 112]. In this chapter, we state an explicit iteration bound, and provide a proof for completeness.

Proposition 49. *For any $A \in \mathbb{R}^{n_y \times n_x}$, $B \in \mathbb{R}^{n_y \times n_z}$, $c \in \mathbb{R}^{n_y}$, $p \in \mathbb{R}^{n_x}$, $q \in \mathbb{R}^{n_z}$, and $D \in \mathbb{R}^{n_x \times n_x}$ satisfying Assumption 42, define m, ℓ according to (6.5). Then ADMM with fixed parameter $\beta = \sqrt{m\ell}$ solves (6.1) with f, g defined in (6.4) to ϵ residual convergence in*

$$2 + \left\lceil (\sqrt{\kappa} + 1) \log(c_1 \kappa_M \epsilon^{-1}) \right\rceil \text{ iterations,}$$

where the condition number is $\kappa = \ell/m$, the scalar c_1 is defined in Lemma 55, and $\kappa_M = \|M\|\|M^{-1}\|$ with M defined in (6.6).

Proof. The proof is provided in Appendix 6.9. □

Furthermore, they show that over-relaxed ADMM can often improve convergence rates. Over-relaxation makes the substitution

$$Ax^{(k+1)} \leftarrow \omega Ax^{(k+1)} + (1 - \omega)(Bz^{(k)} - c),$$

in the global variable and multiplier updates (steps 2 & 3), and setting the relaxation parameter $\omega \in (0, 2]$ to the value of ~ 1.5 [72]. If we write $G_{\mathrm{AD}}(\beta, \omega)$ as the iteration matrix associated with over-relaxed ADMM, with parameters β and ω, then the nonzero eigenvalues of $G_{\mathrm{AD}}(\beta, \omega)$ are related to those of $G_{\mathrm{AD}}(\beta)$ via the relation (cf. Corollary 77)[1]

$$\Lambda_{nz}\{G_{\mathrm{AD}}(\beta, \omega)\} = \omega\, \Lambda_{nz}\{G_{\mathrm{AD}}(\beta)\} + (1 - \omega).$$

Therefore, it is reasonable to expect that further reductions in the spectral radius and the spectral norm may be achieved by a well-chosen, fixed value of ω.

Proposition 50. *Let A, B, D, c, p, q, m, ℓ, κ and β be the same as in Proposition 49. Then over-relaxed ADMM with fixed β and fixed over-relaxation parameter $\omega = 2$ solves (6.1) to ϵ residual convergence in*

$$2 + \left\lceil \frac{1}{2} (\sqrt{\kappa} + 1) \log(c_1 \kappa \epsilon^{-1}) \right\rceil \text{ iterations,}$$

where the scalar c_1 is defined in Lemma 55, and $\kappa_M = \|M\|\|M^{-1}\|$ with M defined in (6.6).

Proof. The proof is provided in Appendix 6.9. □

[1] Ghadimi *et al.* [111] also proved a version of this statement.

6.2.3 Sequence acceleration via GMRES

It is natural to ask: what further speed-ups are achievable by varying the over-relaxation parameter ω between iterations, and possibly also allowing it to become complex? For example, given the iterations, $u^{(k+1)} = Gu^{(k)} + b$, one may consider cycling through a sequence of relaxation coefficients $\omega_0, \omega_1, \omega_2, \ldots$, in a scheme sometimes known as a "high-order" over-relaxation scheme

$$u^{(k+1)} = (1 - \omega_k)u^{(k)} + \omega_k G(u^{(k)} + b). \tag{6.10}$$

In fact, no choice of relaxation coefficients, fixed or varied, complex or real, can converge faster than GMRES, at least in the specific sense of step convergence.

Definition 51 (Step convergence)**.** Given the sequence $u^{(0)}$, $u^{(1)}$, ..., $u^{(k)}$, $u^{(k+1)}$, we say ϵ step convergence is achieved at the k-th iteration if $\|u^{(k+1)} - u^{(k)}\| \leq \epsilon \|u^{(1)} - u^{(0)}\|$.

To sketch this characterization, we define the step-size at the k-th step as $\Delta u^{(k)} \triangleq (Gu^{(k)} + b) - u^{(k)}$ and rearrange (6.10) to reveal,

$$\Delta u^{(k)} = \left[\prod_{i=0}^{k-1}((1 - \omega_i)I + \omega_i G)\right] r^{(0)} = p(G)\Delta u^{(0)}. \tag{6.11}$$

The role of each relaxation coefficient ω_i is to define a zero for the matrix polynomial $p(\cdot)$, and to rescale it to satisfy $p(1) = 1$. The optimal polynomial for step convergence is the minimizer of the step-size $\|\Delta u^{(k)}\| = \|p(G)\Delta u^{(0)}\|$. This is precisely the optimization problem solved by GMRES.

Proposition 52 (Saad & Schultz [125])**.** *Given the linear fixed-point iterations, $u = Gu + b$, and the initial point $u^{(0)}$. Let $u^{(k)}$ be the iterate generated at the k-th iteration of GMRES for the fixed-point equation $u = Gu + b$. Then the following bounds hold*

$$\frac{\|\Delta u^{(k)}\|}{\|\Delta u^{(0)}\|} \leq \min_{\substack{p \in \mathbb{P}_k \\ p(1)=1}} \frac{\|p(G)\Delta u^{(0)}\|}{\|\Delta u^{(0)}\|} \leq \min_{\substack{p \in \mathbb{P}_k \\ p(1)=1}} \|p(G)\|,$$

where $\Delta u^{(k)} = (Gu^{(k)} + b) - u^{(k)}$ and $\mathbb{P}_k$ denotes the space of order-k polynomials.

In the context of ADMM, the notion of step convergence is closely related, but not identical to that of residual convergence in Definition 43. The step-size and the residual norms are connected via a condition number, defined in Lemma 56, so convergence in one notion can be used to imply convergence in the other. Alternatively, in a right-preconditioned problem, the step-size coincides with the residual norm, so the two notions of convergence become identical. A right-preconditioned ADMM is developed in Section 6.7.

6.3 ADMM as a Block Gauss-Seidel Method

Consider the augmented Lagrangian to (6.1),

$$\mathscr{L}_\beta(x,z,y) = f(x) + g(x) + y^T(Ax+Bz-c) + \frac{\beta}{2}\|Ax+Bz-c\|^2, \tag{6.12}$$

whose saddle-point is determined by the *augmented* Karush-Kuhn-Tucker (KKT) equations

$$\begin{bmatrix} D+\beta A^TA & \beta A^TB & A^T \\ \beta B^TA & \beta B^TB & B^T \\ A & B & 0 \end{bmatrix} \begin{bmatrix} x \\ z \\ y \end{bmatrix} = \begin{bmatrix} \beta A^Tc - p \\ \beta B^Tc - q \\ c \end{bmatrix}. \tag{6.13}$$

The iterates generated by ADMM have a convenient interpretation as a block Gauss-Seidel matrix-splitting of (6.13), as in

$$\begin{bmatrix} D+\beta A^TA & 0 & 0 \\ \beta B^TA & \beta B^TB & 0 \\ A & B & -\frac{1}{\beta}I \end{bmatrix} \begin{bmatrix} x \\ z \\ y \end{bmatrix} = \begin{bmatrix} 0 & -\beta A^TB & -A^T \\ 0 & 0 & -B^T \\ 0 & 0 & -\frac{1}{\beta}I \end{bmatrix} \begin{bmatrix} x \\ z \\ y \end{bmatrix} + \begin{bmatrix} \beta A^Tc - p \\ \beta B^Tc - q \\ c \end{bmatrix}.$$

In turn, the convergence of the iterates is dictated by the *ADMM iteration matrix*, $G_{\mathrm{AD}}(\beta)$, defined

$$G_{\mathrm{AD}}(\beta) = \begin{bmatrix} D+\beta A^TA & 0 & 0 \\ \beta B^TA & \beta B^TB & 0 \\ A & B & -\frac{1}{\beta}I \end{bmatrix}^{-1} \begin{bmatrix} 0 & -\beta A^TB & -A^T \\ 0 & 0 & -B^T \\ 0 & 0 & -\frac{1}{\beta}I \end{bmatrix}. \tag{6.14}$$

The three pivot blocks correspond to the three steps of the ADMM algorithm: the local variable update is performed by inverting the block $(D+\beta A^TA)$, the global variable update is performed by inverting the block βB^TB, and the gradient ascent step scales the "constraint violation" by β and accumulates it within the variable.

Alternatively, note that the augmented KKT equations (6.13) may be obtained from the unaugmented KKT equations (6.6) by a left- multiplication with a β-dependent shear transformation matrix

$$T_{\mathrm{aug}}(\beta) = \begin{bmatrix} I & 0 & \beta A^T \\ 0 & I & \beta B^T \\ 0 & 0 & I \end{bmatrix}. \tag{6.15}$$

Therefore, ADMM may be also be viewed as a preconditioner matrix-splitting of the unaugmented KKT equations (6.6), using the *ADMM preconditioner matrix*, $P_{\mathrm{AD}}(\beta)$,

$$\begin{bmatrix} D & -\beta A^TB & A^T \\ 0 & 0 & B^T \\ A & B & -\frac{1}{\beta}I \end{bmatrix} \begin{bmatrix} x \\ z \\ y \end{bmatrix} = \begin{bmatrix} 0 & -\beta A^TB & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{\beta}I \end{bmatrix} \begin{bmatrix} x \\ z \\ y \end{bmatrix} + \begin{bmatrix} -p \\ -q \\ c \end{bmatrix}$$
$$\Leftrightarrow \quad P_{\mathrm{AD}}(\beta)u = [P_{\mathrm{AD}}(\beta) - M]u + r. \tag{6.16}$$

Note that the corresponding iteration matrix for (6.16) coincides with $G_{\mathrm{AD}}(\beta)$ defined earlier in (6.14).

6.3.1 Basic spectral properties

A key feature of ADMM is that dual feasibility is satisfied at every iteration. More specifically, when the gradient ascent step-size is chosen to coincide with the scaling of the quadratic penalty in the augmented Lagrangian, β, the following condition is satisfied at every k-th iteration with $k \geq 1$,

$$B^T y^{(k)} + q = 0, \tag{6.17}$$

which is the second block-row of (6.16). Dual feasibility is an important clue that hints at an eigendecomposition. Returning to the interpretation of ADMM as linear fixed-point iterations, the condition (6.17) can only hold if $[0, 0, B^T]$ spans the left nullspace of the iteration matrix $G_{\mathrm{AD}}(\beta)$. Defining an orthogonal transform based on this insight reveals a block-Schur decomposition of $G_{\mathrm{AD}}(\beta)$.

Lemma 53. *Define the QR decomposition $B = QR$ with $Q \in \mathbb{R}^{n_y \times n_z}$ and $R \in \mathbb{R}^{n_z \times n_z}$, and define $P \in \mathbb{R}^{p \times (n_y - n_z)}$ as its orthogonal complement. Then defining the orthogonal matrix U and the scaling matrix $S(\beta)$,*

$$U = \left[\begin{array}{c|cc|c} I_{n_x} & 0 & 0 & 0 \\ 0 & I_{n_z} & 0 & 0 \\ 0 & 0 & P & Q \end{array}\right], \qquad S(\beta) = \left[\begin{array}{c|ccc} \beta I_{n_x} & 0 & 0 & 0 \\ \hline 0 & \beta R & 0 & 0 \\ 0 & 0 & I & 0 \\ \hline 0 & 0 & 0 & I_{n_z} \end{array}\right] \tag{6.18}$$

yields a block-Schur decomposition of $G_{\mathrm{AD}}(\beta)$

$$U^T G_{\mathrm{AD}}(\beta) U = S^{-1}(\beta) \left[\begin{array}{c|c|c} 0_{n_x} & G_{12}(\beta) & G_{13}(\beta) \\ \hline 0 & G_{22}(\beta) & G_{23}(\beta) \\ \hline 0 & 0 & 0_{n_z} \end{array}\right] S(\beta), \tag{6.19}$$

where the size $n_y \times n_y$ inner iteration matrix $G_{22}(\beta) = \frac{1}{2} I + \frac{1}{2} K(\beta)$ is defined in terms of the matrix

$$K(\beta) = \begin{bmatrix} Q^T \\ -P^T \end{bmatrix} \left[(\beta^{-1}\tilde{D} + I)^{-1} - (\beta \tilde{D}^{-1} + I)^{-1}\right] \begin{bmatrix} Q & P \end{bmatrix}, \tag{6.20}$$

and $\tilde{D} = (AD^{-1}A^T)^{-1}$.

Proof. The proof follows from routine computation and applications of the Woodbury identity; cf. Appendix 6.10. □

We conclude that the ADMM iteration matrix, $G_{\mathrm{AD}}(\beta)$, has $n_x + n_y$ zero eigenvalues and n_y nonzero eigenvalues, all of which exactly coinciding with the eigenvalues of the inner iteration matrix $G_{22}(\beta)$.

Corollary 54 (Disk enclosure)**.** *The nonzero eigenvalues of $G_{\mathrm{AD}}(\beta)$ are enclosed within the disk on the complex plane,*

$$\mathcal{D}(\beta) = \left\{ z \in \mathbb{C} : \left| z - \frac{1}{2} \right| \leq \frac{1}{2} \max \left\{ \frac{\ell - \beta}{\ell + \beta}, \frac{\beta - m}{\beta + m} \right\} \right\}.$$

Proof. The nonzero eigenvalues of $G_{\mathrm{AD}}(\beta)$ are related to the eigenvalues of $K(\beta)$ via $\Lambda_{nz}\{G_{\mathrm{AD}}(\beta)\} = \Lambda\{G_{22}(\beta)\} = \frac{1}{2} + \frac{1}{2}\Lambda\{K(\beta)\}$. We use the spectral norm of $K(\beta)$ to enclose its eigenvalues. Obviously, $\|K(\beta)\| = \|(\beta^{-1}\tilde{D} + I)^{-1} - (\beta\tilde{D}^{-1} + I)^{-1}\|$; substituting $\ell = \lambda_{\max}(\tilde{D})$ and $m = \lambda_{\min}(\tilde{D})$ yields the desired result after some minor manipulations. □

After two ADMM iterations, the convergence behavior of ADMM becomes entirely dependent upon the inner iteration matrix.

Lemma 55. *For any β and any polynomial $p(\cdot)$, we have*

$$\|p(G_{\mathrm{AD}}(\beta))\, G_{\mathrm{AD}}^2(\beta)\| \leq c_1(\beta)\|p(G_{22}(\beta))\|,$$

where $c_1(\beta)$ is defined in terms of the matrices in Lemma 53, as in

$$c_1(\beta) = \|S(\beta)\|\|S^{-1}(\beta)\|\|G_{\mathrm{AD}}(\beta)\|^2.$$

Proof. The following is a standard identity for matrices with nilpotent blocks

$$\left[\begin{array}{c|c|c} 0_{n_x} & G_{12} & G_{13} \\ \hline 0 & G_{22} & G_{23} \\ \hline 0 & 0 & 0_{n_z} \end{array}\right]^{k+2} = \left[\begin{array}{c} G_{12} \\ \hline G_{22} \\ \hline 0 \end{array}\right] G_{22}^k \left[\begin{array}{c|c|c} 0 & G_{22} & G_{23} \end{array}\right]$$

and holds for any $k \geq 0$. Applying this identity to each monomial of $p(\cdot)$ yields the desired result. □

Setting $\beta = \sqrt{m\ell}$ minimizes the radius of the disk in Corollary 54, which in turn minimizes an estimate of the spectral norm for the inner iteration matrix, G_{22}.

6.3.2 Different notions of convergence

In the literature for ADMM, convergence is most commonly measured using the Euclidean norms of the primal and dual residuals,

$$r_{\mathrm{primal}}^{(k)} = Ax^{(k)} + Bz^{(k)} - c, \qquad r_{\mathrm{dual}}^{(k)} = \beta A^T B(x^{(k)} - x^{(k-1)}).$$

In fact, this notion of convergence is identical to that of residual convergence defined in Definition 43. To see this, consider the fixed-point equation in (6.16), and note that our definition of ϵ residual convergence is equivalent to the stopping condition $\|r_{\mathrm{primal}}^{(k)}\|^2 + \|r_{\mathrm{dual}}^{(k)}\|^2 \leq \epsilon^2$.

To convert between the notions of step convergence in Definition 51 and residual convergence in Definition 43, we use the following statement, which is self-explanatory via (6.16).

Lemma 56. *Let M, r be the KKT matrix and vector defined in (6.6), and let $u^{(k)}, u^{(k+1)}$ be two consecutive iterates generated by ADMM with parameter β. Then the step-size and the residual norms are related*

$$\kappa_P^{-1} \leq \frac{\|Mu^{(k)} - r\|}{\|u^{(k+1)} - u^{(k)}\|} \leq \kappa_P$$

where $\kappa_P = \|P_{\mathrm{AD}}(\beta)\|\|P_{\mathrm{AD}}^{-1}(\beta)\|$ and $P_{\mathrm{AD}}(\beta)$ is the ADMM preconditioner matrix as defined in (6.16).

Remark 57. Given any ϵ, define $\epsilon' = \epsilon\kappa_P^{-1}$. Lemma 56 says that ϵ' step convergence implies ϵ residual convergence, and ϵ' residual convergence implies ϵ step convergence.

6.4 GMRES-Accelerated ADMM Converges in $O(\kappa^{\frac{1}{4}})$ Iterations

In order to use Proposition 52 to derive convergence estimates for GMRES, it is common to reduce the polynomial norm-minimization problem into a polynomial approximation problem over the complex plane. For ADMM, this requires the following normality assumption, which is standard within this context.

Assumption 58 (κ_X is bounded)**.** The ADMM inner iteration matrix $G_{22}(\beta)$, defined in Lemma 53, is diagonalizable at $\beta = \sqrt{m\ell}$ with eigendecomposition, $G_{22}(\beta) = X\Lambda X^{-1}$. Furthermore, the condition number for the matrix-of-eigenvectors, $\kappa_X = \|X\|\|X^{-1}\|$, is bounded from above by an absolute constant.

Remark 59. Assumption 58 is implicitly evoked whenever a spectral radius estimate is used to study the convergence rate of ADMM in the Euclidean norm (cf. [121,126] for a detail discussion). Hence it is also present in much of the existing literature on the convergence of ADMM within the quadratic setting [111,119,120]. The considerable predictive power of these existing results suggests that the assumption is not a particularly strong one in practice.

Accordingly, convergence analysis for ADMM-GMRES is reduced to an *eigenvalue approximation problem* over the eigenvalues of the ADMM inner iteration matrix,

$$\min_{\substack{p\in\mathbb{P}_k \\ p(1)=1}} \|p(G_{\mathrm{AD}})\| \leq c_1\kappa_X \min_{\substack{p\in\mathbb{P}_{k-2} \\ p(1)=1}} \max_{\lambda\in\Lambda\{G_{22}\}} |p(\lambda)|, \tag{6.21}$$

where the scalar c_1 is defined in Lemma 55. In this section, we establish the following upper-bound to (6.21), and use it to prove our first main result.

Lemma 60. *Let $\beta = \sqrt{m\ell}$ and $k_0 = 2\min\{n_z, n_y - n_z\}$. Then for all $k \geq k_0$, we have*

$$\min_{\substack{p\in\mathbb{P}_{k-2}\\ p(1)=1}} \max_{\lambda\in\Lambda\{G_{22}\}} |p(\lambda)| \leq \kappa^{k_0/2}\left(\frac{\kappa^{\frac{1}{4}}-1}{\kappa^{\frac{1}{4}}+1}\right)^{k-k_0}. \tag{6.22}$$

Proof. After introducing the necessarily preliminaries in Section 6.4.1, the proof is provided in Section 6.4.2. □

Proof of Theorem 44. GMRES must terminate in $k_{\max} = 2 + n_y$ iterations, because the characteristic polynomial $\chi(\cdot)$ for G_{22} is at most order n_y. Applying Lemma 55 to $\chi(\cdot)$ yields

$$\|G_{\mathrm{AD}}\,\chi(G_{\mathrm{AD}})\,G_{\mathrm{AD}}\| \leq c_1\|\chi(G_{22})\| = 0, \tag{6.23}$$

and substituting (6.23) into Proposition 52 yields convergence in $k_{\max}$ iterations.

To derive the non-trivial estimate, we apply Lemma 60 to the eigenvalue approximation problem in (6.21). Converting the residual tolerance ϵ to the step-size tolerance $\epsilon' = \epsilon\kappa_P^{-1}$ via Lemma 56, taking logarithms, and applying the bound $x(1+x)^{-1} \leq \log(1+x) \leq x$ yields our desired result. □

6.4.1 The Chebyshev approximation

Upper-bounds to the eigenvalue approximation problem (6.21) can be obtained by heuristically approximating an outer enclosure over the eigenvalues. More specifically, given any outer enclosure $\mathcal{S} \subset \mathbb{C}$ satisfying $\mathcal{S} \supseteq \Lambda\{G_{22}\}$ and any heuristic order-k polynomial $q(\cdot)$ satisfying $q(1) = 1$, an upper-bound on (6.21) is established via the inequality chain

$$\min_{\substack{p\in\mathbb{P}_k\\ p(1)=1}} \max_{\lambda\in\Lambda\{G_{22}\}} |p(\lambda)| \leq \min_{\substack{p\in\mathbb{P}_k\\ p(1)=1}} \max_{z\in\mathcal{S}} |p(z)| \leq \max_{z\in\mathcal{S}} |q(z)|. \tag{6.24}$$

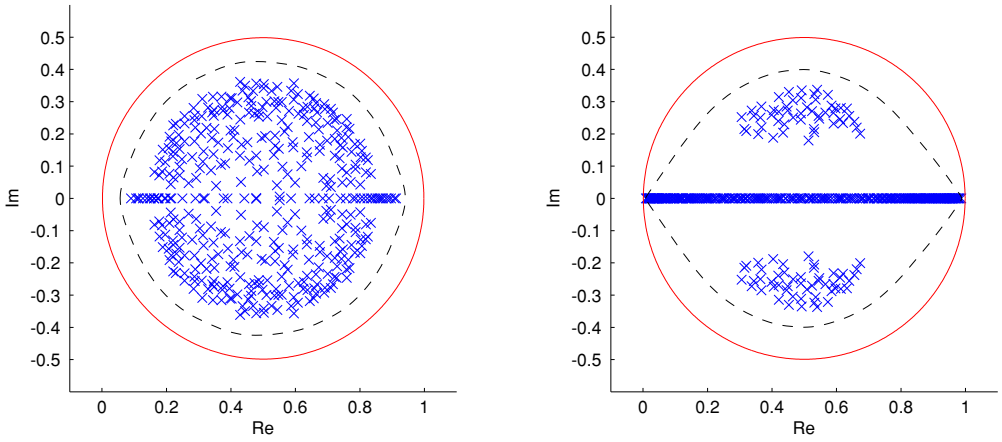
Clearly, any valid choice of outer enclosure $\mathcal{S}$ and heuristic polynomial q will yield upper-bounds. However, better bounds are generated by tighter enclosures and optimal approximations.

Let us begin by considering the disk-shaped enclosure $\mathcal{D} \supset \Lambda\{G_{22}\}$ in Corollary 54. The polynomial approximation problem over a disk has a well-known closed-form solution [93].

Theorem 61 (Circle approximation)**.** *Let $\mathcal{D}$ denote the disk on the complex plane, centered at $c \in \mathbb{C}$ with radius $a \in \mathbb{R}$, and let $\gamma \in \mathbb{C}\backslash\mathcal{D}$. Then the polynomial approximation problem has closed-form solution*

$$\min_{\substack{p\in\mathbb{P}_k\\ p(\gamma)=1}} \max_{z\in\mathcal{D}} |p(z)| = \left(\frac{a}{|\gamma-c|}\right)^k = \left(\frac{\kappa_D-1}{\kappa_D+1}\right)^k,$$

where $\kappa_D = (|\gamma - c| + a)/(|\gamma - c| - a)$ is the condition number for the disk. The minimum is attained by the monomial $p^\star(z) = (z-c)^k/|\gamma-c|^k$.



(a) (b)

Figure 6-2: Nonzero eigenvalues (crosses), field of values (dashed), and predicted disk (solid) of ADMM with $\beta = \sqrt{m\ell}$ for two randomly generated problems: (a) $n_x = 500$, $n_z = 200$, $n_y = 400$; (b) $n_x = 500$, $n_z = 50$, $n_y = 400$.

Plugging $\beta = \sqrt{m\ell}$ into Corollary 54 yields a condition number of $\kappa_D = \sqrt{\kappa}$, and applying Theorem 61 produces the upper-bound

$$\min_{\substack{p\in\mathbb{P}_k \\ p(1)=1}} \max_{\lambda\in\Lambda\{G_{22}\}} |p(\lambda)| \leq \left(1 - \frac{2}{\sqrt{\kappa}+1}\right)^k. \tag{6.25}$$

Repeating the steps in the proof of Theorem 44 using the upper-bound (6.25) produces an iteration estimate of $O(\sqrt{\kappa})$. This is the best iteration estimate achievable using the disk enclosure $\mathcal{D}$, but the result is trivial; we have already established that ADMM (with our specified choice of β) will converge in $O(\sqrt{\kappa})$ iterations on its own, and GMRES-accelerated ADMM will always converge faster than ADMM.

A better iteration estimate requires a tighter enclosure over the eigenvalues of G_{22}. An important observation is that the eigenvalues *tend to be* purely-real, and that the number of eigenvalues with imaginary parts is correlated with the problem dimensions. To illustrate this point, two random problems with different problem dimensions are shown in Figure 6-2. The distribution of eigenvalues are typical; fixing n_x, n_y, n_z and randomly generating new matrices will recover essentially the same plot every time.

Purely-real eigenvalues spread over an interval is a particularly desirable structure in eigenvalue approximation problems, due to the existence of a closed-form optimal solution attributed to Chebyshev [93].

Theorem 62 (Interval Approximation)**.** *Let $\mathcal{I}$ denote the interval $[c-a, c+a]$ on the real line, and let $\gamma \in \mathbb{R}\backslash\mathcal{I}$. Then the polynomial approximation problem has*

closed-form solution

$$\min_{\substack{p\in\mathbb{P}_k\\ p(\gamma)=1}} \max_{z\in\mathcal{I}} |p(z)| = \frac{1}{|T_k((\gamma-c)/a)|} \leq 2\left(\frac{\sqrt{\kappa_I}-1}{\sqrt{\kappa_I}+1}\right)^k,$$

where $T_k(z)$ is the degree-k Chebyshev polynomial of the first kind, and $\kappa_I = (|\gamma-c|+a)/(|\gamma-c|-a)$ is the condition number for the interval. The minimum is attained by the Chebyshev polynomial $p^\star(z) = T_k(\frac{z-c}{a})/|T_k(\frac{\gamma-c}{a})|$.

Suppose that, in addition to Corollary 54, we have *a priori* knowledge that all of our eigenvalues were purely-real. Then there exists a real interval $\mathcal{I} \supset \Lambda\{G_{22}\}$ satisfying $\kappa_I = \kappa_D = \sqrt{\kappa}$ (e.g. the projection of $\mathcal{D}$ onto the real line), and Theorem 62 assures us that there exists an order $O(\kappa^{\frac{1}{4}}\log\epsilon^{-1})$ polynomial to reduce the eigenvalue approximation error associated with $\mathcal{I}$ to below ϵ. In turn, GMRES will converge in $O(\kappa^{\frac{1}{4}})$, which is an entire square-root factor better than ADMM on its own.

6.4.2 Annihilating the complicating eigenvalues

In practice, it is very rare for all of our eigenvalues to be purely-real, i.e. there usually exists a number of eigenvalues with nonzero imaginary parts. These eigenvalues prevent the Chebyshev approximation from being directly applicable, so we refer to them as *complicating* eigenvalues. Fortunately, the number of such eigenvalues can be explicitly bounded through the problem dimensions.

Lemma 63. *The ADMM iteration matrix $G_{\mathrm{AD}}(\beta)$ has at most $2\min(n_z, n_y - n_z)$ eigenvalues with nonzero imaginary parts, counting conjugates, for every choice of $\beta > 0$.*

Proof. Recall from Lemma 53 that $G_{\mathrm{AD}}(\beta)$ has $n_x + n_z$ zero eigenvalues and n_y nonzero eigenvalues, and that the nonzero eigenvalues are shifted-and-scaled from the eigenvalues of the matrix $K(\beta)$, defined in (6.20). Define the symmetric matrix $\tilde{K}(\beta) = (\beta^{-1}\tilde{D}+I)^{-1} - (\beta\tilde{D}^{-1}+I)^{-1}$, and note that $K(\beta)$ as the following structure

$$K(\beta) = \begin{bmatrix} Q^T\tilde{K}(\beta)Q & Q^T\tilde{K}(\beta)P \\ -P^T\tilde{K}(\beta)Q & P^T\tilde{K}(\beta)P \end{bmatrix} = \begin{bmatrix} X(\beta) & Z(\beta) \\ -Z^T(\beta) & Y(\beta) \end{bmatrix}, \tag{6.26}$$

where $X = X^T \in \mathbb{R}^{n_z\times n_z}$ and $Y = Y^T \in \mathbb{R}^{(n_y-n_z)\times(n_y-n_z)}$. A matrix of this form is known as "J-symmetric", because it satisfies the symmetry condition $JK(\beta) = K^T(\beta)J$ with the matrix $J = \mathrm{blkdiag}(I_{n_z}, -I_{(n_y-n_z)})$. An immediate consequence of this J-symmetry is that $K(\beta)$ must have at most $2\min(n_z, n_y - n_z)$ eigenvalues with nonzero imaginary parts, counting conjugates [127, Prop. 2.3]. □

Suppose that the iteration matrix has k_0 complicating eigenvalues. Then, we may expend the first k_0 zeros of our polynomial (i.e. the first k_0 iterations of GMRES) to annihilating these eigenvalues. Once the complicating eigenvalues are removed from consideration, the Chebyshev approximation theorem can be applied to the remaining

eigenvalues, which are all purely-real. Putting these pieces together with a formal iteration bound proves the desired upper-bound on the polynomial approximation problems.

Proof of Lemma 60. Let us label the k_0 complicating eigenvalues as the set $\mathcal{C} = \{\lambda \in \Lambda\{G_{22}\} : \text{Im}\lambda \neq 0\}$, and the projection of $\mathcal{D}$ in Corollary 54 onto the real line as $\mathcal{I}$. Then the union of the two regions must satisfy $\mathcal{C} \cup \mathcal{I} \supset \Lambda\{G_{22}\}$. For a heuristic solution to the accompanying approximation problem, consider the product of the order $(k - k_0)$ Chebyshev polynomial alongside k_0 zeros placed at the complicating eigenvalues, as in

$$p(z) = \frac{T_{k-k_0}((z-c)/a)}{T_{k-k_0}((1-c)/a)} \prod_{\lambda \in \mathcal{C}} \frac{(z-\lambda)}{(1-\lambda)},$$

where $c = \frac{1}{2}$ and $a = \frac{1}{2}\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$ are the center and the radius of $\mathcal{I}$. Clearly, $p(1) = 1$, and the polynomial is zero at every eigenvalue with a nonzero imaginary part, i.e. $\max_{\lambda \in \mathcal{C}} |p(z)| = 0$. The maximum modulus then occurs over the purely-real eigenvalues, which is bound

$$\max_{z \in \mathcal{I}} |p(z)| \leq \frac{1}{T_{k-k_0}\left(\frac{1-c}{a}\right)} \max_{z \in \mathcal{I}} \prod_{\lambda \in \mathcal{C}} \frac{|(z-\lambda)|}{|(1-\lambda)|} \leq 2(\kappa_D)^{\kappa_0} \left(\frac{\sqrt{\kappa_D}-1}{\sqrt{\kappa_D}+1}\right)^{k-k_0},$$

since $\max_{z,\lambda \in \mathcal{D}} \frac{|(z-\lambda)|}{|(1-\lambda)|} \leq \kappa_D$. Substituting $\kappa_D = \sqrt{\kappa}$ yields the desired result. □

6.5 A Sufficient Condition for Convergence in $O(\kappa^{\frac{1}{4}})$ Iterations

In the previous section, we showed that the additional square-root factor speed-up achieved by GMRES arises from a Chebyshev polynomial approximation of the purely-real eigenvalues of the ADMM iteration matrix. Intuitively, we would expect to see the speed-up only in cases where the vast majority of the iteration matrix eigenvalues are purely-real. Yet in our empirical results, e.g. those shown in Figure 6-1, we observed the speed-up over all problems, even in those that do not admit any purely-real eigenvalues.

Upon closer inspection, we find that the iteration matrix eigenvalues with non-zero imaginary parts, which we named *complicating eigenvalues*, are commonly bounded away from the constraint point $z = +1$; an illustration of this description is shown in Figure 6-3. Loosely speaking, the complicating eigenvalues are *better conditioned* than the purely-real eigenvalues; instead of precisely annihilating them, it may be sufficient to simply "dampen" their effect with a number of fixed-point iterations. Then, the Chebyshev approximation can be used to approximate the remaining purely-real but poorly-conditioned eigenvalues.

Exploring this alternative eigenvalue approximation strategy, we arrive at a considerably less conservative heuristic solution to the eigenvalue approximation problem that is independent of the exact number of imaginary eigenvalues.

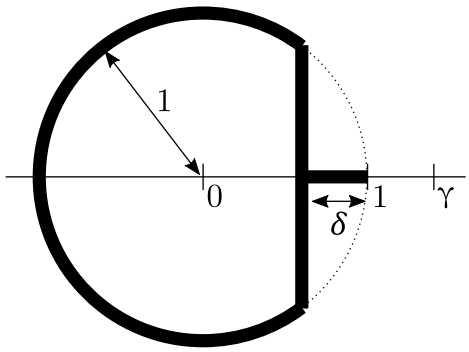


Figure 6-3: Illustration of the distribution of real and complex eigenvalues in the rescaled matrix $K/\|K\|$. The complicating eigenvalues are bounded away from the right-side of the disk by the distance δ.

Lemma 64. *Let $\beta = \sqrt{m\ell}$, and define $\hat{K} \triangleq K(\beta)/\|K(\beta)\|$ as the rescaled version of matrix defined in (6.20). Define $\delta \geq 0$ as the distance from the right-most complicating eigenvalue of $\hat{K}$ to the boundary of the unit disk, as in*

$$\delta = \min\{1 - \mathrm{Re}\lambda : \lambda \in \Lambda\{\hat{K}\}, \mathrm{Im}\{\lambda\} \neq 0\}. \tag{6.27}$$

Then the polynomial approximation problem over the eigenvalues of $\hat{K}$ is bound

$$\min_{\substack{p \in \mathbb{P}_k \\ p(1)=1}} \max_{\lambda \in \Lambda\{G_{22}\}} |p(\lambda)| \leq 2\left(1 - \frac{1}{\kappa^{1/2}}\right)^{\eta} \left(1 - \frac{2}{1+\kappa^{1/4}}\right)^{\xi}, \tag{6.28}$$

where $c_2 = 1/[4\log(1+\sqrt{2})]$ and

$$\eta = \left\lceil \frac{1}{c_2\delta + 1} k \right\rceil, \qquad \xi = \left\lfloor \frac{c_2\delta}{c_2\delta + 1} k \right\rfloor. \tag{6.29}$$

Proof. The proof is provided below in Section 6.5.1. □

The definition of δ is illustrated in Figure 6-3. We will also prove the following statement, which says that the δ_{lb} as stated in Theorem 46 is a lower-bound on δ.

Proposition 65. *Let δ, β and $\hat{K}$ be as in Lemma 64, and the orthogonal matrix $[Q, P]$ according to Lemma 53. Define the scalar δ_{lb} as*

$$1 - \delta_{\mathrm{lb}} = \frac{1}{2}\left[\lambda_{\max}(Q^T\hat{K}Q) + \lambda_{\max}(-P^T\hat{K}P)\right].$$

Then $\delta_{\mathrm{lb}} \leq \delta$.

Proof. Recall from the proof of Lemma 63 that K is J-symmetric. For matrices with this structure, Benzi & Simoncini [127] used a field-of-values type argument to show

that if $\lambda \in \Lambda\{K\}$ and $\mathrm{Im}\{\lambda\} \neq 0$, then

$$\frac{1}{2}\left[\lambda_{\min}(X) + \lambda_{\min}(Y)\right] \leq \mathrm{Re}\lambda \leq \frac{1}{2}\left[\lambda_{\max}(X) + \lambda_{\max}(Y)\right].$$

Substituting the definitions of δ, X, and Y results in the desired bound. □

Our second main result in this chapter, i.e. the iteration bound in Theorem 46, is established by solving the eigenvalue approximation problem in (6.21) using the alternative heuristic solution in Lemma 64.

Proof of Theorem 46. Repeating the same steps as in the proof of Theorem 44, but using the bound in (6.28) in Lemma 64 yields an iteration estimate in terms of δ. Since $A(\beta^{-1}D + A^TA)^{-1}A^T = (\beta^{-1}\tilde{D} + I)^{-1}$, it is easy to verify that the matrix H in Theorem 46 satisfies $Q^THQ = Q^T\hat{K}Q$ and $P^THP = P^T\hat{K}P$. The iteration estimate is monotonously decreasing with respect to δ, and substituting $\delta_{\mathrm{lb}} \leq \delta$ from Proposition 65 yields the desired iteration estimate. □

6.5.1 Damping the complicating eigenvalues

We begin by converting the eigenvalue approximation problem over G_{22} to an equivalent problem over $\hat{K}$.

Lemma 66. *Define $\hat{K}$ as in Lemma 64, and $G_{22} \equiv G_{22}(\beta)$ as in Lemma 53 with fixed $\beta = \sqrt{m\ell}$. Then*

$$\min_{\substack{p\in\mathbb{P}_k \\ p(1)=1}} \max_{z\in\Lambda\{G_{22}\}} |p(z)| = \min_{\substack{p\in\mathbb{P}_k \\ p(\gamma)=1}} \max_{z\in\Lambda\{\hat{K}\}} |p(z)|$$

where $\gamma = (\sqrt{\kappa}+1)/(\sqrt{\kappa}-1)$.

Proof. By definition, we have $\hat{K} = K/\|K\|$ and $G_{22} = \frac{1}{2}I + \frac{1}{2}K$. Hence, the statement follows from the existence of a linear bijection between $\Lambda\{G_{22}\} \cup \{1\}$ and $\Lambda\{\hat{K}\} \cup \{\gamma\}$. □

Let δ be as defined in Lemma 64, and consider the enclosure $\mathcal{S}(\delta) \cup \mathcal{I} \supset \Lambda\{\hat{K}\}$, where

$$\mathcal{S}(\delta) = \{z \in \mathbb{C} : |z| \leq 1, \mathrm{Re}\{z\} \leq \delta\}, \qquad \mathcal{I} = \{z \in \mathbb{R} : |z| \leq 1\}. \tag{6.30}$$

This enclosure is shown in Figure 6-3. The crux of our argument is a heuristic solution for the approximation problem over $\mathcal{S}(\delta) \cup \mathcal{I}$ of the form

$$p(z) = \left(\frac{z+\omega}{\gamma+\omega}\right)^{\eta} \frac{T_\xi(z)}{T_\xi(\gamma)}, \tag{6.31}$$

with polynomial orders chosen to satisfy $\eta + \xi = k$. As before, we expend ξ zeros (hence ξ iterations of GMRES) in the Chebyshev polynomial $T_\xi(z)$, in order to approximate

the purely-real eigenvalues, and to produce the desired $O(\kappa^{\frac{1}{4}})$ factor in the iteration estimate. But the innovation is in the η iterations of *over-relaxation*, $(z+\omega)^\eta/(\gamma+\omega)^\eta$, which are used to resolve the complicating eigenvalues that prevent the Chebyshev approximation from being applicable. As we will soon show, the quantity δ^{-1} serves as a "condition number" over the region $\mathcal{S}(\delta)$. Reducing relative error to ϵ over every point $\lambda \in \mathcal{S}(\delta)$ requires $\eta \in O(\delta^{-1} \log \epsilon^{-1})$ iterations of over-relaxation, independent of the exact number of complicating eigenvalues and the exact value of $\gamma \geq 1$.

Lemma 67. *Define $\mathcal{S}(\delta)$ as as in (6.30), and let $\gamma \geq 1$. Then an over-relaxation solution to the complex polynomial problem yields,*

$$\min_{\substack{p \in \mathbb{P}_k \\ p(\gamma)=1}} \max_{z \in \mathcal{S}(\delta)} |p(z)| \leq \left(1 - \frac{\delta}{2}\right)^{k/2},$$

using the polynomial $p(z) = (1+z)^k/2^k$.

Proof. Define the over-relaxation polynomial, $p(z) = (z+\omega)^k(\gamma+\omega)^{-k}$ to satisfy $p(\gamma) = 1$. For any choice of $\omega \geq 0$, the maximum is attained with the choice of $\gamma = 1$ and $z = (1-\delta) \pm j\sqrt{1-(1-\delta)^2}$, as in $\max_{z \in \mathcal{S}(\delta)} |p(z)| \leq [1 - 2\delta\omega(1+\omega)^{-2}]^{k/2}$. Picking ω to minimize the convergence factor yields $\omega = 1$, which we use as the desired estimate. □

Remark 68. The order $\lceil 4\delta^{-1} \log(\epsilon^{-1}) \rceil \in O(\delta^{-1})$ over-relaxation polynomial satisfies the conditions $\max_{z \in \mathcal{S}(\delta)} |p(z)| \leq \epsilon$ and $p(\gamma) = 1$.

Now, we return to the issue of the Chebyshev polynomial. Recall we had claimed in the previous section that the Chebyshev approximation is not (directly) compatible with eigenvalues with nonzero imaginary parts. The reason is that, with every increment in ξ, the maximum value of $|T_\xi(x)|$ over these eigenvalues is *increased* by a multiplicative factor.

Lemma 69. *The maximum modulus of the n-th order Chebyshev polynomial is bound within the disk centered at the origin with radius 1,*

$$\max_{|z| \leq 1} |T_n(z)| \leq T_n(\sqrt{2}) \leq (1+\sqrt{2})^n, \tag{6.32}$$

and the first inequality is tight for n even.

Proof. The maximum modulus for $T_n(z)$ over the ellipse with unit focal distance and principal axis $a \geq 1$ are attained at $2n$ points along its boundary the points [93, 125]

$$z_k = a\cos\left(\frac{k\pi}{n}\right) + j\sqrt{a^2-1}\sin\left(\frac{k\pi}{n}\right) \qquad k = 1, \ldots, 2n.$$

The ellipse with $a = \sqrt{2}$ is the smallest to enclose the unit disk, and if n is even, then $z_{n/2}$ also lies on its boundary. The second bound follows by definition, e.g. [125, Eqn. 6.111]. □

Examining our heuristic polynomial construction in (6.31), Lemma 69 says that every increment in the Chebyshev polynomial order ξ provides a $O(\kappa^{\frac{1}{4}})$-type global error reduction, but at the cost of *locally increasing* the error about $\mathcal{S}(\delta)$. Fortunately, this error increment is a fixed constant, so may be reverted with $O(\delta^{-1})$ iterations of over-relaxation. Alternating between unit increments of ξ to reduce the global error and $O(\delta^{-1})$ increments of η to reduce the local error completes our main argument.

Proof of Lemma 64. Define the regions $\mathcal{S}(\delta)$ and $\mathcal{I}$ as in (6.30). Suppose we can establish that

$$\min_{\substack{p\in\mathbb{P}_k \\ p(\gamma)=1}} \max_{z\in\mathcal{S}(\delta)\cup\mathcal{I}} |p(z)| \leq 2\left(1-\kappa_I^{-1}\right)^{\eta}\left[1-2(\kappa_I^{1/2}+1)^{-1}\right]^{\xi}, \tag{6.33}$$

where $\kappa_I = (\gamma+1)/(\gamma-1)$ is the condition number for the interval. Then setting $\gamma = (\sqrt{\kappa}+1)/(\sqrt{\kappa}-1)$ and substituting (6.33) into Lemma 66 proves the desired statement. Beginning with $p(z)$ defined in (6.31) with $\omega = 1$, the maximum modulus over $\mathcal{S}(\delta)\cup\mathcal{I}$ is the greater of the two quantities

$$\|p(z)\|_{\mathcal{S}(\delta)} \triangleq \max_{z\in\mathcal{S}(\delta)} |p(z)| \leq \left(\frac{2\sqrt{1-\delta/2}}{\gamma+1}\right)^{\eta} \frac{(1+\sqrt{2})^{\xi}}{T_{\xi}(\gamma)}, \tag{6.34}$$

$$\|p(z)\|_{\mathcal{I}} \triangleq \max_{z\in\mathcal{I}} |p(z)| = \left(\frac{2}{\gamma+1}\right)^{\eta} \frac{1}{T_{\xi}(\gamma)} \leq 2\left(1-\frac{1}{\kappa_I}\right)^{\eta}\left(\frac{\sqrt{\kappa_I}-1}{\sqrt{\kappa_I}+1}\right)^{\xi}, \tag{6.35}$$

where the estimate in (6.34) adopts the bound in Lemma 69. We will pick the ratio η/ξ to guarantee $\|p(z)\|_{\mathcal{I}} \geq \|p(z)\|_{\mathcal{S}(\delta)}$, so that (6.35) may serve as the global error bound in (6.33). Referring (6.34) and (6.35), this means to choosing the ratio η/ξ to satisfy

$$\left(1-\frac{\delta}{2}\right)^{\eta/(2\xi)} \leq \frac{1}{1+\sqrt{2}}. \tag{6.36}$$

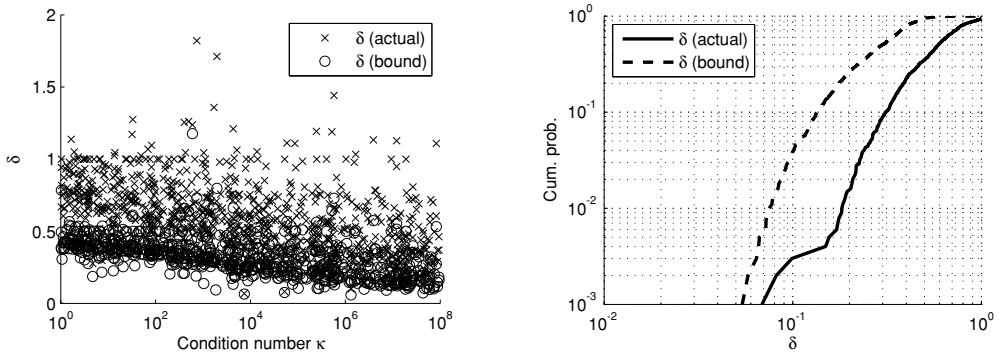
Viewing η/ξ as an "iteration estimate" to guarantee a relative error reduction of $1/(1+\sqrt{2})$ over $\mathcal{S}(\delta)$, we apply Lemma 67 and Remark 68 to obtain $\eta/\xi = 4\log(1+\sqrt{2})/\delta$. This choice yields (6.29) after rounding. □

6.5.2 Explaining the empirical results

Earlier in Figure 6-1, we presented a comparison between ADMM and GMRES-accelerated for 1000 problems. Each of these problems were constructed in the manner described in Construction 70 below. The dimension parameters n_x, n_y, n_z were uniformly sampled from $n_x \in \{1,\ldots,1000\}$, $n_y \in \{1,\ldots,n_x\}$, and $n_z \in \{1,\ldots,n_z\}$, and the log-standard-deviation uniformly swept within the range $s \in [0,2]$.

Construction 70. Begin with nonzero positive integer parameters n_x, $n_y \leq n_x$, $n_z \leq n_y$ and positive real parameter s.

1. Select the orthogonal matrices $U_A, U_B \in \mathbb{R}^{n_y\times n_y}$, $V_A, U_D \in \mathbb{R}^{n_x\times n_x}$, $V_B \in \mathbb{R}^{n_y\times n_z}$ i.i.d. uniformly from their respective orthogonal groups.



(a) (b)

Figure 6-4: Statistics for δ (cf. Theorem 46) and its lower bound (cf. Proposition 65) for 1000 randomly-generated problems: (a) scatter plot; (b) empirical CDF.

2. Select the positive scalars $\sigma_A^{(1)}, \ldots, \sigma_A^{(n_y)}$, $\sigma_B^{(1)}, \ldots, \sigma_B^{(n_z)}$, and $\sigma_D^{(1)}, \ldots, \sigma_D^{(n_x)}$ i.i.d. from the log-normal distribution $\sim \exp(0, s^2)$.

3. Output the matrices $A = U_A \mathrm{diag}(\sigma_A^{(1)}, \ldots, \sigma_A^{(n_y)}) V_A^T$, $B = U_B \mathrm{diag}(\sigma_A^{(1)}, \ldots, \sigma_A^{(n_y)}) V_B^T$, and $D = U_D \mathrm{diag}(\sigma_A^{(1)}, \ldots, \sigma_A^{(n_y)}) U_D^T$.

To check whether Theorem 46 explains the $O(\kappa^{\frac{1}{4}})$ behavior seen in problems generated via Construction 70, we compute the value of δ for each problem considered. Figure 6-4a shows the distribution of δ with respect to the condition number κ for the 1000 problems previously examined in Figure 6-1. The smallest value of δ is 0.07, with mean and median both around 0.6. The associated cumulative probability distribution is shown in Figure 6-4b. An exponentially decaying probability tail can be observed. The rapid roll-off in probability tail is a signature trait for *concentration-of-measure* type results, such as the following statement.

Conjecture 71. *Fix the values of the parameters n_x, n_y, n_z, s, and select the random matrices A, B, D via Construction 70. Then there exists an absolute constant $\alpha > 0$ such that*

$$\mathbf{Pr}\left\{\delta_{\mathrm{lb}}^{-1} \geq t\, \mathbf{E}\delta^{-1}\right\} \leq e^{-\alpha(t-1)} \qquad \forall t > 1.$$

Assuming that $\mathbf{E}\delta^{-1}$ does not become too large, the conjecture suggests that GMRES converges in $O(\kappa^{\frac{1}{4}})$ iterations almost surely, because it would be extremely unlikely for a random problem generated by Construction 70 to produce a value of δ^{-1} so large as to invalidate Theorem 46.

6.6 Worst-case Convergence in $O(\sqrt{\kappa})$ Iterations

When the number of complicating eigenvalues k_0 approaches the total number of eigenvalues $k_{\max}$ in Theorem 44, and when the ratio δ approaches zero too quickly in

Theorem 46, both of our characterizations break down, and fail to predict convergence in $O(\kappa^{\frac{1}{4}})$ iterations. Using only the disk characterization in Corollary 54, the optimal polynomial given by Theorem 61 is

$$p^\star(z) = \left(\frac{z - \frac{1}{2}}{1 - \frac{1}{2}}\right)^k = (2z-1)^k, \tag{6.37}$$

which is precisely over-relaxation with the relaxation parameter set to $\omega = 2$. Its associated iteration estimate coincides with over-relaxed ADMM in Proposition 50 up to a logarithmic factor, and supersedes the estimate in Theorem 46 with $\delta = 0$.

In fact, this result is sharp. In this section, we provide a class of problems (satisfying $k_0 = k_{\max}$ and $\delta = 0$) that forces GMRES to converge at the same rate as over-relaxed ADMM, attaining convergence in $O(\sqrt{\kappa})$ iterations. Since GMRES must converge in at most $O(\sqrt{\kappa})$ iterations to match the convergence of the usual ADMM, these problems represent the worst-case scenario for GMRES.

6.6.1 An explicit worst-case construction

Consider the following choice of A, B, D for any choice of condition number κ.

Construction 72. Begin with nonzero positive integer parameter n, and nonzero real positive parameter $\kappa \geq 1$.

1. Output A as the size-$2n$ identity matrix, and $D = \text{blkdiag}(\kappa^{-\frac{1}{2}} I_n, \kappa^{\frac{1}{2}} I_n)$.
2. Output $B \in \mathbb{R}^{2n \times n}$ in terms of the size-n diagonal matrix Θ,

$$B = \begin{bmatrix} \cos\Theta \\ \sin\Theta \end{bmatrix}, \qquad \Theta = \frac{\pi}{4n}\text{diag}\left(1, 3, 5, \ldots, 2n-1\right).$$

We will show that the optimal polynomial for this particular construction is the over-relaxation polynomial in (6.37).

Proposition 73. *Chose A, B, D according to Construction 72, and set $\beta = \sqrt{m\ell}$. Then for all $k < 2n$,*

$$\min_{\substack{p \in \mathbb{P}_k \\ p(1)=1}} \|p(G_{22})\| = \left(\frac{\kappa^{\frac{1}{2}} - 1}{\kappa^{\frac{1}{2}} + 1}\right)^k,$$

and the optimal polynomial is $p^\star(z) = (2z-1)^k$.

Despite the optimality stated in Proposition 73, we should still expect to GMRES converge in fewer iterations than over-relaxed ADMM, as shown in Figure 6-5. The discrepancy arises because GMRES optimizes convergence for a *specific* initial vector, while the polynomial in (6.37) optimizes convergence for the *worst-case* initial vector. More specifically, GMRES minimizes $\|p(G_{22})\Delta u^{(0)}\|/\|\Delta u^{(0)}\|$ (cf. Proposition 52), while over-relaxed ADMM minimizes $\|p(G_{22})\|$. The latter upper-bound is indeed sharp, and there does exist an initial vector for which the iterates generated

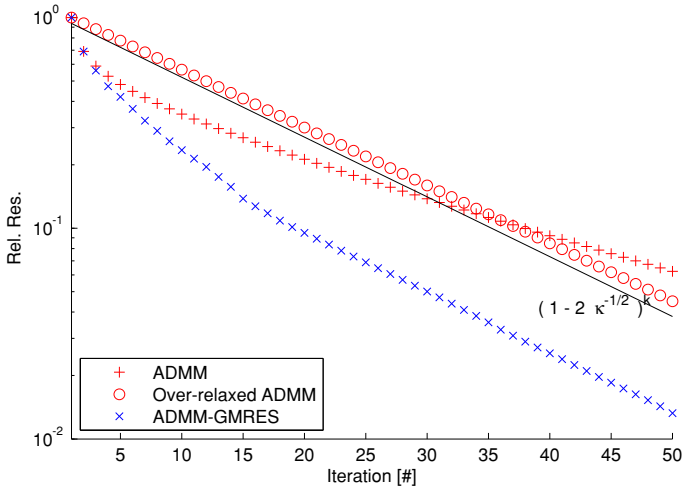


Figure 6-5: The problem construction in Section 6.6 places eigenvalues in a circle, so ADMM-GMRES convergences at the same asymptotic rate as over-relaxed ADMM with $\omega = 2$.

by GMRES and over-relaxed ADMM will coincide. However, in a "typical" instance, GMRES will tend to converge in slightly fewer iterations [121, 126, Ch. 3].

The key step to establish Proposition 73 is to show that the eigenvalues of $\hat{K}$ lie equally spaced on a circle. Under these circumstances, the optimal polynomial is known explicitly to be the over-relaxation polynomial.

Claim 74. The matrix $\hat{K} \triangleq K(\beta)/\|K(\beta\|$ is normal, and its eigenvalues are the (rotated) $2n$-th roots of unity,

$$\Lambda = w^{\frac{1}{2}} \operatorname{diag}(1, w, w^2, \ldots, w^{2n-1}), \qquad w = e^{j\pi/n}. \tag{6.38}$$

Proof. The construction yields $\tilde{D} = (AD^{-1}A^T)^{-1} = D$ and $\beta = \sqrt{m\ell} = 1$. Let us define $J = \operatorname{blkdiag}(I_n, -I_n)$ and $W = \begin{bmatrix} Q & P \end{bmatrix}$. We begin by noting that

$$(\tilde{D} + I)^{-1} - (\tilde{D}^{-1} + I)^{-1} = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \begin{bmatrix} I_n & 0 \\ 0 & -I_n \end{bmatrix} = \|K\| J.$$

Hence, the renormalized matrix is simply $\hat{K} = JW^T JW$. Since Q is defined to span the column space of B, and P to be its orthogonal complement, there must exist orthogonal matrices $\hat{Q}$, $\hat{P}$ satisfying

$$W = \begin{bmatrix} \cos\Theta & -\sin\Theta \\ \sin\Theta & \cos\Theta \end{bmatrix} \begin{bmatrix} \hat{Q} & 0 \\ 0 & \hat{P} \end{bmatrix}.$$

Using the fact that J commutes with $\mathrm{blkdiag}(\hat{Q}, \hat{P})$, we have

$$\begin{aligned}\hat{K} &= \begin{bmatrix}\hat{Q}^T & 0\\ 0 & \hat{P}^T\end{bmatrix} J \begin{bmatrix}\cos\Theta & \sin\Theta\\ -\sin\Theta & \cos\Theta\end{bmatrix} J \begin{bmatrix}\cos\Theta & -\sin\Theta\\ \sin\Theta & \cos\Theta\end{bmatrix} \begin{bmatrix}\hat{Q} & 0\\ 0 & \hat{P}\end{bmatrix}\\ &= \begin{bmatrix}\hat{Q}^T & 0\\ 0 & \hat{P}^T\end{bmatrix} \begin{bmatrix}\cos 2\Theta & -\sin 2\Theta\\ \sin 2\Theta & \cos 2\Theta\end{bmatrix} \begin{bmatrix}\hat{Q} & 0\\ 0 & \hat{P}\end{bmatrix}. \end{aligned} \tag{6.39}$$

As shown, $\hat{K}$ is unitarily similar to $\mathrm{blkdiag}\{\exp(2j\Theta), \exp(-2j\Theta)\}$. Hence it is normal, and its eigenvalues are given as Λ in (6.38). □

Proof of Proposition 73. Since $\hat{K}$ is normal, $G_{22} = \frac{1}{2}I + \frac{1}{2}\|K\|\hat{K}$ is also normal, and the following holds exactly,

$$\|p(G_{22})\| = \max_{\lambda \in \Lambda\{G_{22}\}} |p(\lambda)|.$$

As we have shown the eigenvalues of $\hat{K}$ to be the roots of unity, the eigenvalues $\Lambda\{G_{22}\}$ are $2n$ points equally spaced along the disk $\mathcal{D}$ from Corollary 54. The closed-form solution is known via the discrete analog of Theorem 61 (cf. [93]) to be $p^\star(z) = (z - \frac{1}{2})^k / |1 - \frac{1}{2}|^k = (2z-1)^k$. □

6.6.2 General trends that causes slow-down to $O(\sqrt{\kappa})$

The above is a rather artificial construction that spreads the eigenvalues of $\hat{K}$ evenly spaced around the unit circle, which causes GMRES to lose its extra square-root factor and converge in $O(\sqrt{\kappa})$ iterations. But this "slow-down" can be induced under more general circumstances. For example, consider making a change-of-basis in D, and replacing the carefully chosen B matrix with any random matrix of the same size as follows.

Construction 75. Begin with nonzero positive integer parameter n, and nonzero real positive parameter $\kappa \geq 1$.

1. Output A as the size-$2n$ identity matrix.

2. Select $U_D \in \mathbb{R}^{2n \times 2n}$ uniformly from the size-$2n$ orthogonal group, and output

$$D = U_D \begin{bmatrix}\kappa^{-\frac{1}{2}} I_n & 0\\ 0 & \kappa^{+\frac{1}{2}} I_n\end{bmatrix} U_D^T.$$

3. Select $B \in \mathbb{R}^{2n \times n}$ randomly, independent of U_D, from any desired distribution.

With overwhelming probability, GMRES will also require $O(\sqrt{\kappa})$ iterations to solve a problem from Construction 75. The cause is almost identical to that of Construction 72: the eigenvalues of $\hat{K}$ are "smeared" around the unit circle, and the optimal polynomial for the unit circle is over-relaxation. In this latter construction,

the eigenvalue distribution around the circle is not completely uniform, but this is not necessary for the $O(\kappa^{\frac{1}{4}})$ factor to be lost.

As a general trend, we expect GMRES to converge in $O(\sqrt{\kappa})$ iterations whenever the complicating eigenvalues of $\hat{K}$ spread into a ring of radius ≈ 1, since the optimal polynomial under these conditions is over-relaxation. But if this distribution is not sufficiently "spread-out", then GMRES will be able to exploit clustering, and the $O(\kappa^{\frac{1}{4}})$ factor can often be recovered. A precise characterization of "boundary problems" at the edge of the slow-down remains unknown, and is the subject of future work.

6.7 Left- and Right- Preconditioning

In the presence of finite precision, GMRES may prematurely terminate before finding a sufficiently accurate solution, in the sense of the KKT residual norm in Definition 43. The reason for this is the factor of κ_P that links step convergence with KKT residual convergence, described earlier in Lemma 56. GMRES must terminate once the step-size reaches machine precision, but this does not guarantee that the KKT residual norm has also reach machine precision, particularly once the value of κ_P grows to be large.

Recall from (6.16) in Section 6.3, that the fixed-point equation for ADMM may be written as

$$u = P^{-1}(\beta)[P(\beta) - M]u + b \qquad \Leftrightarrow \qquad P^{-1}(\beta)[Mu - r] = 0, \tag{6.40}$$

where $P(\beta)$ is the ADMM preconditioner matrix. This is simply a left-preconditioned version of the KKT equation $Mu = r$. Note that its own residual vector coincides with our usual definition of the step vector. Its residual norm then coincides with our usual definition of the step-size.

Instead, consider the right-preconditioned system,

$$\hat{u} = [P(\beta) - M]P^{-1}(\beta)\hat{u} + r \qquad \Leftrightarrow \qquad MP^{-1}(\beta)\hat{u} - r = 0, \tag{6.41}$$

whose solution satisfies $P(\beta)u = \hat{u}$. The eigenvalues are clearly the same for both (6.40) and (6.41), and their norms are related by the factor κ_P. However, note that residual vector for the right-preconditioned system *coincides* with the residual of the original KKT system. In other words, consider the iterations

$$\hat{u}^{(k+1)} = [P(\beta) - M]P^{-1}(\beta)\hat{u}^{(k)} + r, \tag{6.42}$$

and observe that its step vector, $\Delta\hat{u}^{(k)} = \hat{u}^{(k+1)} - \hat{u}^{(k)}$, is identical to the residual vector in Definition 43. So if GMRES applied to (6.41) terminates at machine precision, then we can be sure that the solution vector, $u = P^{-1}(\beta)\hat{u}$, also satisfies the KKT equation up to machine precision.

6.8 Application Example: Interior-Point Newton Direction for Semidefinite Programs

In this section, we compare the performance of ADMM and GMRES-accelerated ADMM in their ability to recompute the Newton steps as generated by SeDuMi [123] over 80 problems in the SDPLIB suite [124]. All 80 problems are semidefinite programs (SDP), which are given in linear conic programming form as the primal-dual pair

$$\begin{aligned} \text{Primal: minimize } c^T y \text{ subject to } B^T y = d, \quad y \in \mathcal{K}, \\ \text{Dual: maximize } d^T z \text{ subject to } Bz + x = c, \quad x \in \mathcal{K}^*, \end{aligned} \tag{6.43}$$

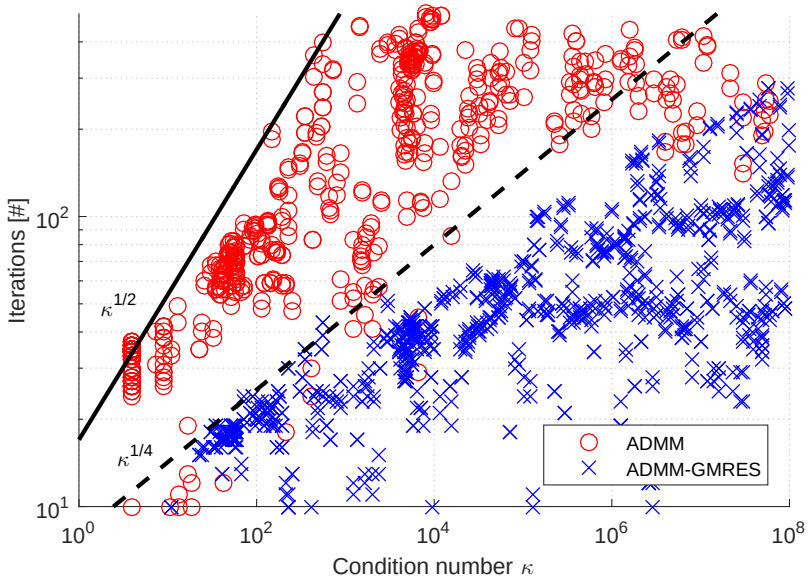
with $\mathcal{K} = \mathcal{K}^*$ set to a Cartesian product of semidefinite cones. The Newton direction at each iteration is defined as the solution to

$$\begin{bmatrix} D & 0 & I \\ 0 & 0 & B^T \\ I & B & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta z \\ \Delta y \end{bmatrix} = \begin{bmatrix} r_c \\ r_p \\ r_d \end{bmatrix}, \tag{6.44}$$

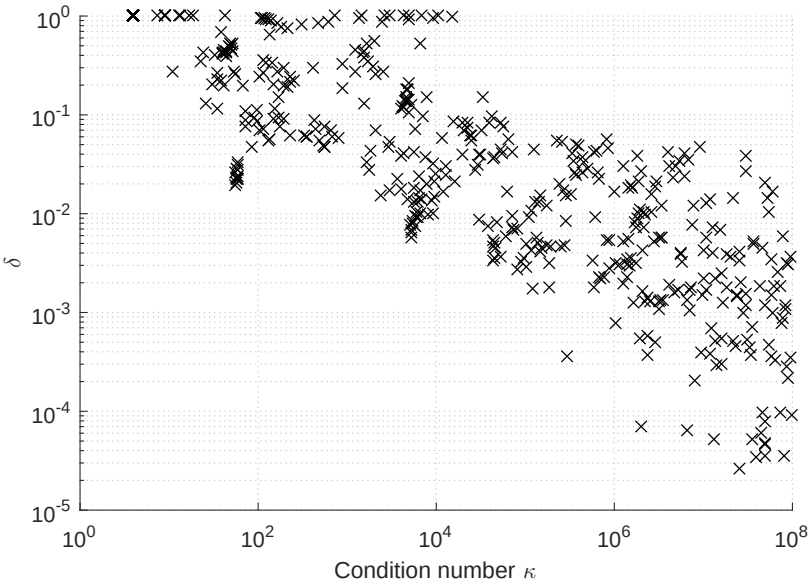
where B is the (usually) *sparse* data matrix, D is the (usually) *dense* scaling matrix specific to the interior-point method, and r_c, r_p, r_d are residual vectors. SeDuMi uses primal-dual Nesterov–Todd (NT) scaling, alongside the Mehrotra predictor-corrector method to minimize the number of interior-point steps taken. For our purposes, this means that two Newton direction subproblems are solved per interior-point step: a predictor step and a corrector step. The data matrix B is fixed for all steps, the scaling matrix D varies between steps but remains fixed between the two subproblems, and the residual vectors r_c, r_p and r_d varies between steps and also between subproblems.

The actual semidefinite programs considered are the 80 problems in the SDPLIB test suite with less than 700 constraints, i.e. with $n_z \leq 700$. For each of these problems, the predictor and corrector step subproblems are extracted, alongside the solutions computed by SeDuMi. As expected, the scaling matrix D became progressively ill-conditioned as the interior-point method progressed, with the condition number κ reaching machine precision within approximately 10 steps. In total, 1038 Newton direction problems had $\kappa \leq 10^8$, all of which we would expect GMRES to solve on the order of hundreds of iterations.

Note that the application of ADMM to semidefinite programs is not new. A number of previous authors have applied ADMM directly (i.e. without the second-order interior-point layer) to solve semidefinite programs [65,128]. The use of ADMM to solve the Newton step problem was considered in [129, 130]. Finally, the Newton system (6.44) can be viewed as a generic saddle-point problem, and in this greater context, preconditioned Krylov-subspace iterations are widespread; cf. [131] for an extensive review.

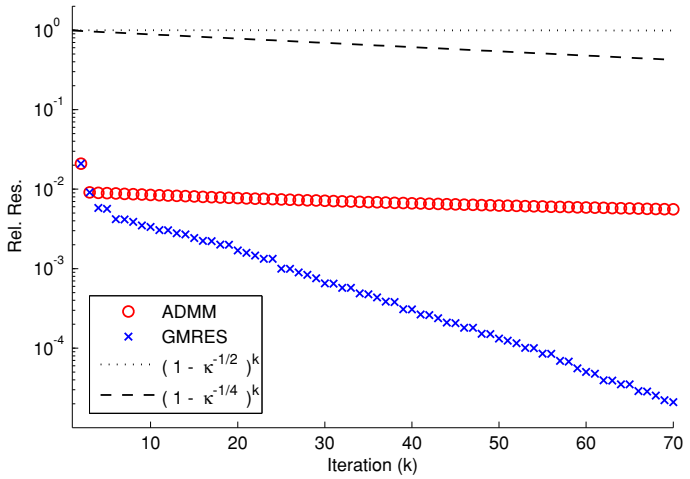


(a)

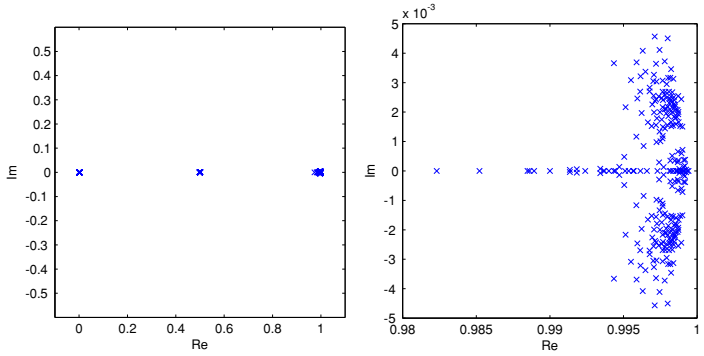


(b)

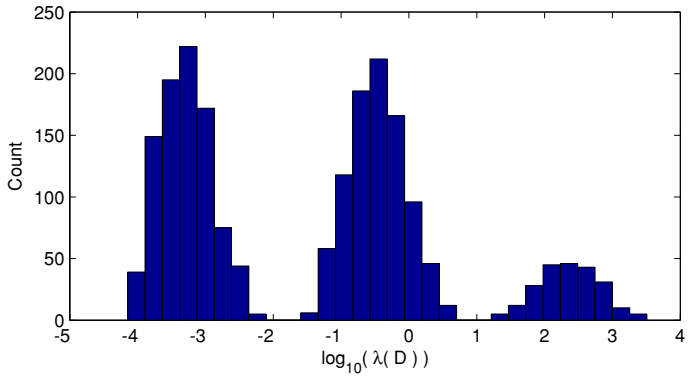
Figure 6-6: Iterations to $\epsilon = 10^{-6}$ residual convergence for the 1038 Newton direction problems described in-text: (a) ADMM vs GMRES-accelerated ADMM; (b) lower bounds on δ via Proposition 65.



(a)

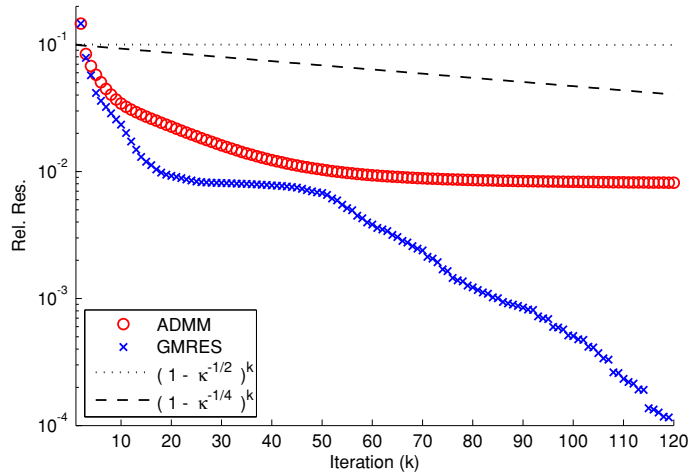


(b) (c)

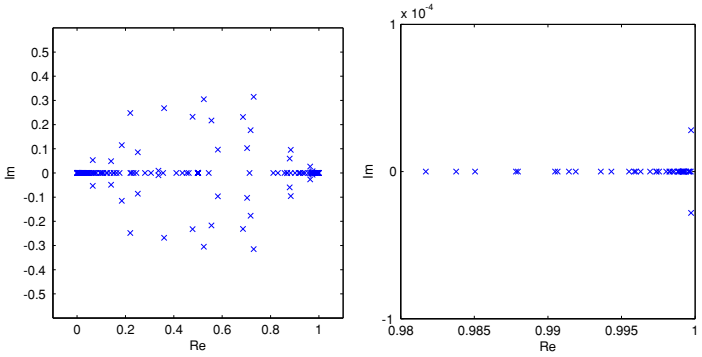


(d)

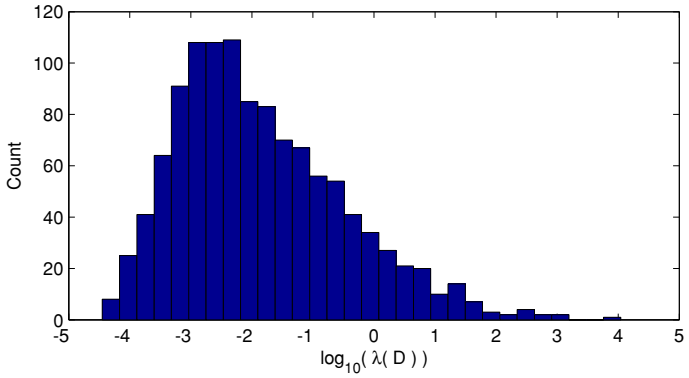
Figure 6-7: Plots for the 5th predictor step of "control3": (a) Convergence comparison; (b) Eigenvalues of G_{22}; (c) Eigenvalues of G_{22}, close-up around +1; (d) Histogram for eigenvalues of $\tilde{D}$.



(a)



(b) (c)



(d)

Figure 6-8: Plots for the 5th predictor step of "hinf14": (a) Convergence comparison; (b) Eigenvalues of G_{22}; (c) Eigenvalues of G_{22}, close-up around +1; (d) Histogram for eigenvalues of $\tilde{D}$.

6.8.1 Efficient implementation of ADMM

A direct solution of the Newton direction via dense Cholesky factorization of the Schur complement has cubic complexity of $O(n_y^{3/2}n_z + n_y n_z^2 + n_z^3)$. In order for ADMM to be competitive with the direct method, the parameter choice β and each iteration update, written

$$\begin{bmatrix}\Delta x^{(i+1)}\\ \Delta z^{(i+1)}\\ \Delta y^{(i+1)}\end{bmatrix} = \begin{bmatrix}D+\beta I & 0 & 0\\ \beta B^T & \beta B^T B & 0\\ I & B & -\frac{1}{\beta}I\end{bmatrix}^{-1}\left(\begin{bmatrix}0 & -\beta B & -I\\ 0 & 0 & -B^T\\ 0 & 0 & -\frac{1}{\beta}I\end{bmatrix}\begin{bmatrix}\Delta x^{(i)}\\ \Delta z^{(i)}\\ \Delta y^{(i)}\end{bmatrix} + \begin{bmatrix}r_c\\ r_p\\ r_d\end{bmatrix}\right),$$

must all be available at less than cubic complexity. To be more specific, we require the following three key ingredients at subcubic complexity:

1. Access to the extremal eigenvalues of $D \equiv \tilde{D}$, written $m = \lambda_{\min}(D)$, $\ell = \lambda_{\max}(D)$, in order to determine the parameter choice $\beta = \sqrt{m\ell}$.

2. Matrix-vector products with $(D+\beta I)^{-1}$ for the local variable update.

3. Matrix-vector products with $(B^T B)^{-1}$ for the global variable update.

In fact, within the context of the Newton direction for interior-point methods, the first two ingredients are always always available at subquadratic complexity of $O(n_y^{3/2})$. To be more specific, assuming that the primal-scaling, dual-scaling, or primal-dual Nesterov-Todd (NT) scaling is used for the underlying interior-point method, the matrix D will be provided in the Kronecker product form

$$D = \text{blkdiag}(W_1 \otimes W_1, W_2 \otimes W_2, \ldots, W_N \otimes W_N).$$

Each of these constituent matrices, $W_1, \ldots, W_N$, is size-$O(n_y^{1/2})$ symmetric positive definite, so diagonalizing them requires just $O(n_y^{3/2})$ operations. Let us write each $W_i = V_i \Lambda_i V_i^T$, and define the matrices

$$\begin{aligned}\Lambda &= \text{blkdiag}(\Lambda_1 \otimes \Lambda_1, \Lambda_2 \otimes \Lambda_2, \ldots, \Lambda_N \otimes \Lambda_N),\\ V_r &= \text{blkdiag}(V_1 \otimes I, V_2 \otimes I, \ldots, V_N \otimes I),\\ V_l &= \text{blkdiag}(I \otimes V_1, I \otimes V_2, \ldots, I \otimes V_N).\end{aligned}$$

Then the eigendecomposition for D is given,

$$D = V_r V_l \Lambda V_l^T V_r^T, \qquad (D+\beta I)^\alpha = V_r V_l (\Lambda + \beta I)^\alpha V_l^T V_r^T.$$

Matrix-vector products with V_r and V_l (as well as their transposes) each requires $O(n_y^{3/2})$ operations. Furthermore, the extremal eigenvalues of D are easily accessible once each Λ_i is known

$$m = \min_{i\in\{1,\ldots,N\}} \|\Lambda_i^{-1}\|^{-2}, \qquad \ell = \min_{i\in\{1,\ldots,N\}} \|\Lambda_i\|^2.$$

Access to the third ingredient, i.e. an efficient matrix-vector product with $(B^TB)^{-1}$, is much more problem specific. In semidefinite relaxations of combinatorial optimization problems, the matrix-vector product may be available at linear complexity of $O(n_z)$, because B^TB is often either diagonal or the identity matrix by construction. But for more general problems in which the matrix B is fully dense, computing its Cholesky factorization requires cubic complexity of $O(n_y n_z^2)$. We note for these problems that the B matrix does not vary between interior-point steps, so once the Cholesky factorization of B^TB is precomputed once, the factorization may be reused for all subsequent steps at quadratic complexity of $O(n_z^2)$.

6.8.2 Newton steps for problems in the SDPLIB suite

The 1038 Newton direction problems with $\kappa \leq 10^8$ are solved using standard ADMM and ADMM-GMRES with right-preconditioning, both to an accuracy of 10^{-6} relative residual for the system (6.44). The maximum number of iterations for all three methods is capped at 500. The number of iterations to convergence are shown in Figure 6-6a.

The results validate the $O(\sqrt{\kappa})$ figure expected of ADMM, and the $O(\kappa^{\frac{1}{4}})$ figure expected of GMRES. In fact, the multiplicative constants associated with each appear to be very similar to the results shown earlier in Figure 6-1. However, examining the values of δ_{lb} in Figure 6-6b, we find that Theorem 46 is unable to explain the $O(\kappa^{\frac{1}{4}})$ convergence rate in all of the problems. Two such cases are given as examples in the discussion below. This result suggests future work to improve the characterization in Theorem 46, in order to explain the enhanced convergence rate in these other problems.

6.8.3 Example 1: 5th predictor step of control3

In this first example, we consider the fifth predictor step for the SDPLIP problem "control3", with 1126 decision variables and 136 constraints, and a condition number of $\kappa = 4.505 \times 10^7$. With $n_y = 1126$ and $n_z = 136$, the value of $k_0 = 136$ does not allow Theorem 44 to predict an enhanced convergence rate. Computing δ yields a bound of 6.11×10^{-5} and an actual value of 1.19×10^{-3}, so Theorem 46 predicts convergence in around $O(\kappa^{\frac{1}{2}})$ iterations. However, the actual convergence performance is closer to $O(\kappa^{\frac{1}{4}})$. Explicitly forming and computing all 1126 eigenvalues of G_{22} reveals that all of the eigenvalues are close to being purely real. Hence, we may consider Theorem 44 to take into effect with $k_0 = 0$.

6.8.4 Example 2: 7th predictor step of hinf14

In this second example, we consider the seventh predictor step for the SDPLIP problem "hinf14", with 421 decision variables and 73 constraints, and a condition number of $\kappa = 3.114 \times 10^8$. With $n_y = 421$ and $n_z = 73$, the value of $k_0 = 73$ does not allow Theorem 44 to predict an enhanced convergence rate. Computing δ yields a bound of 1.636×10^{-4} and an actual value of 4.301×10^{-4}, so Theorem 46 also indicates

convergence in $O(\kappa^{\frac{1}{2}})$ iterations. Again, the actual convergence performance is closer to $O(\kappa^{\frac{1}{4}})$. Explicitly forming and computing all 421 eigenvalues of G_{22} shows that the conservative value of δ is a "false" estimate, because the associated complicating eigenvalue has an imaginary part that is only on the order of 10^{-5}. Eliminating that eigenvalue, e.g. with a single zero of the polynomial, the value of δ is increased to ≈ 0.1. In this latter case, we may consider Theorem 46 to take into effect after a certain number of iterations to improve the value of δ.

6.9 Convergence rates for ADMM and over-relaxed ADMM

For completeness, we present short proofs for Propositions 49 & 50, i.e. that both ADMM and over-relaxed ADMM converge in $O(\sqrt{\kappa}\log\epsilon^{-1})$ iterations. We begin by proving the case for ADMM, using the much of the existing machinery throughout this chapter. We then make some minor adjustments to this machinery in order to extend the proof to over-relaxed ADMM.

Proof of Proposition 49. Recall that ADMM generates iterates via the ADMM iteration matrix $G_{\text{AD}}(\beta)$ defined in (6.14),

$$u^{(k+1)} = G_{\text{AD}}(\beta)u^{(k)} + b, \tag{6.45}$$

where $u^{(k)} = [x^{(k)}; z^{(k)}; y^{(k)}]$, and that ϵ residual convergence is achieved if

$$\|Mu^{(k)} - r\| \leq \epsilon\|Mu^{(0)} - r\|,$$

where M and r are the KKT matrix and residual in (6.6).

Denote the fixed-point of the sequence as $u^\star$. By definition, $u^\star$ satisfies both $u^\star = G_{\text{AD}}(\beta)u^\star + b$ as well as $Mu^\star = r$. Subtracting $u^\star$ from each side of (6.45) yields the error update equation and the residual update equation

$$(u^{(k+1)} - u^\star) = G_{\text{AD}}(\beta)(u^{(k)} - u^\star) \tag{6.46}$$

$$Mu^{(k+1)} - r = [MG_{\text{AD}}(\beta)M^{-1}](Mu^{(k)} - r). \tag{6.47}$$

Inductively repeating this argument the relative residual at the k-th iteration

$$\frac{\|Mu^{(k)} - r\|}{\|Mu^{(0)} - r\|} \leq \|MG^k_{\text{AD}}(\beta)M^{-1}\| \leq \kappa_M\|G^k_{\text{AD}}(\beta)\|, \tag{6.48}$$

where $\kappa_M = \|M\|\|M^{-1}\|$. To resolve this last bound we use $\|G^k_{\text{AD}}(\beta)\| \leq c_1\|G^{k-2}_{22}(\beta)\|$ from Lemma 55, and substitute $\beta = \sqrt{m\ell}$ into Lemma 53 to produce $\|G_{22}(\beta)\|$ to yield

$$(6.48) \leq c_1\kappa_M\|G_{22}(\beta)\|^{k-2} \leq c_1\kappa_M\left(1 - \frac{1}{1+\kappa^{1/2}}\right)^{k-2}. \tag{6.49}$$

Solving for ϵ using $x(1+x)^{-1} \leq \log(1+x) \leq x$ yields the desired iteration estimate. $\square$

To proceed, we will write $G_{\mathrm{AD}}(\beta, \omega)$ as the iteration matrix associated with over-relaxed ADMM with parameter ω. To extend the same proof to over-relaxed ADMM, we will need the following Lemma, which very slightly extends Lemma 53.

Lemma 76. *Define $Q, P, R, U, S(\beta)$ as in Lemma 53. Then the matrix U produces a block-Schur decomposition of $G_{\mathrm{AD}}(\beta, \omega)$*

$$U^T G_{\mathrm{AD}}(\beta,\omega) U = S^{-1}(\beta) \left[\begin{array}{c|c|c} 0_{n_x} & G_{12}(\beta,\omega) & G_{13}(\beta,\omega) \\ \hline 0 & G_{22}(\beta,\omega) & G_{23}(\beta,\omega) \\ \hline 0 & 0 & 0_{n_z} \end{array}\right] S(\beta), \tag{6.50}$$

where the size $n_y \times n_y$ inner iteration matrix is defined in terms of $\tilde{D} = (AD^{-1}A^T)^{-1}$

$$G_{22}(\beta) = \left(1 - \frac{\omega}{2}\right) I + \frac{\omega}{2} \begin{bmatrix} Q^T \\ -P^T \end{bmatrix} \left[(\beta^{-1}\tilde{D} + I)^{-1} - (\beta \tilde{D}^{-1} + I)^{-1}\right] \begin{bmatrix} Q & P \end{bmatrix}. \tag{6.51}$$

Corollary 77. *Let $\Lambda_{nz}\{\cdot\}$ denote the nonzero eigenvalues of a matrix. Then*

$$\Lambda_{nz}\{G_{\mathrm{AD}}(\beta,\omega)\} = \omega\, \Lambda_{nz}\{G_{\mathrm{AD}}(\beta)\} + (1-\omega).$$

Proof of Proposition 50. Repeating the same argument as the proof of Proposition 49, but replacing all instances of Lemma 53 with Lemma 76, we arrive at

$$\frac{\|Mu^{(k)} - r\|}{\|Mu^{(0)} - r\|} \leq c_1 \kappa_M \|G_{22}(\beta,\omega)\|^{k-2} \leq c_1 \kappa_M \left(1 - \frac{\omega}{1+\kappa^{1/2}}\right)^{k-2}, \tag{6.52}$$

where $\kappa_M = \|M\|\|M^{-1}\|$ and $c_1 \leq \|S^{-1}(\beta) G_{\mathrm{AD}}(\beta,\omega)\| \|G_{\mathrm{AD}}(\beta,\omega) S(\beta)\|$. Solving for ϵ using $x(1+x)^{-1} \leq \log(1+x) \leq x$ yields the desired iteration estimate. $\square$

6.10 Proof of Lemmas 53 & 76

We will only consider the over-relaxed case Lemma 76, since the case of regular ADMM in Lemma 53 is an immediate corollary obtained by fixing $\omega = 1$. Substituting the definitions in Section 6.2 yields the iteration matrix

$$G_{\mathrm{AD}}(\beta,\omega) = \begin{bmatrix} D + \beta A^T A & 0 & 0 \\ \omega \beta B^T A & \beta B^T B & 0 \\ \omega A & B & -\frac{1}{\beta} I \end{bmatrix}^{-1} \begin{bmatrix} 0 & -\beta A^T B & -A^T \\ 0 & (\omega-1)\beta B^T B & -B^T \\ 0 & (\omega-1) B & -\frac{1}{\beta} I \end{bmatrix}. \tag{6.53}$$

Multiplying by U, U^T, then factoring from the right by $S(\beta)$ yields

$$U^T G_{\mathrm{AD}}(\beta) U = S^{-1}(\beta) \left[\begin{array}{c|c|c} 0 & G_{12}(\beta,\omega) & G_{13}(\beta,\omega) \\ \hline 0 & G_{22}(\beta,\omega) & G_{23}(\beta,\omega) \\ \hline 0 & 0 & 0 \end{array}\right] S(\beta), \tag{6.54}$$

where the sub-blocks are respectively

$$G_{12}(\beta,\omega) = -(\beta^{-1}D + A^TA)^{-1}A^T\begin{bmatrix}Q & P\end{bmatrix}, \tag{6.55}$$

$$G_{13}(\beta,\omega) = -(\beta^{-1}D + A^TA)^{-1}A^TQ, \tag{6.56}$$

$$G_{22}(\beta,\omega) = \begin{bmatrix}(1-\omega)I & 0\\ 0 & I\end{bmatrix} + \omega\begin{bmatrix}Q^T\\ -P^T\end{bmatrix}A(\beta^{-1}D + A^TA)^{-1}A^T\begin{bmatrix}Q & P\end{bmatrix}, \tag{6.57}$$

$$G_{23}(\beta,\omega) = \begin{bmatrix}-B^TQ\\ 0\end{bmatrix} + \omega\begin{bmatrix}Q^T\\ -P^T\end{bmatrix}A(\beta^{-1}D + A^TA)^{-1}A^TQ. \tag{6.58}$$

We will make the following claim to simplify the expressions

Claim 78. Let AA^T and D be invertible. Then

$$A(\beta^{-1}D + A^TA)^{-1}A^T = (\beta^{-1}\tilde{D} + I)^{-1}$$

where $\tilde{D} = (AD^{-1}A^T)^{-1}$.

Proof. The factorization arises from two applications of the Woodbury formula,

$$\begin{aligned}A(\beta^{-1}D + A^TA)^{-1}A^T &= \beta AD^{-1}A^T + \beta AD^{-1}A^T(\beta AD^{-1}A^T + I)^{-1}\beta AD^{-1}A^T\\ &= \beta\tilde{D}^{-1} + \beta\tilde{D}^{-1}(\beta\tilde{D}^{-1} + I)^{-1}\beta\tilde{D}^{-1}\\ &= (\beta^{-1}\tilde{D} + I)^{-1}.\end{aligned}$$

Finally, the Woodbury formula gives $I - (\beta^{-1}\tilde{D} + I)^{-1} = (\beta\tilde{D}^{-1} + I)^{-1}$. □

Applying the above then yields

$$G_{22}(\beta,\omega) = \left(1 - \frac{\omega}{2}\right)I + \frac{\omega}{2}\begin{bmatrix}Q^T\\ -P^T\end{bmatrix}\left[(\beta^{-1}\tilde{D} + I)^{-1} - (\beta\tilde{D}^{-1} + I)^{-1}\right]\begin{bmatrix}Q & P\end{bmatrix}, \tag{6.59}$$

which completes our proof of Lemmas 53 & 76.

Chapter 7

Conclusions and Future Work

This thesis began by examining robust stability analysis for power systems from an engineering perspective. In Chapter 3, we presented two realistic case studies to show how robust stability analysis may be used to provide situational awareness to the grid operator. The technique simultaneously guaranteed many uncertain scenarios to be stable all at once. In the IEEE 118-bus test case, we identified and bounded the worst-case scenario (in terms of the decay rate), and in the microgrid test case, we computed stability margins in terms of renewable penetrations.

Our computational results found robust stability analysis to be computationally intensive. Using an interior-point method, the time complexity of the technique scales $O(n^6)$, where n is the number of state variables in the system. This put realistic-sized power systems entirely out of reach.

The second part of this thesis re-examined robust stability analysis from a computational perspective. In Chapter 4, we developed first-order methods and mixed first-order / interior-point methods, and showed that they can be used to perform robust stability analysis on large-scale systems. Our key insight was to relate the bounded tree-width property of power systems to a certain hierarchical structure in its Jacobian matrices. In Chapter 5, we used this hierarchy to reduce the per-iteration cost of first-order methods to $O(n^3)$. This was the core mechanism that allowed us to extend the techniques to large-scale power systems.

7.1 Engineering Applications

The natural next step is to apply these techniques to real-life power systems. Our research code was able to compute Lyapunov functions for a size $n = 375$ problem in a few hours. With some further development, our techniques can be extended to real-life power systems, with $n \approx 1000$ state variables. It remains to be confirmed whether robust stability analysis will continue to be useful when used on these systems.

Robust analysis of any type is naturally conservative. In this thesis, we took on a largely model-agnostic approach, working with generic sets of matrices. This was motivated by the desire to encompass a wide range of models, without being restricted to a particular structure. On the other hand, much of the existing work

on direct energy methods (which uses a similar Lyapunov-based approach to robust stability analysis) are formulated for specific models in order to reduce conservatism considerably below that of generic approaches. The best trade-off between modeling flexibility and conservatism will likely come from a mixed approach.

At a minimum, this thesis showed robust stability analysis to be an invaluable extension for small-signal stability analysis on large-scale power systems. Our second case study also investigated its use for large-signal stability analysis, for a small microgrid model. At least in principle, the same strategy can be extended to large-scale systems, and used to analyze stability problems traditionally considered using simulation-based transient stability analysis.

7.2 Computational Considerations

The crucial insight of Chapter 5 is that the bounded tree-width property of power systems naturally imply a certain hierarchy in the linearized matrices. The hierarchy in this thesis can be improved in a number of ways. A two-level hierarchical structure in Chapter 5 is of the form

$$M = \begin{bmatrix} \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} M_3 & 0 \\ 0 & M_4 \end{bmatrix} \end{bmatrix} + \begin{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} U_3 \\ U_4 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} V_3 \\ V_4 \end{bmatrix} \end{bmatrix} + \begin{bmatrix} \begin{bmatrix} U_5 \\ U_6 \end{bmatrix} \\ \begin{bmatrix} U_7 \\ U_8 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} U_5 \\ U_6 \end{bmatrix} \\ \begin{bmatrix} U_7 \\ U_8 \end{bmatrix} \end{bmatrix}^T,$$

and this yields a typical $O(n \log n)$ complexity in e.g. the storage requirement. This can be improved to $O(n)$ using a telescoping formulation (see [97, 100])

$$\begin{aligned} M = & \begin{bmatrix} \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} M_3 & 0 \\ 0 & M_4 \end{bmatrix} \end{bmatrix} \\ & + \begin{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} U_3 \\ U_4 \end{bmatrix} \end{bmatrix} \left(\begin{bmatrix} M_5 & 0 \\ 0 & M_6 \end{bmatrix} + \begin{bmatrix} U_5 \\ U_6 \end{bmatrix} \begin{bmatrix} U_5 \\ U_6 \end{bmatrix}^T \right) \begin{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} V_3 \\ V_4 \end{bmatrix} \end{bmatrix}^T. \end{aligned}$$

potentially reducing the complexity of the first-order methods from $O(n^4)$ factorization and $O(n^3)$ per-iteration to $O(n^3)$ factorization and $O(n^2 \log n)$ per iteration. If this were achieved, then the solution of the Lyapunov inequality will become comparable to that of Lyapunov equations.

Finally, one of our algorithms (PCG-Schur) experienced an enhanced rate of convergence due to a low-rank property in its dual matrix variables. Future work should look to exploit such a low-rank structure more directly. For example, it is possible to forgo convexity altogether, and to reformulate the problem into a smooth nonconvex optimization upon a low-rank Riemann manifold. These techniques have the poten-

tial of reducing the complexity below $O(n^3)$. If such an approach is successful, then the speed of robust stability analysis can be dramatically improved.

Bibliography

[1] J. G. Kassakian, R. Schmalensee, G. Desgroseilliers, T. D. Heidel, K. Afridi, A. Farid, J. Grochow, W. Hogan, H. Jacoby, J. Kirtley *et al.*, "The future of the electric grid," Massachusetts Institute of Technology, Tech. Rep., 2011.

[2] P. Kundur, J. Paserba, V. Ajjarapu, G. Andersson, A. Bose, C. Canizares, N. Hatziargyriou, D. Hill, A. Stankovic, C. Taylor *et al.*, "Definition and classification of power system stability ieee/cigre joint task force on stability terms and definitions," *Power Systems, IEEE Transactions on*, vol. 19, no. 3, pp. 1387–1401, 2004.

[3] A. Silverstein, Y. Zhang, and D. Corbus, "Synchrophasor technology and renewables integration – naspi technical workshop," NASPI, Tech. Rep., 06 2012.

[4] ERCOT, "Pmu event analysis report: January 10, 2014," ERCOT, Tech. Rep., 2014.

[5] ——, "Pmu event analysis report: March 20, 2014," ERCOT, Tech. Rep., 2014.

[6] R. Walling *et al.*, "Analysis of wind generation impact on ercot ancillary services requirements," ERCOT and GE Energy, Tech. Rep., 2008.

[7] J. Boemer, K. Burges, C. Nabe, and M. Pöller, "All island tso facilitation of renewables studies," EirGrid and DIgSILENT, Tech. Rep., 2010.

[8] J. O'Sullivan, A. Rogers, D. Flynn, P. Smith, A. Mullane, and M. O'Malley, "Studying the maximum instantaneous non-synchronous generation in an island system—frequency stability challenges in ireland," *IEEE Transactions on Power Systems*, vol. 29, no. 6, pp. 2943–2951, 2014.

[9] D. N. Kosterev, C. W. Taylor, and W. A. Mittelstadt, "Model validation for the august 10, 1996 wscc system outage," *IEEE transactions on power systems*, vol. 14, no. 3, pp. 967–979, 1999.

[10] M. Patel, S. Aivaliotis, E. Ellen *et al.*, "Real-time application of synchrophasors for improving reliability," NERC, Tech. Rep., 10 2010.

[11] K. Zhou, J. C. Doyle, and K. Glover, *Robust and optimal control*. Prentice hall New Jersey, 1996, vol. 40.

[12] C. Scherer and S. Weiland, "Linear matrix inequalities in control," *Lecture Notes, Dutch Institute for Systems and Control, Delft, The Netherlands*, 1999.

[13] B. Pal and B. Chaudhuri, *Robust control in power systems*. Springer Science & Business Media, 2006.

[14] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization*. Princeton University Press, 2009.

[15] S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994, vol. 15.

[16] D. Henrion and A. Garulli, *Positive polynomials in control*. Springer Science & Business Media, 2005, vol. 312.

[17] G. Chesi, A. Garulli, A. Tesi, and A. Vicino, *Homogeneous Polynomial Forms for Robustness Analysis of Uncertain Systems*. Springer, 2009.

[18] G. Chesi, "LMI techniques for optimization over polynomials in control: a survey," *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2500–2510, 2010.

[19] C. W. Scherer, "LMI relaxations in robust control," *European Journal of Control*, vol. 12, no. 1, pp. 3–29, 2006.

[20] P. Kundur, N. Balu, and M. Lauby, *Power system stability and control*, 1994.

[21] R. D. Zimmerman and C. E. Murillo-Sanchez, "Matpower 5.1-user's manual," *Power Systems Engineering Research Center (PSERC)*, 2015.

[22] C.-W. Ho, A. Ruehli, and P. Brennan, "The modified nodal approach to network analysis," *IEEE Transactions on circuits and systems*, vol. 22, no. 6, pp. 504–509, 1975.

[23] P. Kundur, N. J. Balu, and M. G. Lauby, *Power system stability and control*. McGraw-hill New York, 1994, vol. 7.

[24] F. Milano, *Power system modelling and scripting*. Springer Science & Business Media, 2010.

[25] H.-D. Chiang, *Direct methods for stability analysis of electric power systems: theoretical foundation, BCU methodologies, and applications*. John Wiley & Sons, 2011.

[26] P. C. Magnusson, "The transient-energy method of calculating stability," *American Institute of Electrical Engineers, Transactions of the*, vol. 66, no. 1, pp. 747–755, 1947.

[27] G. Gless, "Direct method of liapunov applied to transient power system stability," *Power Apparatus and Systems, IEEE Transactions on*, no. 2, pp. 159–168, 1966.

[28] H.-D. Chiang, F. F. Wu, and P. P. Varaiya, "Foundations of direct methods for power system transient stability analysis," *Circuits and Systems, IEEE Transactions on*, vol. 34, no. 2, pp. 160–173, 1987.

[29] M. Pai, *Energy function analysis for power system stability*. Springer Science & Business Media, 1989.

[30] H.-D. Chiang, F. F. Wu, and P. P. Varaiya, "A bcu method for direct analysis of power system transient stability," *Power Systems, IEEE Transactions on*, vol. 9, no. 3, pp. 1194–1208, 1994.

[31] L. El Ghaoui and G. Scorletti, "Control of rational systems using linear-fractional representations and linear matrix inequalities," *Automatica*, vol. 32, no. 9, pp. 1273–1284, 1996.

[32] L. El Ghaoui, F. Oustry, and H. Lebret, "Robust solutions to uncertain semidefinite programs," *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 33–52, 1998.

[33] J. Magni, "Linear fractional representation toolbox for use with matlab," 2006, available with the SMAC Toolbox at http://w3.onera.fr/smac/lfrt.

[34] C. Roos, G. Hardier, and J.-M. Biannic, "Polynomial and rational approximation with the apricot library of the smac toolbox," in *2014 IEEE Conference on Control Applications (CCA)*. IEEE, 2014, pp. 1473–1478.

[35] R. Tóth, *Modeling and identification of linear parameter-varying systems*. Springer, 2010, vol. 403.

[36] C. Briat, "Linear parameter-varying and time-delay systems," *Analysis, Observation, Filtering & Control*, vol. 3, 2014.

[37] D. J. Leith and W. E. Leithead, "Survey of gain-scheduling analysis and design," *International journal of control*, vol. 73, no. 11, pp. 1001–1025, 2000.

[38] W. J. Rugh and J. S. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, no. 10, pp. 1401–1425, 2000.

[39] B. Tamimi, C. Canizares, and K. Bhattacharya, "System stability impact of large-scale and distributed solar photovoltaic generation: The case of Ontario, Canada," *Sustainable Energy, IEEE Transactions on*, vol. 4, no. 3, pp. 680–688, 2013.

[40] R. Shah, N. Mithulananthan, R. Bansal, and V. Ramachandaramurthy, "A review of key power system stability challenges for large-scale pv integration," *Renewable and Sustainable Energy Reviews*, vol. 41, pp. 1423–1436, 2015.

[41] J. L. Rueda, D. G. Colomé, and I. Erlich, "Assessment and enhancement of small signal stability considering uncertainties," *Power Systems, IEEE Transactions on*, vol. 24, no. 1, pp. 198–207, 2009.

[42] R. Preece, K. Huang, and J. Milanovic, "Probabilistic small-disturbance stability assessment of uncertain power systems using efficient estimation methods," *Power Systems, IEEE Transactions on*, vol. 29, no. 5, pp. 2509–2517, Sept 2014.

[43] R. Christie. (2000) Power systems test case archive.

[44] J. Jackson, "Interpretation and use of generator reactive capability diagrams," *Industry and General Applications, IEEE Transactions on*, no. 6, pp. 729–732, 1971.

[45] D. Lee, "Ieee recommended practice for excitation system models for power system stability studies (ieee std 421.5-1992)," *Energy Development and Power Generating Committee of the Power Engineering Society*, 1992.

[46] S. Eftekharnejad, V. Vittal, G. T. Heydt, B. Keel, and J. Loehr, "Small signal stability assessment of power systems with increased penetration of photovoltaic generation: A case study," *Sustainable Energy, IEEE Transactions on*, vol. 4, no. 4, pp. 960–967, 2013.

[47] P. Apkarian and D. Noll, "Nonsmooth H_∞ synthesis," *Automatic Control, IEEE Transactions on*, vol. 51, no. 1, pp. 71–86, Jan 2006.

[48] G. Verghese, F. Schweppe *et al.*, "Selective modal analysis with applications to electric power systems, part i: Heuristic introduction," *Power Apparatus and Systems, IEEE Transactions on*, no. 9, pp. 3117–3125, 1982.

[49] K. Wang, A. N. Michel, and D. Liu, "Necessary and sufficient conditions for the hurwitz and schur stability of interval matrices," *IEEE Transactions on Automatic Control*, vol. 39, no. 6, pp. 1251–1255, 1994.

[50] V. Balakrishnan, S. Boyd, and S. Balemi, "Branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems," *International Journal of Robust and Nonlinear Control*, vol. 1, no. 4, pp. 295–317, 1991.

[51] L. Ravanbod, D. Noll, and P. Apkarian, "Branch and bound algorithm for the robustness analysis of uncertain systems," *IFAC-PapersOnLine*, vol. 48, no. 25, pp. 85–90, 2015.

[52] A. Monticelli, M. Pereira, and S. Granville, "Security-constrained optimal power flow with post-contingency corrective rescheduling," *Power Systems, IEEE Transactions on*, vol. 2, no. 1, pp. 175–180, 1987.

[53] H. Ma and S. Shahidehpour, "Unit commitment with transmission security and voltage constraints," *Power Systems, IEEE Transactions on*, vol. 14, no. 2, pp. 757–764, 1999.

[54] N. Mithulananthan, R. Shah, and K. Y. Lee, "Small-disturbance angle stability control with high penetration of renewable generations," *Power Systems, IEEE Transactions on*, vol. 29, no. 3, pp. 1463–1472, 2014.

[55] A. Mills, "Implications of wide-area geographic diversity for short-term variability of solar power," *Lawrence Berkeley National Laboratory*, 2010.

[56] H. Holttinen, P. Meibom, A. Orths, B. Lange, M. O'Malley, J. O. Tande, A. Estanqueiro, E. Gomez, L. Söder, G. Strbac *et al.*, "Impacts of large amounts of wind power on design and operation of power systems, results of IEA collaboration," *Wind Energy*, vol. 14, no. 2, pp. 179–192, 2011.

[57] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*. IEEE, 2004, pp. 284–289.

[58] M. ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28).*, 2015.

[59] E. Davison and E. Kurak, "A computational method for determining quadratic lyapunov functions for non-linear systems," *Automatica*, vol. 7, no. 5, pp. 627–636, 1971.

[60] A. C. Antoulas, *Approximation of large-scale dynamical systems*. Siam, 2005, vol. 6.

[61] Y. Nesterov, A. Nemirovskii, and Y. Ye, *Interior-point polynomial algorithms in convex programming*. SIAM, 1994, vol. 13.

[62] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM review*, vol. 38, no. 1, pp. 49–95, 1996.

[63] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[64] A. d'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. Lanckriet, "A direct formulation for sparse pca using semidefinite programming," *SIAM review*, vol. 49, no. 3, pp. 434–448, 2007.

[65] Z. Wen, D. Goldfarb, and W. Yin, "Alternating direction augmented lagrangian methods for semidefinite programming," *Mathematical Programming Computation*, vol. 2, no. 3-4, pp. 203–230, 2010.

[66] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, "Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 746–768, 1998.

[67] M. Todd, K. Toh, and R. Tütüncü, "On the nesterov–todd direction in semidefinite programming," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 769–796, 1998.

[68] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.

[69] N. Parikh, S. P. Boyd *et al.*, "Proximal algorithms." *Foundations and Trends in optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[70] S. Becker, J. Bobin, and E. J. Candès, "Nesta: a fast and accurate first-order method for sparse recovery," *SIAM Journal on Imaging Sciences*, vol. 4, no. 1, pp. 1–39, 2011.

[71] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[72] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[73] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd, "Conic optimization via operator splitting and homogeneous self-dual embedding," *Journal of Optimization Theory and Applications*, pp. 1–27, 2016.

[74] O. Devolder, F. Glineur, and Y. Nesterov, "First-order methods of smooth convex optimization with inexact oracle," *Mathematical Programming*, vol. 146, no. 1-2, pp. 37–75, 2014.

[75] Y. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical programming*, vol. 103, no. 1, pp. 127–152, 2005.

[76] ——, "Smoothing technique and its applications in semidefinite optimization," *Mathematical Programming*, vol. 110, no. 2, pp. 245–259, 2007.

[77] ——, "Gradient methods for minimizing composite functions," *Mathematical Programming*, vol. 140, no. 1, pp. 125–161, 2013.

[78] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *Journal of Scientific Computing*, vol. 66, no. 3, pp. 889–916, 2016.

[79] B. He and X. Yuan, "On the o(1/n) convergence rate of the douglas-rachford alternating direction method," *SIAM Journal on Numerical Analysis*, vol. 50, no. 2, pp. 700–709, 2012.

[80] Y. Nesterov, "A method of solving a convex programming problem with convergence rate o (1/k2)," vol. 27, no. 2, pp. 372–376, 1983.

[81] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.

[82] Y. Saad, *Iterative methods for sparse linear systems*. Siam, 2003.

[83] A. Greenbaum, *Iterative methods for solving linear systems*. Siam, 1997, vol. 17.

[84] M. H. Wright, "Some properties of the hessian of the logarithmic barrier function," *Mathematical Programming*, vol. 67, no. 1-3, pp. 265–295, 1994.

[85] S. J. Wright, "On the convergence of the newton/log-barrier method," *Mathematical Programming*, vol. 90, no. 1, pp. 71–100, 2001.

[86] K.-C. Toh and M. Kojima, "Solving some large scale semidefinite programs via the conjugate residual method," *SIAM Journal on Optimization*, vol. 12, no. 3, pp. 669–691, 2002.

[87] K.-C. Toh, "Solving large scale semidefinite programs via an iterative solver on the augmented systems," *SIAM Journal on Optimization*, vol. 14, no. 3, pp. 670–698, 2004.

[88] N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, no. 4, pp. 373–395, 1984.

[89] L. Vandenberghe and S. Boyd, "A primal-dual potential reduction method for problems involving matrix inequalities," *Mathematical Programming*, vol. 69, no. 1-3, pp. 205–236, 1995.

[90] K.-C. Toh and M. Kojima, "Solving some large scale semidefinite programs via the conjugate residual method," *SIAM Journal on Optimization*, vol. 12, no. 3, pp. 669–691, 2002.

[91] K.-C. Toh, "Solving large scale semidefinite programs via an iterative solver on the augmented systems," *SIAM Journal on Optimization*, vol. 14, no. 3, pp. 670–698, 2004.

[92] Y. Saad, *Iterative methods for sparse linear systems*. Siam, 2003.

[93] T. J. Rivlin, *The Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory*. John Wiley & Sons, 1974.

[94] L. Vandenberghe, S. Boyd, and S.-P. Wu, "Determinant maximization with linear matrix inequality constraints," *SIAM journal on matrix analysis and applications*, vol. 19, no. 2, pp. 499–533, 1998.

[95] W. Hackbusch, "A sparse matrix arithmetic based on\ cal h-matrices. part i: Introduction to {\ Cal H}-matrices," *Computing*, vol. 62, no. 2, pp. 89–108, 1999.

[96] W. C. Chew, E. Michielssen, J. Song, and J.-M. Jin, *Fast and efficient algorithms in computational electromagnetics*. Artech House, Inc., 2001.

[97] P.-G. Martinsson and V. Rokhlin, "A fast direct solver for boundary integral equations in two dimensions," *Journal of Computational Physics*, vol. 205, no. 1, pp. 1–23, 2005.

[98] S. Chandrasekaran, M. Gu, and T. Pals, "A fast ulv decomposition solver for hierarchically semiseparable representations," *SIAM Journal on Matrix Analysis and Applications*, vol. 28, no. 3, pp. 603–622, 2006.

[99] L. Greengard, D. Gueyffier, P.-G. Martinsson, and V. Rokhlin, "Fast direct solvers for integral equations in complex three-dimensional domains," *Acta Numerica*, vol. 18, pp. 243–275, 2009.

[100] K. L. Ho and L. Greengard, "A fast direct solver for structured linear systems by recursive skeletonization," *SIAM Journal on Scientific Computing*, vol. 34, no. 5, pp. A2507–A2532, 2012.

[101] T. Kloks, H. Bodlaender, H. Müller, and D. Kratsch, "Computing treewidth and minimum fill-in: All you need are the minimal separators," in *European Symposium on Algorithms*. Springer, 1993, pp. 260–271.

[102] H. L. Bodlaender, J. R. Gilbert, H. Hafsteinsson, and T. Kloks, "Approximating treewidth, pathwidth, frontsize, and shortest elimination tree," *Journal of Algorithms*, vol. 18, no. 2, pp. 238–255, 1995.

[103] V. Bouchitté and I. Todinca, "Treewidth and minimum fill-in: Grouping the minimal separators," *SIAM Journal on Computing*, vol. 31, no. 1, pp. 212–232, 2001.

[104] R. Madani, M. Ashraphijuo, and J. Lavaei, "Promises of conic relaxation for contingency-constrained optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1297–1307, 2016.

[105] H. L. Bodlaender, "A tourist guide through treewidth," *Acta cybernetica*, vol. 11, no. 1-2, p. 1, 1994.

[106] R. J. Lipton, D. J. Rose, and R. E. Tarjan, "Generalized nested dissection," *SIAM journal on numerical analysis*, vol. 16, no. 2, pp. 346–358, 1979.

[107] G. Strang and T. Nguyen, "The interplay of ranks of submatrices," *SIAM review*, vol. 46, no. 4, pp. 637–646, 2004.

[108] R. J. Vanderbei, "Symmetric quasidefinite matrices," *SIAM Journal on Optimization*, vol. 5, no. 1, pp. 100–113, 1995.

[109] Y. Nesterov, *Introductory lectures on convex optimization*. Springer Science & Business Media, 2004, vol. 87.

[110] P. Giselsson and S. Boyd, "Diagonal scaling in Douglas-Rachford splitting and ADMM," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 5033–5039.

[111] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems," *Automatic Control, IEEE Transactions on*, vol. 60, no. 3, pp. 644–658, 2015.

[112] R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. I. Jordan, "A general analysis of the convergence of ADMM," *arXiv preprint arXiv:1502.02009*, 2015.

[113] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1-3, pp. 293–318, 1992.

[114] T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk, "Fast alternating direction optimization methods," *SIAM Journal on Imaging Sciences*, vol. 7, no. 3, pp. 1588–1623, 2014.

[115] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *Journal of Scientific Computing*, pp. 1–28, 2012.

[116] D. Han and X. Yuan, "Local linear convergence of the alternating direction method of multipliers for quadratic programs," *SIAM Journal on numerical analysis*, vol. 51, no. 6, pp. 3446–3457, 2013.

[117] D. Davis and W. Yin, "Convergence rate analysis of several splitting schemes," *arXiv preprint arXiv:1406.4834*, 2014.

[118] ——, "Faster convergence rates of relaxed Peaceman-Rachford and ADMM under regularity assumptions," *arXiv preprint arXiv:1407.5210*, 2014.

[119] D. Boley, "Local linear convergence of the alternating direction method of multipliers on quadratic or linear programs," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2183–2207, 2013.

[120] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *Signal Processing, IEEE Transactions on*, vol. 62, no. 7, pp. 1750–1761, 2014.

[121] A. Greenbaum, *Iterative methods for solving linear systems*. Siam, 1997, vol. 17.

[122] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *Information Theory, IEEE Transactions on*, vol. 47, no. 7, pp. 2845–2862, 2001.

[123] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.

[124] B. Borchers, "SDPLIB 1.2, a library of semidefinite programming test problems," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 683–690, 1999.

[125] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on scientific and statistical computing*, vol. 7, no. 3, pp. 856–869, 1986.

[126] T. A. Driscoll, K.-C. Toh, and L. N. Trefethen, "From potential theory to matrix iterations in six steps," *SIAM review*, vol. 40, no. 3, pp. 547–578, 1998.

[127] M. Benzi and V. Simoncini, "On the eigenvalues of a class of saddle point matrices," *Numerische Mathematik*, vol. 103, no. 2, pp. 173–196, 2006.

[128] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd, "Operator splitting for conic optimization via homogeneous self-dual embedding," *arXiv preprint arXiv:1312.3039*, 2013.

[129] M. Annergren, S. K. Pakazad, A. Hansson, and B. Wahlberg, "A distributed primal-dual interior-point method for loosely coupled problems using ADMM," *arXiv preprint arXiv:1406.2192*, 2014.

[130] S. K. Pakazad, A. Hansson, and M. S. Andersen, "Distributed primal-dual interior-point methods for solving loosely coupled problems using message passing," *arXiv preprint arXiv:1502.06384*, 2015.

[131] M. Benzi, G. H. Golub, and J. Liesen, "Numerical solution of saddle point problems," *Acta numerica*, vol. 14, no. 1, pp. 1–137, 2005.