AFRL-AFOSR-VA-TR-2019-0025

Optimization of Complex Systems using Imperfect Data from Large-scale Simulations

Jason Hicken
RENSSELAER POLYTECHNIC INST TROY NY

11/28/2018
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ RTA2
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
10/26/2018	Final Report	Aug 2015 - Jul 2018

4. TITLE AND SUBTITLE
Optimization of Complex Systems using Imperfect Outputs from Large-Scale Simulations

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-15-1-0242

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Hicken, Jason, E
Ashley, Anthony

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Rensselaer Polytechnic Institute
110 8th Street
Troy NY 12180-3522

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
USAF, AFRL DUNS 143574726
Air Force Office of Scientific Research
875 North Randolph Street, RM 3112
Arlington VA 22203-1954

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution A - Approved for Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Hicken, Jason, E
U	U	U	UU		**19b. TELEPHONE NUMBER** *(Include area code)* 518-276-4893

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g. 30-06-1998; xx-06-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATE COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33315-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report. e.g. AFOSR-82-1234.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. 61101A.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, J, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/ monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/ restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.

FA9550-15-1-0242 Final Report: Optimization of Complex Systems using Imperfect Outputs from Large-Scale Simulations

Jason E. Hicken (PI) and Anthony Ashley

October, 2018

Contents

List of Figures

List of Tables

Executive Summary

This report summarizes our efforts to develop an optimization framework that can use imperfect (*i. e.* inaccurate) data to support the design of complex, high-dimensional aerospace systems. We are primarily interested in the optimization of systems governed by chaotic dynamics, such as gas-turbine combustors and aircraft during stall; however, the framework is also suitable for inaccurate data that may arise during design under uncertainty.

Chapters 1 and 2 describe our work to develop gradient-based optimization algorithms that are suitable for large-scale, high-dimensional problems with inaccurate data. Our preliminary investigations into Anderson Acceleration, summarized in Chapter 1, demonstrate that this root-finding algorithm is well-suited for unconstrained optimization in the context of inaccurate and noisy data. When considering constrained optimization problems, we broadened our investigation to the more general class of multi-secant methods. This is the subject of Chapter 2. In that chapter we show that, when a suitable preconditioner is available, multi-secant methods are highly effective for constrained optimization with inaccurate derivative data. Furthermore, when accurate derivatives are available, our proposed algorithm outperforms a state-of-the-art inexact-Newton-Krylov method.

In addition to an error-tolerant algorithm, the optimization of chaotic systems requires a scalable sensitivity analysis method. To this end, Chapters 3 and 4 describe our efforts to develop a suitable sensitivity analysis method. The initial investigation, presented in Chapter 3, considers a reduced-order approximation to the least-squares shadowing (LSS) adjoint. The idea is to approximate the LSS adjoint using a modal (Fourier) expansion to reduce the size of the LSS space-time boundary-value problem. The expansion succeeds in reducing the size of the problem, while maintaining accuracy, but it produces a fully-coupled dense-in-time system. More recently, we have investigated a straightforward approach that finds the minimal perturbation that stabilizes the adjoint or direct sensitivity equation. This promising stabilization approach is described in Chapter 4.

Chapter 1

Adapting Anderson acceleration to optimization with inaccurate gradients

1.1 Introduction

Gradient-based optimization can be a valuable tool in the engineering design process. This is especially true when the target design is governed by complex (nonlinear) physics and there are many design parameters. In such circumstances, even a seasoned engineer will be challenged to choose the parameters using intuition alone. Exemplar applications of gradient-based optimization include aerodynamic shape optimization [41, 64, 65, 61, 6, 58, 59], aerostructural design [54, 52, 55, 44, 43], structural topology optimization [12, 80, 70, 73], and satellite design [40], to name a few.

A practical challenge in applying gradient-based optimization is the efficient and accurate evaluation of derivatives. Conventional optimization methods rely on accurate derivative information for computing search directions and for globalization strategies, *e. g.* line-search and trust-region methods. Inaccurate gradients and/or objectives produce inconsistencies during the globalization that ultimately lead to failure of the optimization algorithm. The failure can occur well before the optimization has made significant progress toward improving the design or satisfying the constraints.

Given the importance of accurate and efficient gradients, it is hardly surprising that considerable work has been devoted to methods for differentiating engineering analysis codes. For example, analytic sensitivity methods like the direct and adjoint sensitivity methods [62, 35, 41] are now used

routinely in partial-differential equation (PDE) solvers, including some commercial codes[1]. In addition, algorithmic differentiation [34] has matured to the point that it can be applied to a large, complex PDE solver library [3].

Nevertheless, despite the progress in computing derivatives, there remain industrially relevant applications for which inaccuracies in the gradient are theoretically or practically unavoidable:

- Problems governed by chaotic dynamics, in which the objective is a time-averaged quantity, cannot be treated using conventional sensitivity analysis methods [48]. Instead, methods like the least-square-shadowing (LSS) adjoint are needed [78]; however, even the LSS adjoint produces gradients with errors that cannot be eliminated.

- Difficult nonlinear analyses often suffer from incomplete convergence of their algebraic solvers. An example is the solution of the Reynolds-averaged Navier-Stokes equations for configurations at high lift conditions, where two to three orders reduction in the nonlinear residual may be acceptable. Incomplete convergence leads to errors in both the objective function and the gradient.

- When the continuous-adjoint method [62, 41] is used, the resulting gradient is inconsistent with the objective function in that it differs from the true gradient by errors on the order of the discretization [20].

- If the analysis mesh is regenerated during a line search, the gradient based on the previous mesh and the objective based on the new mesh will be inconsistent; this can occur when using a Cartesian adaptive flow solver [1, 2] for aerodynamic shape optimization [57].

- In a multifidelity analysis [10], the objective may be evaluated using a high-fidelity model and the gradient may be evaluated using a low-fidelity model, again leading to inconsistencies.

While some of the above problems can be ameliorated in theory, *e.g.* mesh refinement in the case of the continuous adjoint or mesh regeneration, the computational cost may not be acceptable in practice.

In this chapter, we describe our first efforts to devise an optimization algorithm that is tolerant of inaccuracies in the objective and gradient in the context of high-dimensional, computationally expensive simulations. In Section 1.2 we review Anderson Acceleration (AA), a root-finding multisecant method, and adapt it to optimization. Using a synthetic quadratic,

[1] For example, ANSYS CFD now provides adjoint capabilities

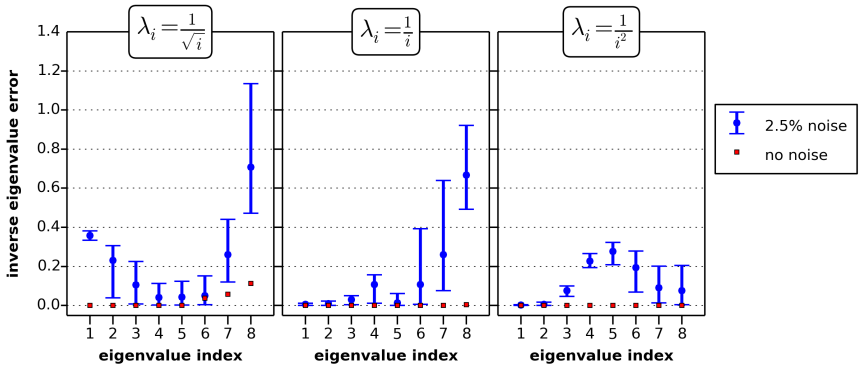


Figure 1.1: Relative accuracy in the inverse of the eigenvalues approximated using Arnoldi sampling with a noise magnitude of 2.5% for three different eigenvalue distributions.

we demonstrate that AA is robust to inaccuracies in the gradient. To handle nonconvex objective functions, we propose a homotopy path-following globalization in Section 1.3. We apply the globalized AA optimization algorithm to a chaotic-differential-equation constrained optimization problem in Section 1.4.

1.2 Anderson Acceleration

In the months leading to the start of this award, the PI and A. Ashley had been investigating Arnoldi's method in the context of optimization with inaccurate gradients. A summary of this investigation can be found in Reference[37]. In brief, our strategy was to replace Hessian-vector products in Arnoldi's method with relatively large perturbations in the gradient, and then take the symmetric part of the upper Hessenberg matrix to form a quadratic model for the objective in the "Krylov" subspace. We showed that this Stochastic Arnoldi's Method (SAM) was able to estimate the dominant eigenvalues of the Hessian with surprising accuracy, even in the presence of error; see Figure 1.1.

Despite the accuracy of its eigenvalue estimates, SAM was found to be unreliable for noisy optimization problems with ill-conditioned Hessians: any errors in the gradient were significantly amplified by the approximate inverse, thus leading to poor steps. Another issue with SAM was the selection of the step size for the gradient perturbation. We developed an effective

Algorithm 1: Anderson Acceleration.

Data: Given the initial design, $x_0 \in \mathbb{R}^n$, and $m \geq 1$.

Set $g_0 = g(x_0)$, $x_1 = x_0 - g_0$, and $g_1 = g(x_1)$
for $k = 1, 2, \ldots$ **do**
  Set $m_k = \min(m, k)$
  Set $\mathsf{X}_k = [\Delta x_{k-m_k}, \ldots, \Delta x_{k-1}]$
  Set $\mathsf{G}_k = [\Delta g_{k-m_k}, \ldots, \Delta g_{k-1}]$
  Solve $\gamma_k = \mathsf{argmin}_\gamma \|g_k + \mathsf{G}_k \gamma\|$
  Set $x_{k+1} = x_k - g_k + (\mathsf{X}_k - \mathsf{G}_k)\gamma$
end

error estimate to select the optimal step size, but it was computationally expensive.

The issues with SAM lead us to focus on a different, albeit related, set of algorithms: multi-secant methods. Multi-secant methods are used in several fields that must contend with data that is noisy, computationally expensive, and high dimensional. In particular, Anderson acceleration (AA) [5], also known as Anderson mixing, is a popular multi-secant method for electronic structure calculations [19]. AA has also found uses in complex transportation system studies [13]. The overlap in problem characteristics — noisy, high-dimensional, and expensive — suggests that AA and other multi-secant methods would be good candidates for the current work.

Algorithm 1 lists the AA method adapted to the unconstrained optimization problem

$$\min_x \ f(x),$$

where $x \in \mathbb{R}^n$ denotes the design or control variable. We assume that $f(x)$ has continuous first-order derivatives and denote its gradient by $g : \mathbb{R}^n \to \mathbb{R}^n$, *i.e.* $g = \nabla f$. The algorithm introduces the matrices $\mathsf{X}_k, \mathsf{G}_k \in \mathbb{R}^{n \times m_k}$, which are constructed from the differences

$$\Delta x_i = x_{i+1} - x_i, \qquad \text{and} \qquad \Delta g_i = g_{i+1} - g_i,$$

for each $i \in \{k - m_k, k - m_k + 1, \ldots, k - 1\}$. Since we are using AA to solve $g(x) = 0$, there is some ambiguity regarding the sign of $g(x)$ (we could also solve $-g(x) = 0$). In Algorithm 1, we have adopted the sign that leads to a steepest descent direction in the first iterate.

To explore the potential of AA on noisy, high-dimensional optimization problems, we compared both AA and SAM [37] on the optimization of a set

of quadratic objectives with different eigenvalue distributions. In particular, the objective was defined by

$$F(x) = x^T \mathsf{E}\Sigma\mathsf{E}^T x, \tag{1.1}$$

where E denotes the $2^p \times 2^p$ orthonormalized Hadamard matrix, whose columns are the synthetic eigenvectors. The diagonal matrix $\Sigma \in \mathbb{R}^{2^p \times 2^p}$ holds the prescribed eigenvalues, given by

$$\Sigma_{i,i} = \frac{1}{i^q},$$

where $q = \frac{1}{2}, 1$, or 2.

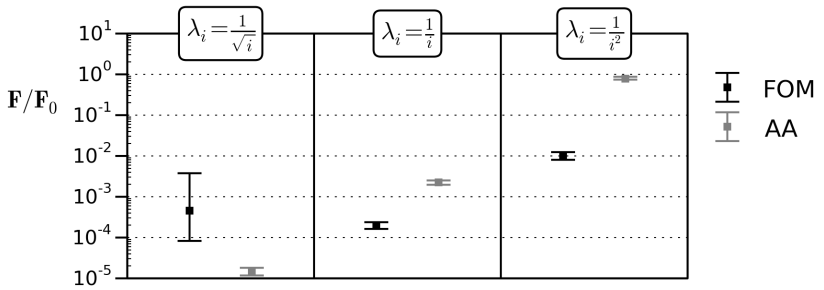
For this investigation, we considered $n = 256$ variables. The initial guess was set to $x_0 = \frac{2}{n}\mathsf{E}\Sigma^{-1}\mathbf{1}$, which ensures that $f(x_0) = 1$. Noise with a Gaussian distribution was added to each component of the gradient. The noise had zero mean and its standard deviation was either 0.05%, 0.5% or 5%, relative to the norm of the initial gradient, $\|g_0\|$.

Figure 1.2 shows the objective-function reduction statistics for SAM and AA. Both algorithms used $m = 40$ iterations, so their computational cost is roughly the same in terms of gradient evaluations. For low-noise cases, SAM outperforms AA, which is not surprising because it can exhibit its underlying Newton-Krylov behavior. However, for the high-noise cases of interest here, AA performs much better: unlike SAM it reduces or at least maintains the objective value, whereas SAM can diverge due to the ill-conditioning of the Hessian.

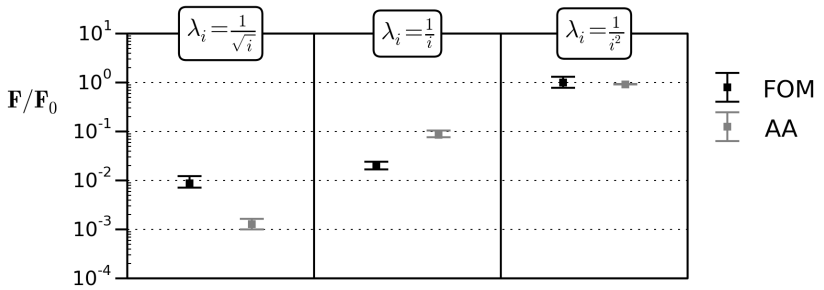
1.3 Globalization of Anderson Acceleration

The encouraging results in Figure 1.2 motivated us to continue work with AA. Indeed, given these results, a natural question arises: "why has Anderson Acceleration not found broader use within the optimization community?" We believe one answer to this question is globalization. AA is a method for solving fixed-point or nonlinear equations. In the case of optimization, it solves $g(x) = 0$ and does not distinguish between local maximizers and minimizers. Moreover, standard methods of globalizing AA for optimization problems, namely line-search and trust-region methods, cannot be applied directly to AA, because i) it does not guarantee descent directions (required for line-searches) and ii) it does not produce a symmetric Hessian approximation (required for trust-region models).

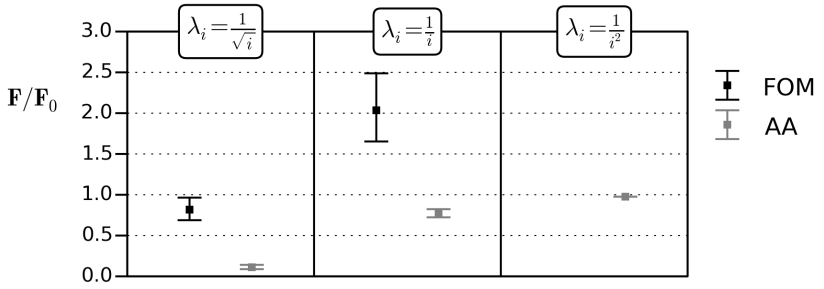
In the following sections, we describe an approach we have been developing to globalize AA for optimization problems.



(a) 0.05% noise



(b) 0.5% noise



(c) 5% noise

Figure 1.2: Mean and 95% confidence intervals in the reduction of the objective, F/F_0, using the Stochastic Arnoldi Method (FOM) and Anderson Acceleration (AA). Note the different scales for F/F_0 used in (a), (b), and (c). The FOM results do not appear in the $\lambda_i = 1/i^2$ case of Figure 1.2(c) because its 95% confidence interval is $[165, 126970]$.

1.3.1 Homotopy path-following continuation

Homotopy-based globalization methods introduce a nonlinear mapping whose solution traces a path from an "easy" problem to the nonlinear problem of interest [4]. In the context of optimization, we can consider the parameterized problem

$$\min_x \nu f(x) + (1-\nu)\frac{1}{2}(x-x_0)^T(x-x_0), \tag{1.2}$$

where $\nu \in [0,1]$. Let x_ν denote the solution to (1.2). When $\nu = 0$, the solution is clearly $x_\nu = x_0$, while for $\nu = 1$ the solution is the desired x. The idea is to move along the path defined by x_ν from the easy problem defined by $\nu = 0$ to the (possibly) hard problem $\nu = 1$; hence the name path-following. See [79] for a review of the theory of homotopies for nonlinear optimization.

Homotopy path-following methods are attractive for AA, and noisy optimization more generally, because they only require a sequence of stationary points be found. For convex problems, this sequence of stationary points leads from one local minimizer (x_0) to the desired solution. Moreover, there is no need for line-search or trust-region methods.

There are several possible strategies for following the path defined by x_ν. In this work we have implemented a predictor-corrector algorithm [4], in which we use AA to approximately solve an Euler-based predictor and then use AA again for the corrector phase.

To illustrate this globalization method, we consider the following optimization problem:

$$\min_x \ f(x) = \frac{1}{2}x^2 - \frac{1}{3}x^3.$$

We assume that the variable is restricted to $[-1,1]$. In this domain there is a local minimizer at $x = 0$ and a local maximizer at $x = 1$. Figure 1.3 show several paths beginning at different initial guesses along $\nu = 0$. This example includes initial guesses within 0.001 of the local maximizer. In each case, the globalization is able to converge to the local minimizer.

1.3.2 Resilience to Noise

The globalization strategy must be resilient to noise; that is, it should converge to a local minimizer with high probability when noise is present. To investigate the resilience of the path-following globalization, we solved the problem

$$\min_{x,y} f(x,y) = \frac{1}{2}x^2 - \frac{1}{3}x^3 + \frac{1}{2}y^2$$

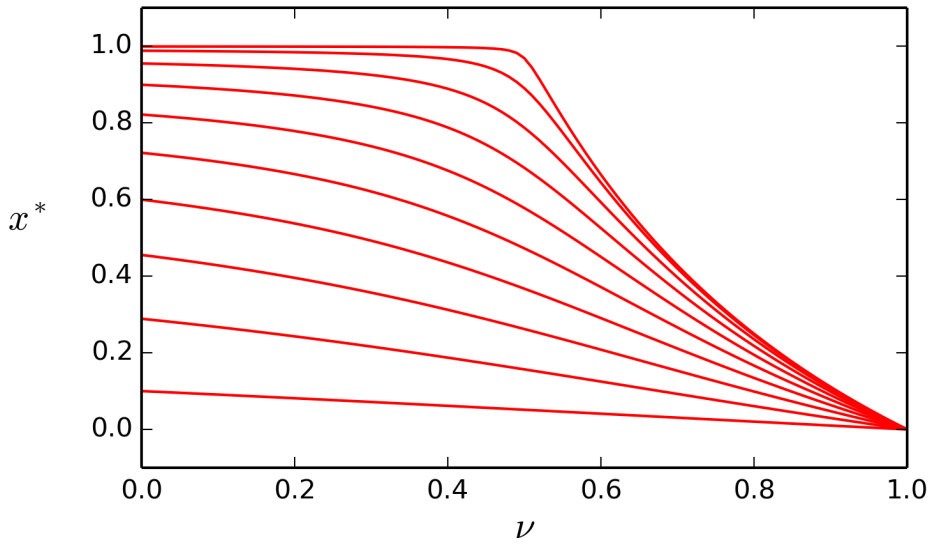


Figure 1.3: Example homotopy paths for the problem $\min_x \ f(x) = \frac{1}{2}x^2 - \frac{1}{3}x^3$ beginning at different x_0.

with Gaussian noise added to the gradient. For each initial value $x_0 \in \{i/100\}_{i=0}^{100}$, one hundred random values of y were selected with uniform probability from the interval $[-1, 1]$, and the noise standard deviation was set to $0.5\|g_0\|$. The path-following algorithm was then applied to find the local minimizer using relative tolerances of 0.5 for the corrector stages.

Figure 1.3.2 plots the median of the solution determined by AA versus x_0. The 50% and 95% confidence intervals are also plotted. The local minimizer has a value of $f(x, y) = 0$, so values close to this indicate success. The figure shows that the path-following algorithm determines the local minimizer with 95% probability up to $x_0 \approx 0.75$. It determines the minimizer with greater than 50% probability up to $x_0 \approx 0.97$.

1.4 Anderson Acceleration applied to a chaotic problem

In this section, we present the results of applying our globalized AA algorithm to an objective function based on the Lorenz dynamical system. The Lorenz system is described in more detail in Chapter 3. The objective

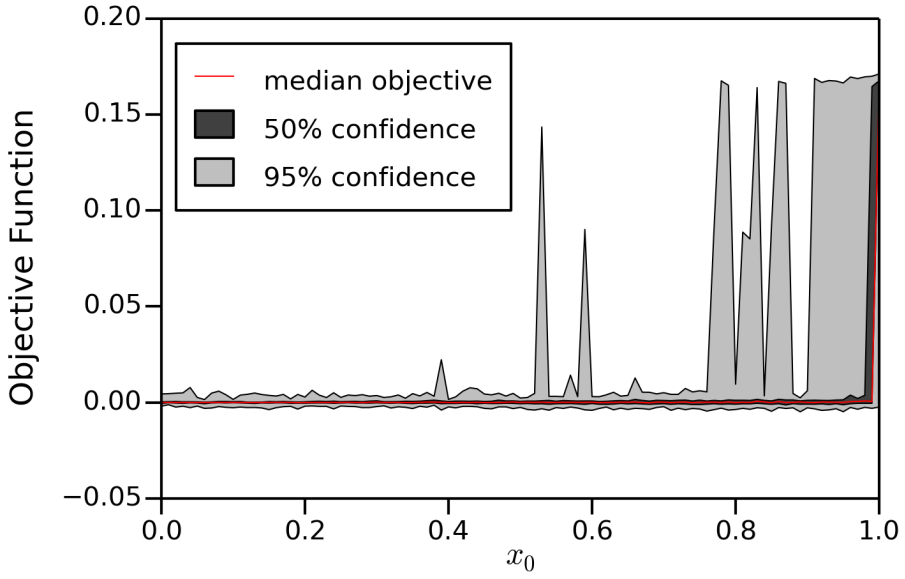


Figure 1.4: Median, 50% and 95% confidence intervals of the optimal objective value as determined by AA globalized with homotopy continuation. There is a 0.5% noise added to the gradient based on the initial gradient norm.

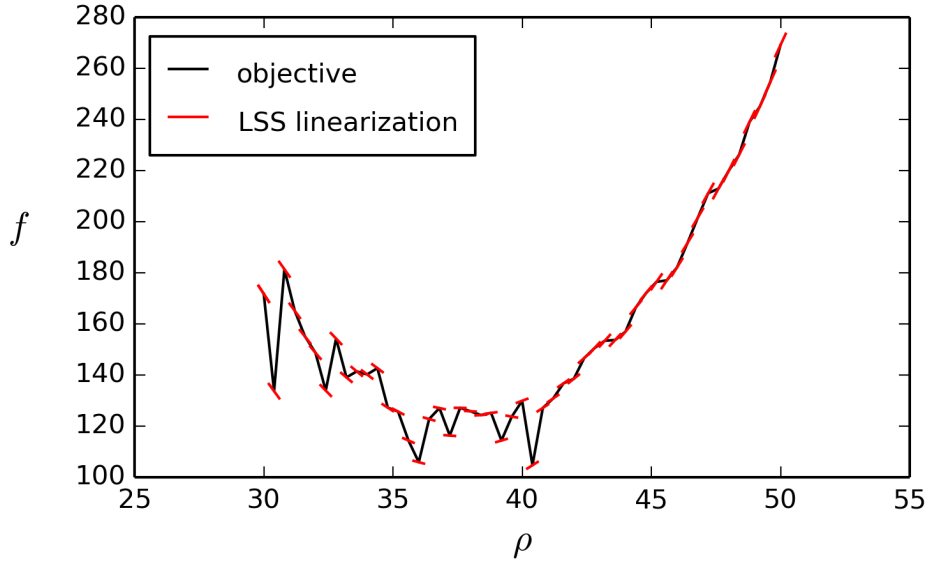


Figure 1.5: $\mathcal{J}$ versus ρ for constant $\beta = 8/3$. The red line segments show the LSS-based linearization.

function we consider is

$$\mathcal{J}(\rho, \beta) = \frac{1}{T} \int_0^T (z - z_{\mathsf{targ}})^2 \, dt + \frac{40}{\beta},$$

where the design variables, ρ and β, are parameters in the Lorenz ODE; see (3.1) and (3.3). The time period is $T = 16$ units, and we use a "spin-up" time of $T_{\mathsf{spin}} = 10$ units. The Lorenz ODE is discretized using Crank-Nicolson with a step size of $\Delta t = 0.0064$.

Derivatives of $\mathcal{J}$ are estimated using the LSS method [78] with a time-dilation parameter of $\alpha = 100$. Figure 1.4 plots the objective $\mathcal{J}$ and its linearization based on the LSS gradient for a range of ρ values and $\beta = 8/3$. The LSS gradients are remarkably accurate, despite the large fluctuations in the objective function. The fluctuations in $\mathcal{J}$ illustrate why line-search and trust-region methods are difficult to use in the context of noisy optimization.

We applied AA to the Lorenz problem starting from 100 randomly selected initial guesses from $\rho \in [30, 50]$ and $\beta \in [0.25, 4]$. These bound constraints were enforced explicitly in AA. Each run used a maximum of $m = 5$ saved vectors in AA, and at most 3 corrector iterations. The ν parameter was increased by 0.1 from one homotopy iteration to the next.

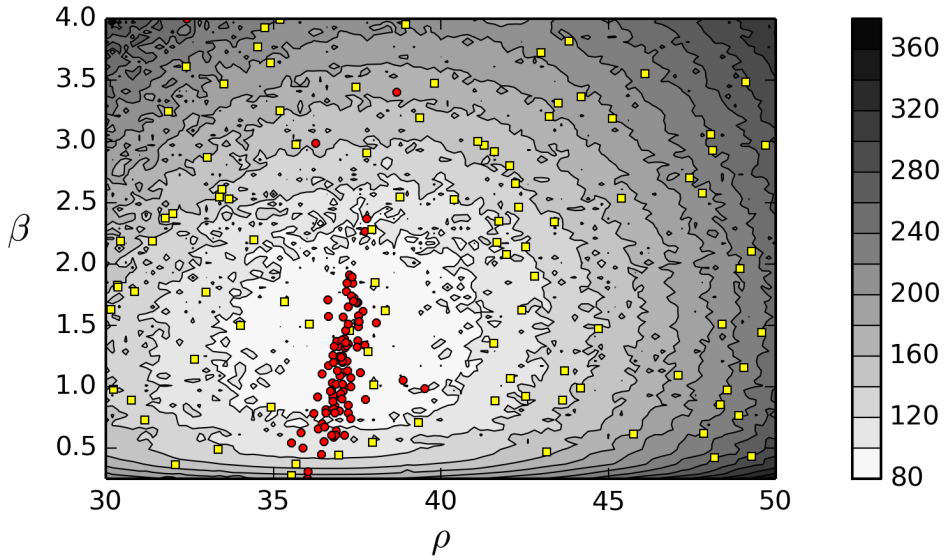


Figure 1.6: Initial guesses (yellow squares) and approximate solutions (red circles) for 100 runs of AA applied to the Lorenz problem.

Figure 1.6 plots the 100 x_0 values (yellow squares) and the x^* solutions from AA (red circles) on top of the contours of $\mathcal{J}(\rho, \beta)$. This figure demonstrates, at least qualitatively, that the globalized AA algorithm is effective on a simple chaotic optimization problem. Note the apparent clustering in the β direction is an artifact of the plotting scale used in the figure.

We point out that not every run of AA was acceptable. For example, at least one red circle lies along the upper boundary, $\beta = 4$. Therefore, for a more quantitative assessment of the algorithm, we have plotted the convergence statistics of the 100 runs in Figure 1.7. This figure shows, for each iteration, the median value and 50% and 95% confidence intervals for the objective. More than half the runs terminate within 20 units of the optimal value, and 95% of the time the final value is half the worst-case value on the design domain.

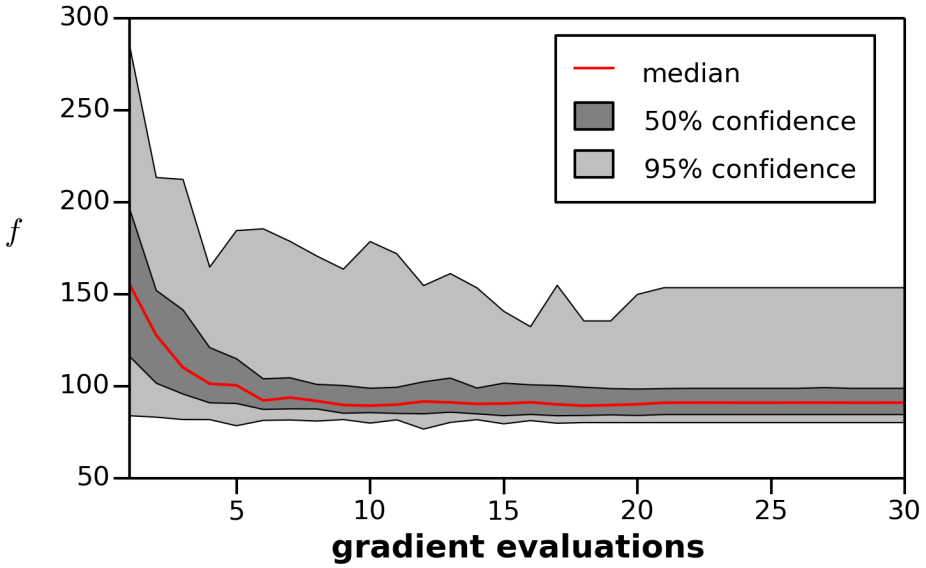


Figure 1.7: Median value, and 50% and 95% confidence intervals, at each AA iteration for Lorenz problem.

Chapter 2

Error-tolerant Multisecant Method for Nonlinearly Constrained Optimization

2.1 Introduction

This chapter builds on our preliminary investigations of Anderson Acceleration for unconstrained optimization with inaccurate data. In particular, we present a multisecant quasi-Newton algorithm designed to address derivative-based optimization in the presence of inaccurate data. Multisecant methods [74, 28] are a generalization of Broyden's method for nonlinear equations [16], and they have been shown to be particularly effective for solving nonlinear equations that are high dimensional, expensive to evaluate, and potentially noisy [13, 19, 30].

To solve constrained optimization problems using multisecant methods, we formulate the first-order necessary optimality conditions as an equivalent set of nonlinear equations and apply the multisecant update directly to these equations. An advantage of this approach is that it is matrix-free in the sense that it does not require the constraint Jacobian or Hessian of the Lagrangian. This should be contrasted with conventional algorithms that require, at the least, the explicit constraint Jacobian. The constraint Jacobian is especially problematic in PDE-constrained optimization problems with many state-dependent constraints, because each such constraint requires the solution of an adjoint equation.

The rest of the chapter is organized as follows. Section 2.2 formally defines the optimization problem and shows how the first-order necessary

conditions can be recast as an equivalent set of semi-smooth nonlinear equations. Section 2.3 begins with a general review of multisecant methods, and then describes the particulars of our implementation, including how we incorporate preconditioning and how we handle nonconvexity. Some numerical experiments are provided in Section 2.4 to verify the method and demonstrate its effectiveness. Finally, Section 2.5 summarizes our findings and discusses future work.

2.2 Preliminaries

2.2.1 Problem Definition

We consider nonlinear optimization problems of the form

$$\begin{aligned} \min_{x} \quad & f(x), \\ \text{s.t.} \quad & h(x) = 0, \\ & g(x) \geq 0, \end{aligned} \tag{P}$$

where $x \in \mathbb{R}^n$ denotes the optimization variables, $f : \mathbb{R}^n \to \mathbb{R}$ is the objective, $h : \mathbb{R}^n \to \mathbb{R}^{m_h}$ are the equality constraints, and $g : \mathbb{R}^n \to \mathbb{R}^{m_g}$ are the inequality constraints. We will assume that the objective and constraints are continuously differentiable.

Our strategy for solving (P) is based on Newton's method, which requires that we recast the problem as a system of nonlinear equations. To that end, we recall that a solution to (P) must satisfy the first-order optimality conditions, also known as the Karush-Kuhn-Tucker (KKT) conditions.

Theorem 1 (Karush-Kuhn-Tucker (KKT) conditions)**.** *Let x^* denote a local solution of (P). If the set of active constraint gradients, namely*

$$\{\nabla h_i(x^*) \mid i = 1, \ldots, m_h\} \bigcup \{\nabla g_i(x^*) \mid g_i(x^*) = 0,\ i = 1, \ldots, m_g\},$$

is linearly independent, then there are multiplier values λ^* and μ^* such that

$$\nabla L(x^*, \lambda^*, \mu^*) = 0, \tag{2.1a}$$
$$h(x^*) = 0, \tag{2.1b}$$
$$g_i(x^*)\mu_i^* = 0, \quad \forall i = 1, \ldots, m_g, \tag{2.1c}$$
$$g_i(x^*) \geq 0, \quad \forall i = 1, \ldots, m_g, \tag{2.1d}$$
$$\mu_i^* \geq 0, \quad \forall i = 1, \ldots, m_g, \tag{2.1e}$$

where $L(x, \lambda, \mu) \equiv f(x) - h(x)^T\lambda - g(x)^T\mu$ is the Lagrangian.

Newton's method cannot be applied directly to the KKT conditions, because it cannot enforce the bounds (2.1d) and (2.1e). That said, Newton's method can be applied indirectly. For example, interior-point methods deal with the bounds (2.1d) and (2.1e) by introducing a homotopy map with a barrier term and using a sequence of Newton solves, while active-set methods attempt to predict the active inequality constraints and treat them as equality constraints during a Newton-like iteration.

Our approach for dealing with inequality constraints is related to the active-set approach and is based on the following theorem due to Mangasarin [51].

Theorem 2. *Let $G : \mathbb{R} \to \mathbb{R}$ be any strictly increasing function, that is $a > b \Leftrightarrow G(a) > G(b)$, and let $G(0) = 0$. Then $g_i(x^*)$ and μ_i^* satisfy the complementarity conditions (2.1c), (2.1d), and (2.1e) if and only if*

$$G(|g_i(x^*) - \mu_i^*|) - G(g_i(x^*)) - G(\mu_i^*) = 0.$$

Proof. The original proof given in [51] holds with $g_i(x^*)$ taking the role of $F_i(z)$ and μ_i taking the role of z_i. □

Theorem 2 is significant, because it shows that we can replace the problematic complementarity conditions with an equivalent set of nonlinear equations. Furthermore, these nonlinear equations will be amenable to Newton-like solution methods if we make an appropriate choice for the function G appearing in the theorem. Here we adopt the simple choice $G(z) = \frac{1}{2}z$. This choice leads to a nonlinear system that is differentiable almost everywhere and, therefore, is suitable for Newton-like methods.

Remark 1. *The factor of $\frac{1}{2}$ in $G(z) = \frac{1}{2}z$ is not strictly necessary; it merely avoids the factor of 2 in the nonlinear equation $|g_i(x) - \mu_i| - g_i(x) - \mu_i$, which evaluates to $2g_i(x)$ or $2\mu_i$, depending on the sign of $g_i(x) - \mu_i$.*

Remark 2. *With $G(z) = \frac{1}{2}z$, the nonlinear function $\frac{1}{2}(|g_i(x) - \mu_i| - g_i(x) - \mu_i)$ is not differentiable along $g_i(x) = \mu_i$. However, if we have strict complementarity at the solution, i.e. $g_i(x^*) - \mu^* \neq 0$, the function is locally differentiable as required by Newton's method. Thus, our approach is in the class of semi-smooth Newton methods [63].*

We conclude this section by summarizing the above results and introducing some notation and definitions to make the subsequent presentation more concise. Let $y \in \mathbb{R}^N$, with $N \equiv n + m_h + m_g$, be the compound vector composed of the primal variables and multipliers:

$$y \equiv \begin{bmatrix} x^T & \lambda^T & \mu^T \end{bmatrix}^T.$$

In addition, let the nonlinear residual function $r : \mathbb{R}^N \to \mathbb{R}^N$ be defined by

$$r(y) \equiv \begin{bmatrix} \nabla L(x, \lambda, \mu) \\ -h(x) \\ \frac{1}{2}\left(|g(x) - \mu| - g(x) - \mu\right) \end{bmatrix}, \tag{2.2}$$

where the Lagrangian, $L(x, \lambda, \mu)$ was defined in Theorem 1, and the absolute value in the last block is to be interpreted componentwise. Using these definitions, Theorems 1 and 2 imply the following result.

Corollary 1. *Let x^* be a local solution of (P), and assume that active constraint gradients are linearly independent. Then there exists multipliers λ^* and μ^* such that $y^* = \begin{bmatrix} x^{*T} & \lambda^{*T} & \mu^{*T} \end{bmatrix}^T$ satisfies*

$$r(y^*) = 0.$$

Our basic approach to finding a local solution to the optimization problem (P) is to solve $r(y) = 0$. The following section describes the multisecant algorithm that we use for this purpose.

2.3 Algorithm Description

2.3.1 Newton's Method and Mutisecant Methods

Before describing our particular algorithm, we briefly review the class of multisecant methods upon which it is based. For a more complete review of multisecant methods see [30].

In this subsection, we consider the generic problem of solving $r(y) = 0$, where $r : \mathbb{R}^N \to \mathbb{R}^N$ is continuously differentiable almost everywhere. In particular, we will not be concerned with avoiding stationary points that are not local minimizers of (P); we will address nonconvexity in 2.3.3.

Multisecant methods are approximations to Newton's method. Newton's method itself is based on the linear approximation[1]

$$r(y_k + \Delta y_k) \approx r(y_k) + \mathsf{J}(y_k)\Delta y_k, \tag{2.3}$$

where y_k is the estimated solution at iteration k, and $\mathsf{J} : \mathbb{R}^N \to \mathbb{R}^{N \times N}$ is the Jacobian of $r(y)$. The Newton step is found by setting the right-hand side above to zero and solving for Δy_k:

$$\Delta y_k = -\mathsf{J}(y_k)^{-1} r_k, \tag{2.4}$$

[1] Subscripts in this section refer to iteration number, not the component of the vector as they did in the previous section.

where $r_k \equiv r(y_k)$. The next iterate of Newton's method is then obtained as $y_{k+1} = y_k + \Delta y_k$, usually with some safe-guards on the step to ensure globalization.

A potential disadvantage of Newton's method is that forming and/or inverting the Jacobian can be expensive. There are several strategies for reducing or avoiding these costs, one being the inexact-Newton-Krylov class of methods [46]; Newton-Krylov methods require only Jacobian-vector products, and therefore avoid the need to form the Jacobian explicitly. Quasi-Newton methods offer an alternative strategy that avoids the need to form the exact Jacobian; they store an approximation to the Jacobian (or its inverse), $\mathsf{J}_k \approx \mathsf{J}(\boldsymbol{y}_k)$, and update this approximation at each iteration using low-rank matrices. Multisecant methods belong to the class of quasi-Newton methods.

Multisecant methods get their name from the secant condition, which is obtained by replacing the approximation in (2.3) with an equality:

$$\mathsf{J}_{k+1}\Delta y_k = \Delta r_k, \tag{2.5}$$

where $\Delta r_k \equiv r_{k+1} - r_k$. Alternatively, when approximating the inverse Jacobian with $\mathsf{G}_{k+1} \approx \mathsf{J}(\boldsymbol{y}_{k+1})^{-1}$, the secant condition becomes

$$\mathsf{G}_{k+1}\Delta r_k = \Delta y_k. \tag{2.6}$$

The secant condition is the basis for Broyden's method for nonlinear equations [16], as well as several popular quasi-Newton methods for optimization, namely DFP [24], BFGS [17, 31, 33, 71], and SR1 [21].

Rather than a single secant condition, Vanderbilt and Louie [74] and Eyert [28] proposed generalizations of Broyden's method that require G_k (or J_k) to satisfy a set of q secant equations. If we define

$$\mathsf{Y}_k = \begin{bmatrix}\Delta y_{k-q} & \Delta y_{k-q+1} & \cdots & \Delta y_{k-1}\end{bmatrix}$$
$$\text{and} \qquad \mathsf{R}_k = \begin{bmatrix}\Delta r_{k-q} & \Delta r_{k-q+1} & \cdots & \Delta r_{k-1}\end{bmatrix},$$

then the q previous secant conditions can be written succinctly as

$$\mathsf{G}_k\mathsf{R}_k = \mathsf{Y}_k. \tag{2.7}$$

In general $q < N$, so the q conditions in (2.7) are insufficient to define G_k, and additional conditions are necessary.

In the context of optimization, the Jacobian of the KKT conditions is symmetric, so symmetry provides another condition we might consider imposing on G_k. Indeed, most quasi-Newton methods for optimization that

are based on the single secant condition, (2.5) or (2.6), do impose symmetry on the approximation of the Hessian, KKT matrix, or their inverses. Therefore, it is interesting to consider whether symmetry can be imposed on a multisecant quasi-Newton approximation. Unfortunately, it is easy to show that the answer is negative.

Proposition 1. *A quasi-Newton approximation* G_k *that satisfies the multisecant condition* (2.7) *cannot be symmetric, in general, if* $q > 1$.

Proof. Assume that $\mathsf{G}_k = \mathsf{G}_k^T$. Then, left multiplying (2.7) by R_k^T, we have

$$\mathsf{R}_k^T\mathsf{Y}_k = \mathsf{R}_k^T\mathsf{G}_k\mathsf{R}_k = \mathsf{R}_k^T\mathsf{G}_k^T\mathsf{R}_k = \mathsf{Y}_k^T\mathsf{R}_k.$$

This implies that $\mathsf{R}_k^T\mathsf{Y}_k$ is symmetric, which is only true, in general, if this product is a scalar, *i.e.*, $q = 1$. Therefore, our assumption on the possible symmetry of G_k must be false. □

While we cannot impose symmetry on G_k, we can follow the approach used in the generalized Broyden's method [28]; specifically, G_k is required to be as close as possible, in the Frobenious norm, to some estimate of the inverse Jacobian, which we will denote by $\tilde{\mathsf{G}}_k$. For example, [28] proposes using the previous estimate for the inverse Jacobian, that is $\tilde{\mathsf{G}}_k = \mathsf{G}_{k-q}$. We will discuss other possible choices for $\tilde{\mathsf{G}}_k$ in the next subsection.

The requirement that G_k be as close as possible to $\tilde{\mathsf{G}}_k$, together with the secant conditions (2.7), produces the closed-form expression for G_k provided in the following theorem.

Theorem 3. *The approximate inverse of the Jacobian at iteration* k *of the generalized Broyden's method is*

$$\mathsf{G}_k = \tilde{\mathsf{G}}_k + (\mathsf{Y}_k - \tilde{\mathsf{G}}_k\mathsf{R}_k)(\mathsf{R}_k^T\mathsf{R}_k)^{-1}\mathsf{R}_k^T, \tag{2.8}$$

which is the solution of

$$\min_{\mathsf{G}\in\mathbb{R}^{N\times N}} \quad \|\mathsf{G} - \tilde{\mathsf{G}}_k\|_F, \qquad s.t \quad \mathsf{G}\mathsf{R}_k = \mathsf{Y}_k,$$

where $\|\cdot\|_F$ *denotes the Frobenious norm.*

Proof. The proof of this result can be found in [28] and [30]. □

To summarize, a multisecant method makes the approximation $\mathsf{G}_k \approx \mathsf{J}(y_k)^{-1}$ in the Newton update (2.4), where G_k is defined by (2.8). Thus, the next iterate in a multisecant method is given by

$$y_{k+1} = y_k - \left[\tilde{\mathsf{G}}_k + (\mathsf{Y}_k - \tilde{\mathsf{G}}_k\mathsf{R}_k)(\mathsf{R}_k^T\mathsf{R}_k)^{-1}\mathsf{R}_k^T\right] r(y_k). \tag{2.9}$$

2.3.2 Choosing $\tilde{\mathsf{G}}_k$

In order to complete the definition of the multisecant update we must specify a choice for $\tilde{\mathsf{G}}_k$. The simplest choice for $\tilde{\mathsf{G}}_k$ is the scaled identity:

$$\tilde{\mathsf{G}}_k = \alpha \mathsf{I}, \tag{2.10}$$

where $\alpha > 0$. This choice makes the update (2.9) equivalent to Anderson-mixing [28]. Furthermore, in the case of unconstrained optimization with no secant conditions on G_k (*i.e.* $q = 0$), the multisecant update reduces to steepest descent, and the parameter α determines the step length via $\|\Delta y_k\| = \alpha \|r(y_k)\|$.

Even when secant conditions are imposed on G_k ($q > 0$), α can (and should) be chosen to influence the step lengths, preventing overly conservative or aggressive updates; however, when $q > 0$, there is no guarantee that the step at iteration k will have length $\alpha\|r(y_k)\|$. This is because G_k must respect the secant conditions, whereas $\|\mathsf{G}_k - \alpha\mathsf{I}\|$ is only minimized; see Theorem 3. This is potentially beneficially, because it is possible for the algorithm to overcome a poor choice for α if q is sufficiently large.

We have found that using the scaled identity for $\tilde{\mathsf{G}}_k$ is adequate for reasonably well-scaled unconstrained problems, as well as constrained problems with modest numbers of active constraints (fewer than 10); however, for ill-conditioned problems, a more sophisticated choice is necessary. Ideally, we would use $\tilde{\mathsf{G}}_k = \mathsf{J}(y_k)^{-1}$. While this choice is not practical, it does suggest a class of options for $\tilde{\mathsf{G}}_k$: preconditioners.

Consider the iterative solution of the Newton update (2.4) using a Krylov method. For Krylov methods, the number of iterations is related to the condition number of the linear system [68], so preconditioners are employed to cluster the eigenvalues and/or reduce the conditioning of the system. Preconditioners are usually based on approximations to $\mathsf{J}(y_k)^{-1}$, which is precisely what we want for $\tilde{\mathsf{G}}_k$. Thus, any suitable preconditioner designed for the Krylov-iterative solution of (2.4) can be adopted for $\tilde{\mathsf{G}}_k$. We will explore this possibility in the numerical results.

Remark 3. *The connection between Krylov and multisecant methods goes beyond their mutual need for preconditioning. For instance, Walker and Ni [76] have shown that the iterates of Anderson acceleration can be obtained from those of GMRES [69], and vice versa, when Anderson acceleration is used to solve linear systems.*

Remark 4. $\tilde{\mathsf{G}}_k$ *does not need to be the same matrix at every iteration of the multisecant method, that is, nonstationary preconditioners are permitted. In*

this regard, multisecant methods are similar to flexible Krylov iterate methods like FGMRES [67].

As in the simple case $\tilde{\mathsf{G}}_k = \alpha \mathsf{I}$, scaling the chosen preconditioner by α can help control the step size in the early stages of the multisecant method. If we let P_k^{-1} denote a generic preconditioner, then the choice

$$\tilde{\mathsf{G}}_k = \alpha \mathsf{P}_k^{-1},$$

with $\alpha > 0$, encapsulates both the scaled diagonal ($\mathsf{P}_k^{-1} = \mathsf{I}$) as well as more elaborate preconditioners. This is the form for $\tilde{\mathsf{G}}_k$ we will use throughout the remaining chapter.

2.3.3 Handling Nonconvex Problems

Recall that the KKT conditions (2.1), as well as the equivalent conditions $r(y) = 0$, are *necessary*, but not sufficient, conditions for a solution to (P). Other stationary points, including local maximizers and saddle points, also satisfy $r(y) = 0$. Newton's method has no way to distinguish between these different types of stationary points, and it can easily converge to the wrong type. A multisecant method, being based on Newton's method, will suffer the same fate if it is not safe-guarded.

Before we describe how we handle nonconvexity, we first review common strategies used by existing algorithms, and explain why these are not suitable for a multisecant method.

- Quasi-Newton methods like BFGS are updated in such a way that the approximate inverse of the Lagrangian Hessian remains positive definite. Consequently, they are guaranteed to produce a descent direction. Unfortunately, multisecant methods cannot produce a symmetric Hessian, in general, let alone one that is positive definite; see Theorem 1.

- For unconstrained problems, a step direction can easily be checked to see if it is a descent direction. If the step is not a descent direction, it can be discarded and we can resort to a steepest-descent step, for example. This approach is viable for our algorithm, but it is limited to unconstrained problems.

- Many optimization algorithms for constrained problems require the user to provide the constraint Jacobian, which can then be factored to determine a basis for its null-space. Using this basis, an algorithm can

project the problem onto a reduced-space, effectively turning it into a unconstrained problem. In this reduced space, a step can be checked to see if it is a descent direction, analogous to the unconstrained case. This approach is not possible for multisecant methods, because the Jacobian is not explicitly available.

Instead of the above methods, we use a simple Hessian-regularization approach to address nonconvexity. In the context of the Newton update (2.4), undesirable steps caused by indefinite Hessians[2] can be prevented by adding a scaled identity, $\beta \mathsf{I}$, to the Hessian of the Lagrangian, provided $\beta > 0$ is larger than the most negative eigenvalue of the projected Hessian.

While we do not have direct access to the Hessian of the Lagrangian in the multisecant update, the effect of adding $\beta \mathsf{I}$ to the Hessian can be mimicked by modifying the difference vectors Δr_k as follows:

$$\Delta r_k \equiv r(y_{k+1}) - r(y_k) + \begin{bmatrix} \beta(x_{k+1} - x_k) \\ 0 \\ 0 \end{bmatrix}.$$

There are two drawbacks to Hessian regularization. First, it will limit the asymptotic rate of convergence to linear, and, second, the ideal value for β requires an estimate of the negative eigenvalue of greatest magnitude. In our experience, the impact on the asymptotic rate of convergence is of minor practical concern: superlinear convergence of the unregularized method is typically limited to the last two or three iterates.

The estimate of β is a more serious concern. In this work we have used trial and error to determine a suitable value for β. A more methodical approach would be to use a few iterates of the Lanzcos method [47] applied to the Hessian of L after convergence, in order to estimate the negative eigenvalues of largest magnitude, if any. The Lanzcos method is attractive here, because it requires only Hessian-vector products, which can be approximated in a matrix-free manner using a forward difference applied to the Lagrangian gradient (∇L is already required by our algorithm). If negative eigenvalues are found, the algorithm can be restarted with the value of β set appropriately. However, this posterior Lanzcos approach may be overly conservative, since the only negative eigenvalues of significance are those in the null-space of the constraint Jacobian.

[2] Specifically, Hessians that are not positive-definite in the null-space of the linearized constraints.

2.3.4 The Multisecant Accelerated Descent (MAD) Algorithm

Our proposed optimization method, Multisecant Accelerated Descent (MAD), is summarized in Algorithm 2. There are several implementation details that are important to highlight.

- We assume that the initial multiplier values are zero; see Line 1.
- There are different criteria that can be used to assess convergence of the first-order optimality conditions in Line 3. In our implementation, we accept the solution if relative and absolute tolerances on primal optimality and feasibility are met, specifically

$$\begin{aligned} &\|\nabla f(x_k) - (\nabla h(x_k)^T)\lambda_k - (\nabla g(x_k)^T)\mu_k\| \leq \epsilon_r \|\nabla f(x_0)\| + \epsilon_a \\ &\left\| \begin{matrix} h(x_k) \\ \frac{1}{2}(|g(x_k) - \lambda_k| - g(x_k) - \lambda_k \end{matrix} \right\| \leq \epsilon_r \left\| \begin{matrix} h(x_0) \\ \frac{1}{2}(|g(x_0) - \lambda_0| - g(x_0) - \lambda_0 \end{matrix} \right\| + \epsilon_a, \end{aligned} \tag{2.11}$$

  where $\epsilon_r \in (0, 1)$ and $\epsilon_a > 0$ are relative and absolute tolerances, respectively. We use the same tolerances for both primal optimality and feasibility in this work, because our problems are relatively well scaled; in general, different tolerances may be necessary for the two criteria.
- For problems with noisy/inaccurate data, the computational budget defined by the maximum number of iterations, $K_{\max}$, may be exceeded before the convergence criteria in Line 3 are met.
- The least-squares subproblem on Line 10 corresponds to the vector

$$(\mathsf{R}_k^T \mathsf{R}_k)^{-1} \mathsf{R}_k^T r_k = \gamma,$$

  seen in the multisecant update (2.9). The above expression is the normal-equation solution to the overdetermined problem $\mathsf{R}_k \gamma = r_k$. While the normal-equation solution is convenient theoretically, it is not advisable in practice [30]. This is because the columns of R_k can become close to linearly dependent, leading to ill-conditioning in $\mathsf{R}_k^T \mathsf{R}_k$. A common solution to this possible ill-conditioning, and the one adopted here, is to use a truncated singular-value decomposition. In particular, we truncate singular values that are smaller than 10^{-6} relative to the largest singular value.

- After evaluating the full multisecant step in Line 11, we check its magnitude in Line 12 and limit the step to a maximum length of $\Delta_{\max}$, if necessary.

Algorithm 2: Multisecant Accelerated Descent.

Data: x_0, $\alpha > 0$, $\beta > 0$, $\Delta_{\max}$, $q \geq 0$, $K_{\max} \geq 0$, and operator P_k^{-1}
Result: y, a approximate solution to $r(y) = 0$

1 set $y_0 = \begin{bmatrix} x_0^T & 0^T & 0^T \end{bmatrix}^T$ and compute and store $r_0 = r(y_0)$
2 **for** $k = 0, 1, 2, \ldots, K_{\max}$ **do**
3 **if** $\|r_k\|$ *is sufficiently small* **then**
4 return y_k
5 **end**
6 **for** $j = 1, 2, \ldots, \min(k, q)$ **do**
7 $\Delta y_{k-j} \leftarrow y_{k-j+1} - y_{k-j}$
8 $\Delta r_{k-j} \leftarrow r_{k-j+1} - r_{k-j} + \beta \begin{bmatrix} \Delta x_{k-j} \\ 0 \\ 0 \end{bmatrix}$
9 **end**
10 solve $\operatorname{argmin}_\gamma \|r_k - \mathsf{R}_k\gamma\|$
11 $\Delta y_k = -\alpha \mathsf{P}_k^{-1} r_k - (\mathsf{Y}_k - \alpha \mathsf{P}_k^{-1} \mathsf{R}_k)\gamma$
12 **if** $\|\Delta y_k\| > \Delta_{\max}$ **then**
13 $\Delta y_k \leftarrow \frac{\Delta_{\max}}{\|\Delta y_k\|} \Delta y_k$
14 **end**
15 $y_{k+1} = y_k + \Delta y_k$
16 $r_{k+1} = r(y_{k+1})$
17 **end**

Other than the rudimentary step-length safeguard in Line 12, our algorithm has no globalization strategies, such as line-search or trust-region methods. This is unusual and demands some justification.

When the application has noisy or inaccurate data[3], it is difficult to distinguish a good step from a poor step. Consider an unconstrained optimization algorithm in which a sufficient-decrease line search is implemented, and suppose a step causes the objective function to violate the sufficient-decrease condition. Did this violation happen because the step was poor and the "true" value of f increased, or did this happen because of inaccurate

[3]Data here refers to $r(y)$ and $f(x)$.

data? If we know enough about the nature of the error in the data, we may be able to answer this question; however, in general, the behavior of the error will be unknown and globalization methods will be unreliable.

When the data is accurate, our justification is more pragmatic: the method seems to more efficient without globalization. One possible explanation is that multisecant methods may, sometimes, benefit from "bad" steps, since these steps provide information about the curvature of the problem. Furthermore, recall that α does provide some control over the step-length size, so it acts as a kind of implicit globalization. That said, a more rigorous and efficient globalization for error-free problems may be possible with further analysis and investigation.

Remark 5. *We are not the first to forego standard globalization techniques when using a multisecant method. Fang and Saad [30] also avoid explicit globalization when using multisecant methods to solve electronic structure calculations. Their motivation is to minimize unnecessary evaluations of the self-consistent field iteration, which is expensive. Our target application, PDE-constrained optimization, also leads to expensive function evaluations, so we are also sensitive to the additional evaluations required by globalization methods.*

2.4 Numerical Experiments

2.4.1 Multidisciplinary Design Optimization Problem

In the following sets of experiments, we consider a model multidisciplinary design optimization (MDO) problem. The problem consists of finding a nozzle geometry such that the quasi-one-dimensional Euler equations produce a pressure that is as close as possible to a target pressure. The nozzle itself can (statically) deform under the pressure loading, and this deformation is modeled using a one-dimensional finite-element beam. The aerostructural optimization problem is described in detail in Reference [25]; below we provide a brief description as needed for the present study.

The MDO problem is posed using the individual-discipline feasible (IDF) formulation [36, 22]. The IDF formulation introduces additional optimization variables, called coupling variables, that allow the disciplinary state equations to be solved independently at each optimization iteration. This helps maintain modularity and also avoids coupled multidisciplinary analyses and coupled adjoints; however, the IDF formulation makes the optimization problem more difficult, because there are many more variables,

and it introduces state-based constraints whose Jacobian is expensive to evaluate. This makes matrix-free optimization methods attractive for this type of problem.

For the elastic-nozzle problem, the IDF optimization statement is

$$\begin{aligned} \underset{b,\bar{p},\bar{u}_y}{\text{minimize}} \quad & f(b,\bar{p},\bar{u}_y), \\ \text{s.t.} \quad & p(b,\bar{u}_y) - \bar{p} = 0, \\ & u_y(b,\bar{p}) - \bar{u}_y = 0. \end{aligned} \tag{IDF}$$

The objective function is a discretization of $\int (p - p_t)^2 \, dx$, where p is the pressure and p_t is the target pressure; again, see [25] for the details. The optimization variables consist of 1) the b-spline control points, b, that define the unloaded shape of the nozzle, 2) the pressure coupling variables, $\bar{p}$, that are used by the structural model to define the loading, and 3) the nozzle vertical-displacement coupling variables, $\bar{u}_y$, that are used by the flow model to define the nozzle shape (static + displacement). For a valid solution to (IDF), the coupling variables $\bar{p}$ and $\bar{u}_y$ must agree with the values of pressure and vertical displacement, respectively, predicted by the analyses. This requirement is expressed by the (vector) equality constraints in (IDF).

In order to evaluate the objective and constraints in (IDF) at the kth optimization iteration, we must first solve the disciplinary state equations based on the given values of b_k, $\bar{p}_k$, and $(\bar{u}_y)_k$. Furthermore, gradients of the objective and constraints require the solution of adjoint equations. In this work, both the state and adjoint equations are solved iteratively, the former with a Newton solver and the latter with a preconditioned Krylov method. We will use the iterative solvers' tolerances to control the accuracy of the state and adjoint solutions when we study the impact of inaccurate data on our algorithm.

We benchmark the MAD solution of (IDF) against a previously developed inexact-Newton-Krylov algorithm [38, 27]. Dener and Hicken [26, 25] recently developed a specialized preconditioner that takes advantage of the structure in the IDF formulation. This preconditioner is used in Algorithm 2 for all solutions of (IDF), unless stated otherwise.

We use 5 b-spline control points and the mesh has 31 nodes, so $\bar{p}, \bar{u}_y \in \mathbb{R}^{31}$ and $m_h = 2 \times 31 = 62$; including the multipliers, this gives a problem with $N = 5 + 4 \times 31 = 129$ variables. For all experiments below, the initial guess for x_0 is the same used in [25]. The relative and absolute tolerances are $\epsilon_r = 10^{-4}$ and $\epsilon_a = 10^{-6}$, respectively. The maximum allowable primal step is $\Delta_{\max} = 1.0$, and the maximum number of iterations is $K_{\max} = 2000$.

Table 2.1: Number of iterations required by MAD on problem (IDF) with accurate data, for different combinations of q and α. The Hessian regularization parameter is fixed at $\beta = 0.5$. An "∞" denotes a run that did not converge in fewer than $K_{\max}$ iterations, and "nan" denotes a run that diverged.

	q				
α	5	10	15	20	25
0.01	1773	∞	432	156	73
0.05	600	562	254	231	263
0.1	277	78	42	35	35
1.0	nan	nan	nan	nan	nan

Parameter Study We begin by investigating, in the context of (IDF), the effect of varying the primary parameters in Algorithm 2. These parameters are 1) the number of saved vectors[4], q, 2) the preconditioner scaling parameter, α, and 3) the Hessian regularization parameter β.

Table 2.1 lists the number of iterations used by MAD for a range of q and α values, with $\beta = 0.5$. As q increases for fixed α, the number of iterations generally decreases; the only exception to this is $q = 10$ and $\alpha = 0.01$, which did not converge in $K_{\max} = 2000$ iterations. Intuitively, increasing the number of secant conditions that are satisfied should increase the accuracy of the approximate inverse G_k and, therefore, reduce the number of iterations, at least when sufficiently close to the solution that the linear approximation to $r(y)$ is accurate.

Increasing α also improves performance, up to a point, although the trends are less consistent. Recall that α influences the step length, particularly during the first few iterations. Consequently, relatively large α may lead to aggressive steps; indeed, increasing α beyond $\alpha = 0.1$ leads MAD to diverge on problem (IDF). Conversely, small values of α can lead to conservative steps and many iterations. These general trends are reflected in the data.

The results of varying β and q are listed in Table 2.2. For all runs in the table, α was held fixed at 0.1. As before, we see improved performance as q increases. However, the result of increasing β is not as expected. Increasing the Hessian regularization should reduce the effective step size, increasing the number of iterations and improving robustness. Instead, the method

[4]Alternatively, $q - 1$ is the number of columns in R_k and Y_k, once $k \geq q$.

Table 2.2: Number of iterations required by MAD on problem (IDF) with accurate data, for different combinations of q and β. The scaling parameter is fixed at $\alpha = 0.1$. The symbol "nan" denotes a run that diverged.

	q				
β	5	10	15	20	25
0.0	224	571	111	45	50
0.1	329	254	156	80	138
0.5	277	78	42	35	35
1.0	354	414	179	114	70
5.0	nan	nan	nan	nan	nan

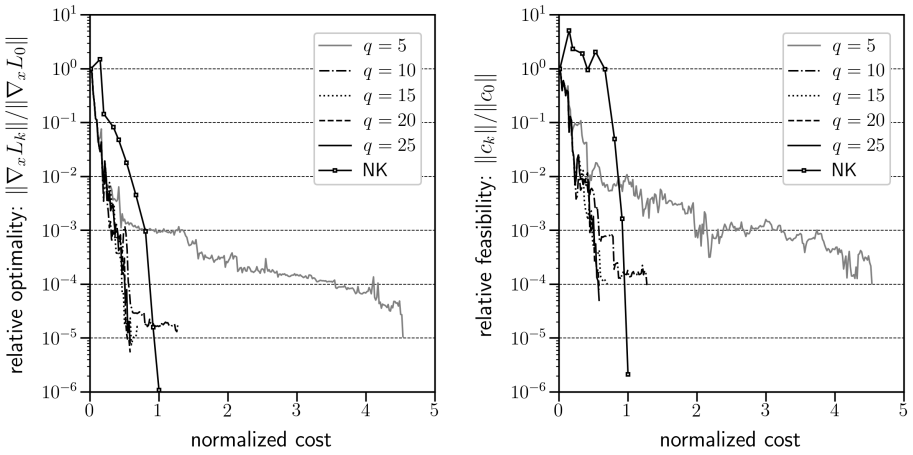
becomes unstable and diverges as β is increased beyond a value of one. For the values considered, $\beta = 0.5$ appears to be optimal, but it is not clear why. Further investigation into the role of β is necessary.

Performance Using Accurate Data Figure 2.1 shows the convergence histories of the MAD algorithm applied to the multidisciplinary design problem (IDF). Specifically, the plots show the optimality and feasibility norms on the left side of (2.11), normalized by their initial values. Histories are plotted for a range of q values from 5 to 25, with fixed values of $\alpha = 0.1$ and $\beta = 0.5$. The convergence history produced by the Newton-Krylov (NK) algorithm [25] is included for comparison. The abcissa is the computational cost normalized by the cost of the NK method.

The results in Figure 2.1 are based on tightly converged state and adjoint residuals with relative tolerances of 10^{-10} and 10^{-6}, respectively. Consequently, the data in this case is sufficiently accurate that conventional optimization methods, like the NK algorithm, will not experience issues with globalization.

The results show that, for this particular case with accurate data, the MAD algorithm is competitive with the NK algorithm, especially for values of q larger than 15. Indeed, for $q \geq 15$ the asymptotic convergence rate appears similar to that of the inexact-Newton method, *i.e.* superlinear.

Performance Using Inaccurate Data Next, we repeat the experiments described above, but we loosen the tolerances on the state and adjoint residuals to 10^{-3} and 10^{-2}, respectively. The convergence histories in this case are plotted in Figure 2.2.



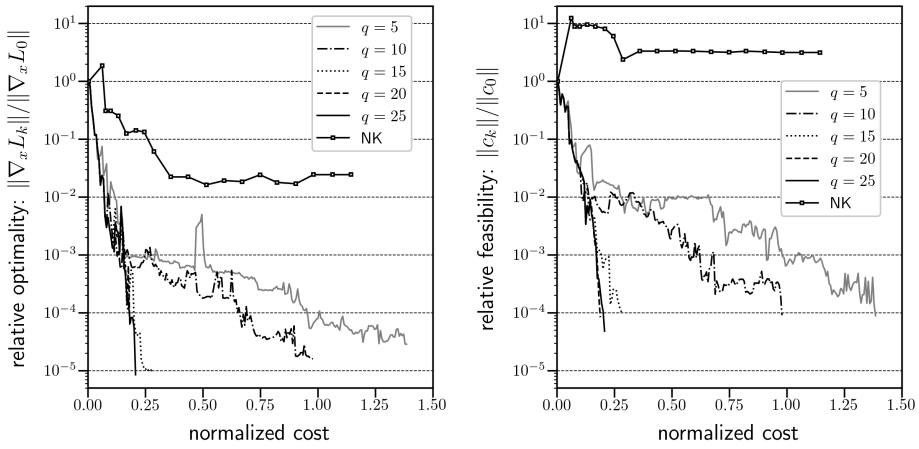
(a) relative optimality

(b) relative feasibility

Figure 2.1: Convergence histories of MAD, for a range of q values, applied to (IDF) with accurate data. All variants use $\alpha = 0.1$ and $\beta = 0.5$. Results from a Newton-Krylov algorithm are included for reference, and the cost is normalized using the cost of the Newton-Krylov optimization.

With the loose tolerances on the state and adjoint residuals, the objective, constraints, and gradients have sufficiently large errors that conventional globalization methods have difficulties. This is reflected in the convergence history of the NK algorithm, which shows that the filter-based globalization stalls. In contrast, the MAD algorithm proceeds without difficulty in the presence of the inaccurate data.

The cost in Figure 2.2 is normalized by the computational cost of the NK algorithm using *accurate data*. Thus, it is possible to compare the accurate-data results in Figure 2.1 with the inaccurate-data results in Figure 2.2. Comparing the figures, we see that the MAD algorithm is significantly faster when using inaccurate data. This is because the MAD runs with and without accurate data use approximately the same number of iterations (for this problem); consequently, the optimization cost is directly proportional to the cost of the state and adjoint solves. Since the state and adjoint solves with loose tolerances are more than twice as fast as those with tight tolerances, this reduction in state/adjoint cost is translated directly into a reduction in MAD optimization cost.



(a) relative optimality

(b) relative feasibility

Figure 2.2: Convergence histories of MAD, for a range of q values, applied to (IDF) with inaccurate data. All variants use $\alpha = 0.1$ and $\beta = 0.5$. Results from a Newton-Krylov algorithm are included for reference, and the cost is normalized using the cost of the Newton-Krylov optimization *with accurate data*.

2.5 Summary and Future Work

Many design optimization problems involve computationally expensive simulations, and while the objective and constraint derivatives may be available, they may be inaccurate or inconsistent. These errors in the data can cause conventional optimization algorithms to fail. To enable the use of inaccurate/noisy data in optimization, we have proposed an error-tolerant optimization algorithm based on a multisecant quasi-Newton framework.

The algorithm solves a set of semi-smooth nonlinear equations that are equivalent to the KKT first-order necessary conditions. By recasting the KKT conditions as nonlinear equations, it is possible to apply multisecant methods directly to the entire set of equations, in contrast to conventional quasi-Newton optimization algorithms that apply the approximation to the Hessian of the Lagrangian only. The algorithm was enhanced by incorporating preconditioning into the multisecant update equation, and a regularization was introduced to handle nonconvex problems.

To demonstrate the method, it was applied to a multidisciplinary design optimization problem that has state-dependent constraints and over a hundred variables. Using optimal parameter values, the multisecant method was found to be competitive with an inexact-Newton-Krylov algorithm when accurate data was provided. When inaccurate data was used, the performance of the multisecant method was virtually unchanged, whereas the Newton-Krylov algorithm stalled.

The numerical experiments suggest that the proposed multisecant algorithm is a promising method for nonlinearly constrained optimization problems, both with and without errors in the data. Nevertheless, some important issues for future research remain. The parameters α and β were found to have a significant impact on the success and performance of the algorithm, yet we do not have an automated means of selecting optimal values for these parameters. Furthermore, the influence of β on regularization is counter-intuitive and demands further investigation. Finally, the algorithm benefits from preconditioning, but there are few effective, general-purpose preconditioners for constrained optimization problems.

Chapter 3

Sensitivity Analysis of Chaotic Problems using a Fourier Approximation of the Least-Squares Adjoint

3.1 Introduction

In this chapter, we describe our efforts to reduce the cost of the Least-Squares Shadowing (LSS) adjoint. Our original, proposed plan was to use the LSS method, or some variant, to estimate the gradients for turbulent channel-flow and airfoil optimization. However, the computational cost of LSS is high, as it requires the solution of a second-order boundary-value problem in time. Liao [49] partially addressed cost by developing a block-based lower-upper factorization of the LSS system. Here, we present a method in which we allow the primal and dual solutions to be of different dimensions. Specifically, we represent the adjoint solution for our model problem using a Fourier sine series, although other choices are possible.

We begin by introducing the model problem for sensitivity analysis and reviewing the conventional adjoint in Section 3.2. The LSS adjoint and our proposed Fourier approximation to it are presented in Section 3.3. Numerical results are presented in Section 3.4.

3.2 Model problem and review of the conventional adjoint

3.2.1 The generic and model problems

Our intent is to find the sensitivity of some objective function, $\mathcal{J}$, to a design variable. The objective function of interest will be governed by a system of equations that exhibits chaotic behavior. We represent the chaotic dynamical system with the generic initial-value problem (IVP)

$$\begin{aligned} \mathcal{R}(u,\nu,t) \equiv \frac{du}{dt} - \mathcal{F}(u,\nu,t) = 0, \qquad & \forall t \in [0,T], \\ u(0) = u_{\mathsf{IC}}, & \end{aligned} \tag{3.1}$$

where $\mathcal{R}$ is the nonlinear residual, and the symbol $\nu \in \mathbb{R}^n$ appearing in $\mathcal{F}$ denotes a control/design variable. One can consider the solution to (3.1) to be a function of both time and the parameter ν, *i.e.* $u = u(t,\nu)$.

Suppose we are interested in minimizing some functional that depends on the solution $u(t,\nu)$. Specifically, consider the time-integrated quantity

$$\mathcal{J}(u,\nu) = \int_0^T \mathcal{G}(u,\nu,t)\, dt, \tag{3.2}$$

where $\mathcal{G}$ is a nonlinear function. In the fluid mechanics applications we are interested in, $\mathcal{J}(u,\nu)$ may be a time-averaged force or moment.

In order to both demonstrate chaotic behavior and validate our method, we select the Lorenz system for investigation, which is defined by

$$\mathcal{F} = \begin{pmatrix} \sigma(y-x) \\ x(\rho - z) - y \\ xy - \beta z \end{pmatrix}; \tag{3.3}$$

the state variables of (3.3) are $u = [x, y, z]^T$, and the parameters are $\nu = [\sigma, \rho, \beta]^T$. The system has initial conditions $\boldsymbol{x}(0) = \boldsymbol{x}_0$.

The objective function selected for the Lorenz model problem is

$$\mathcal{J} = \frac{1}{T}\int_0^T z\, dt, \tag{3.4}$$

as it is well studied and behaves in a consistent manner — the gradient $\partial\mathcal{J}/\partial\rho$ remains relatively constant at a value of 0.96 for $\rho \in [0, 100]$ [48]. Note that (3.4) is in the form (3.2) with $\mathcal{G}(u,\nu,t) = z/T$.

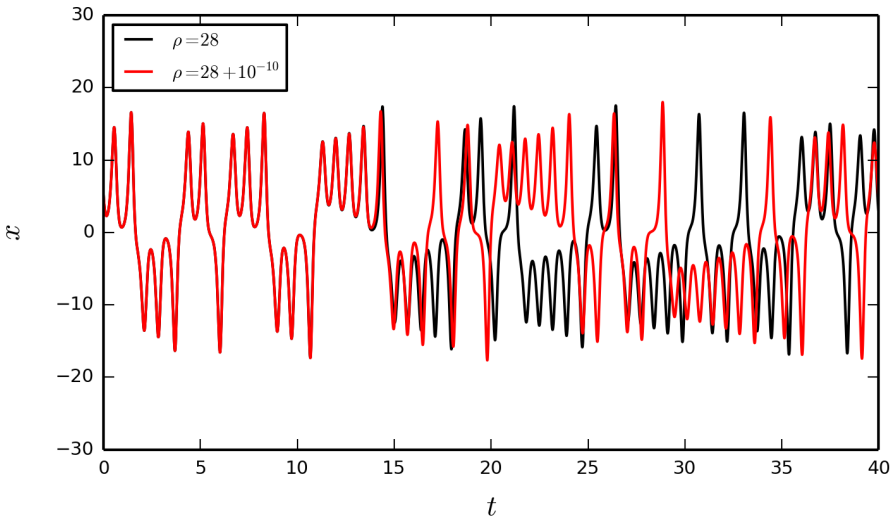


Figure 3.1: A demonstration of the characteristics of the Lorenz system. Two solutions of the Lorenz system are shown, with the x state variable plotted against time t. Despite being solved with almost identical values of the design variable ρ, the trajectories rapidly diverge. β and σ were set to their typical values of 8/3 and 10, respectively.

The sensitive dependence of chaotic systems to initial conditions is fairly well known. What is less widely known is that these systems can be sensitive to changes in other parameters. Figure 3.1 illustrates this sensitivity. In both runs displayed, the initial conditions and the parameters β and σ were fixed, while the ρ parameters differ by 10^{-10}. After only 15 units of time, the trajectories are significantly different.

3.2.2 The conventional adjoint

When there are many inputs and few outputs, the *de facto* sensitivity analysis method in PDE-governed gradient-based optimization is the standard adjoint method. We are interested in such a scenario — although our model problem only has three inputs (ρ, σ, and β) — as we seek a method that will work in aerospace applications that have hundreds or even thousands of design variables. We now review the conventional adjoint for completeness and demonstrate it on the Lorenz problem.

In order to minimize $\mathcal{J}(u, \nu)$ with respect to ν using efficient gradient-based methods, we need the gradient $\nabla_\nu \mathcal{J}$, which is a total derivative. In the case of the functional defined above, $\mathcal{J}$ may depend directly on ν, via its appearance in $\mathcal{G}(\cdot, \nu, \cdot)$, or indirectly through $u(t, \nu)$. In order to account for the indirect dependence, we introduce the Lagrangian

$$\mathcal{L}(u, \psi, \nu) = \mathcal{J}(u, \nu) + \int_0^T \psi^T \mathcal{R}(u, \nu, t)\, dt + \left[\psi^T (u - u_{\mathsf{IC}})\right]_{t=0}, \tag{3.5}$$

where ψ denotes the adjoint. As long as $\mathcal{R}(u, \nu, t) = 0$ and $u(0) = u_{\mathsf{IC}}$, we have that $\mathcal{L} = \mathcal{J}$; however, their partial derivatives are not necessarily equal.

The gradient we want is given by

$$\nabla_\nu \mathcal{J} \equiv \frac{\partial \mathcal{L}}{\partial \nu} = \int_0^T \frac{\partial \mathcal{G}}{\partial \nu}\, dt + \int_0^T \psi^T \frac{\partial \mathcal{R}}{\partial \nu}\, dt. \tag{3.6}$$

However, we cannot compute this without first solving for ψ. To find an equation for ψ, we take the first variation of $\mathcal{L}$ with respect to u and set the

result to zero:

$$\begin{aligned}\delta\mathcal{L} &= \int_0^T \mathcal{G}(\delta u, \nu)\, dt + \int_0^T \psi^T \mathcal{R}(\delta u, t; \nu)\, dt + \left[\psi^T \delta u\right]_{t=0} \\ &= \int_0^T \frac{\partial \mathcal{G}}{\partial u}\delta u\, dt + \int_0^T \psi^T \frac{\partial \mathcal{R}}{\partial u}\delta u\, dt + \left[\psi^T \delta u\right]_{t=0} \\ &= \int_0^T \left[\frac{\partial \mathcal{G}}{\partial u}^T - \frac{d\psi}{dt} - \frac{\partial \mathcal{F}}{\partial u}^T \psi\right]^T \delta u\, dt + \left[\psi^T \delta u\right]_{t=T} \\ &= 0,\end{aligned}$$

where (3.1) and integration by parts (in time) were used to arrive at the second last line. Since $\delta\mathcal{L} = 0$ must be true for all variations δu, we conclude that

$$\begin{aligned} -\frac{d\psi}{dt} - \frac{\partial \mathcal{F}}{\partial u}^T \psi + \frac{\partial G}{\partial u}^T &= 0, \qquad \forall t \in [0, T], \\ \psi(T) &= 0. \end{aligned} \tag{3.7}$$

The differential equation (3.7) is the conventional adjoint equation for the objective (3.2) and the forward IVP (3.1).

In a chaotic problem, however, the conventional adjoint grows unbounded (backward) in time, making the derivative information unusable. This unbounded growth is related to the sensitive dependence of the state on the initial condition. Or, stated another way, the time-integrated objective function $\mathcal{J}$ looks noisy for finite T, and the adjoint-based gradient accurately reflects this noise, since the derivative (and the adjoint) of a noisy function is large.

Returning to our model problem of the Lorenz system (3.3) and our selected objective function (3.4), we can apply the conventional adjoint approach in an attempt to obtain gradient information. Plotted in Figure 3.2 are discrete solutions of the objective function for a range of ρ, along with its gradient with respect to ρ, calculated using the conventional adjoint method. Clearly the adjoint-based derivative does not capture the large-scale trend in the objective function and instead reflects the noise.

3.3 Least-squares shadowing and its Fourier approximation

3.3.1 Least-squares shadowing adjoint

To overcome the instability of the conventional adjoint, the least-squares shadowing (LSS) adjoint method modifies the Lagrangian (3.5) in a cou-

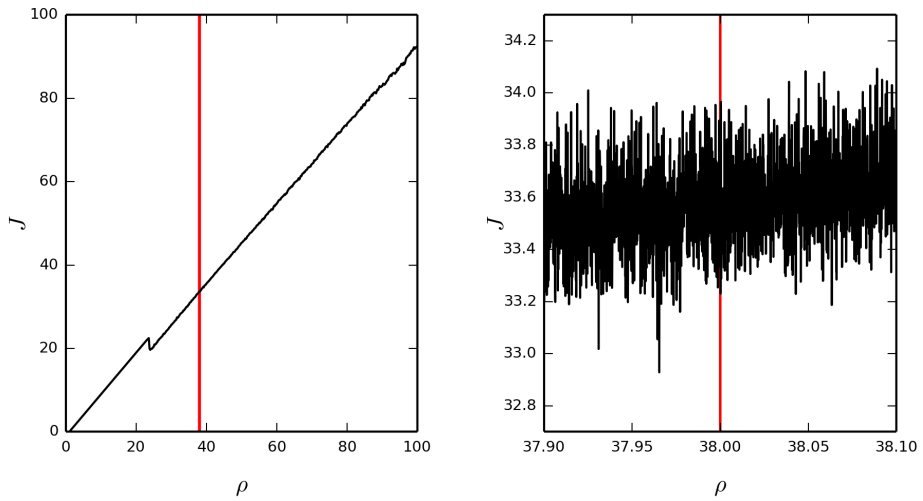


Figure 3.2: A plot of the objective function $\mathcal{J} = 1/T \int_0^T z$ versus ρ. The left plot shows $\mathcal{J}$ over a range of $\rho \in [0, 100]$, while the right plot shows J over a range of $\rho \in [37.9, 38.1]$. Overlaid in red is a representative example of the gradient calculated with the conventional adjoint. Immediately visible is the problematic nature of these gradients.

ple ways. In order to describe the LSS adjoint, it is helpful to show how the conventional adjoint is related to a fully-constrained optimization problem and associated saddle-point problem. Taking the Lagrangian (3.5) and linearizing about $\bar{u}$ using perturbations u', we obtain

$$\mathcal{L}(\bar{u}, u', \psi, \nu) = \mathcal{J}'(u') + \int_0^T \psi^T \mathcal{R}'(u', t)\, dt + \psi^T u', \tag{3.8}$$

where $\mathcal{R}'$ is given by

$$\mathcal{R}'(u', t) \equiv \left[\frac{d}{dt} - \left.\frac{\partial \mathcal{F}}{\partial u}\right|_{\bar{u}}\right] u' = 0, \tag{3.9}$$

and $\mathcal{J}'$ is defined as

$$\mathcal{J}'(u') = \int_0^T \mathcal{G}(\bar{u})\, dt + \int_0^T \frac{\partial \mathcal{G}}{\partial u} u'\, dt.$$

If we take the derivative of $\mathcal{L}$ in (3.8) with respect to u', we obtain the standard adjoint.

In the above definition, $\bar{u}$ is the frozen, or reference, state: it is the state about which we want to evaluate the gradient. The perturbation to the state is the function $u'(t)$. To simplify notation, the dependence of various quantities on ν is not shown in the following derivation.

The issue with the conventional adjoint is that the differential equation (DE) associated with $\mathcal{R}'(u', t)$, (3.9), is linearly unstable. To regularize this differential equation, the LSS formulation requires the perturbation, u', to satisfy the linearized differential equation without the initial condition. However, with the initial condition removed, the linearized differential equation is also ill-posed; this necessitates the addition of regularization.

The regularization ensures that, of all possible perturbations to the state, u' is the one that i) satisfies the linearized differential equation and ii) minimizes the composite objective consisting of the linearized objective and the L^2 norm of u'. Thus, u' is the solution to the DE-constrained optimization problem

$$\begin{aligned} &\min_{u'} \ \mathcal{J}'(u') + \frac{1}{2}\|u'\|^2 \\ \text{s.t.} \qquad &\mathcal{R}'(u', t) \equiv \left[\frac{d}{dt} - \left.\frac{\partial \mathcal{F}}{\partial u}\right|_{\bar{u}}\right] u' = 0, \qquad \forall t \in [0, T], \end{aligned} \tag{3.10}$$

where $\|u'\|^2 = \int_0^T (u')^2\, dt$. Based on the optimization problem (3.10), we can construct the associated Lagrangian:

$$\mathcal{L}'(u', \psi) = \mathcal{J}'(u') + \frac{1}{2}\|u'\|^2 + \int_0^T \psi^T \mathcal{R}'(u', t)\, dt. \tag{3.11}$$

The LSS adjoint equation can be found by taking the first-variation of $\mathcal{L}'$ with respect to u', which produces the LSS problem

$$\begin{aligned} -\frac{d\psi}{dt} - \frac{\partial \mathcal{F}}{\partial u}^T \psi + \frac{\partial G}{\partial u}^T + u' = 0, \qquad \forall t \in [0,T], \\ \psi(0) = 0, \qquad \text{and} \qquad \psi(T) = 0. \end{aligned} \tag{3.12}$$

This DE is almost identical to the ill-posed one we found earlier, *i. e.* (3.7), except for the presence of u'. We can eliminate u' from the adjoint DE by recalling that $\mathcal{R}'(u',t) = 0$, that is, the perturbation is in the null space of the linearized forward problem. Thus, applying the linear operator corresponding to $\mathcal{R}'$ to the above equation, we get the differential equation

$$\begin{aligned} \left[\frac{d}{dt} - \frac{\partial \mathcal{F}}{\partial u}\right]\left[-\frac{d\psi}{dt} - \frac{\partial \mathcal{F}}{\partial u}^T \psi + \frac{\partial G}{\partial u}^T\right] = 0, \qquad \forall t \in [0,T], \\ \psi(0) = 0, \qquad \text{and} \qquad \psi(T) = 0. \end{aligned} \tag{3.13}$$

Equation (3.13) is a linear boundary-value problem (BVP) in time; therefore, two boundary conditions (in time) are necessary, and the conditions at $t = 0$ and $t = T$ are consistent with this requirement.

For completeness, we expand the BVP fully to obtain

$$-\frac{d^2\psi}{dt^2} - \frac{d}{dt}\left[\frac{\partial \mathcal{F}}{\partial u}^T \psi\right] + \frac{\partial \mathcal{F}}{\partial u}\frac{d\psi}{dt} + \frac{\partial \mathcal{F}}{\partial u}\frac{\partial \mathcal{F}}{\partial u}^T \psi = -\frac{\partial^2 \mathcal{G}}{\partial t \partial u} + \frac{\partial \mathcal{F}}{\partial u}\frac{\partial \mathcal{G}}{\partial u}^T.$$

Thus, we are dealing with a second-order in time linear BVP problem.

It should also be noted that, in this part of the report, we are considering a specific form of the LSS sensitivity method without time dilation and its associated weighting parameters[1]. Although this may have an effect on the accuracy of the LSS adjoint, our proposed methods remain applicable to the LSS method with the time dilation term included.

3.3.2 Discretization of the LSS adjoint

Primal-Problem Discretization The LSS method does not require a particular discretization method, but in order to make our explanation more concrete, we choose to demonstrate the LSS method using a Crank-Nicolson discretization. We discretize the IVP (3.1) using N uniform time steps; thus

[1] The time-dilation term is included in the results of Part I

$\Delta t = T/N$ and $t_i = (i-1)\Delta t$, where $i = 1, 2, \ldots, N+1$. Including the initial condition, we have to solve

$$\begin{aligned} u_1 &= u_{\mathsf{IC}}, \\ R_i(\boldsymbol{u}, \nu, t) = \frac{u_{i+1} - u_i}{\Delta t} &- \frac{1}{2}\left[\mathcal{F}(u_i, \nu, t_i) + \mathcal{F}(u_{i+1}, \nu, t_{i+1})\right] = 0, \end{aligned} \tag{3.14}$$

$\forall i = 1, 2, \ldots, N$. In addition, we discretize the functional $\mathcal{J}$ using the midpoint rule[2]:

$$J(\boldsymbol{u}) = \sum_{i=1}^{N} \frac{1}{2}\left[\mathcal{G}(u_i, \nu, t_i) + \mathcal{G}(u_{i+1}, \nu, t_{i+1})\right] \Delta t, \tag{3.15}$$

where $\boldsymbol{u} = [u_1, u_2, \ldots, u_{N+1}]^T$ is the discrete solution.

Standard Adjoint Discretization We can also use the midpoint rule for the term $\int_0^T \psi^T \mathcal{R}\, dt$ in the Lagrangian, see (3.5). Thus, the discretization of the Lagrangian becomes

$$\begin{aligned} L(u, \psi, \nu) = &\sum_{i=1}^{N} \frac{1}{2}\left[\mathcal{G}(u_i, \nu, t_i) + \mathcal{G}(u_{i+1}, \nu, t_{i+1})\right] \Delta t \\ &+ \sum_{i=1}^{N} \psi_{i+1}^T \left\{ \frac{u_{i+1} - u_i}{\Delta t} - \frac{1}{2}\left[\mathcal{F}(u_i, \nu, t_i) + \mathcal{F}(u_{i+1}, \nu, t_{i+1})\right] \right\} \Delta t \\ &+ \psi_1^T (u_1 - u_{\mathsf{IC}}). \end{aligned}$$

We obtain the discrete adjoint equation by differentiating L with respect to the u_i and setting each partial derivative to zero. For $i = 1$ we get

$$\frac{\Delta t}{2} \left.\frac{\partial \mathcal{G}}{\partial u}\right|_1 - (\psi_2^T - \psi_1^T) - \psi_2^T \left[\frac{\Delta t}{2} \left.\frac{\partial \mathcal{F}}{\partial u}\right|_1\right] = 0.$$

Transposing and rearranging we have

$$\psi_1 = \psi_2 + \mathrm{O}(\Delta t),$$

which is a first-order accurate extrapolation of ψ_1 based on ψ_2.

Next, differentiating L with respect to u_{N+1} we get

$$\frac{\Delta t}{2} \left.\frac{\partial \mathcal{G}}{\partial u}\right|_{N+1} + \psi_{N+1}^T - \psi_{N+1}^T \left[\frac{\Delta t}{2} \left.\frac{\partial \mathcal{F}}{\partial u}\right|_{N+1}\right] = 0.$$

[2]It should be noted that when fully expanded, Equation (3.15) becomes the trapezoid rule.

Transposing and rearranging gives

$$\psi_{N+1} = \frac{\Delta t}{2}\left[\psi_{N+1}^T \left.\frac{\partial \mathcal{F}}{\partial u}\right|_{N+1} - \left.\frac{\partial \mathcal{G}}{\partial u}\right|_{N+1}\right].$$

As $\Delta t \to 0$, we recover $\psi_{N+1} = 0$. Thus, the above is a first-order approximation to the terminal condition in (3.7).

Finally, differentiating L with respect to u_i, where $i \neq 1$ and $i \neq N+1$, we get

$$\Delta t \left.\frac{\partial \mathcal{G}}{\partial u}\right|_i - \left(\psi_{i+1}^T - \psi_i^T\right) - \frac{\Delta t(\psi_{i+1}^T + \psi_i^T)}{2}\left.\frac{\partial \mathcal{F}}{\partial u}\right|_i = 0.$$

Transposing, dividing by Δt, and rearranging this equation, we get

$$-\frac{\psi_{i+1} - \psi_i}{\Delta t} - \left[\frac{\partial \mathcal{F}}{\partial u}\right]_1^T \frac{(\psi_{i+1} + \psi_i)}{2} + \left[\frac{\partial \mathcal{G}}{\partial u}\right]_i = 0,$$

which is the Crank-Nicholson discretization of the DE in (3.7).

Later, when discussing the discretization of the LSS adjoint, it will be helpful to have a compact representation of the discretization of the conventional discrete adjoint. To this end we write

$$\mathsf{A}^T\boldsymbol{\psi} + \boldsymbol{b} = \mathbf{0},$$

where

$$\boldsymbol{b} = \left[\frac{\Delta t}{2}\left[\frac{\partial \mathcal{G}}{\partial u}\right]_1, \Delta t\left[\frac{\partial \mathcal{G}}{\partial u}\right]_2, \ldots, \Delta t\left[\frac{\partial \mathcal{G}}{\partial u}\right]_N, \frac{\Delta t}{2}\left[\frac{\partial \mathcal{G}}{\partial u}\right]_{N+1}\right]^T,$$

and for the Jacobian we have

$$\mathsf{A} = \begin{bmatrix} \mathsf{I} & & & & \\ \left[-\mathsf{I} - \frac{\Delta t}{2}\left.\frac{\partial \mathcal{F}}{\partial u}\right|_1\right] & \left[\mathsf{I} - \frac{\Delta t}{2}\left.\frac{\partial \mathcal{F}}{\partial u}\right|_2\right] & & & \\ & \left[-\mathsf{I} - \frac{\Delta t}{2}\left.\frac{\partial \mathcal{F}}{\partial u}\right|_2\right] & \left[\mathsf{I} - \frac{\Delta t}{2}\left.\frac{\partial \mathcal{F}}{\partial u}\right|_3\right] & & \\ & & \ddots & \ddots & \\ & & & \left[-\mathsf{I} - \frac{\Delta t}{2}\left.\frac{\partial \mathcal{F}}{\partial u}\right|_N\right] & \left[\mathsf{I} - \frac{\Delta t}{2}\left.\frac{\partial \mathcal{F}}{\partial u}\right|_{N+1}\right] \end{bmatrix}.$$

Once the adjoint $\boldsymbol{\psi}$ is obtained, a discrete version of (3.6) is used to obtain the gradient:

$$\sum_{i=1}^{N} \frac{1}{2}\left(\left.\frac{\partial \mathcal{G}}{\partial \nu}\right|_i + \left.\frac{\partial \mathcal{G}}{\partial \nu}\right|_{i+1}\right)\Delta t + \sum_{i=1}^{N} \psi_{i+1}\left.\frac{\partial R_i}{\partial \nu}\right|_i^T, \tag{3.16}$$

where R_i is defined in (3.14).

Discretization of the LSS Adjoint Referring to (3.10), the optimization problem associated with the discretized LSS adjoint method is

$$\begin{aligned} \min_{\boldsymbol{u}'} \; & J(\bar{\boldsymbol{u}}) + \boldsymbol{b}^T \boldsymbol{u}' + \frac{1}{2} \left(\boldsymbol{u}'\right)^T \mathsf{H} \boldsymbol{u}' \\ \text{s.t.} \quad & \mathsf{A}' \boldsymbol{u}' = \boldsymbol{0}. \end{aligned} \tag{3.17}$$

In Equation (3.17), $J(\bar{\boldsymbol{u}}) + \boldsymbol{b}^T \boldsymbol{u}'$ is the discrete version of the linearized objective function; here, $\bar{\boldsymbol{u}}$ is the discretized reference state and $\boldsymbol{u}'$ is the discretized perturbation. The term $\frac{1}{2} \left(\boldsymbol{u}'\right)^T \mathsf{H} \boldsymbol{u}'$ is the discretized regularization term, where $\mathsf{H} = \Delta t \, \mathsf{diag}\left(1/2, 1, 1, \ldots, 1, 1/2\right)$ holds the quadrature weights corresponding to the trapezoid rule. Again, since the LSS method removes the initial condition from the linearized problem, we remove the first block row from A to give A' in (3.17). To be explicit, A' is defined as

$$\mathsf{A}' = \begin{bmatrix} \left[-\mathsf{I} - \frac{\Delta t}{2} \frac{\partial \mathcal{F}}{\partial u}\Big|_1\right] & \left[\mathsf{I} - \frac{\Delta t}{2} \frac{\partial \mathcal{F}}{\partial u}\Big|_2\right] & & & \\ & \left[-\mathsf{I} - \frac{\Delta t}{2} \frac{\partial \mathcal{F}}{\partial u}\Big|_2\right] & \left[\mathsf{I} - \frac{\Delta t}{2} \frac{\partial \mathcal{F}}{\partial u}\Big|_3\right] & & \\ & & \ddots & \ddots & \\ & & & \left[-\mathsf{I} - \frac{\Delta t}{2} \frac{\partial \mathcal{F}}{\partial u}\Big|_N\right] & \left[\mathsf{I} - \frac{\Delta t}{2} \frac{\partial \mathcal{F}}{\partial u}\Big|_{N+1}\right] \end{bmatrix}$$

The optimization problem (3.17) is a quadratic optimization problem with a convex objective. The solution to such an optimization problem is given by the solution of the following saddle-point linear system:

$$\begin{bmatrix} \mathsf{H} & \left(\mathsf{A}'\right)^T \\ \mathsf{A}' & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{u}' \\ \boldsymbol{\psi} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix}$$

However, since $\mathsf{A}'\boldsymbol{u}' = 0$ (the second row of the above), we can multiply the first block row by $\mathsf{A}'\mathsf{H}^{-1}$ to get a single equation for the discrete LSS:

$$\mathsf{A}'\mathsf{H}^{-1} \left(\mathsf{A}'\right)^T \boldsymbol{\psi} = -\mathsf{A}'\mathsf{H}^{-1}\boldsymbol{b}. \tag{3.18}$$

This is effectively forming the Schur complement of the block system. With the LSS adjoint found, (3.16) is used in the same way as the conventional adjoint to obtain the desired derivative.

3.3.3 A Petrov-Galerkin approach to the LSS method: Fourier approximation of the adjoint

Our general approach is to use a Petrov-Galerkin projection of the linearized state equation in (3.10). That is, we select different solution spaces for the

state and LSS adjoint. In particular, and unlike typical finite-element discretizations, the dimension of the state and adjoint spaces can be different.

To illustrate the more general idea, we approximate the LSS adjoint in the Lorenz system using a truncated Fourier sine series. In the discrete case, this is done by projecting the LSS Jacobian A' (which, again, is A with the first block row removed) onto a series of sine terms. In matrix form, this projection is defined by:

$$\mathsf{A}_{\text{sin}} = \mathsf{F}^T \mathsf{A}', \tag{3.19}$$

where

$$\mathsf{F}^T = \begin{bmatrix} \sin\left(\pi \frac{t_1}{T}\right) & \sin\left(\pi \frac{t_2}{T}\right) & \dots & \sin\left(\pi \frac{t_N}{T}\right) \\ \sin\left(2\pi \frac{t_1}{T}\right) & \sin\left(2\pi \frac{t_2}{T}\right) & \dots & \sin\left(2\pi \frac{t_N}{T}\right) \\ \vdots & \vdots & & \vdots \\ \sin\left(M\pi \frac{t_1}{T}\right) & \sin\left(M\pi \frac{t_2}{T}\right) & \dots & \sin\left(M\pi \frac{t_N}{T}\right) \end{bmatrix}, \tag{3.20}$$

where M is the number of Fourier modes. The approximated Jacobian is then used to compute the adjoint vector in a similar manner to the original LSS method; see (3.18):

$$\left(\mathsf{F}^T \mathsf{A}'\right) \mathsf{H}^{-1} \left(\mathsf{F}^T \mathsf{A}'\right)^T \boldsymbol{\psi}_{\text{sin}} = -\left(\mathsf{F}^T \mathsf{A}'\right) \mathsf{H}^{-1} \boldsymbol{b}, \tag{3.21}$$

which can then be used to obtain values for the gradient.

The process of computing the gradient from the adjoint is similar to that in (3.16), but involves the basis function values to reconstitute the adjoint using

$$\psi_i = \sum_{m=1}^{M} \psi_{\text{sin}} \sin\left(m\pi t_i\right). \tag{3.22}$$

Solving Equation (3.18) for the adjoint is the most computationally intensive aspect of the LSS method, as it requires a matrix solve on the typically quite large A. It is our hypothesis that the Fourier system will be at least an order of magnitude smaller than the standard system, and thus the computation of the adjoint using the Fourier-approximated method (3.21) has the potential to offer significant computational advantages.

3.4 Numerical results

In order to compare the Fourier LSS method with the standard LSS method, both adjoints were computed and compared for simulations of the Lorenz model problem. In all simulations, the integration period was $T = 16$ and

the time step was $\Delta t = 0.004$. Additionally, the solution of the Lorenz system was allowed to undergo a spin-up period of $T = 10$, to ensure that the gathered statistics better reflect the long-time behavior of the system. Typical values for the Lorenz system parameters were used: $\sigma = 10$, $\beta = 8/3$, and $\rho = 28$ (except where stated otherwise).

3.4.1 The standard LSS applied to the Lorenz problem

For future reference, Figure 3.3 shows a sample of LSS-based gradients for the objective function with respect to ρ, over a range of $[0, 100]$. At each value of ρ, a Lorenz simulation was started with an initial condition of $[x, y, z] = [\Delta x, \Delta y, \rho + \Delta z]$, where Δx, Δy, and Δz are normally distributed random variables with a mean of zero and a standard deviation of 0.05. After setting the initial condition and solving the primal problem, the LSS adjoint was computed and used to obtain gradient information. An ensemble of twenty simulations was conducted at each value of ρ. While the sensitivity to changes in initial conditions can be seen, the figure clearly shows the clustering of gradient values around $d\mathcal{J}/d\rho = 0.96$, which is the approximate value of the gradient for the chosen objective [48]. Recall that our LSS implementation does not include the time-dilation term, and we believe this explains the increased spread in the gradient values relative to the literature.

3.4.2 Fourier-approximated LSS

With the implementation of the standard LSS adjoint verified, we can begin investigating the quality of our Fourier approximation method. In all of the following results, the number of Fourier modes was selected as $M = 460$ as a tradeoff between minimizing the relative gradient error and keeping the dimensions of the dense matrix suitably small. A qualitative comparison can be seen in Figure 3.4, in which the LSS-based adjoint and the Fourier-approximated adjoint are plotted in phase space. It is evident from Figure 3.4 that the approximation mirrors the standard LSS quite well; this is also observed by the more quantitative comparison of Figure 3.5, in which the difference in the x-coordinate of the adjoints over time is shown.

The same aggregate analysis shown in Figure 3.3 was conducted for the Fourier-approximated LSS, and these results can be seen in Figure 3.6. Again, the clustering of gradient values about $d\mathcal{J}/d\rho = 0.96$ is seen. The dip in the mean gradient at high values of ρ, compared to the standard LSS gradient sampling, is due to an insufficient number of modes to approximate the dominant dynamics of the problem. It should be noted that as ρ

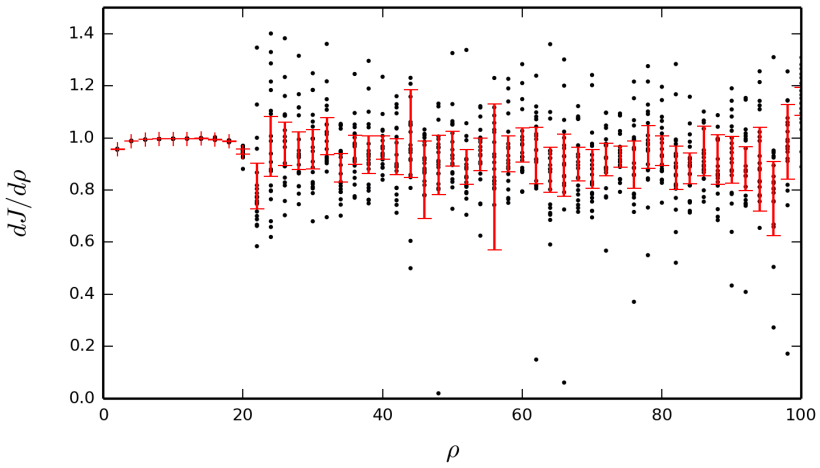


Figure 3.3: A statistical gathering of the gradient of the objective function with respect to ρ, calculated using the standard LSS adjoint. Each dot represents one simulation of the Lorenz system with $T = 16$; there are 20 simulations performed at each value of ρ, with slight randomness given to the initial conditions. Error bars, centered about the mean and with half-length one standard deviation, are shown. The locations in which the error bars appear longer than the dataset contain negative gradients which affect the calculation of the standard deviation.

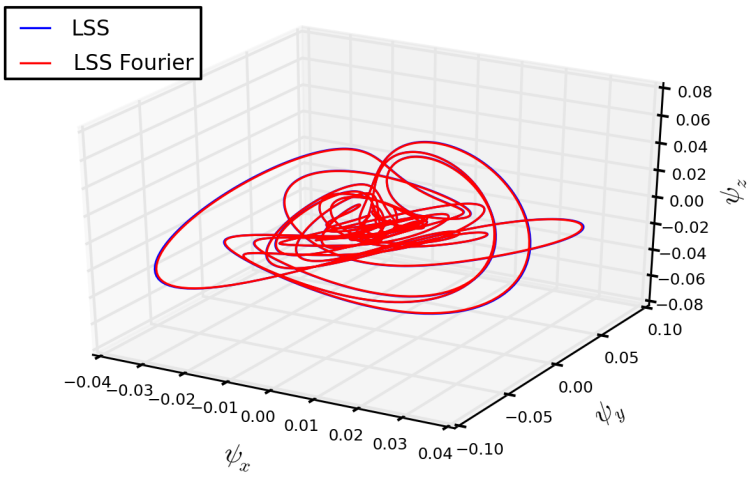


Figure 3.4: The standard LSS adjoint plotted in state space, alongside its Fourier approximation using 460 modes and a solution time of $T = 16$. Qualitatively, the closeness of the approximation can be seen.

increases, the number of modes required to drive the relative gradient error down tends to increase.

A separate numerical experiment, using 800 modes instead of the 460 modes in Figure 3.6, confirmed that high values of ρ require more modes to approximate the LSS adjoint. Additional work is necessary to determine an automated way of selecting the appropriate number of modes in the approximation.

Another verification of the Fourier approximation was conducted by analyzing the relative gradient error between the standard LSS and the Fourier-approximated LSS, defined in Equation (3.23), as a function of the number of sine modes.

$$\epsilon_{dJ/d\rho} = \frac{\left|\left(\frac{dJ}{d\rho}\right)_{\text{LSS}} - \left(\frac{dJ}{d\rho}\right)_{\text{Fourier}}\right|}{\left|\left(\frac{dJ}{d\rho}\right)_{\text{LSS}}\right|}. \tag{3.23}$$

As can be seen in Figure 3.7, the relative gradient error decreases significantly once a sufficient number of modes are used, eventually plateauing on the order of 10^{-3}. This particular study was conducted at a parameter value of $\rho = 28$.

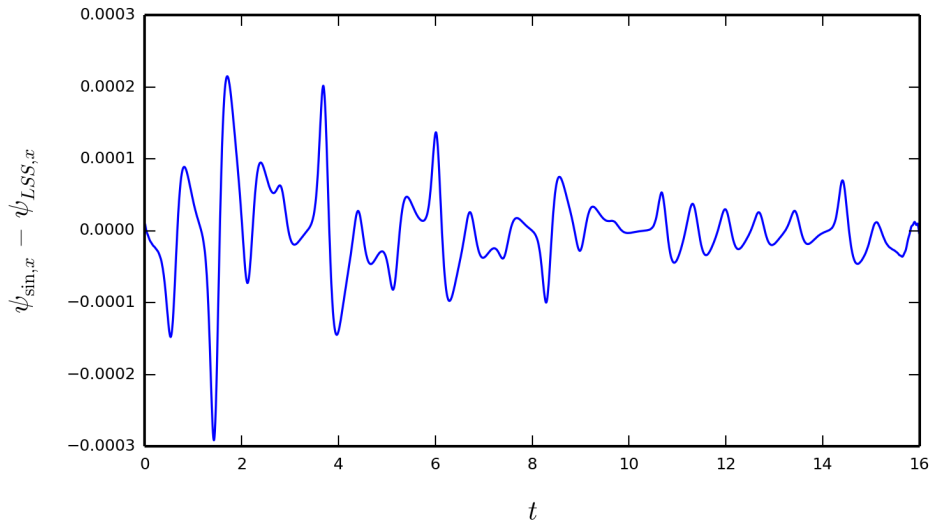


Figure 3.5: The error between the LSS adjoint and the Fourier-approximated LSS adjoint for the component of ψ_x corresponding to the x state variable plotted versus time.

One disadvantage of the Fourier approximation is the condition number of the Schur complement matrix in (3.21). The condition number of the Schur complement is plotted against the number of modes in Figure 3.8 for $\rho = 28$. The condition number becomes high ($\approx 10^7$) as the number of modes is increased, which has a negative effect on accuracy and preconditioning for iterative methods.

A brief discussion of the "plateau" in Figures 3.7 and 3.8 is warranted. At around 460 modes for this selection of ρ, the relative gradient error of the Fourier approximation stops decreasing. The main reason for this is that there will be some error inherent to the standard LSS method, due to the choice in discretization, and that the Fourier approximation is simply trying to capture high-frequency, broadband discretization error. This is supported by the fact that smaller time step values in the discretization drives downward the resting value of the plateau that the relative gradient error reaches, visible in Figure 3.7.

Although this approximation method displays promise when compared to the standard LSS approach, it must be stated that the Fourier system matrix is dense. This could be a significant drawback, and additional work

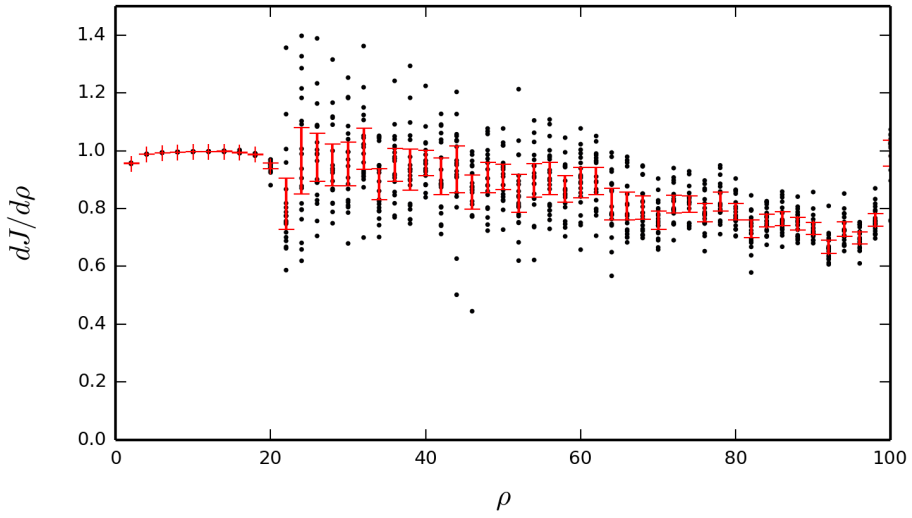


Figure 3.6: A statistical gathering of the gradient of the objective function with respect to ρ, calculated using the Fourier approximation to the LSS adjoint. Each dot represents one simulation of the Lorenz system with $T = 16$; there are 20 simulations performed at each value of ρ, with slight randomness given to the initial conditions. Error bars, centered about the mean and with half-length one standard deviation, are shown.

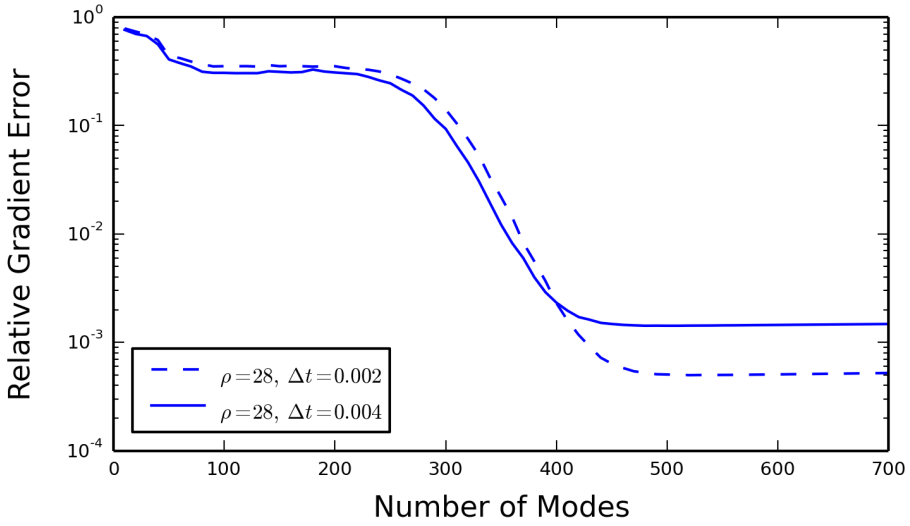


Figure 3.7: Relative error in gradient in the Fourier approximation of the least-squares shadowing adjoint matrix as the number of Fourier modes is varied, for differing values of Δt. Simulation time remains $T = 16$.

is needed to investigate strategies to solve this system in the context of large-scale simulations.

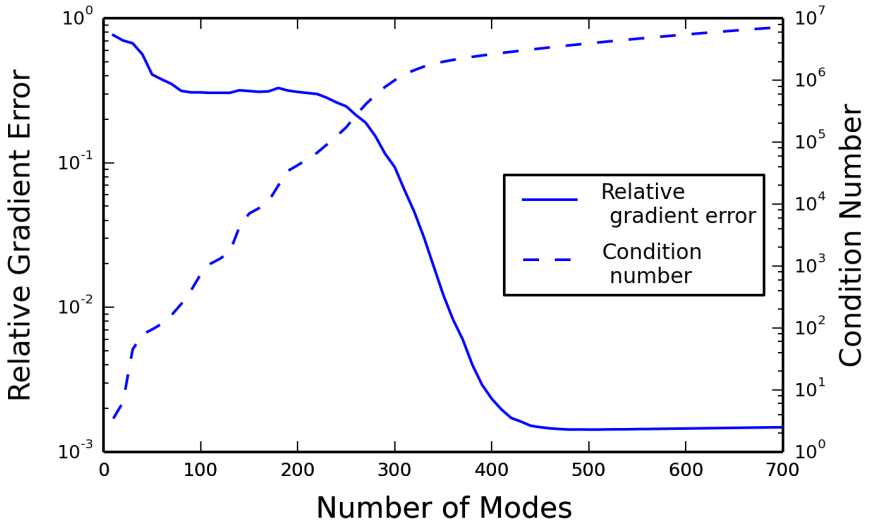


Figure 3.8: Relative error in gradient (blue) and condition number of the approximation of the least-squares shadowing adjoint matrix (red) as the number of Fourier modes is varied. Simulation time $T = 16$, and $\rho = 28$.

Chapter 4

Energy-stable sensitivities for chaotic systems

4.1 Introduction

The potential for simulation-based optimization has been demonstrated on a range of aircraft design problems, from structural optimization [11, 75, 42], to aerodynamic [61, 7, 56, 59, 39] and aerostructural optimization [53, 55, 45, 43]. However, there is an important class of problems that simulation-based optimization has failed to adequately address: problems exhibiting chaotic dynamics. This class of problem cannot be ignored, because it arises in many applications; examples relevant to aircraft design include the flow control of boundary layers, flow-separation on high-lift devices, broad-band noise due to turbulence, and dynamic aeroelasticity.

The challenge posed by chaotic dynamics for simulation-based optimization is not merely one of insufficient computational resources or human effort. Instead, there is a fundamental breakdown of the sensitivity analysis typically used in simulation-based optimization. We elaborate on this breakdown in the following subsection; readers familiar with the challenges posed by sensitivity analysis of chaotic systems can proceed to Subsection 4.1.2.

4.1.1 The breakdown of conventional sensitivity analysis

Simulation-based optimization of aircraft hinges on two key requirements. First, the simulation must be sufficiently accurate with respect to the outputs of interest (*e.g.*, drag, stresses). Second, the simulation must be able to efficiently evaluate sensitivity derivatives of these same outputs. Sensitiv-

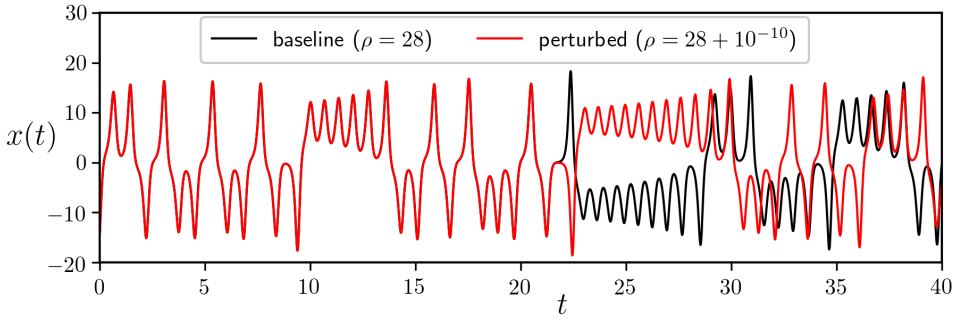


Figure 4.1: Effect of small parameter perturbation on Lorenz state variable $x(t)$.

ity derivatives are needed for gradient-based optimization algorithms, which are the most efficient algorithms for optimizing complex systems involving many (> 100) parameters.

These two requirements — accurate outputs and derivatives — conflict with each other in the case of chaotic dynamical systems and time-averaged outputs. To explain the origins of this conflict, we will use the Lorenz dynamical system, which is a canonical example of a chaotic system. The Lorenz system [50] consists of three state variables, $x(t)$, $y(t)$, and $z(t)$, governed by

$$\begin{aligned} \dot{x} &= \sigma(y - x), & x(0) &= x_0, \\ \dot{y} &= x(\rho - z) - y, & y(0) &= y_0, \\ \dot{z} &= xy - \beta z, & z(0) &= z_0, \end{aligned}$$

where $\dot{x}$ denotes the time derivative of x, for example, and (σ, ρ, β) are parameters.

Chaotic dynamical systems, like Lorenz's, are partially characterized by a sensitive dependence on parameters. This sensitive dependence is illustrated in Figure 4.1, which plots the Lorenz state $x(t)$ for two values of the parameter ρ that differ by 10^{-10}; all other parameters and initial conditions are identical. While the two trajectories track each other up to $t \approx 21$, they subsequently diverge from one another and become distinct.

This high sensitivity to parameter changes is disastrous for obtaining useful derivative information of time-averaged outputs of interest, *e.g.*, derivatives of lift, drag, and moment. To observe this in the case of the Lorenz

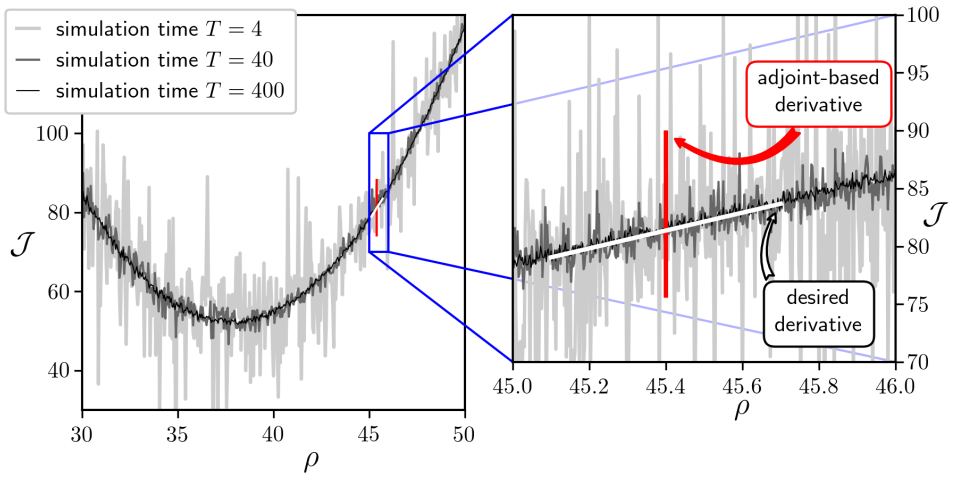


Figure 4.2: Model objective function $\mathcal{J}_T$ for the Lorenz system using three different integration periods T. The zoom on the right shows the adjoint-based derivative and the desired derivative.

system, consider the time averaged output

$$\mathcal{J}_T(\rho) = \frac{1}{T}\int_0^T \frac{1}{2}(z(t;\rho) - 35)^2\, dt.$$

This output acts similar to time-averaged drag in unsteady computational fluid dynamics simulations. For this example, we consider the state $z(t;\rho)$ and, therefore, $\mathcal{J}_T(\rho)$ to be implicit functions of the parameter ρ.

Figure 4.2 plots $\mathcal{J}_T$ versus ρ. Each data point corresponds to solving the Lorenz system over some period[1] of time T. We can see a minimizer in the data, a value $\rho \approx 38$ that produces the smallest $\mathcal{J}_T$ on average. The figure also illustrates that, as one would expect, a longer integration time T reduces the deterministic "noise" in the simulation.

Unfortunately, increasing T does not improve the situation for derivatives. As the zoomed figure on the right demonstrates, increasing T actually increases the frequency of the oscillations. A conventional sensitivity analysis produces a derivative that accurately reflects this noise (red line in zoomed figure), but this derivative is not useful for design. What is needed is a methodology that produces the white line in the figure.

[1]We precede each integration period with a spin-up period to approach the Lorenz attractor and eliminate bias due to the initial conditions.

4.1.2 State-of-Knowledge

There have been several efforts to compute useful sensitivity derivatives for chaotic systems, and these are summarized below. However, the solutions presented so far have either been inaccurate or computationally impractical. Our goal is to address this trade-off between accuracy and cost.

The earliest paper on the subject of sensitivity analysis of chaotic systems was by Lea *et al.* [48]. They proposed the ensemble adjoint, in which the integration period T is partitioned into P smaller periods. On each smaller time domain, independent adjoints are solved and then used to compute sensitivities that are averaged. While the ensemble adjoint helps avoid unbounded growth in the adjoint, the method converges at a rate slower than $1/\sqrt{P}$, which is even slower than Monte–Carlo methods [29]. Second-order ensemble adjoints were investigated by Ashley and Hicken [8] in a Newton–Krylov trust-region optimization.

A more recent development is the least-squares shadowing (LSS) method of Wang *et al.* [78]. The LSS method addresses the issue of sensitive dependence by essentially eliminating the initial condition from the perturbed state. This technique is justified by the Shadowing Lemma [66], which states that for a given parameter perturbation, *e.g.*, $\rho + \delta\rho$, one can find a perturbed state that is "nearby" the baseline state. Blonigan *et al.* [15] successfully applied the LSS method to a computational fluid dynamics (CFD) simulation around an airfoil. While the original LSS method requires the determination of a time-dilation term, Chater *et al.* [18] have shown that windowing can be used instead.

The primary drawback of the LSS approach is computational cost. For the CFD applications of interest to the aerospace industry, the LSS formulation produces impractically large space-time boundary-value problems. For example, a modest turbulent-flow simulation consisting of 10^6 state variables and 10^4 time steps leads to an LSS problem with approximately 10^{11} degrees of freedom[2]. Assuming 10^4 degrees of freedom per processor, this would require 10^7 processors; for reference, the corresponding forward simulation would typically be run with O(100) processors.

To help reduce the cost of the LSS sensitivity method, Ashley and Hicken [9] used a Fourier approximation of the least-squares adjoint, which is described in Chapter 3. In the case of the Lorenz system, they showed that the number of degrees of freedom could be reduced by an order of magnitude at the cost of coupling the time steps.

[2]Assuming 5 state variables per node, we have 10 DOF per mesh node including the adjoint and the LSS multipliers.

The cost of the LSS approach can also be ameliorated, at least in part, by using the non-intrusive LSS (NILSS) method [60, 14]. The NILSS approach tries to identify the unstable modes of the direct or adjoint sensitivity and remove them from the solution. This approach works well if there are only a few unstable modes, that is, only a few positive Lyapunov exponents. However, because the NILSS must track and solve all unstable modes, the method becomes computational expensive on practical problems. Blonigan [14] showed that even a simple turbulent channel flow at $\mathrm{Re}_\tau = 140$ can have more than 150 modes, which would make the NILSS approximately two orders more expensive than the conventional adjoint.

Finally, Talnikar *et al.* [72] introduced numerical dissipation into the unsteady adjoint equation in order to stabilize the conventional adjoint. They used an energy-stability analysis to determine where and when to introduce dissipation; however, a parameter was necessary to control the magnitude of the dissipation, and the choice of this parameter has a significant influence on the accuracy of the resulting derivatives.

4.2 A method to stabilize the adjoint

The conventional adjoint fails to provide useful sensitivity derivatives because the adjoint equation is linearly unstable. The key insight of this work is that applying a small perturbation to the adjoint equation is sufficient to stabilize the adjoint and obtain useful derivatives. This small perturbation is practical to evaluate because, in the case of the Euler and Navier–Stokes equations, the instability is caused by a spatially localized source term.

Our approach is explained in greater detail in the following sections. Subsection 4.2.1 uses the Lorenz system to explain the linear instability present in the adjoint equations. We then present our proposed stabilization idea in Subsection 4.2.2, and explain its generalization to fluid flows in Subsection 4.3.

4.2.1 A Deeper Look at the Source of the Problem

The problem with conventional sensitivity analysis methods in the context of chaotic dynamics is that the governing linear equations are unstable. Understanding and managing this instability is at the heart of the proposed method, so we will briefly elaborate on this phenomena using the Lorenz problem.

In order to relate our discussion to the Euler equations later, we express

the Lorenz system in the following abstract form:

$$R(u,\rho) \equiv \frac{du}{dt} - F(u;\rho) = 0, \qquad u(0) = u_0,$$

where $u = [x, y, z]^T$ and $F(u;\rho) \equiv \left[\sigma(y-x), \quad x(\rho - z) - y, \quad xy - \beta z\right]^T$ in the case of the Lorenz system. Using this abstract formulation, we can write the (total) derivative of $\mathcal{J}_T(\rho)$ from Sec. 4.1.1 as [32]

$$\frac{d\mathcal{J}_T}{d\rho} = \underbrace{\frac{\partial \mathcal{J}_T}{\partial \rho} - \int_0^T \psi^T \frac{\partial F}{\partial \rho}\, dt}_{\text{General}} = \underbrace{-\int_0^T \psi_y x\, dt}_{\text{Lorenz}},$$

where ψ denotes the adjoint. The adjoint is governed by the linear ordinary differential equation (ODE)

$$-\frac{d\psi}{dt} - \left(\frac{\partial F}{\partial u}\right)^T \psi = -\frac{\partial \mathcal{J}_T}{\partial u}, \qquad \psi(T) = 0. \tag{4.1}$$

Note that the adjoint ODE must be solved backward in time, and that the terminal condition $\psi(T) = 0$ is due to the particular form of $\mathcal{J}_T$.

The adjoint ODE (4.1) is unstable for chaotic dynamical systems. To elucidate the nature of the instability, we multiply the homogeneous adjoint ODE from the left by ψ^T to obtain an equation for the adjoint "energy," that is, an equation for the L^2 norm of the adjoint:

$$-\frac{d}{dt}\underbrace{\left(\frac{1}{2}\|\psi\|^2\right)}_{\text{adjoint "energy"}} - \frac{1}{2}\psi^T \underbrace{\left[\left(\frac{\partial F}{\partial u}\right) + \left(\frac{\partial F}{\partial u}\right)^T\right]}_{\text{symmetric part of Jacobian}} \psi = 0.$$

The skew-symmetric part of the Jacobian does not contribute to the adjoint "energy" because $\psi^T \mathsf{A}\psi = -\psi^T \mathsf{A}\psi = 0$ for any skew-symmetric matrix A.

Remark 6. *We consider the homogeneous adjoint ODE, because it is sufficient to explain the exponential growth in $\|\psi\|$. The inhomogenous term, $\partial \mathcal{J}_T / \partial u$, while vital for the accuracy of the adjoint, does not lead to exponential growth; thus, it is excluded from the present analysis.*

All symmetric matrices are orthogonally diagonalizable, so we can rewrite the ODE for adjoint energy as

$$-\frac{d}{dt}\left(\frac{1}{2}\|\psi\|^2\right) = \psi^T(\mathsf{V}\Lambda\mathsf{V}^T)\psi = \sum_{i=1}^{3}(\mathsf{V}^T\psi)_i^2 \lambda_i,$$

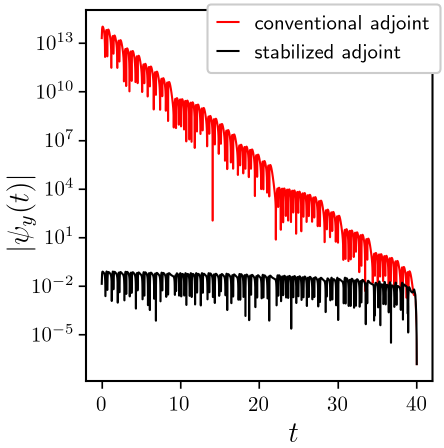


Figure 4.3: Conventional adjoint (red) and proposed, stabilized adjoint (black). Note the y-axis scale.

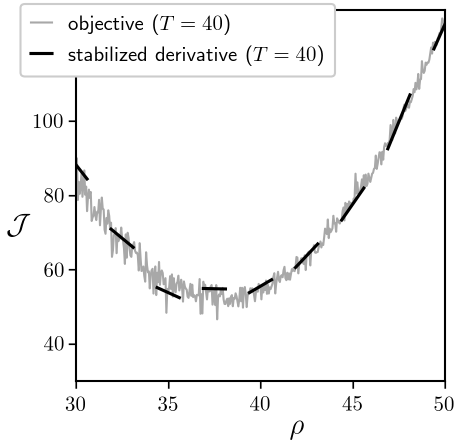


Figure 4.4: Objective and derivatives based on the stabilized adjoint.

where $\mathsf{V}\Lambda\mathsf{V}$ is the eigenvalue decomposition of the symmetric part of $\partial F/\partial u$, with eigenvectors in the columns of V and eigenvalues, λ_i, along the diagonal of Λ. Since $(V^T\psi)_i^2 \geq 0$, the adjoint energy will decrease *backward in time* if all the eigenvalues are negative; however, if some eigenvalues are positive, the boundedness of the adjoint is less clear, and it may grow exponentially backward in time.

Chaotic dynamical systems have positive eigenvalues that, on balance, overwhelm the negative eigenvalues. More precisely, chaotic dynamical systems have positive Lyapunov exponents [66]. Consequently, for long-time averaged outputs, the adjoint of chaotic dynamical systems grows exponentially as it is evolved backward in time. The red curve in Figure 4.3 demonstrates this exponential growth for the adjoint ψ_y in the Lorenz problem; recall that ψ_y is the adjoint variable used to evaluate the derivative $d\mathcal{J}_T/d\rho$. If this adjoint is used to compute $d\mathcal{J}_T/d\rho$, we obtain the (unusable) vertical derivative seen earlier in Figure 4.2.

4.2.2 Idea: minimal stabilization of the adjoint

Based on the eigenvalue analysis of the Lorenz adjoint, we propose a simple but effective solution to the instability: add a small perturbation to the Jacobian such that the adjoint energy does not grow. There are several

possible ways to apply such a perturbation. Here, we determine the minimal symmetric matrix that, when added to the Jacobian, produces a stable system.

To illustrate the idea, we once again consider the Lorenz system. If $\mathsf{S}(t) \in \mathbb{R}^{3\times 3}$ denotes the symmetric perturbation matrix, then the stabilized version of the adjoint equation (4.1) becomes

$$-\frac{d\psi}{dt} - \left(\frac{\partial F}{\partial u} + \mathsf{S}\right)^T \psi = -\frac{\partial \mathcal{J}_T}{\partial u}, \qquad \psi(T) = 0.$$

If we repeat the energy analysis on the homogeneous version of the above equation, we arrive at the following:

$$-\frac{d}{dt}\left(\frac{1}{2}\|\psi\|^2\right) - \underbrace{\frac{1}{2}\psi^T\left[\left(\frac{\partial F}{\partial u}\right) + \left(\frac{\partial F}{\partial u}\right)^T + \mathsf{S}\right]\psi}_{\text{must be negative for stability}} = 0.$$

To ensure that the adjoint energy does not grow exponentially backward in time, we need to ensure that the product of ψ with the sum of S and the symmetric part of the Jacobian is negative. On the other hand, we want to avoid large S that would negatively impact the accuracy of the sensitivities. This leads us to the following quadratic optimization problem for the entries in S:

$$\begin{aligned} \min_{\mathsf{S}_{ij}} \quad & \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{i}\mathsf{S}_{ij}^2, \\ \text{s.t.} \quad & \psi^T\left(\mathsf{J}_{\text{sym}} + \mathsf{S}\right)\psi < 0, \end{aligned} \tag{4.2}$$

where J_{sym} denotes the symmetric part of the Jacobian.

Since the problem (4.2) is a quadratic program with a convex objective, it has a unique closed form solution. If $\psi^T\mathsf{J}_{\text{sym}}\psi < 0$, the solution is simply $\mathsf{S} = \mathsf{0}$. Otherwise, the constraint is active and we need to solve an equality constrained quadratic program. Let $s \in \mathbb{R}^6$ denote the vector of unknown entries in the symmetric matrix S, ordered as follows:

$$\mathsf{S}_{ij} = s_{i(i-1)/2+j}, \qquad i = 1,\ldots,3,\ j = 1,\ldots,i.$$

Then, the solution to the equality-constrained quadratic program corresponds to the solution of the saddle-point problem

$$\begin{bmatrix} \mathsf{I} & \mathsf{A}^T \\ \mathsf{A} & \mathsf{0} \end{bmatrix}\begin{bmatrix} s \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ -\psi^T\mathsf{J}_{\text{sym}}\psi \end{bmatrix},$$

where the constraint Jacobian is a row matrix whose entries are given by

$$\mathsf{A}_{i(i-1)/2+j} = \begin{cases} \psi_i^2, & \text{if } i = j \\ 2\psi_i\psi_j, & \text{if } i \neq j. \end{cases}$$

Thus, the solution to (4.2) is (in terms of the entries s)

$$s = \begin{cases} 0, & \text{if } \psi^T \mathsf{J}_{\text{sym}} \psi < 0, \\ -\left(\dfrac{\psi^T \mathsf{J}_{\text{sym}} \psi}{\mathsf{A}\mathsf{A}^T} \right) \mathsf{A}^T, & \text{otherwise.} \end{cases}$$

Remark 7. *The stabilization matrix* S *depends on the adjoint, consequently, the stabilized adjoint equation is nonlinear.*

Figure 4.3 includes the evolution of the stabilized adjoint obtained with the proposed method. The exponential growth of the conventional adjoint has been eliminated. Such a stabilized adjoint can also be obtained using the LSS method described in Sec. 4.1.2. However, there is a significant difference in cost between the LSS adjoint and the solution proposed here. Indeed, the stabilized adjoint shown in Figure 4.3 is only a small fraction more expensive than the conventional adjoint, and it is orders of magnitude less expensive than the LSS adjoint.

Although we can stabilize the adjoint, this does not necessarily imply that it will produce accurate, useful sensitivity derivatives. However, the proposed mechanism is such that the perturbation is minimized. The magnitude of the perturbation to the adjoint equation is controlled by the entries in S, which are, by definition, the smallest possible perturbations that that ensure non-positive growth in ψ.

The potential of this approach is demonstrated in Figure 4.4, which shows the objective $\mathcal{J}_T$ for $T = 40$ and the sensitivity derivatives computed using the proposed stabilized adjoint. The derivatives are plotted as "linearizations" about several values of ρ. Qualitatively we observe good agreement with the slope of the objective. Note that the linearizations are shifted up or down depending on the value of $\mathcal{J}_T$ at the corresponding ρ.

4.3 Generalizing to Fluid Dynamics

The proposed method is promising, but it is not immediately clear that it will generalize to the Euler or Navier-Stokes equations. In particular, the perturbation to the Jacobian used in the Lorenz example relied on solving a

quadratic optimization with as many variables as the (symmetric part of the) Jacobian. In the case of an Euler or Navier–Stokes simulation, the Jacobian will have potentially billions (or more) of rows and columns, making such an optimization impractical.

In the following section, we explain why this issue is mitigated by the nature of the instability in fluid flows. Subsequently, we describe a practical implementation of the stabilization method in the case of the Euler equations.

4.3.1 A localized instability

As described in the introduction to this section, the proposed stabilization method appears to be impractical for the Euler or Navier-Stokes equations, because the size of the Jacobian matrix precludes the necessary optimization. Fortunately, the terms responsible for destabilizing the Euler and Navier-Stokes adjoints are spatially-local source terms. To see this, consider the (homogeneous) adjoint equation for the incompressible Navier–Stokes equations (with the summation-convention on repeated indices):

$$\rho\frac{\partial\psi_i}{\partial t}+\rho u_j\frac{\partial\psi_i}{\partial x_j}-\underbrace{\rho\frac{\partial u_j}{\partial x_i}\psi_j}_{\text{unstable}}+\frac{\partial\pi}{\partial x_i}=\nu\frac{\partial^2\psi_i}{\partial x_j\partial x_j},\qquad \frac{\partial\psi_j}{\partial x_j}=0, \tag{4.3}$$

where $\psi=[\psi_1,\psi_2,\psi_3]^T$ is the adjoint corresponding to velocity and π is the adjoint corresponding to pressure. Using an energy-stability analysis, Wang and Gao [77] showed that the third term in this equation is responsible for the adjoint's exponential growth. The symmetric part of the tensor multiplying ψ in the unstable term is

$$\frac{\rho}{2}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right),$$

which is the density times the strain-rate tensor. While this tensor depends on derivatives of the velocity field, it is entirely local from the perspective of the adjoint. Therefore, we can perturb this small, 3×3 tensor at each node in exactly the same way that we did to stabilize the adjoint of the Lorenz system.

Talnikar *et al.* [72] performed an energy-stability analysis of the compressible Navier–Stokes adjoint and showed that its destabilizing term is also a spatially localized source term. Consequently, the proposed approach is equally valid for stabilizing the types of flow problems of interest to the aerospace industry.

At this point, it is worth highlighting the differences between the proposed stabilization and the dissipation-based stabilization used in [72]. Introducing dissipation into the adjoint requires a subtle balance between stabilizing the equation and obtaining accurate sensitivities: too little dissipation and the adjoint will grow exponentially, too much dissipation and the sensitivities will be inaccurate. The difficulty in achieving this balance was documented in [72].

In contrast, the proposed stabilization is, by definition, the minimum perturbation necessary to avoid exponential growth in the adjoint. Our hypothesis is that this will lead to accurate sensitivities, as it does for the Lorenz problem. Furthermore, the proposed method is parameter-free, which is a significant advantage in practice.

4.3.2 Implementation case-study: SBP discretization of the Euler equations

The proposed stabilization can be applied to a wide range of discretizations, but the details will vary depending on the choice of discretization. For concreteness, we will focus on an entropy-stable summation-by-parts (SBP) discretization. Furthermore, the destabilizing term in the sensitivity equations, for both incompressible and compressible flows, is due to the inviscid terms. Therefore, to describe our implementation it is sufficient to consider the Euler equations. Finally, since the destabilization mechanism is the same for both the tangent and adjoint sensitivity, we will consider the tangent sensitivity for simplicity.

The entropy-stable SBP discretization of the Euler equations [23] produces a semi-discretization of the form

$$\mathsf{H}\frac{du_h}{dt} - \sum_{\kappa\in\mathcal{T}} \mathsf{P}_\kappa^T F_\kappa(\mathsf{P}_\kappa u_h) - B(u_h,\alpha) = 0, \tag{4.4}$$

where $u_h(t) \in \mathbb{R}^n$ is the discrete solution evaluated at the collocation points, and $\alpha \in \mathbb{R}$ is a parameter, *e.g.* the Mach number. Here we consider a continuous SBP solution space over the tesselation of the domain $\mathcal{T}$, which is analogous to a continuous Galerkin finite-element method (in contrast with the discontinuous SBP space considered in [23]). The matrix $\mathsf{H} \in \mathbb{R}^{n\times n}$ is the diagonal mass matrix, the matrix $\mathsf{P}_\kappa \in \mathbb{R}^{n_\kappa\times n}$ restricts the global solution to the n_κ degrees of freedom on element κ, and $F_\kappa : \mathbb{R}^{n_\kappa} \to \mathbb{R}^{n_\kappa}$ denotes the element-local residual vector that discretizes the spatial derivatives. Finally, $B : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ denotes the boundary terms; the boundary conditions are

imposed weakly using penalty terms and we assume they depend on the parameter α.

The tangent-sensitivity equation is obtained by (total) differentiation of the discretization (4.4) with respect to α. Thus, if $v_h(t) \in \mathbb{R}^n$ denotes the sensitivity, the tangent-sensitivity equation is given by

$$\mathsf{H}\frac{dv_h}{dt} - \sum_{\kappa\in\mathcal{T}} \mathsf{P}_\kappa^T \mathsf{J}_\kappa \mathsf{P}_\kappa v_h - \frac{\partial B}{\partial u_h} v_h - \frac{\partial B}{\partial \alpha} = 0, \tag{4.5}$$

where $\mathsf{J}_\kappa \equiv \partial F_\kappa / \partial u_\kappa$ is the element-local Jacobian, with $u_\kappa = \mathsf{P}_\kappa u_\kappa$. In the following analysis, we will focus on stabilizing the terms involving J_κ, since they contain the localized instability described earlier. We will not stabilize the derivatives of the boundary terms, which are responsible for driving the tangent (and adjoint) sensitivities.

Next, we perform an energy-stability analysis of (4.5). Ignoring the boundary-term derivatives and left-multiplying by v_h^T, we find

$$\begin{aligned} & v_h^T \mathsf{H}\frac{dv_h}{dt} - \sum_{\kappa\in\mathcal{T}} v_h^T \mathsf{P}_\kappa^T \mathsf{J}_\kappa \mathsf{P}_\kappa v_h = 0 \\ \Rightarrow \qquad & \frac{d}{dt}\left(\frac{1}{2}\|v_h\|_{\mathsf{H}}^2\right) = \sum_{\kappa\in\mathcal{T}} v_\kappa^T \mathsf{J}_\kappa v_\kappa, \end{aligned}$$

where $\|v_h\|_{\mathsf{H}}$ is the SBP approximation of the L^2 norm of v_h, and $v_\kappa \equiv \mathsf{P}_\kappa v_h$ is the tangent sensitivity projected onto the degrees of freedom of element κ. In general, $v_\kappa^T \mathsf{J}_\kappa v_\kappa$ may be positive, leading to growth in the sensitivity. To combat this, following the Lorenz-problem example, we add a symmetric matrix $\mathsf{S}_\kappa \in \mathbb{R}^{n_\kappa \times n_\kappa}$ to each J_κ such that $v_\kappa^T(\mathsf{J}_\kappa + \mathsf{S}_\kappa)v_\kappa \leq 0$. It then follows that

$$\frac{d}{dt}\left(\frac{1}{2}\|v_h\|_{\mathsf{H}}^2\right) = \sum_{\kappa\in\mathcal{T}} v_\kappa^T(\mathsf{J}_\kappa + \mathsf{S}_\kappa)v_\kappa \leq 0,$$

which implies the tangent sensitivity remains bounded. The S_κ are found by solving a quadratic optimization analogous to (4.2), with J_{sym} replaced with $(\mathsf{J}_\kappa + \mathsf{J}_\kappa^T)/2$ and ψ replaced with v_κ.

Notice that, rather than stabilizing the global Jacobian, $\sum_{\kappa\in\mathcal{T}} \mathsf{P}_\kappa^T \mathsf{J}_\kappa \mathsf{P}_\kappa$, we instead stabilize $|\mathcal{T}|$ element Jacobians, J_κ. This represents a significant reduction in cost, since the former requires solving an optimization problem with $\mathrm{O}(n^2)$ variables, while the latter requires only $\mathrm{O}(nn_\kappa)$ variables. Furthermore, the element stabilization is easily parallelized.

4.4 Current and Future Work

The analysis presented above proves that the proposed method will prevent unbounded growth in the direct and adjoint sensitivities. Therefore, the only unanswered question is whether or not the stabilized sensitivities yield accurate derivatives. While the Lorenz-problem results are promising in this regard, the full potential of the method will not be clear until it has been applied to a flow simulation.

In light of this, Mr. Ashley is currently implementing the stabilized sensitivity-analysis method for the Euler equations as part of his thesis. We will apply the proposed method to the chaotic NACA0012 flow studied by Blonigan *et al.* [15]. Since they used the LSS method, this problem will provide a benchmark against which we can compare the accuracy and cost of the stabilized sensitivity method.

Bibliography

[1] M. J. Aftosmis. Lecture notes for the 28th computational fluid dynamics lecture series: solution adaptive Cartesian grid methods for aerodynamic flows with complex geometries. Technical report, von Kármán Institute for Fluid Dynamics, Rhode-Saint-Genèse, Belgium, Mar. 1997.

[2] M. J. Aftosmis, M. J. Berger, and J. E. Melton. Robust and efficient Cartesian mesh generation for component-based geometry. *AIAA journal*, 36(6):952–960, 1998.

[3] T. A. Albring, M. Sagebaum, and N. R. Gauger. Efficient Aerodynamic Design using the Discrete Adjoint Method in SU2. In *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pages 15+. American Institute of Aeronautics and Astronautics, June 2016. ISBN 978-1-62410-439-8. doi: 10.2514/6.2016-3518.

[4] E. L. Allgower and K. Georg. *Acta Numerica*, chapter Continuation and Path Following, pages 1–64. Cambridge University Press, 1993.

[5] D. G. Anderson. Iterative Procedures for Nonlinear Integral Equations. *Journal of the Association for Computing Machinery*, 12(4):547–560, Oct. 1965. ISSN 0004-5411. doi: 10.1145/321296.321305. URL http://dx.doi.org/10.1145/321296.321305.

[6] W. K. Anderson and D. L. Bonhaus. Airfoil design on unstructured grids for turbulent flows. *AIAA Journal*, 37(2):185–191, Feb. 1999.

[7] W. K. Anderson and D. L. Bonhaus. Airfoil design on unstructured grids for turbulent flows. *AIAA Journal*, 37(2):185–191, Feb. 1999.

[8] A. Ashley and J. E. Hicken. Optimization Algorithm for Systems Governed by Chaotic Dynamics. In *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 13, Atlanta, Georgia, United States, June 2014. doi: 10.2514/6.2014-2434.

URL https://dl.dropboxusercontent.com/u/38627164/Website/papers/AIAA-2014-2434.pdf. AIAA 2014-2434.

[9] A. Ashley and J. E. Hicken. Sensitivity Analysis of Chaotic Problems using a Fourier Approximation of the Least-Squares Adjoint. In *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 14, Washington, D.C., United States, June 2016. American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-439-8. doi: 10.2514/6.2016-4409. URL http://dx.doi.org/10.2514/6.2016-4409. AIAA 2016-4409.

[10] V. Balabanov and G. Venter. Multi-fidelity optimization with high-fidelity analysis and low-fidelity gradients. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, New York, 2004.

[11] Barthelemy and R. T. Haftka. Approximation concepts for optimum structural design — a review. *Structural and Multidisciplinary Optimization*, 5(3):129–144, Sept. 1993. ISSN 0934-4373. doi: 10.1007/bf01743349. URL http://dx.doi.org/10.1007/bf01743349.

[12] M. P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197–224, Nov. 1988. ISSN 00457825. doi: 10.1016/0045-7825(88)90086-2.

[13] M. Bierlaire and F. Crittin. Solving Noisy, Large-Scale Fixed-Point Problems and Systems of Nonlinear Equations. *Transportation Science*, 40(1):44–63, Feb. 2006. ISSN 0041-1655. doi: 10.1287/trsc.1050.0119. URL http://dx.doi.org/10.1287/trsc.1050.0119.

[14] P. J. Blonigan. Adjoint sensitivity analysis of chaotic dynamical systems with non-intrusive least squares shadowing. *Journal of Computational Physics*, 348:803–826, Nov. 2017. ISSN 00219991. doi: 10.1016/j.jcp.2017.08.002. URL http://dx.doi.org/10.1016/j.jcp.2017.08.002.

[15] P. J. Blonigan, Q. Wang, E. J. Nielsen, and B. Diskin. Least-Squares Shadowing Sensitivity Analysis of Chaotic Flow Around a Two-Dimensional Airfoil. *AIAA Journal*, 56(2):658–672, Feb. 2018. ISSN 0001-1452. doi: 10.2514/1.j055389. URL http://dx.doi.org/10.2514/1.j055389.

[16] C. G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19(92):577–593, 1965.

[17] C. G. Broyden. The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90, Mar. 1970. ISSN 0272-4960. doi: 10.1093/imamat/6.1.76.

[18] M. Chater, A. Ni, and Q. Wang. Simplified Least Squares Shadowing sensitivity analysis for chaotic ODEs and PDEs. *Journal of Computational Physics*, 329:126–140, Jan. 2017. ISSN 00219991. doi: 10.1016/j.jcp.2016.10.035. URL http://dx.doi.org/10.1016/j.jcp.2016.10.035.

[19] J. R. Chelikowsky and S. G. Louie. *Quantum theory of real materials*, volume 348. Springer Science & Business Media, 1996.

[20] S. S. Collis and M. Heinkenschloss. Analysis of the streamline upwind/Petrov Galerkin method applied to the solution of optimal control problems. Technical Report TR02-01, Houston, Texas, 2002. URL http://www.caam.rice.edu/.

[21] A. R. Conn, N. I. M. Gould, and P. Toint. Convergence of quasi-Newton matrices generated by the symmetric rank one update. *Mathematical Programming*, 50(1-3):177–195, 1991. doi: 10.1007/bf01594934.

[22] E. J. Cramer, J. E. Dennis, Jr, P. D. Frank, R. M. Lewis, and G. R. Shubin. Problem formulation for multidisciplinary optimization. *SIAM Journal on Optimization*, 4(4):754–776, 1994.

[23] J. Crean, J. E. Hicken, D. C. Del Rey Fernández, D. W. Zingg, and M. H. Carpenter. Entropy-stable summation-by-parts discretization of the Euler equations on general curved elements. *Journal of Computational Physics*, 356:410–438, Mar. 2018. ISSN 00219991. doi: 10.1016/j.jcp.2017.12.015. URL http://dx.doi.org/10.1016/j.jcp.2017.12.015.

[24] W. C. Davidon. Variable metric method for minimization. *SIAM Journal on Optimization*, 1(1):1–17, Feb. 1991.

[25] A. Dener and J. Hicken. Matrix-free algorithm for the optimization of multidisciplinary systems. *Structural and Multidisciplinary Optimization*, pages 1–18, 2017. doi: 10.1007/s00158-017-1734-0.

[26] A. Dener and J. E. Hicken. Revisiting individual discipline feasible using matrix-free Inexact-Newton-Krylov. In *10th AIAA Multidisciplinary Design Optimization Conference*. American Institute of Aeronautics and Astronautics, Jan. 2014. doi: 10.2514/6.2014-0110. AIAA 2014–0110.

[27] A. Dener, J. E. Hicken, P. Meng, G. J. Kennedy, J. Hwang, and J. Gray. Kona: a parallel optimization library for engineering-design problems. In *AIAA SciTech Conference*, Jan. 2016. doi: 10.2514/6.2016-1422. AIAA 2016-1422.

[28] V. Eyert. A Comparative Study on Methods for Convergence Acceleration of Iterative Vector Sequences. *Journal of Computational Physics*, 124(2):271–285, Mar. 1996. ISSN 00219991. doi: 10.1006/jcph.1996.0059.

[29] G. L. Eyink, T. W. N. Haine, and D. J. Lea. Ruelle's linear response formula, ensemble adjoint schemes and Lévy flights. *Nonlinearity*, 17 (5):1867–1889, Sept. 2004. ISSN 0951-7715. doi: 10.1088/0951-7715/17/5/016. URL http://dx.doi.org/10.1088/0951-7715/17/5/016.

[30] H.-r. Fang and Y. Saad. Two classes of multisecant methods for nonlinear acceleration. *Numerical Linear Algebra with Applications*, 16(3):197–221, Mar. 2009. ISSN 10705325. doi: 10.1002/nla.617.

[31] R. Fletcher. A new approach to variable metric algorithms. *The Computer Journal*, 13(3):317–322, Mar. 1970. ISSN 0010-4620. doi: 10.1093/comjnl/13.3.317.

[32] M. B. Giles and N. A. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*, 65(3):393–415, 2000. doi: 10.1023/A:1011430410075. URL http://dx.doi.org/10.1023/A:1011430410075.

[33] D. Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of Computation*, 24(109):23–26, 1970. ISSN 0025-5718. doi: 10.1090/s0025-5718-1970-0258249-6.

[34] A. Griewank and A. Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. Society for Industrial and Applied Mathematics, 2008. ISBN 9780898716597. URL http://www.worldcat.org/isbn/9780898716597.

[35] R. T. Haftka and R. V. Grandhi. Structural shape optimizationA survey. *Computer Methods in Applied Mechanics and Engineering*, 57 (1):91–106, Aug. 1986. ISSN 00457825. doi: 10.1016/0045-7825(86) 90072-1.

[36] R. T. Haftka, J. Sobieszczanski-Sobieski, and S. L. Padula. On options for interdisciplinary analysis and design optimization. *Structural optimization*, 4(2):65–74, 1992.

[37] J. E. Hicken and A. Ashley. Arnoldi-based Sampling for High-dimensional Optimization using Imperfect Data. In *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. American Institute of Aeronautics and Astronautics, June 2015. ISBN 978-1-62410-368-1. doi: 10.2514/6.2015-2943. URL https://dl.dropboxusercontent.com/u/38627164/Website/papers/AIAA-2014-2434.pdf.

[38] J. E. Hicken and A. Dener. A flexible iterative solver for nonconvex, equality-constrained quadratic subproblems. *SIAM Journal on Scientific Computing*, 37(4):A1801–A1824, 2015.

[39] J. E. Hicken and D. W. Zingg. Induced-drag minimization of nonplanar geometries based on the Euler equations. *AIAA Journal*, 48(11):2564–2575, Nov. 2010. doi: 10.2514/1.52436. URL http://dx.doi.org/10.2514/1.52436.

[40] J. T. Hwang, D. Y. Lee, J. W. Cutler, and J. R. R. A. Martins. Large-Scale Multidisciplinary Optimization of a Small Satellite's Design and Operation. *Journal of Spacecraft and Rockets*, 51(5):1648–1663, Sept. 2014. doi: 10.2514/1.a32751.

[41] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3):233–260, 1988.

[42] G. J. Kennedy and J. R. R. A. Martins. A parallel finite-element framework for large-scale gradient-based design optimization of high-performance structures. *Finite Elements in Analysis and Design*, 87: 56–73, September 2014. doi: 10.1016/j.finel.2014.04.011.

[43] G. K. W. Kenway and J. R. R. A. Martins. Multipoint high-fidelity aerostructural optimization of a transport aircraft configuration. *Journal of Aircraft*, 51(1):144–160, January 2014. doi: 10.2514/1.C032150.

[44] G. K. W. Kenway, G. J. Kennedy, and J. R. R. A. Martins. Scalable parallel approach for high-fidelity steady-state aeroelastic analysis and adjoint derivative computations. *AIAA Journal*, 52(5):935–951, 2014. doi: 10.2514/1.J052255.

[45] G. K. W. Kenway, G. J. Kennedy, and J. R. R. A. Martins. Scalable parallel approach for high-fidelity steady-state aeroelastic analysis and derivative computations. *AIAA Journal*, 52(5):935–951, May 2014. doi: 10.2514/1.J052255.

[46] D. A. Knoll and D. E. Keyes. Jacobian-free Newton-Krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, 2004. doi: 10.1016/j.jcp.2003.08.010.

[47] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4):255–282, Oct. 1950.

[48] D. J. Lea, M. R. Allen, and T. W. N. Haine. Sensitivity analysis of the climate of a chaotic system. *Tellus A*, 52(5):523–532, Oct. 2000. ISSN 1600-0870. doi: 10.1034/j.1600-0870.2000.01137.x. URL http://dx.doi.org/10.1034/j.1600-0870.2000.01137.x.

[49] H. Liao. Efficient sensitivity analysis method for chaotic dynamical systems. *Journal of Computational Physics*, 313:57–75, May 2016. ISSN 00219991. doi: 10.1016/j.jcp.2016.02.016. URL http://dx.doi.org/10.1016/j.jcp.2016.02.016.

[50] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, 1963.

[51] O. L. Mangasarian. Equivalence of the Complementarity Problem to a System of Nonlinear Equations. *SIAM Journal on Applied Mathematics*, 31(1):89–92, July 1976. ISSN 0036-1399. doi: 10.1137/0131009.

[52] J. R. R. A. Martins. *A coupled-adjoint method for high-fidelity aero-structural optimization*. PhD thesis, Stanford University, Palo Alto, California, 2002.

[53] J. R. R. A. Martins, J. J. Alonso, and J. J. Reuther. High-fidelity aerostructural design optimization of a supersonic business jet. *Journal of Aircraft*, 41(3):523–530, May 2004. doi: 10.2514/1.11478.

[54] K. Maute, M. Nikbay, and C. Farhat. Coupled Analytical Sensitivity Analysis and Optimization of Three-Dimensional Nonlinear Aeroelastic Systems. *AIAA Journal*, 39(11):2051–2061, Nov. 2001. ISSN 0001-1452. doi: 10.2514/2.1227.

[55] K. Maute, M. Nikbay, and C. Farhat. Sensitivity analysis and design optimization of three-dimensional non-linear aeroelastic systems by the adjoint method. *International Journal for Numerical Methods in Engineering*, 56(6):911–933, Feb. 2003. doi: 10.1002/nme.599.

[56] S. Nadarajah and A. Jameson. Optimal Control of Unsteady Flows Using a Time Accurate Method. In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. American Institute of Aeronautics and Astronautics, Sept. 2002. doi: 10.2514/6.2002-5436. URL http://dx.doi.org/10.2514/6.2002-5436.

[57] M. Nemec and M. J. Aftosmis. Aerodynamic shape optimization using a Cartesian adjoint method and CAD geometry. In *24th AIAA Applied Aerodynamics Conference*, number AIAA–2006–3456, San Francisco, California, United States, June 2006.

[58] M. Nemec and D. W. Zingg. Newton-Krylov algorithm for aerodynamic design using the Navier-Stokes equations. *AIAA Journal*, 40(6):1146–1154, June 2002.

[59] M. Nemec, D. W. Zingg, and T. H. Pulliam. Multipoint and multi-objective aerodynamic shape optimization. *AIAA Journal*, 42(6):1057–1065, 2004.

[60] A. Ni and Q. Wang. Sensitivity analysis on chaotic dynamical systems by Non-Intrusive Least Squares Shadowing (NILSS). *Journal of Computational Physics*, 347:56–77, Oct. 2017. ISSN 00219991. doi: 10.1016/j.jcp.2017.06.033. URL http://dx.doi.org/10.1016/j.jcp.2017.06.033.

[61] E. J. Nielsen and W. K. Anderson. Aerodynamic design optimization on unstructured meshes using the Navier-Stokes equations. *AIAA Journal*, 37(11):1411–1419, Nov. 1999.

[62] O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64(1):97–110, 1974.

[63] L. Qi and J. Sun. A nonsmooth version of Newton's method. *Mathematical Programming*, 58(1-3):353–367, 1993. doi: 10.1007/bf01581275.

[64] J. J. Reuther, A. Jameson, J. J. Alonso, M. J. Rimlinger, and D. Saunders. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 1. *AIAA Journal*, 36 (1):51–60, Jan. 1999.

[65] J. J. Reuther, A. Jameson, J. J. Alonso, M. J. Rimlinger, and D. Saunders. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 2. *AIAA Journal*, 36 (1):61–74, Jan. 1999.

[66] C. Robinson. *Dynamical Systems: Stability, Symbolic Dynamics, and Chaos*. Taylor & Francis Group, 1998.

[67] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM Journal on Scientific and Statistical Computing*, 14(2):461–469, 1993.

[68] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, PA, second edition, 2003.

[69] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, July 1986.

[70] J. A. Sethian and A. Wiegmann. Structural Boundary Design via Level Set and Immersed Interface Methods. *Journal of Computational Physics*, 163(2):489–528, Sept. 2000. ISSN 00219991. doi: 10.1006/jcph.2000.6581.

[71] D. F. Shanno. Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation*, 24(111):647–656, 1970. ISSN 0025-5718. doi: 10.1090/s0025-5718-1970-0274029-x.

[72] C. Talnikar, Q. Wang, and G. M. Laskowski. Unsteady Adjoint of Pressure Loss for a Fundamental Transonic Turbine Vane. *Journal of Turbomachinery*, 139(3):031001+, Nov. 2016. ISSN 0889-504X. doi: 10.1115/1.4034800. URL http://dx.doi.org/10.1115/1.4034800.

[73] N. P. van Dijk, K. Maute, M. Langelaar, and F. van Keulen. Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimization*, 48(3):437–472, 2013. doi: 10.1007/s00158-013-0912-y.

[74] D. Vanderbilt and S. G. Louie. Total energies of diamond (111) surface reconstructions by a linear combination of atomic orbitals method. *Physical Review B*, 30(10):6118–6130, Nov. 1984. ISSN 0163-1829. doi: 10.1103/physrevb.30.6118.

[75] S. Venkatamaran and R. T. Haftka. Structural optimization complexity: what has Moore's law done for us? *Structural and Multidisciplinary Optimization*, 28:375–387, 2004.

[76] H. F. Walker and P. Ni. Anderson Acceleration for Fixed-Point Iterations. *SIAM Journal on Numerical Analysis*, 49(4):1715–1735, Jan. 2011. ISSN 0036-1429. doi: 10.1137/10078356x.

[77] Q. Wang and J.-H. Gao. The drag-adjoint field of a circular cylinder wake at Reynolds numbers 20, 100 and 500. *Journal of Fluid Mechanics*, 730:145–161, July 2013. ISSN 0022-1120. doi: 10.1017/jfm.2013.323. URL http://dx.doi.org/10.1017/jfm.2013.323.

[78] Q. Wang, R. Hu, and P. Blonigan. Least Squares Shadowing sensitivity analysis of chaotic limit cycle oscillations. *Journal of Computational Physics*, 267:210–224, June 2014. ISSN 00219991. doi: 10.1016/j.jcp.2014.03.002. URL http://dx.doi.org/10.1016/j.jcp.2014.03.002.

[79] L. T. Watson. Theory of Globally Convergent Probability-One Homotopies for Nonlinear Programming. *SIAM Journal on Optimization*, 11(3):761–780, Jan. 2001. ISSN 1052-6234. doi: 10.1137/s105262349936121x. URL http://dx.doi.org/10.1137/s105262349936121x.

[80] M. Zhou and G. I. N. Rozvany. The COC algorithm, Part II: Topological, geometrical and generalized shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 89(1-3):309–336, Aug. 1991. ISSN 00457825. doi: 10.1016/0045-7825(91)90046-9.