2-14-2014

Quantification of Stability in Systems of Nonlinear Ordinary Differential Equations

Jason Terry

Jason Terry

Candidate

Mathematics and Statistics

Department

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

Approved by the Dissertation Committee:

Dr. Jens Lorenz, Chairperson

Dr. M. Cristina Pereyra

Dr. Stephen Lau

Dr. Francesco Sorrentino

Quantification of Stability in Systems of Nonlinear Ordinary Differential Equations

by

Jason Terry

B.A., Mathematics, California State University Fresno, 2003
B.S., Computer Science, California State University Fresno, 2003
M.A., Interdiscipline Studies, California State University Fresno, 2005
M.S., Applied Mathematics, University of New Mexico, 2009

DISSERTATION

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy
Mathematics

The University of New Mexico

Albuquerque, New Mexico

December, 2013

Dedication

To my mom.

Acknowledgments

I would like to thank my professors, Dr. M. Cristina Pereyra and Dr. Pedro Embid. And I would especially like to thank my professor and advisor, Dr. Jens Lorenz. This was possible with their help and support.

Quantification of Stability in Systems of Nonlinear Ordinary Differential Equations

by

Jason Terry

B.A., Mathematics, California State University Fresno, 2003

B.S., Computer Science, California State University Fresno, 2003

M.A., Interdiscipline Studies, California State University Fresno, 2005

M.S., Applied Mathematics, University of New Mexico, 2009

Ph.D., Mathematics, University of New Mexico, 2013

Abstract

A common process in ODE theory is to linearize an ODE system about an equilibrium point to determine the local stability properties of its orbits. Less common are results that quantify the domain of stability in the original system. We study a class of ODE systems where the domain of nonlinear stability is significantly small given the parameters of the problem. The aim of this paper is to attempt to quantify this region of stability.

Contents

List of Figures **ix**

List of Tables **x**

1 Introduction **1**

1.1 Class of Problems 1

1.2 Previous Result 4

2 Preliminaries **7**

3 Rescaling of the Problem Class **9**

4 Numerical Investigation **14**

5 Linear Subspace Method of Proof **22**

5.1 The Stable Linear Subspace 24

5.2 The Stable Linear Subspace in the Plane 27

5.3 The Stable Linear Subspace in 3-Space 29

6 Linear System Comparison Method of Proof **31**

6.1 Solution to the Linear System 32

6.2 Comparison of the Nonlinear and Linear Systems 35

7 Conclusion **39**

7.1 Main Result 39

7.2 Future Research 40

Appendices **42**

A Inverse of a Vandermonde Matrix **43**

B Product of Differences of the Roots of Unity **48**

C Code **49**

C.1 ode45.m 49

C.2 RandSphere.m 51

C.3 odesphere.m 52

References **55**

List of Figures

4.1 3D Representation of our MATLAB Routine: Experimental Radius of Stability of an ODE . . . 15

4.2 Log-Log Plot of R_{num} vs δ in Dimension $n = 2$ . . . 18

4.3 Log-Log Plot of R_{num} vs δ in Dimension $n = 3$ . . . 20

5.1 Finding the Critical Value of Stability on the u_n-axis Numerically . 24

5.2 The Stable Linear Subspace in the Plane . . . 28

5.3 The Stable Linear Subspace in 3-Space . . . 29

List of Tables

4.1 Theoretical vs. Experimental Radii of Stability for Dimension $n = 2$ 16

4.2 Theoretical vs. Experimental Radii of Stability for Dimension $n = 3$ 19

5.1 Numerical Results for the Critical Value of Stability c_n on the u_n-axis 25

5.2 Numerical Results for where the Stable Linear Subspace Intersects the u_n-axis 27

Chapter 1

Introduction

When analyzing the behavior of a nonlinear system of ordinary differential equations near an equilibrium point, the most common process is to approximate it with a linear system. If the eigenvalues of the linearized system at this point have negative real part, then the point is locally stable, and the discussion may end there. But we want to further study this stability. How large is this region of stability? Does it depend on the size of the eigenvalues, i.e., with greater negative real part comes a larger region of stability? The investigation of these types of questions are less common, so we will attempt to understand them. We begin by defining the class of problems this paper will investigate and summarizing previous results.

1.1 Class of Problems

Consider the nonlinear autonomous ODE system:

$$x' = Ax + Q(x), \quad A \in \mathbb{R}^{n \times n}, \quad x \in \mathbb{R}^n, \quad Q(x) \in \mathbb{R}^n, \tag{1.1}$$

where $Q(x)$ is quadratic in x and vanishes quadratically at the origin. The eigenvalues of A lie in the left-half of the complex plane and the origin is a stable equilibrium point.

Our aim is to quantify the region of stability around the origin in the euclidean norm, which we will denote by $|\cdot|$. We can mathematically pose this question by determining the radius of the largest n-dimensional ball around the origin such that any orbit with initial condition inside the ball will converge to the origin. If we denote this radius by r^*, then the problem can be stated as finding:

$$r^*(A, Q) := \sup\{r : |x(0)| < r \Rightarrow \lim_{t\to\infty} |x(t)| = 0\}. \tag{1.2}$$

We will call r^* the radius of stability.

In the case where A is a diagonal or normal matrix, the problem is solved. We begin with the diagonal case.

Theorem 1. *Consider the nonlinear ODE system $x' = \Lambda x + Q(x)$, where $Q(x)$ is quadratic in x that vanishes at the origin, $|Q(x)| \le |x|^2$, $\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_n)$, and each $\lambda_i < 0$. Define $\alpha(\Lambda) := \min_{1\le i\le n} |\lambda_i|$. If $|x(0)| < \alpha(\Lambda)$, then $\lim_{t\to\infty} |x(t)| = 0$, i.e., $r^*(\Lambda, Q) \ge \alpha(\Lambda)$.*

Proof. Consider:

$$\begin{aligned}
\frac{d}{dt}|x(t)| = \frac{d}{dt}[x_1^2(t) + \cdots + x_n^2(t)]^{1/2} &= \frac{2x_1(t)x_1'(t) + \cdots + 2x_n(t)x_n'(t)}{2[x_1^2(t) + \cdots + x_n^2(t)]^{1/2}} \\
&= \frac{\langle x(t), x'(t)\rangle}{|x(t)|} = \frac{\langle x(t), \Lambda x + Q(x)\rangle}{|x(t)|} \\
&= \frac{\langle x(t), \Lambda x\rangle + \langle x(t), Q(x)\rangle}{|x(t)|} \\
&= \frac{[\lambda_1 x_1^2(t) + \cdots + \lambda_n x_n^2(t)] + \langle x(t), Q(x)\rangle}{|x(t)|} \\
&\leq \frac{-\alpha(\Lambda)|x(t)|^2 + |x(t)|^3}{|x(t)|} \\
&= |x(t)|[|x(t)| - \alpha(\Lambda)]
\end{aligned}$$

Evaluating this inequality at $t = 0$ gives:

$$\frac{d}{dt}|x(0)| \leq |x(0)|[|x(0)| - \alpha(\Lambda)] < 0$$

Hence, $|x(t)|$ is initially decreasing. We claim it is always decreasing. Otherwise, there is an initial time $T > 0$ where $x(T) = \alpha(\Lambda)$ and $\frac{d}{dt}|x(t)| < 0$ on the interval $[0, T]$. However, if we consider:

$$\int_0^T \frac{d}{dt}|x(s)|ds < 0 \Rightarrow |x(T)| - |x(0)| < 0 \Rightarrow |x(T)| < |x(0)| < \alpha(\Lambda)$$

which is a contradiction. Finally, since $|x(t)|$ is always decreasing and positive, it is bounded. So let $|x(t)| - \alpha(\Lambda) \leq -M$. Then the earlier computation gives:

$$\frac{d}{dt}|x(t)| \leq |x(t)|[|x(t)| - \alpha(\Lambda)] \leq -M|x(t)|$$

Solving this differential inequality yields:

$$|x(t)| \leq Ce^{-Mt}$$

Therefore, $\lim_{t\to\infty} |x(t)| = 0$. $\square$

Note for the 1-dimensional case $x' = -\delta x + x^2$, we require $|x(0)| < \delta$ for an orbit to converge to the origin. And so it turns out that the radius of stability satisfies $r^*(\delta, x^2) = \delta$. So if one has no further information but $|Q(x)| \leq |x|^2$, then the lower bound on $\alpha(\Lambda)$ is sharp.

This proof can be extended when the matrix of the nonlinear system is normal.

Theorem 2. *Under the conditions of Theorem 1 with Λ replaced by a normal matrix A, the same result holds.*

Proof. Since A is normal, there exists a diagonal matrix Λ and a unitary matrix U such that $UA = \Lambda U$. If we define the transformation $y = Ux$, then $y' = Ux' = UAx + UQ(x) = \Lambda Ux + UQ(x) = \Lambda y + UQ(U^{-1}y) = \Lambda y + \tilde{Q}(y)$. Theorem 1 now applies to this transformed system. $\square$

So the question remains to study when A is not normal.

1.2 Previous Result

In [KL], it was proposed to study (1.1) with the assumptions $\operatorname{Re}\lambda \leq -2\delta < 0$ for all $\lambda \in \sigma(A)$ and $|Q(x)| \leq \frac{1}{2}C_Q|x|^2$ for all $x \in \mathbb{C}^n$, where $\sigma(A)$ denotes the set of eigenvalues of A and C_Q is some positive constant that depends on $Q(x)$. Their aim was to determine a realistic value for ϵ such that:

$$|x(0)| < \epsilon \Rightarrow \lim_{t\to\infty} |x(t)| = 0. \tag{1.3}$$

In the paper, they showed the condition:

$$|x(0)| < \frac{2\delta^{2n-1}}{C_Q} \tag{1.4}$$

is sufficient for (1.3) and reasoned it was essentially necessary unless one makes more specific assumptions on $Q(x)$. When A is not normal, then the size of ϵ may become extremely small. This can be illustrated by considering (1.1) with:

$$A = \begin{bmatrix} -2\delta & 1 & & 0 \\ & \ddots & \ddots & \\ & & -2\delta & 1 \\ 0 & & & -2\delta \end{bmatrix}, \quad 0 < \delta \leq 1, \quad Q(x) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ x_1^2 \end{bmatrix}. \tag{1.5}$$

In this case where $C_Q = 2$, $\delta = 0.01$, and $n = 5$, we have the surprisingly restrictive condition $|x(0)| < 10^{-18}$.

An issue that is worth discussing at this point is the highly degenerate nature of the eigenvalues of A. One may speculate that this may be a principle cause for this restrictive result. However, we benefit from the fact that the nonlinear system is a relatively straightforward system to inspect. Consider if we were to replace A with the matrix:

$$A = \begin{bmatrix} -2\delta + \epsilon_1 & 1 & & 0 \\ & \ddots & \ddots & \\ & & -2\delta + \epsilon_{n-1} & 1 \\ 0 & & & -2\delta + \epsilon_n \end{bmatrix},$$

where each ϵ_i is a small and distinct perturbation, then all the eigenvalues of A become distinct, but the dynamics of the perturbated system are virtually identical.

Due to this sufficient condition, we have a lower bound on the radius of stability. So in the terms of our proposed problem (1.2), this result shows that for some constant c:

$$c\delta^{2n-1} \leq r^*(A, Q).$$

It remains to investigate if the condition is actually necessary, which this paper will attempt. In order to simplify our investigation of the class of problems (1.1), we will now introduce rescalings of this problem that allow us to study a simpler subclass of problems.

Chapter 2

Preliminaries

In this chapter we will review some basic definitions, results, and introduce the notation that we will use. These definitions can be found in most standard texts such as [Mey], [Wat], and [TP]. We summarize them here.

Given the multivariable function $F : \mathbb{R}^n \to \mathbb{R}^n$, where

$$F(x_1, x_2, \cdots, x_n) = \begin{bmatrix} f_1(x_1, x_2, \cdots, x_n) \\ f_2(x_1, x_2, \cdots, x_n) \\ \vdots \\ f_n(x_1, x_2, \cdots, x_n) \end{bmatrix},$$

its Jacobian is the $n \times n$ matrix of partial derivatives:

$$J_F(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}.$$

A nonlinear autonomous n-th order system of first order differential equations has the form $x' = F(x)$ or:

$$\begin{cases} x_1' &= f_1(x_1, x_2, \cdots, x_n) \\ x_2' &= f_2(x_1, x_2, \cdots, x_n) \\ &\vdots \\ x_n' &= f_n(x_1, x_2, \cdots, x_n) \end{cases}.$$

Solutions to the system of differential equations are called trajectories or orbits. A point $a = (a_1, a_2, \cdots, a_n)$ that satisfies $f_i(a) = 0$ for $i = 1, 2, \cdots, n$ is called an equilibrium point or fixed point of the system. The linearization of the system at an equilibrium point a is $x' = Ax$, where $A = J_F(a)$. Near an equilibrium point, the dynamics of the nonlinear system are similar to the linearized system, provided no eigenvalues of A have zero real part.

The matrix A defines three subspaces of $\mathbb{C}^n$. The stable subspace is spanned by the generalized eigenvectors corresponding to the eigenvalues λ with $\text{Re}\,\lambda < 0$. The unstable subspace is spanned by the generalized eigenvectors corresponding to the eigenvalues λ with $\text{Re}\,\lambda > 0$. The center subspace is spanned by the generalized eigenvectors corresponding to the eigenvalues λ with $\text{Re}\,\lambda = 0$.

According to Schur's Decomposition Theorem, given any square $n \times n$ matrix A, there exists a unitary matrix U and an upper-triangular matrix R such that $A = URU^*$, where the eigenvalues of A lie on the diagonal of R. If A is normal, then $A = U\Lambda U^*$, where Λ is a diagonal matrix consisting of the eigenvalues of A.

Chapter 3

Rescaling of the Problem Class

The class of problems (1.1) is quite large. Thus, we will attempt to narrow the parameters of the problem while maintaining generality. It turns out that the example (1.5) proposed in [KL] is in fact a quintessential demonstration of the ideas of this paper.

The rescaling of (1.1) is accomplished in three steps. Our first task is to use Schur's decomposition on the constant matrix to obtain $A = URU^*$ or $R = U^*AU$, where U is unitary and R is upper-triangular with the eigenvalues of A on its diagonal. If we let $x = Uy$, then the ODE is transformed to:

$$Uy' = AUy + Q(Uy).$$

After left-multiplying by U^*, we obtain:

$$y' = Ry + U^*Q(Uy).$$

If we call $\tilde{Q}(y) = U^*Q(Uy)$, which is also quadratic in y, then we have:

$$y' = Ry + \tilde{Q}(y).$$

Hence, our first transformation yields:

$$x' = Ax + Q(x) \Rightarrow y' = Ry + \tilde{Q}(y).$$

So after this transformation, we may now consider the ODE system:

$$x' = Rx + Q(x),$$

where R is upper triangular with its eigenvalues on the diagonal and $Q(x)$ is quadratic in x. Note that the transformation $x = Uy$ by a unitary matrix does not change the size of balls.

Our second task is a time rescaling accomplished by letting $y(t) = x(\alpha t)$. Then the ODE becomes:

$$y' = \alpha x'(\alpha t) = \alpha[Rx(\alpha t) + Q(x(\alpha t))] = (\alpha R)y + \alpha Q(y).$$

Thus, we may choose α so that the eigenvalues of αR assume any value we choose. Since a time rescaling does not affect the radius of stability, this computation shows that the size of the eigenvalues alone cannot be the determining factor since the eigenvalue bound can take on any arbitrary value without changing the radius of stability! We now see that it is necessary to obtain a standardization before we study this problem. So after this rescaling (after choosing α), we may now consider the ODE system:

$$x' = Rx + Q(x),$$

where R is upper triangular with the eigenvalues of A on its diagonal, $Q(x)$ is quadratic in x, and the standardization we choose is for R to satisfy:

$$\max_i \sum_{j=i+1}^{n} |r_{ij}| = 1.$$

We may assume that R is not diagonal since that would make it a normal matrix.

Our last task is a vector rescaling accomplished by letting $x = \beta y$. Then $x' = \beta y'$ and the ODE becomes:

$$\beta y' = R\beta y + Q(\beta y) = \beta R y + \beta^2 \hat{Q}(y),$$

where $\hat{Q}(y)$ is also quadratic in y. Dividing both sides of this equation by β yields:

$$y' = Ry + \beta\hat{Q}(y).$$

Thus, β can be chosen so that the size of $\beta\hat{Q}(y)$ can be anything we choose. Since a vector rescaling can arbitrarily affect the radius of stability, this computation shows that the size of the nonlinear perturbation is arbitrary as well! It is once again necessary to obtain a standardization. We choose β such that:

$$\max_{y\neq 0} \frac{|\beta\hat{Q}(y)|}{|y|^2} = 1.$$

The following example demonstrates the rescaling process on a 3×3 example. Consider the nonlinear ODE system $x' = Ax + Q(x)$, where:

$$A = \begin{bmatrix} -1 & 1 & 1 \\ 1 & -1 & 0 \\ -1 & 0 & -1 \end{bmatrix}, \quad Q(x) = \begin{bmatrix} 0 \\ 0 \\ x_1^2 \end{bmatrix}.$$

When we apply Schur's decomposition to the constant matrix, it is factored as $A = URU^T$, where:

$$R = \begin{bmatrix} -1 & \sqrt{2} & 0 \\ 0 & -1 & \sqrt{2} \\ 0 & 0 & -1 \end{bmatrix}, \quad U = \begin{bmatrix} 0 & 1 & 0 \\ \sqrt{2}/2 & 0 & \sqrt{2}/2 \\ -\sqrt{2}/2 & 0 & \sqrt{2}/2 \end{bmatrix}.$$

If we let $x = Uy$, substitute this into the ODE system, and use the fact that $AU = UR$, we obtain:

$$Uy' = AUy + Q(Uy) \Rightarrow Uy' = URy + Q(Uy).$$

Left-multiplying by U^T yields:

$$y' = Ry + U^T Q(Uy)$$

We define $\tilde{Q}(y) = U^T Q(Uy)$. And so:

$$y' = Ry + \tilde{Q}(y), \quad \tilde{Q}(y) = \begin{bmatrix} -\frac{\sqrt{2}}{2} y_2^2 \\ 0 \\ \frac{\sqrt{2}}{2} y_2^2 \end{bmatrix}$$

Now we do the time rescaling $z = y(\alpha t)$ with $\alpha = \frac{1}{\sqrt{2}}$. Then $z' = \alpha y'(\alpha t)$ and the ODE system is transformed to:

$$z' = \alpha[Ry(\alpha t) + \tilde{Q}(y(\alpha t))] = (\alpha R)z + \alpha\tilde{Q}(z).$$

This simplifies to $z' = R_2 z + \tilde{Q}_2(z)$, where:

$$R_2 = \begin{bmatrix} -1/\sqrt{2} & 1 & 0 \\ 0 & -1/\sqrt{2} & 1 \\ 0 & 0 & -1/\sqrt{2} \end{bmatrix}, \quad \tilde{Q}_2(z) = \begin{bmatrix} -\frac{1}{2} z_2^2 \\ 0 \\ \frac{1}{2} z_2^2 \end{bmatrix}.$$

Now we do the vector rescaling $z = \beta w$ with $\beta = \sqrt{2}$. Then $z' = \beta w'$, which we substitute into the ODE system:

$$\beta w' = \beta R_2 w + \tilde{Q}_2(\beta w).$$

Note that:

$$\tilde{Q}_2(\beta w) = \begin{bmatrix} -\frac{1}{2}\beta^2 w_2^2 \\ 0 \\ \frac{1}{2}\beta^2 w_2^2 \end{bmatrix}.$$

So then if we divide both sides of the ODE system by β, it becomes:

$$w' = R_2 w + \frac{1}{\beta}\tilde{Q}_2(\beta w).$$

Furthermore:

$$\frac{1}{\beta}\tilde{Q}_2(\beta w) = \begin{bmatrix} -\frac{1}{2}\beta w_2^2 \\ 0 \\ \frac{1}{2}\beta w_2^2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sqrt{2}} w_2^2 \\ 0 \\ \frac{1}{\sqrt{2}} w_2^2 \end{bmatrix} = \tilde{Q}(w).$$

So finally, the ODE system has become:

$$w' = R_2 w + \tilde{Q}(w).$$

The ODE system has now been transformed into a problem within our class of study.

In summary, these rescalings introduce a normalization. It is only after the rescaling process where it is meaningful to consider the size of:

$$\max_j \operatorname{Re} \lambda_j = -\delta < 0$$

and try to relate δ to the radius of stability. So this paper now considers the subclass of problems:

$$x' = Ax + Q(x), \tag{3.1}$$

where A is upper triangular, $Q(x)$ is quadratic in x that vanishes at the origin, A satisfies:

$$\max_i \sum_{j=i+1}^{n} |a_{ij}| = 1,$$

and $Q(x)$ satisfies:

$$\max_{x \neq 0} \frac{|Q(x)|}{|x|^2} = 1.$$

Chapter 4

Numerical Investigation

To begin our quest to find the radius of stability, we attempt to numerically test the result (1.4) from [KL]. Is this bound for $|x(0)|$ necessary to assure convergence or too pessimistic? We use an ODE solver in MATLAB to simulate the paths of orbits in a nonlinear ODE. We wrote our own routine to randomly generate points on the surface of an n-dimensional sphere to provide the initial conditions to the solver. We then manually adjusted the radius of the sphere until we found the maximum radius for which all orbits converged to the origin. Confirmation of this was generally accepted when at least 1,000 points met the convergence condition in the code. Full documentation of the code and criteria used can be found in the appendix. Figure 4.1 provides a 3-dimensional demonstration of the routine's function.

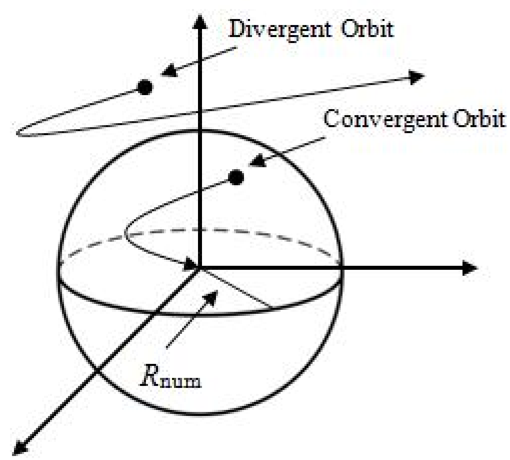

Figure 4.1: 3D Representation of our MATLAB Routine: Experimental Radius of Stability of an ODE

$\delta^{2(2)-1}$	R_{num}
0.125	1.391
0.0156	0.241
0.00463	0.0765
0.00195	0.0321
0.001	0.0159
0.000579	0.00891
0.000364	0.00547
0.000244	0.00362
0.000171	0.00255
0.000125	0.00188

Table 4.1: Theoretical vs. Experimental Radii of Stability for Dimension $n = 2$

Using the system (1.5), we will call the right-hand side of (1.4) the theoretical radius of stability, which is δ^{2n-1}. We will call the numerical radius found by the MATLAB routine the experimental radius of stability, denoted R_{num}. Table 4.1 summarizes our results when $n = 2$ using ten chosen values for δ.

Table 4.1 shows there is a wide discrepancy between the theoretical and experimental radii, which could imply that (1.4) is indeed too pessimistic. However, an important numerical result we notice is that the δ^{2n-1} scaling in (1.4) appears to be valid in all ten cases. In mathematical terms, let us hypothesize that for some constant C:

$$R_{\text{num}} \sim C\delta^{2n-1}.$$

Taking the logarithm of this relationship will give:

$$\log(R_{\text{num}}) \sim \log(C\delta^{2n-1}).$$

Rearranging the terms gives:

$$\log(R_{\text{num}}) \sim (2n-1)\log(\delta) + \log(C).$$

Testing this hypothesis, we use the data in Table (4.1) to construct a log-log plot of the experimental versus theoretical radii of stability for dimension $n = 2$ in the hopes that we obtain a slope of approximately $2n - 1 = 3$. Indeed, Figure (4.2) shows that the linear regression of these points yields a slope of 2.92

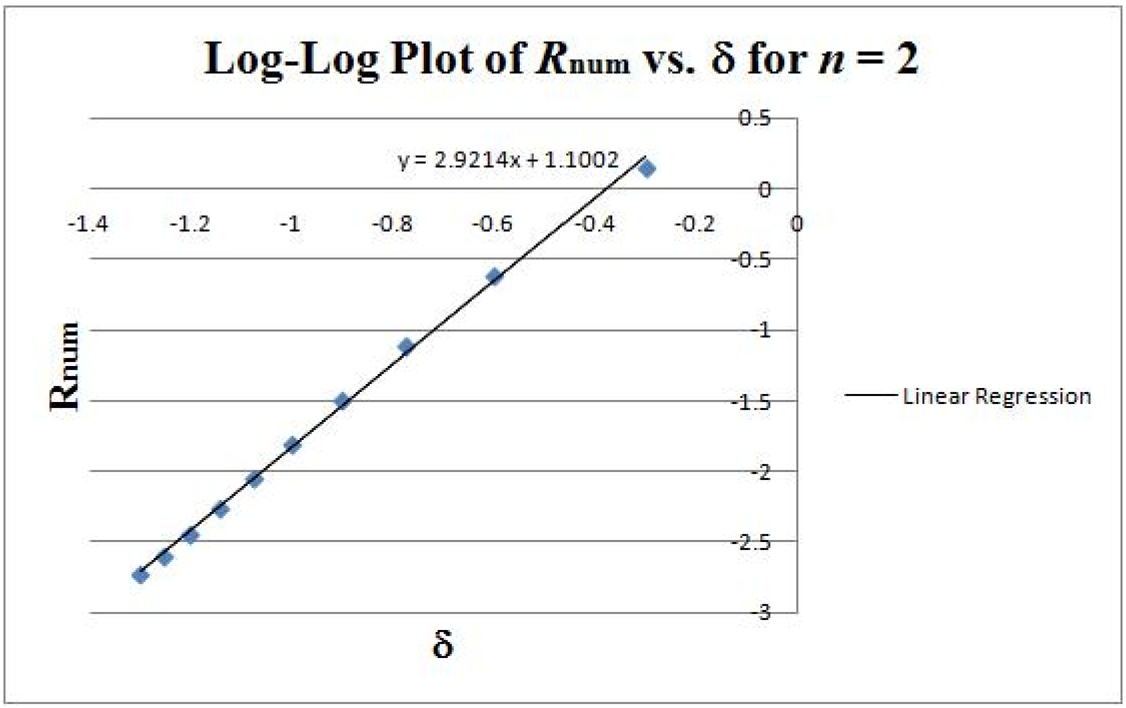

Figure 4.2: Log-Log Plot of R_{num} vs δ in Dimension $n = 2$

$\delta^{2(3)-1}$	R_{num}
0.000977	0.0898
0.000129	0.0127
0.0000305	0.00311
0.00001	0.00103
0.00000402	0.000418
0.00000186	0.000193

Table 4.2: Theoretical vs. Experimental Radii of Stability for Dimension $n = 3$

With this hypothesis in hand, we test it again by conducting a similar numerical test in dimension $n = 3$. Table 4.2 summarizes our results when $n = 3$ using six chosen values for δ. And again, we construct a similar log-log plot in order to obtain slope of approximately $2n - 1 = 5$. And once again, Figure (4.3) shows that the linear regression of these points yields of slope 4.90.

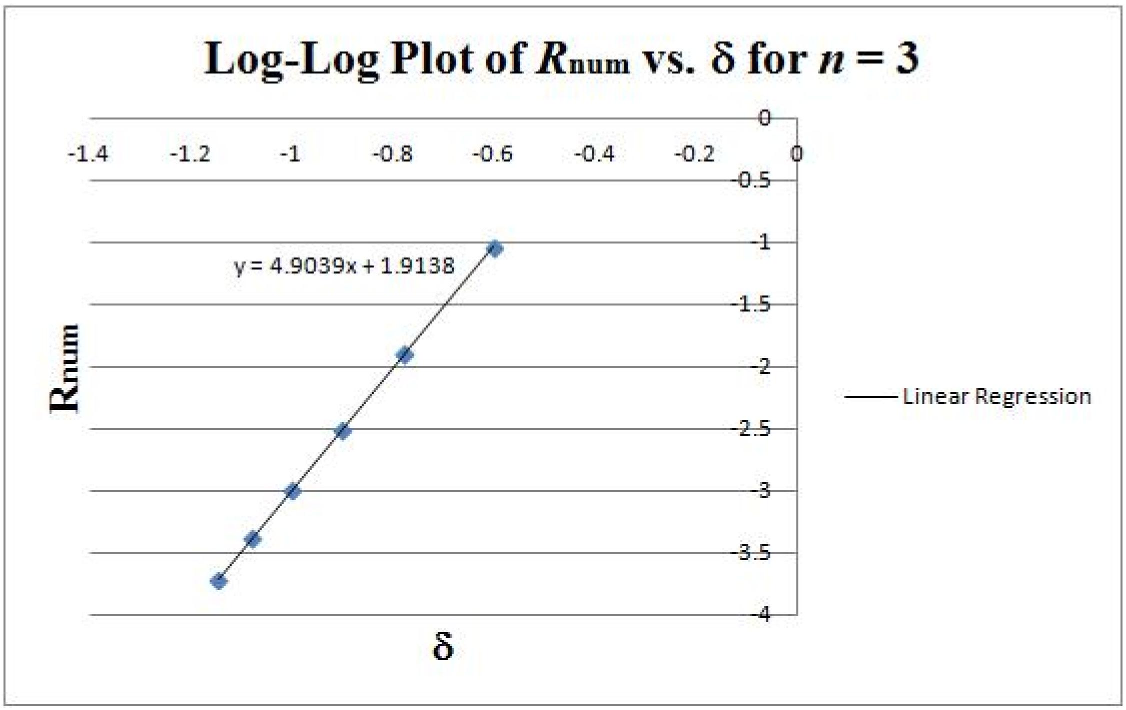

Figure 4.3: Log-Log Plot of R_{num} vs δ in Dimension $n = 3$

So while the theoretical and experimental results are not a match, it appears that the δ^{2n-1} relationship is not coincidental. Earlier we mentioned that, in terms of our posed problem, it has already been shown that there exists a constant c such that the radius of stability satisfies:

$$c\delta^{2n-1} \leq r^*(A, Q).$$

The numerical work up to this point seems to suggest the worthwhile result that there also exists a system and a constant C_n (depending on n, but not on $\delta > 0$) such that:

$$r^*(A, Q) \leq C_n \delta^{2n-1}. \tag{4.1}$$

We will, indeed, prove this result, which will be the focus of the rest of this paper. The next two chapters will demonstrate two methods of proof, the first method having potential and the second method being successful. After these chapters, the main result (4.1) will be formally stated in the conclusion.

Chapter 5

Linear Subspace Method of Proof

After the numerical investigation gave us a deeper understanding of the problem, our first rigorous approach to prove (4.1) will be based on a familiar technique. As we mentioned before, the most common process for understanding equilibrium points in a nonlinear ODE system is to linearize it and study its Jacobian. So, we hypothesized that it may be possible to use the information from the linearized system to give us useful information about the nonlinear system. Specifically, would the stable and unstable subspaces of the linearized system give us any information about the radius of stability about the origin?

With this idea in mind, we now study an example motivated by the one proposed in [KL]. Consider (3.1) with:

$$A = \begin{bmatrix} -\delta & 1 & & 0 \\ & \ddots & \ddots & \\ & & -\delta & 1 \\ 0 & & & -\delta \end{bmatrix}, \quad Q(x) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ x_1^2 \end{bmatrix}. \tag{5.1}$$

Since δ is normally a small quantity, we can use the transformation $x_j = \delta^{n+j-1} y_j$

for $j = 1, 2, \cdots, n$ to eliminate it. Written out, the transformation is:

$$x_1 = \delta^n y_1, \quad x_2 = \delta^{n+1} y_2, \quad \cdots, \quad x_n = \delta^{2n-1} y_n.$$

Note that this transformation changes the size of balls in a simple way. Substituting these equations into the ODE system gives:

$$\begin{cases} \delta^n y_1' & = -\delta^{n+1} y_1 + \delta^{n+1} y_2 \\ \delta^{n-1} y_2' & = -\delta^n y_2 + \delta^n y_3 \\ & \vdots \\ \delta^{2n-1} y_n' & = -\delta^{2n} y_n + \delta^{2n} y_1^2 \end{cases}.$$

Dividing each equation by the appropriate power of δ, the system is transformed to:

$$y' = \delta[By + Q(y)], \quad B = \begin{bmatrix} -1 & 1 & & 0 \\ & \ddots & \ddots & \\ & & -1 & 1 \\ 0 & & & -1 \end{bmatrix}.$$

We now do a second transformation using the time rescaling $u(t) = y(t/\delta)$. Then $\delta u'(t) = y'(t/\delta)$. Substituting these two equations into the ODE system allows us to cancel the δ factor and obtain:

$$u' = Bu + Q(u) \tag{5.2}$$

This simplified system will make studying this problem less numerically intractable. Note that the transformation changes A in a clear way. But one only has to study one system, namely (5.2). Once we have a result, we may reverse the transformation and draw a conclusion.

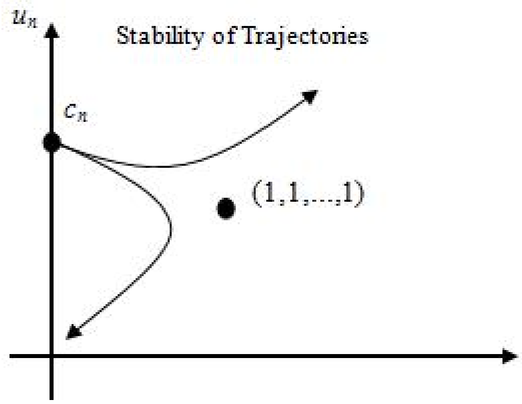

Figure 5.1: Finding the Critical Value of Stability on the u_n-axis Numerically

5.1 The Stable Linear Subspace

We now wish to find the values of ϵ that satisfies (1.3) for (5.2). To get an idea of these values, we will numerically experiment with this system in the hopes that it will lead us to a provable hypothesis. Note that $x_n(t) = \delta^{2n-1}y_n(t) = \delta^{2n-1}u_n(\delta t)$, so we will consider initial values of the form $u(0) = (0, 0, \ldots, c)$ since the last component has the largest perturbation in the transformation back to $x(t)$. Using a MATLAB ODE solver, we numerically determine the values c_n for which $0 \leq c < c_n$ implies convergence to the origin and $c > c_n$ implies divergence. Figure 5.1 shows the significance of c_n on the u_n-axis.

After many numerical trials, the results for c_n up through dimension $n = 12$ are summarized in Table 5.1. At first inspection, it appears $c_n \approx n$. To help better understand this phenomenon, we continue our study of (5.2) by linearizing about the

n	c_n
2	2.213
3	3.289
4	4.310
5	5.294
6	6.248
7	7.176
8	8.080
9	8.959
10	9.814
11	10.644
12	11.448

Table 5.1: Numerical Results for the Critical Value of Stability c_n on the u_n-axis

unstable equilibrium point $(1, 1, \cdots, 1)$. The linearized system is $u' = A_L u$, with:

$$A_L = \begin{bmatrix} -1 & 1 & & 0 \\ 0 & \ddots & \ddots & \\ & \ddots & -1 & 1 \\ 2 & & 0 & -1 \end{bmatrix}.$$

To find its eigenvalues, we must solve the characteristic equation $|A_L - \lambda I| = 0$, which yields:

$$(-1 - \lambda)^n + 2(-1)^{n+1} = 0 \Rightarrow (1 + \lambda)^n = 2.$$

Solving this equation yields:

$$\lambda = -1 + 2^{1/n}[\cos(2\pi k/n) + i \sin(2\pi k/n)], \quad k = 0, 1, \ldots, n-1.$$

By direct computation, this equation reveals that the linearized system has exactly one eigenvalue with a positive real part (unstable) up through dimension $n = 28$. The unstable linear subspace spanned by its corresponding eigenvector will be one-dimensional. Thus, the stable linear subspace will be $(n-1)$-dimensional and should intersect the u_n-axis, unless by some miracle this subspace turns out to be parallel to the u_n-axis. We hypothesize that there is relationship between this intersection point, which we will denote by s_n, and the values c_n we computed earlier.

To confirm this hypothesis, we next attempt to find the stable linear subspace spanned by the eigenvectors with negative real part at $(1, 1, \ldots, 1)$ to determine the point s_n where it intersects the u_n-axis up through the 28th dimension. We will numerically compute the eigenvalues, eigenvectors, and s_n with MATLAB. Let $v_1, v_2, \ldots, v_{n-1} \in \mathbb{R}^n$ span the $(n-1)$-dimension stable linear subspace at $(1, 1, \ldots, 1)$. The point s_n where the space intersects the u_n-axis must satisfy the system:

$$\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + c_1 v_1 + c_2 v_2 + \ldots + c_{n-1} v_{n-1} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ s_n \end{bmatrix}$$

Taking the first $n-1$ rows, we obtain the system:

$$c_1 v_1 + c_2 v_2 + \ldots + c_{n-1} v_{n-1} = \begin{bmatrix} -1 \\ -1 \\ \vdots \\ -1 \end{bmatrix}$$

Solving the second system gives us the coefficients $c_1, c_2, \ldots, c_{n-1}$ that we insert into the first system to compute s_n. The numerical results for s_n up through dimension $n = 7$ is summarized in Table 5.2.

These numerical results lead the observer to hypothesize that the points s_n where the stable linear subspace intersects the u_n-axis is an upper bound to the points c_n. This provides us our first direction to attempt a rigorous proof.

n	s_n
2	2.414
3	3.847
4	5.285
5	6.725
6	8.166
7	9.607

Table 5.2: Numerical Results for where the Stable Linear Subspace Intersects the u_n-axis

5.2 The Stable Linear Subspace in the Plane

Before attempting a general proof, we start with a rigorous proof in the $n = 2$ case. The stable linear subspace is the line $y = -\sqrt{2}x + 1 + \sqrt{2}$, which implies the value where this space intersects the y-axis is $s_2 = 1 + \sqrt{2}$. Figure 5.2 explicitly shows that s_2 is above the critical value of stability c_2. To prove that any trajectory with an initial value above this subspace diverges to infinity, it suffices to show that the region in the first quadrant above the subspace is an invariant region.

Theorem 3. *For the nonlinear ODE system:*

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ x^2 \end{bmatrix}$$

the set $\{(x,y)|x \geq 0, y \geq 0, y \geq -\sqrt{2}x + 1 + \sqrt{2}\}$ *is an invariant region.*

Proof. It suffices to show that all trajectories along the three borders of the set in the first quadrant travel inward toward the region itself. Along the x-axis, this border is described by $y = 0$ and $x \geq (1+\sqrt{2})/\sqrt{2}$. So the ODE gives $y' = x^2 > 0$, and hence, trajectories on this border will travel upward into the region. Along the y-axis, this border is described by $x = 0$ and $y \geq 1 + \sqrt{2}$. So the ODE gives $x' = y > 0$, and hence, trajectories on this border will travel rightward into the region. Along the

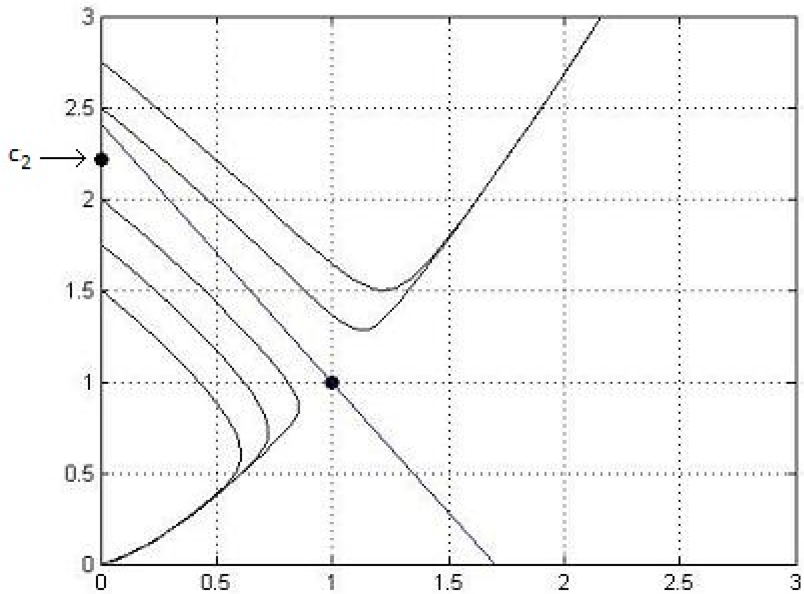

Figure 5.2: The Stable Linear Subspace in the Plane

stable linear subspace (excluding the equilibrium point), this border is described by $y = -\sqrt{2}x + 1 + \sqrt{2}$. Since the slope of this line is $-\sqrt{2}$, showing that no trajectory can travel below this line and achieve a lower slope is equivalent to showing that $|dy/dx| < \sqrt{2}$. Indeed, we confirm this condition by considering:

$$\begin{aligned}
\left|\frac{dy}{dx}\right| = \left|\frac{dy/dt}{dx/dt}\right| = \left|\frac{x^2 - y}{y - x}\right| = \frac{y - x^2}{y - x} < \sqrt{2} \\
&\Rightarrow y - x^2 < \sqrt{2}y - \sqrt{2}x \\
&\Rightarrow -\sqrt{2}x + 1 + \sqrt{2} - x^2 < -2x + \sqrt{2} + 2 - \sqrt{2}x \\
&\Rightarrow x^2 - 2x + 1 = (x-1)^2 > 0.
\end{aligned}$$

Therefore, all trajectories along the three borders of the set in the first quadrant travel inward toward the region itself and thus, it is invariant. $\square$

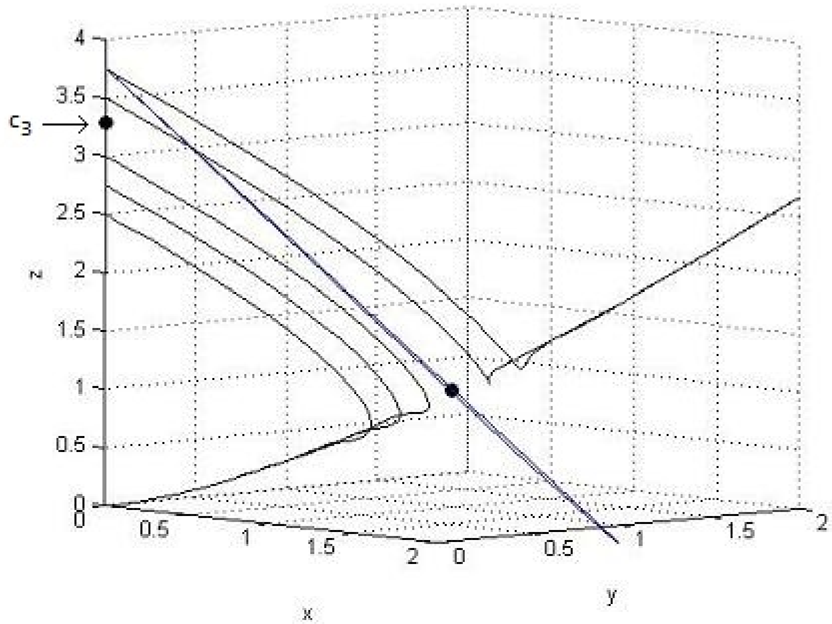

Figure 5.3: The Stable Linear Subspace in 3-Space

5.3 The Stable Linear Subspace in 3-Space

We now consider the $n = 3$ case to discuss the similarities with the previous proof and highlight the difficulties in generalizing this method of proof. Figure 5.3 visually confirms that our hypothesis still holds. But it will soon become computationally more difficult to prove.

Let us continue the process and find where the difficulties arise. The nonlinear ODE system in this case is:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ x^2 \end{bmatrix}$$

By direct calculation, the eigenvectors that span the plane corresponding to the stable linear subspace at the equilibrium point $(1, 1, 1)$ are found to be:

$$v_1 = (-2^{5/3}, -2^{-4/3}, 1)^T \quad \text{and} \quad v_2 = (-2^{-5/3}\sqrt{3}, -2^{-4/3}\sqrt{3}, 0)^T.$$

Thus, the plane of the stable linear subspace is described by:

$$\{(1, 1, 1) + c_1 v_1 + c_2 v_2 | c_1, c_2 \in \mathbb{R}\}.$$

Performing the cross $v_1 \times v_2$ lets us obtain the plane's (scaled) normal:

$$N = (2^{5/3}, 2^{4/3}\sqrt{3}, 1 + \sqrt{3}).$$

To proceed as before and prove that the region in the first octant above the plane is an invariant region, we would again attempt to prove that all trajectories along the borders travel inward toward this region. This can be accomplished by showing the angle between the trajectories and normal are always positive. This leads us to the condition:

$$\langle (x', y', z')^T, N \rangle = 2^{5/3}(-x + y) + 2^{4/3}\sqrt{3}(-y + z) + (1 + \sqrt{3})(-z + x^2) > 0$$

One can now see that this method of proof has become computationally difficult. And the difficulty will increase as the dimension increases. Further, even if this method is successful, it would only be valid up through dimension $n = 28$. Although, this is still interesting since one may obtain insight into the constant c_n.

Chapter 6

Linear System Comparison Method of Proof

With the previous method's apparent intractability, we attempt another method to prove (4.1), but no attempt is made to study how c_n depends on n. The concept for this proof is that if a trajectory begins with a high enough initial condition in the n-dimensional space, it cannot reach the origin and will eventually diverge to infinity. We accomplish this by comparing the nonlinear system with a similar linear system.

Before beginning this proof, we give the linear system and establish some notation. Considering (5.2), let us introduce:

$$J = \begin{bmatrix} 0 & 1 & & 0 \\ & \ddots & \ddots & \\ & & 0 & 1 \\ 0 & & & 0 \end{bmatrix}, \quad P = \begin{bmatrix} 0 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \\ 0 & & 0 & 0 \\ a & 0 & \dots & 0 \end{bmatrix},$$

where $a > 1$ is arbitrary, but fixed. So (5.2) can be expressed as:

$$u' = (-I + J)u + Q(u), \quad Q(u) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ u_1^2 \end{bmatrix}. \tag{6.1}$$

We will be comparing it with the linear ODE system:

$$w' = (-I + J + P)w. \tag{6.2}$$

Note that these two systems are identical except for the last component where the nonlinear equation is $u_n' = -u_n + u_1^2$ and the linear equation is $w_n' = -w_n + aw_1$.

6.1 Solution to the Linear System

We first show that the first component of the linear system will diverge to infinity. We must essentially solve the linear system to accomplish this.

Theorem 4. *Consider the linear system (6.2) with the initial condition:*

$$w(0) = (0, \cdots, 0, m), \quad m > 0.$$

Then the first component of the solution satisfies $w_1(t) \to \infty$ as $t \to \infty$.

Proof. The solution of the linear system (6.2) can be represented as:

$$w(t) = e^{(-I+J+P)t}w(0) = e^{-t}e^{(J+P)t}w(0)$$

To simplify this expression, we diagonalize $J + P$ to exponentiate it. We compute its eigenvalues by solving $(J + P)v = \lambda v$. The components are:

$$v_2 = \lambda v_1, \quad v_3 = \lambda v_2, \quad \cdots, \quad v_n = \lambda v_{n-1}, \quad av_1 = \lambda v_n$$

A nontrivial solution to this system is:

$$v_1 = 1, \quad v_2 = \lambda, \quad v_3 = \lambda^2, \quad \cdots, \quad v_n = \lambda^{n-1}, \quad a = \lambda^n$$

The last equation provides n eigenvalues of the matrix $J + P$, each being one of the roots of a. Let those eigenvalues be denoted by:

$$\lambda_k = \sqrt[n]{a} e^{2\pi i(k-1)/n}, \quad k = 1, 2, \cdots, n.$$

The matrix of eigenvectors will be denoted by V and the diagonal matrix of eigenvalues shall be denoted by D . Thus:

$$V = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_n \\ \lambda_1^2 & \lambda_2^2 & \cdots & \lambda_n^2 \\ \vdots & \vdots & & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \cdots & \lambda_n^{n-1} \end{bmatrix}, \quad D = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}.$$

Note that V is a Vandermonde matrix. Let $(V^{-1})_{ij}$ denote the entry in the i-th row and j-th column of the inverse of V. So $J + P = VDV^{-1}$ and the solution to the linear system $w' = (-I + J + P)w$ with initial condition $w(0) = (0, \cdots, 0, m)$ can be written as:

$$\begin{aligned} w(t) &= e^{-t} e^{(VDV^{-1})t} w(0) = e^{-t} V e^{Dt} V^{-1} w(0) \\ &= e^{-t} V \begin{bmatrix} e^{\lambda_1 t} & & & \\ & e^{\lambda_2 t} & & \\ & & \ddots & \\ & & & e^{\lambda_n t} \end{bmatrix} \begin{bmatrix} m(V^{-1})_{1n} \\ m(V^{-1})_{2n} \\ \vdots \\ m(V^{-1})_{nn} \end{bmatrix} \\ &= m e^{-t} V \begin{bmatrix} e^{\lambda_1 t}(V^{-1})_{1n} \\ e^{\lambda_2 t}(V^{-1})_{2n} \\ \vdots \\ e^{\lambda_n t}(V^{-1})_{nn} \end{bmatrix} \end{aligned}$$

From the inverse of the Vandermonde matrix (see Theorem 9 in appendix), we have that:

$$(V^{-1})_{1n} = \frac{(-1)^{n+1}}{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)\cdots(\lambda_n - \lambda_1)}$$

We claim that $(V^{-1})_{1n} > 0$, that is:

$$(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)\cdots(\lambda_n - \lambda_1)(-1)^{n+1} > 0. \tag{6.3}$$

If we let $z_1 = 1, z_2 = e^{2\pi i \cdot 1/n}, z_3 = e^{2\pi i \cdot 2/n}, \cdots, z_n = e^{2\pi i \cdot (n-1)/n}$ be the n roots of unity, then the left-side (6.3) is equal to:

$$(\sqrt[n]{a}z_2 - \sqrt[n]{a}z_1)(\sqrt[n]{a}z_3 - \sqrt[n]{a}z_1)\cdots(\sqrt[n]{a}z_n - \sqrt[n]{a}z_1)(-1)^{n+1}$$

Factoring the $\sqrt[n]{a}$ terms, the left-side of (6.3) is now equal to:

$$(\sqrt[n]{a^{n-1}})(z_2 - z_1)(z_3 - z_1)\cdots(z_n - z_1)(-1)^{n+1}$$

From Theorem 10 in the appendix, we indeed have that $(V^{-1})_{1n} > 0$. So now, the first component of $w(t)$ is:

$$\begin{aligned} w_1(t) &= me^{-t}[e^{\sqrt[n]{a}t}(V^{-1})_{1n} + e^{\lambda_2 t}(V^{-1})_{2n} + \cdots + e^{\lambda_n t}(V^{-1})_{nn}] \\ &= m[e^{(\sqrt[n]{a}-1)t}(V^{-1})_{1n} + e^{(\lambda_2-1)t}(V^{-1})_{2n} + \cdots + e^{(\lambda_n-1)t}(V^{-1})_{nn}] \end{aligned}$$

Note that $\sqrt[n]{a} > \operatorname{Re} \lambda_k$ for $k = 2, 3, \cdots, n$. So the term $m(V^{-1})_{1n}e^{(\sqrt[n]{a}-1)t}$ is the dominant term in $w_1(t)$. Furthermore, $\sqrt[n]{a} - 1 > 0$ since $a > 1$. Since $(V^{-1})_{1n} > 0$ and $m > 0$, we have that:

$$w_1(t) \approx m(V^{-1})_{1n}e^{(\sqrt[n]{a}-1)t} \to \infty \quad \text{as} \quad t \to \infty$$

□

It is worth noting that while the expression for $w_1(t)$ may contain complex values, it is indeed real. The terms $e^{(\lambda_2-1)t}(V^{-1})_{2n} + \cdots + e^{(\lambda_n-1)t}(V^{-1})_{nn}$ come in complex conjugate pairs whose imaginary parts will vanish, if n is odd. Otherwise, it will contain an additional pure real term if n is even. The argument would be similar to Theorem 10 in the appendix.

6.2 Comparison of the Nonlinear and Linear Systems

Recall Duhamel's formula which states that if $A \in \mathbb{R}^{n\times n}$, $F : \mathbb{R}^n \to \mathbb{R}^n$, and $F \in C^\infty$, then the solution to the general inhomogeneous equation $v' = Av + F(v)$ satisfies:

$$v(t) = e^{At}v(0) + \int_0^t e^{A(t-s)}F(v(s))ds$$

We next prove an inequality for the components of the nonlinear system (6.1).

Theorem 5. *Consider the nonlinear ODE system (6.1) with initial condition:*

$$u(0) = (0, \cdots, 0, M), \quad M > 0.$$

Each component of the solution $u(t)$ of the system satisfies:

$$u_k(t) \geq \frac{Mt^{n-k}}{(n-k)!}e^{-t}, \quad k = 1, 2, \cdots, n.$$

Proof. Consider the last component of the system with $u_n(0) = M$:

$$u_n' = -u_n + u_1^2 \Rightarrow u_n' \geq -u_n \Rightarrow u_n(t) \geq Me^{-t}$$

We use Duhamel on the second-to-last component $u_{n-1}' = -u_{n-1} + u_n$ to obtain:

$$u_{n-1}(t) = e^{-t}u_{n-1}(0) + \int_0^t e^{-(t-s)}u_n(s)ds \geq \int_0^t e^{-t+s}Me^{-s}ds = Me^{-t}\int_0^t ds = Mte^{-t}$$

Using Duhamel again on the third-to-last component $u_{n-2}' = -u_{n-2} + u_{n-1}$ to obtain:

$$\begin{aligned} u_{n-2}(t) &= e^{-t}u_{n-2}(0) + \int_0^t e^{-(t-s)}u_{n-1}(s)ds \geq \int_0^t e^{-t+s}Mse^{-s}ds \\ &= Me^{-t}\int_0^t sds = \frac{Mt^2}{2}e^{-t} \end{aligned}$$

Continuing in this manner on each component gives the desired result. □

We now have the ingredients necessary to prove that the nonlinear system will diverge to infinity.

Theorem 6. *For the nonlinear ODE system (6.1) with initial condition:*

$$u(0) = (0, \cdots, 0, M),$$

and the linear ODE system (6.2) with initial condition:

$$w(0) = (0, \cdots, 0, m), \quad m > 0,$$

we may pick $M > 0$ large enough so that there exists a time T such that $u_1(t) > w_1(t)$ for all $t > T$.

Proof. By Theorem 4, we know $w_1(t) \to \infty$. So there exists a time T such that $w_1(t) > a$ for all $t > T$. From Theorem 5, we can pick M large enough so that for each component in the nonlinear system, $u_k(T) > w_k(T)$ for $k = 1, 2, \cdots, n$. In particular, $u_1(T) > w_1(T) > a$. We now claim that $u_1(t) > w_1(t)$ for all $t > T$. Seeking a contradiction, suppose there is a first time $T^* > T$ where $u_1(T^*) = w_1(T^*)$. From Duhamel's formula, the solution to the nonlinear system starting at T satisfies:

$$u(t) = e^{(J+P)(t-T)}u(T) + \int_T^t e^{(J+P)(t-s)} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ u_1^2(s) \end{bmatrix} ds$$

From Duhamel again, the solution to the linear system starting at T satisfies:

$$w(t) = e^{(J+P)(t-T)}w(T) + \int_T^t e^{(J+P)(t-s)} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ aw_1(s) \end{bmatrix} ds$$

The difference of these two solutions is:

$$(u-w)(t) = e^{(J+P)(t-T)}(u-w)(T) + \int_T^t e^{(J+P)(t-s)} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ (u_1^2 - aw_1)(s) \end{bmatrix} ds$$

Now we look at the first component of the difference and evaluate at T^*:

$$\begin{aligned}(u_1 - w_1)(T^*) &= \left(e^{(J+P)(T^*-T)}(u-w)(T)\right)_1 \\ &\quad + \int_T^{T^*} \left(e^{(J+P)(T^*-s)} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ (u_1^2 - aw_1)(s) \end{bmatrix} \right)_1 ds\end{aligned}$$

Note that all the exponentiated matrices above have only non-negative entries. The vector $(u-w)(T)$ has only positive entries since $u_k(T) > w_k(T)$ for $k = 1, 2, \cdots, n$ by our choice of M. The matrix entry $(u_1^2 - aw_1)(s)$ in the integrand is also positive on the interval $T \le s \le T^*$ by assumption because (recall $a > 1$):

$$u_1(s) > w_1(s) > a \Rightarrow u_1^2(s) > au_1(s) > aw_1(s)$$

Thus, all the values in the first component are positive. Therefore, $(u_1 - w_1)(T^*) > 0$, which is a contradiction, and the result is proved. $\square$

Connecting the last three theorems now gives the following corollary:

Theorem 7. *For the nonlinear ODE system (6.1) with initial condition:*

$$u(0) = (0, \cdots, 0, M), \quad M > 0,$$

there exists a value $M = M(n)$ large enough such that $|u(t)| \to \infty$ as $t \to \infty$.

We now know that if a trajectory begins with a high enough initial condition in n-dimensional space, it will indeed diverge to infinity. Once we take the rescalings of the problem into consideration, we are now prepared to formally state the consequences in terms of the radius of stability.

Chapter 7

Conclusion

7.1 Main Result

Theorem 7 says that there exists $M_n > 0$ large enough such that (5.2) with initial condition $u(0) = (0, \cdots, 0, M_n)$ implies that $|u(t)| \to \infty$ as $t \to \infty$. Suppose we now reverse the transformation $x_j(t) = \delta^{n+j-1} y_j(t) = \delta^{n+j-1} u_j(\delta t)$ of (5.2) back to (5.1). Then we have $x(0) = (0, \cdots, 0, M_n \delta^{2n-1})$ implies that $|x(t)| \to \infty$ as $t \to \infty$.

This result assures there is an upper bound on the radius of stability of the order δ^{2n-1}. Thus, while the δ^{2n-1} scaling from (1.4) at first appears too pessimistic, it is indeed necessary, unless one makes more specific assumptions on $Q(x)$. In terms of our proposed problem (1.2), we have shown that there exists a constant M_n such that:

$$r^*(A, Q) \leq M_n \delta^{2n-1}.$$

Combining this with the result of [KL], we have:

Theorem 8. *Suppose we have the nonlinear autonomous ODE system*

$$x' = Ax + Q(x), \quad A \in \mathbb{R}^{n \times n}, \quad x \in \mathbb{R}^n, \quad Q(x) \in \mathbb{R}^n,$$

with a stable equilibrium point at the origin and $Q(x)$ quadratic in x. Define:

$$r^*(A, Q) := \sup\{r : |x(0)| < r \Rightarrow \lim_{t\to\infty} |x(t)| = 0\}.$$

If $Q(x) \leq |x|^2$ and Re $\lambda \leq -\delta < 0$ for all $\lambda \in \sigma(A)$, then there exists a constant c and a nonlinear system with constant C_n such that:

$$c\delta^{2n-1} \leq r^*(A, Q) \leq C_n\delta^{2n-1}.$$

This is all one can say unless one makes more specific assumptions on $Q(x)$.

7.2 Future Research

In this chapter we will highlight potential areas of future research that were discovered while investigating our problem.

The first question involves a pattern found during the numerical investigation of the stable linear subspace of (5.2). Recall that we used an ODE solver to numerically determine the initial values c_n on the u_n-axis for which an orbit would converge to the origin, as depicted in Figure 5.1. The results listed in Table 5.1 up through dimension $n = 12$ lead one to hypothesize that $c_n \approx n$. If this could be proven, we would have a more precise measurement for the upper bound on the radius of stability.

The next question involves our method of proof using the stable linear subspace. Recall that we hypothesized that the intersection point of stable linear subspace and the u_n-axis provided an upper bound for the radius of stability. We successfully proved this in the $n = 2$ case. But the method became computationally intractable in the $n = 3$ case. Furthermore, the method relied on the fact that the linearized system had exactly one eigenvalue with positive real part up through dimension $n = 28$. Is it possible to overcome the computational difficulty? Can this method be generalized to all dimensions? Can it be shown that the stable linear subspace will always intersect the u_n-axis?

The next question involves our class of problems. Recall that, after rescalings, we require the nonlinear component $Q(x)$ of the ODE system (3.1) to satisfy:

$$\max_{x \neq 0} \frac{|Q(x)|}{|x|^2} = 1.$$

The natural extension of this is to loosen the criteria to the more general case of:

$$\max_{x \neq 0} \frac{|Q(x)|}{|x|^k} = 1, \quad k \geq 2.$$

The last question also involves the class of problems. Recall that the matrix A in (1.1) is assumed to be not normal, otherwise the problem is solved. Is it possible to specify the non-normality of A and investigate its effect on the radius of stability? One approach could be to say A is the perturbation of a normal matrix, i.e., $A = N + \epsilon B$. Another approach could be to say A is the sum of a symmetric and antisymmetric matrix, provided the sum is not normal.

Appendices

A Inverse of a Vandermonde Matrix **43**

B Product of Differences of the Roots of Unity **48**

C Code **49**

Appendix A

Inverse of a Vandermonde Matrix

The following proof is based on [Knu]. First recall that given the set of n points:

$$(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n) \in \mathbb{R}^2$$

with distinct $x_1, x_2, \cdots, x_n$, the Lagrange polynomial that interpolates these points is:

$$L(x) = \sum_{i=1}^{n} y_i l_i(x), \quad l_i(x) = \prod_{\substack{1 \le m \le n \\ m \ne i}} \frac{x - x_m}{x_i - x_m}$$

Theorem 9. *Let $x_1, x_2, \cdots, x_n \in \mathbb{R}$ be distinct. Consider the Vandermonde matrix:*

$$V = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & & \vdots \\ x_1^{n-1} & x_2^{n-1} & \cdots & x_n^{n-1} \end{bmatrix}.$$

Define $f(i, j) = 1$ for $j = n$, and otherwise:

$$f(i, j) = \sum_{\substack{1 \le m_1 < \cdots < m_{n-j} \le n \\ m_1, \cdots, m_{n-j} \ne i}} x_{m_1} \cdots x_{m_{n-j}}, \quad j < n.$$

Then each entry of the inverse of the Vandermonde matrix is:

$$(V^{-1})_{ij} = \frac{(-1)^{j+1} f(i,j)}{\prod_{\substack{1 \le m \le n \\ m \ne i}} (x_m - x_i)}$$

Proof. We first consider the 4×4 case, and then generalize it to the $n \times n$ case. The 4×4 Vandermonde matrix is:

$$V = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ x_1^2 & x_2^2 & x_3^2 & x_4^2 \\ x_1^3 & x_2^3 & x_3^3 & x_4^3 \end{bmatrix}$$

Let $B = V^{-1}$ so that $BV = I$ or:

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ x_1^2 & x_2^2 & x_3^2 & x_4^2 \\ x_1^3 & x_2^3 & x_3^3 & x_4^3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

If we multiply B and V, we may interpret each entry of BV in the ij-th position as a polynomial function with coefficients $b_{i1}, \cdots, b_{i4}$, input x_j, and whose outputs are either 0 or 1. Thus, we observe the following polynomials and points they interpolate:

$$p_1(x) = \sum_{k=1}^{4} b_{1k} x^{k-1} = b_{11} + b_{12}x + b_{13}x^2 + b_{14}x^3; \quad (x_1, 1), (x_2, 0), (x_3, 0), (x_4, 0)$$

$$p_2(x) = \sum_{k=1}^{4} b_{2k} x^{k-1} = b_{21} + b_{22}x + b_{23}x^2 + b_{24}x^3; \quad (x_1, 0), (x_2, 1), (x_3, 0), (x_4, 0)$$

$$p_3(x) = \sum_{k=1}^{4} b_{3k} x^{k-1} = b_{31} + b_{32}x + b_{33}x^2 + b_{34}x^3; \quad (x_1, 0), (x_2, 0), (x_3, 1), (x_4, 0)$$

$$p_4(x) = \sum_{k=1}^{4} b_{4k} x^{k-1} = b_{41} + b_{42}x + b_{43}x^2 + b_{44}x^3; \quad (x_1, 0), (x_2, 0), (x_3, 0), (x_4, 1)$$

Another representation of these functions are their Lagrange polynomials:

$$p_1(x) = \frac{x - x_2}{x_1 - x_2} \cdot \frac{x - x_3}{x_1 - x_3} \cdot \frac{x - x_4}{x_1 - x_4}$$

$$p_2(x) = \frac{x - x_1}{x_2 - x_1} \cdot \frac{x - x_3}{x_2 - x_3} \cdot \frac{x - x_4}{x_2 - x_4}$$

$$p_3(x) = \frac{x - x_1}{x_3 - x_1} \cdot \frac{x - x_2}{x_3 - x_2} \cdot \frac{x - x_4}{x_3 - x_4}$$

$$p_4(x) = \frac{x - x_1}{x_4 - x_1} \cdot \frac{x - x_2}{x_4 - x_2} \cdot \frac{x - x_3}{x_4 - x_3}$$

Rewriting these expressions:

$$p_1(x) = \frac{x^3 - (x_2 + x_3 + x_4)x^2 + (x_2x_3 + x_2x_4 + x_3x_4)x - x_2x_3x_4}{(x_1 - x_2)(x_1 - x_3)(x_1 - x_4)}$$

$$p_2(x) = \frac{x^3 - (x_1 + x_3 + x_4)x^2 + (x_1x_3 + x_1x_4 + x_3x_4)x - x_1x_3x_4}{(x_2 - x_1)(x_2 - x_3)(x_2 - x_4)}$$

$$p_3(x) = \frac{x^3 - (x_1 + x_2 + x_4)x^2 + (x_1x_2 + x_1x_4 + x_2x_4)x - x_1x_2x_4}{(x_3 - x_1)(x_3 - x_2)(x_3 - x_4)}$$

$$p_4(x) = \frac{x^3 - (x_1 + x_2 + x_3)x^2 + (x_1x_2 + x_1x_3 + x_2x_3)x - x_1x_2x_3}{(x_4 - x_1)(x_4 - x_2)(x_4 - x_3)}$$

Matching the coefficients of the two representations for $p_1(x)$:

$$b_{11} = \frac{-x_2x_3x_4}{(x_1 - x_2)(x_1 - x_3)(x_1 - x_4)}, \quad b_{12} = \frac{x_2x_3 + x_2x_4 + x_3x_4}{(x_1 - x_2)(x_1 - x_3)(x_1 - x_4)}$$

$$b_{13} = \frac{-(x_2 + x_3 + x_4)}{(x_1 - x_2)(x_1 - x_3)(x_1 - x_4)}, \quad b_{14} = \frac{1}{(x_1 - x_2)(x_1 - x_3)(x_1 - x_4)}$$

Matching the coefficients of the two representations for $p_2(x)$:

$$b_{21} = \frac{-x_1x_3x_4}{(x_2 - x_1)(x_2 - x_3)(x_2 - x_4)}, \quad b_{22} = \frac{x_1x_3 + x_1x_4 + x_3x_4}{(x_2 - x_1)(x_2 - x_3)(x_2 - x_4)}$$

$$b_{23} = \frac{-(x_1 + x_3 + x_4)}{(x_2 - x_1)(x_2 - x_3)(x_2 - x_4)}, \quad b_{24} = \frac{1}{(x_2 - x_1)(x_2 - x_3)(x_2 - x_4)}$$

Matching the coefficients of the two representations for $p_3(x)$:

$$b_{31} = \frac{-x_1x_2x_4}{(x_3 - x_1)(x_3 - x_2)(x_3 - x_4)}, \quad b_{32} = \frac{x_1x_2 + x_1x_4 + x_2x_4}{(x_3 - x_1)(x_3 - x_2)(x_3 - x_4)}$$

$$b_{33} = \frac{-(x_1 + x_2 + x_4)}{(x_3 - x_1)(x_3 - x_2)(x_3 - x_4)}, \quad b_{34} = \frac{1}{(x_3 - x_1)(x_3 - x_2)(x_3 - x_4)}$$

Matching the coefficients of the two representations for $p_4(x)$:

$$b_{41} = \frac{-x_1 x_2 x_3}{(x_4 - x_1)(x_4 - x_2)(x_4 - x_3)}, \quad b_{42} = \frac{x_1 x_2 + x_1 x_3 + x_2 x_3}{(x_4 - x_1)(x_4 - x_2)(x_4 - x_3)}$$

$$b_{43} = \frac{-(x_1 + x_2 + x_3)}{(x_4 - x_1)(x_4 - x_2)(x_4 - x_3)}, \quad b_{44} = \frac{1}{(x_4 - x_1)(x_4 - x_2)(x_4 - x_3)}$$

Now the pattern for deriving the coefficients is apparent. Their numerators are the product of linear factors with alternating signs and their denominators are the products of the differences of the Vandermonde entries x_1, x_2, x_3, x_4. So in the general case, the $n \times n$ Vandermonde matrix is:

$$V = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & & \vdots \\ x_1^{n-1} & x_2^{n-1} & \cdots & x_n^{n-1} \end{bmatrix}$$

Let $B = [b_{ij}]$ be the inverse of V so that $BV = I$. Multiplying B and V, the entries along the i-th row can be interpreted as the polynomial:

$$p_i(x) = \sum_{k=1}^{n} b_{ik} x^{k-1} = b_{i1} + b_{i2} x + b_{i3} x^2 + \cdots + b_{in} x^{n-1},$$

that interpolates the following points:

$$(x_1, 0), \cdots, (x_{i-1}, 0), (x_i, 1), (x_{i+1}, 0), \cdots, (x_n, 0).$$

Its Lagrange polynomial is:

$$p_i(x) = \prod_{\substack{1 \leq m \leq n \\ m \neq i}} \frac{x - x_m}{x_i - x_m} = \frac{x - x_1}{x_i - x_1} \cdots \frac{x - x_{i-1}}{x_i - x_{i-1}} \cdot \frac{x - x_{i+1}}{x_i - x_{i+1}} \cdots \frac{x - x_n}{x_i - x_n}$$

To match the coefficients of these two representations, we need an expression for the numerator of each b_{ij} term, which is a product of linear factors and whose sign depends alternately on its column. Define $f(i,j) = 1$ for $j = n$, and otherwise:

$$f(i,j) = \sum_{\substack{1 \leq m_1 < \cdots < m_{n-j} \leq n \\ m_1, \cdots, m_{n-j} \neq i}} x_{m_1} \cdots x_{m_{n-j}}, \quad j < n$$

Then the inverse of the Vandermonde matrix is:

$$(V^{-1})_{ij} = \frac{(-1)^{j+1} f(i,j)}{\prod_{\substack{1 \leq m \leq n \\ m \neq i}} (x_m - x_i)}$$

□

Appendix B

Product of Differences of the Roots of Unity

Theorem 10. *Let* $z_1 = 1, z_2 = e^{2\pi i\cdot 1/n}, z_3 = e^{2\pi i\cdot 2/n}, \cdots, z_n = e^{2\pi i\cdot (n-1)/n}$ *be the* n *roots of unity. Then:*

$$(z_2 - z_1)(z_3 - z_1)\cdots(z_n - z_1)(-1)^{n+1} > 0.$$

Proof. Case 1: Suppose n is odd. Then $(-1)^{n+1} = 1 > 0$ and each term $(z_k - z_1)$ of the product has a complex conjugate pairing $(\bar{z}_k - z_1)$. Let $z_k = c_k + d_k i$, where $0 < c_k, d_k < 1$ and $c_k^2 + d_k^2 = 1$. The product of each pairing will be:

$$(z_k - z_1)(\bar{z}_k - z_1) = (c_k + d_k i - 1)(c_k - d_k i - 1) = c_k^2 - 2c_k + d_k^2 + 1 = 2 - 2c_k = 2(1 - c_k) > 0.$$

And so the result is proved.

Case 2: Suppose n is even. Then one root $z_{(\frac{n}{2}+1)} = -1$ and $(-1)^{n+1} = -1$. So $(z_{(\frac{n}{2}+1)} - z_1)(-1)^{n+1} = 2 > 0$. The rest of the terms of the product come in complex conjugate pairings just as in Case 1, whose products are also positive. And so the result is proved. $\square$

Appendix C

Code

C.1 ode45.m

Code originates from [Com], so only (abbreviated) specifications provided:

```
function [tout,xout] = ode45(
          FUN,tspan,x0,pair,ode_fcn_format,tol,trace,count)
%
% Copyright (C) 2001, 2000 Marc Compere
% ode45.m is free software; you can redistribute it and/or
% modify it under the terms of the GNU General Public License
% as published by the Free Software Foundation; either version
% 2, or (at your option) any later version.
%
% INPUT:
% FUN - String containing name of user-supplied problem
%       description.
%       Call: xprime = fun(t,x) where FUN = 'fun'.
```

```
%       t      - Time (scalar).
%       x      - Solution column-vector.
%       xprime - Returned derivative COLUMN-vector;
%                xprime(i) = dx(i)/dt.
% tspan - [ tstart, tfinal ]
% x0    - Initial value COLUMN-vector.
% pair  - flag specifying which integrator coefficients to use:
%            0 --> use Dormand-Prince 4(5) pair (default)
%            1 --> use Fehlberg pair 4(5) pair
% ode_fcn_format - this specifies if the user-defined ode
%                  function is in
%    the form:  xprime = fun(t,x) (ode_fcn_format=0, default)
%    or:        xprime = fun(x,t) (ode_fcn_format=1)
%    Matlab's solvers comply with ode_fcn_format=0 while
%    Octave's lsode() and sdirk4() solvers comply with
%    ode_fcn_format=1.
% tol   - The desired accuracy. (optional, default:
%         tol = 1.e-6).
% trace - If nonzero, each step is printed. (optional, default:
%         trace = 0).
% count - if nonzero, variable 'rhs_counter' is initalized,
%         made global and counts the number of state-dot
%         function evaluations 'rhs_counter' is incremented
%         in here, not in the state-dot file simply make
%        'rhs_counter' global in the file that calls ode45
%
% OUTPUT:
% tout  - Returned integration time points (column-vector).
```

```
% xout  - Returned solution, one solution column-vector per
%         tout-value.
%
% The result can be displayed by: plot(tout, xout).
%
% Marc Compere
% CompereM@asme.org
% created : 06 October 1999
% modified: 17 January 2001
```

C.2 RandSphere.m

Code originates from [GH], so only (abbreviated) specifications provided:

```
function X=RandSphere(N,dim)
% RANDSPHERE
%
% RandSphere generates uniform random points on the surface of
% a unit radius N-dim sphere centered in the origin. This
% script uses different algorithms according to the dimensions
% of point:
%
%    -2D:  random generation of theta [0 2*pi]
%    -3D:  the "trig method".
%    -nD:  Gaussian distribution
%
% SYNOPSYS:
%
```

```
% INPUT:
%
%    N: integer number representing the number of points to
%       be generated
%    dim: dimension of points, if omitted 3D is assumed as
%         default
%
% OUTPUT:
%
%   X: Nxdim double matrix representing the coordinates of
%      random points generated
%
% Authors: Luigi Giaccari, Ed Hoyle
```

C.3 odesphere.m

Full code:

```
function odesphere(odefun,r,p,n,tspan)
% function odesphere(odefun,r,p,n,tspan)
% Will generate random points on the surface of an n-dimesional
% sphere of a given radius to be used as the inital conditions for
% orbits of a given ODE system. The data will be run through the ODE
% solver 'ode45' over a given time interval. odesphere will stop if
% (1) all points have been run through the solver, or (2) one point's
% orbit diverges to infinity. If all points are run through the
% solverm the program will display the count of the number of orbits
% that converge to the orgin.
```

```
%
% NOTE: Will call two library functions: ode45.m and RandSphere.m
%
% INPUT:
%   odefun: The address of the m-file representing the ODE system
%           (the format of the m-file is specified by the 'ode45.m'
%            function)
%   r: Radius of the n-dimensional sphere
%   p: Number of points to be generated on the n-dimensional sphere
%   n: Dimension of the sphere/ODE system
%   tspan: 1x2 matrix specifying the time inteval for the ODE solver

% Generate p random points on the surface of n-dim sphere of radius r
X = r*RandSphere(p,n);

% Initialize counter for orbits that converge to the origin
conv = 0;

% For each point, run as the initial condition through the ODE solver
% Convergence condition: Final position has norm < 10^(-5)
% Divergence condition: Final position has norm > 10^(1)
% "Between" condition: Does not advance counter, but continues
% Loop immediately stops if a divergent orbit is encountered
for k=1:p
    [T,Y] = ode45(odefun,tspan,X(k,:));
    [mm,nn] = size(Y);
    a = norm(Y(mm,:));
    if (a < 0.00001)
```

```
      conv = conv+1;
    elseif (a > 10)
      disp(['Diverge at iteration #: ',num2str(k)]);
      return;
    end
end

% Display counter of the orbits that converged
disp(['# Coverge: ',num2str(conv)]);
```

References

[Com] Mark Compere. ode45.m. Online Publication, 2001.

[GH] Luigi Giaccari and Ed Hoyle. Randsphere.m. Online Publication, No date provided.

[KL] Heinz-Otto Kreiss and Jens Lorenz. Resolvent estimates and quantification of nonlinear stability. *Acta Mathematica*, 16(1):1–20, 2000.

[Knu] Donald E. Knuth. *The Art of Computer Programming.* Addison-Wesley, 1997.

[Mey] Carl D. Meyer. *Matrix Analysis and Applied Linear Algebra.* Siam, 2000.

[TP] Morris Tenenbaum and Harry Pollard. *Ordinary Differential Equations.* Dover, 1963.

[Wat] David S. Watkins. *Fundamentals of Matrix Computations.* John Wiley and Sons, 2002.