Claremont Colleges
Scholarship @ Claremont

HMC Senior Theses | HMC Student Scholarship

2007

Analyzing Traveling Waves in a Viscoelastic Generalization of Burgers' Equation

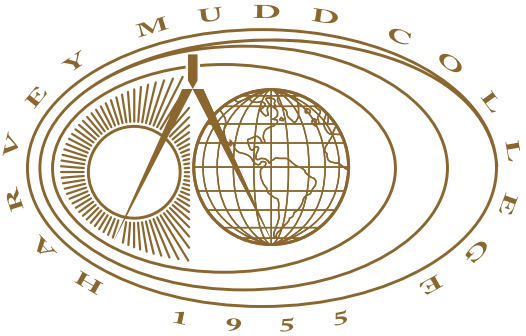
Victor Camacho
Harvey Mudd College

Follow this and additional works at: https://scholarship.claremont.edu/hmc_theses

Recommended Citation

Camacho, Victor, "Analyzing Traveling Waves in a Viscoelastic Generalization of Burgers' Equation" (2007). *HMC Senior Theses*. 193.
https://scholarship.claremont.edu/hmc_theses/193

This Open Access Senior Thesis is brought to you for free and open access by the HMC Student Scholarship at Scholarship @ Claremont. It has been accepted for inclusion in HMC Senior Theses by an authorized administrator of Scholarship @ Claremont. For more information, please contact scholarship@claremont.edu.



Analyzing Traveling Waves in a Viscoelastic Generalization of Burgers' Equation

Victor Camacho

Darryl Yong, Advisor

Jon Jacobsen, Reader

May, 2007

HARVEY MUDD
COLLEGE
Department of Mathematics

Copyright © 2007 Victor Camacho.

The author grants Harvey Mudd College the nonexclusive right to make this work available for noncommercial, educational purposes, provided that this copyright statement appears on the reproduced materials and notice is given that the copying is by permission of the author. To disseminate otherwise or to republish requires written permission from the author.

Abstract

We analyze a pair of nonlinear PDEs describing viscoelastic fluid flow in one dimension. We give a summary of the physical derivation and nondimensionlize the PDE system. Based on the boundary conditions and parameters, we are able to classify three different categories of traveling wave solutions, consistent with the results in [**?**]. We extend this work by analyzing the stability of the traveling waves. We thoroughly describe the numerical schemes and software program, VISCO, that were designed specifically to analyze the model we study in this paper. Our simulations lead us to conjecture that the traveling wave solutions found in [**?**] are globally stable for all sets of initial conditions with the appropriate asymptotic boundary conditions. We are able give some analytical evidence in support of this hypothesis but are unsuccessful in providing a complete proof.

Contents

List of Figures

Acknowledgments

I would like to first and foremost thank my advisor, Professor Darryl Yong, for his positive attitude, patience, and devotion to my personal developement through this research experience. His guidance, as well as that of Jon Jacobsen and Bob Guy, led me to discover much of what I have published in this thesis. His most notable technical contributions include discovering how to write the PDE in terms of Riemann invariants and a classification of all possible similiarity solutions of the PDE. I also owe much of what I know about numerical methods and scientific computing to his teachings.

I would also like to thank my summer reasearch advisor Jon Jacobsen who got me started on this project in the summer of 2006. He has taught me most of what I know about PDEs and dynamical system between the classes I have taken with him and our work together last summer. I am grateful that he and Professor Bob Guy have taken the time to submit our combined work to the SIAM Journal on Applied Mathematics.

Professor Bob Guy was also another major contributor to my work on this thesis. He has spent several hours with me explaining both his physical derivation of the model and the finite volume method he developed. These conversations inspired me to read Randall LeVeque's book on finite volume methods from which I learned a lot about numerical schemes for hyperbolic differential equations. I have used these tools in developing two of my own numerical schemes for the PDE system I studied in this thesis project.

I would like to thank the Baker Foundation for providing the funding necessary for Jon Jacobsen to hire me over the summer to begin my work on this project. These funds have also allowed me to purchase several relevent textbooks and have paid my travel expenses to two different mathematics conferences, one in which I presented my project.

Finally, I would like to thank Harvey Mudd College and the mathematics department for providing me with the necessary facilities I have needed to complete this thesis project. Furthermore, several professors in the department including Professor Andrew Bernoff and Professor Alfonso Cas-

tro have aided in my understanding of certain topics regarding my thesis project. In general, thank you too all those who have given me the opportunity to make the most of my thesis experience.

Chapter 1

Introduction

Viscoelastic fluids are a broad class of fluids that exhibit both viscous and elastic properties. Fluid elasticity is the measure of a fluid's tendency to return to its original shape in the absence of external forces, and fluid viscosity is the measure of its resistance to flow. Common examples of viscoelastic fluids include bio fluids, tempurpedic mattresses, egg white, and corn starch in water. What distinguishes viscoelastic fluids from Newtonian fluids, such as water, is that their viscosity changes when stress is applied. A Newtonian fluid has no elastic property and thus keeps a constant viscosity under shear stress[1].

The philosophy behind the model studied in this paper is to give a viscoelastic generalization to well-studied Newtonian fluid flow. This entire physical derivation is a basic summary of the more complete derivation which can be found in [**?**]. The Navier-Stokes equation[2] for incompressible fluid flow of Newtonian fluids in three dimensions is

$$\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} = \frac{\nabla P}{\rho} + \nu \Delta \mathbf{u}, \tag{1.1}$$

where $\mathbf{u}$ is the fluid velocity, P is the pressure and ν is the fluid viscosity. In one dimensional flow we may ignore the pressure gradient, so the above equation simplifies to

$$u_t + u u_x = \nu u_{xx}. \tag{1.2}$$

For Newtonian fluids, one makes the assumption that the shear stress on the fluid, σ, is proportional to the velocity gradient (i.e. $a u_x = \sigma$), and thus

[1] The shear stress on a fluid is the projection of the stress vector onto the direction of fluid flow.

[2] See [**?**] for a discussion of the physical derivation and history of the Navier-Stokes equations.

the shear stress does not need to be accounted for in (**??**). However, for viscoelastic fluids we must relax this assumption since the fluid viscosity changes with shear stress. We modify (**??**) to generalize the law of incompressible flow:

$$u_t + uu_x = \sigma_x. \tag{1.3}$$

What we need now is an equation governing the shear stress based on known parameters. In order to model the shear stress we construct a mechanical analog based on known physical properties of viscoelastic fluids. To account for both the viscous and elastic properties of a viscoelastic fluid, we assume that each fluid particle can be approximately represented by a spring and damper connected in series, known as a Maxwell element, as shown in Figure **??**. The strain on any object is calculated as the difference between its current length and rest length, divided by its rest length. In other words it is the percentage of the rest length by which the object is being stretched.

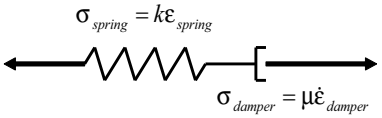


Figure 1.1: Maxwell Element: A spring and damper connected in series. In our model we assume this to be the mechanical analog to viscoelastic fluid particles. The spring constant represents the elasticity while the damper constant represents the polymeric viscosity.

In Figure **??**, ϵ denotes the strain and σ denotes the shear stress. The force exerted on the spring is proportional to the displacement from its rest length, and analogously the shear stress on the spring is proportional to the strain. Thus $\sigma_s = k\epsilon$, where k is the spring constant. It can also be shown that the shear stress on the damper is proportional to the strain rate so that $\sigma_d = \mu\epsilon_t$. When two objects are placed in series, the total strain is calculated as the sum of the strain on the two objects ($\epsilon = \epsilon_s + \epsilon_d$), and the total shear stress is equivalent to the shear stress on either object ($\sigma = \sigma_s = \sigma_d$). Using these two facts we find that

$$\lambda\sigma_t + \sigma = \mu\epsilon_t, \tag{1.4}$$

where $\lambda = k/\mu$. The parameter λ has the units of time represents the relaxation time of the fluid, that is, the relative timescale on which the fluid

recalls its original shape. We may further simplify equation (**??**) as follows. Let w denote the displacement of a Maxwell element from its resting length. Then the strain, ϵ, at any point is simply w_x. So the time derivative of the strain is,

$$\epsilon_t = w_{xt} = (w_t)_x = u_x.$$

So we can rewrite (**??**) as

$$\lambda\sigma_t + \sigma = \mu u_x. \tag{1.5}$$

One may observe that the above relation is not frame invariant, which is not physically acceptable. To account for the moving frame of the fluid we replace the time derivative of the stress, σ_t, with an upper convected time derivative. The one dimensional analog of this is

$$\sigma_t \to \sigma_t + u\sigma_x - \sigma u_x.$$

A precise explanation of why this works is beyond the scope of this paper but the inquisitive reader may refer to the derivation of this model in [**?**]. After making this substitution we now have a complete governing equation for the shear stress. So our generalization of incompressible fluid flow for viscoelastic fluids is given as

$$u_t + uu_x = \sigma_x \tag{1.6a}$$

$$\lambda(\sigma_t + u\sigma_x - \sigma u_x) = \mu u_x - \sigma. \tag{1.6b}$$

Recall that the parameter μ was the proportionality constant of the damper in our physical analog. In the context of fluids, it represents the polymeric viscosity[3].

One small test of validity for our model is to see that it works for Newtonian fluids. Newtonian fluids have no memory and thus the relaxation time is zero since the fluid will not return to its original shape without the help of external forces. Thus $\lambda = 0$ for Newtonian fluids. In this case, (**??**b) simplifies to $\mu u_x = \sigma$ which implies that the stress is equivalent to the velocity gradient. Recall that this is the exact assumption one makes when modeling Newtonian fluids. With this substitution, (**??**a) simplifies to the one dimensional version of the Navier-Stokes equation describing incompressible flow of Newtonian fluids, as desired. This paper investigates the various properties of (**??**) to test its validity in all other cases.

[3]The polymeric viscosity of a fluid is a measure of the friction that the individual particles experience when moving past each other, which is similar to but not the same as fluid viscosity, the fluid's resistance to flow.

1.1 Non-Dimensionalization

In this paper we shall study solutions to (**??**) that have asymptotically constant boundary conditions. In other words, we assume that

$$\lim_{|x|\to\infty} \sigma(x,t) = 0, \qquad \lim_{x\to-\infty} u(x,t) = u_L \qquad \lim_{x\to\infty} u(x,t) = u_R,$$

where u_L and u_R are real constants. To simplify the appearance of (**??**), we shall non-dimensionalize all of the variables. As demonstrated above, the units of each variable are

$$[u] = \frac{\text{length}}{\text{time}} \quad [\sigma] = \frac{\text{length}^2}{\text{time}^2} \quad [\mu] = \frac{\text{length}^2}{\text{time}} \quad [\lambda] = \text{time}.$$

First we shall prove translational invariance of the solutions. Suppose $u(x,t)$ and $\sigma(x,t)$ solve (**??**). Consider the moving coordinate system $(\zeta,\tau) = (x-\gamma t, t)$ and the functions $u^*(x,t) = u(\zeta,\tau) + \gamma$ and $\sigma^*(x,t) = \sigma(\zeta,\tau)$. We see that γ is a velocity, with the same units as u. Thus ζ has units length and τ has units time. We find that $u^*(x,t)$ and $\sigma^*(x,t)$ solve (**??**) as well. Substituting this into (**??**a) gives

$$\begin{aligned} u^*_t + u^* u^*_x &= \sigma^*_x \\ (u_t - \gamma u_x) + (u+\gamma)u_x &= \sigma_x \\ u_t + u u_x &= \sigma_x, \end{aligned}$$

which is true by assumption. Plugging into (**??**b) gives

$$\begin{aligned} \lambda(\sigma^*_t + u^*\sigma^*_x - \sigma^* u^*_x) &= \mu u^*_x - \sigma^* \\ \lambda((\sigma_t - \gamma\sigma_x) + (u+\gamma)\sigma_x - \sigma u_x) &= \mu u_x - \sigma \\ \lambda(\sigma_t + u\sigma_x - \sigma u_x) &= \mu u_x - \sigma, \end{aligned}$$

which is also true by assumption. Thus we may set γ arbitrarily and expect to observe the same behavior for solutions. One very convenient translation is $\gamma = -(u_L + u_R)/2$. The new boundary conditions in this case are

$$u^*_L = \frac{u_L - u_R}{2} \qquad u^*_R = \frac{u_R - u_L}{2}.$$

What is special about this case is that $u^*_L = -u^*_R$. We may therefore assume without loss of generality that solutions to (**??**) have the property that $u_L = -u_R$.

Next, we shall non-dimensionalize all of the variables in this problem. We do several rescalings to produce the following dimensionless variables denoted with an overline.

$$u = u_L \overline{u} \quad \sigma = u_L^2 \overline{\sigma} \quad x = \lambda u_L \overline{x} \quad t = \lambda \overline{t} \quad \mu = u_L^2 \overline{A}$$

Substituting these dimensionless quantities into (**??**) and simplifying gives the following dimensionless PDE.

$$\overline{u}_{\overline{t}} + \overline{u}\overline{u}_{\overline{x}} = \overline{\sigma}_{\overline{x}} \tag{1.7a}$$

$$\overline{\sigma}_{\overline{t}} + \overline{u}\overline{\sigma}_{\overline{x}} = (\overline{\sigma} + \overline{A})\overline{u}_{\overline{x}} - \overline{\sigma} \tag{1.7b}$$

Notice that λ was completely eliminated from the equation and that the dimensionless parameter $\overline{A}$ is defined as $\overline{A} = \mu / u_L^2$. Further, the boundary conditions become $\overline{u}_L = -\overline{u}_R = 1$. It is (**??**) with the aforementioned boundary conditions that shall be studied for the remainder of this paper.

Chapter 2

Traveling Wave Solutions

2.1 ODE Simplification

The goal of this section is to describe the traveling wave solutions of the PDE system introduced in the last chapter:

$$u_t + uu_x = \sigma_x \tag{2.1a}$$
$$\sigma_t + u\sigma_x = (A + \sigma)u_x - \sigma, \tag{2.1b}$$

subject to the boundary conditions

$$\lim_{x \to -\infty} u(x,t) = 1 \qquad \lim_{x \to \infty} u(x,t) = -1 \qquad \lim_{|x| \to \infty} \sigma(x,t) = 0,$$

where $A > 0$. Let us assume that there exists a traveling wave of speed c for both $u(x,t)$ and $\sigma(x,t)$ in the form $u(x,t) = U(\zeta)$ and $\sigma(x,t) = S(\zeta)$ where $\zeta \stackrel{\text{def}}{=} x - ct$. Plugging these functions into the above system yields

$$S' = U'(U - c) \tag{2.2a}$$
$$S'(U - c) = (A + S)U' - S. \tag{2.2b}$$

Integrating (**??**a) gives

$$S = \frac{U^2}{2} - cU + \theta, \tag{2.3}$$

where θ is a constant of integration. The boundary conditions of the PDE imply the following boundary conditions for U and S.

$$\lim_{\zeta\to\infty} U(\zeta) = -1 \tag{2.4a}$$

$$\lim_{\zeta\to-\infty} U(\zeta) = 1 \tag{2.4b}$$

$$\lim_{|\zeta|\to\infty} S(\zeta) = 0 \tag{2.4c}$$

For this model we assume that t is bounded so that $|\zeta| \to \infty$ implies that $|x| \to \infty$. By substituting these boundary conditions into (**??**) and solving the resulting system we find that

$$c = 0 \tag{2.5a}$$

$$\theta = -\frac{1}{2}. \tag{2.5b}$$

Substituting these quantities into (**??**) gives

$$S = \frac{U^2 - 1}{2}. \tag{2.6}$$

2.2 Phase Plane Analysis

Although S can be solved for completely in terms of U, a phase plane analysis of the system will provide some very useful information. By rearranging (**??**), we obtain

$$U' = -\frac{S}{S - (A - 1)} \tag{2.7a}$$

$$S' = -\frac{SU}{S - (A - 1)}. \tag{2.7b}$$

The equilibrium points of this system correspond with $S = 0$. We want both $(-1, 0)$ and $(1, 0)$ to be equilibrium points in the phase plane[1] since they represent the asymptotic boundary conditions.

We need to be careful about the possibility of the denominator of (**??**) being zero. This happens when $S = A - 1$. We know that the solution must obey (**??**), which is a quadratic function relating S and U. The equation

[1] In the phase plane, we shall always assume U to be on the horizontal axis and S on the vertical axis.

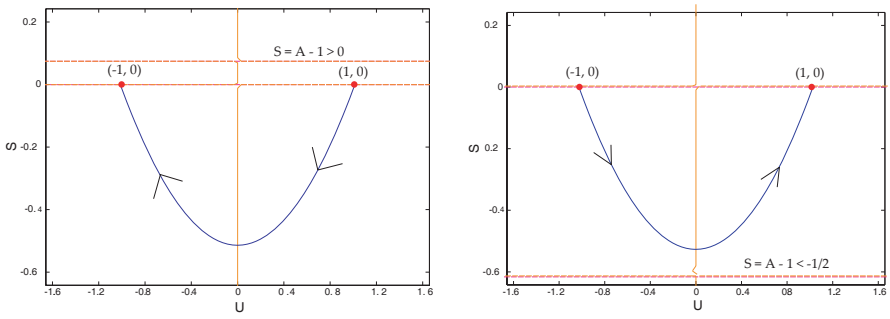


Figure 2.1: Phase portrait of the ODE system for two cases: (a) $A > 1$, (b) $0 < A < 1/2$

has roots at $(1, 0)$ and $(-1, 0)$, which correspond to the initial and terminal points of the solution trajectory. This quadratic function is concave up so to satisfy the boundary conditions the trajectory must only exist in the plane corresponding to $S < 0$, as shown in Figure **??**. The minimum value of this quadratic function is $S_{min} \stackrel{\text{def}}{=} -1/2$, which does not depend on A. Thus the range of possible S values in a valid solution is $[S_{min}, 0]$. Therefore, whenever $(A - 1) \in [S_{min}, 0]$, it will be possible for $S = A - 1$ which will result in a zero denominator of (**??**).

In the case when $A > 1$, all conditions are satisfied and there is no possibility for $S = A - 1$, as seen in Figure **??**a. When $S_{min} \leq A - 1 \leq 0$, or stated differently, $1/2 < A < 1$, there is a singularity in (**??**) and so a more careful analysis will be required in order to investigate this problem. When $0 < A < 1/2$ everything is satisfied except for the boundary conditions. As seen in Figure **??**b, the solution trajectory will begin at the point $(-1, 0)$ and terminate at $(1, 0)$, which is the reverse of what it should do. Thus $0 < A < 1/2$ does not satisfy the boundary conditions and so we will need to analyze this case more carefully as well. The three cases described above will form the basis for our classification of the types of traveling wave solutions of our model.

2.3 Classification of Traveling Waves

Recall from the first section of this chapter that S can be written completely in terms of U as shown in (**??**). Substituting this expression into (**??**b) and

simplifying, we obtain the following ODE for U:

$$U' = \frac{1 - U^2}{U^2 + 1 - 2A}. \tag{2.8}$$

The particular translation of the solution is not important so we arbitrarily choose to have the center of the wave at $\zeta = 0$. We therefore set $U(0) = 0$. Using this initial data we can implicitly solve (**??**) as

$$2(A-1)\log\left|\frac{U-1}{U+1}\right| + U + \zeta = 0. \tag{2.9}$$

In Figure **??** we have implicitly graphed U versus ζ for five different values of A which were chosen based on the results of the previous section. That is, we select one sample of A from each of the intervals $(0, 1/2)$, $(1/2, 1)$, and $(1, \infty)$, and we test the two boundary cases, $A = 1/2$ and $A = 1$. When $A > 1$ the implicit solution is analytic. When $A = 1$ the solution is the line $U = -\zeta$, which does not satisfy the boundary conditions. When $1/2 < A < 1$ the solution is multivalued in the central region and the boundary conditions are reversed. When $A = 1/2$ the solution has an infinite slope at $\zeta = 0$, which can be shown by implicitly differentiating (**??**). Finally, when $0 < A < 1/2$, the solution is analytic with reversed boundary conditions. This figure clearly marks the three categories of traveling wave solutions we want to study. We can describe the first category analytically by (**??**), but for the last two categories we will need find non-classical[2] solutions. In the next chapter we regularize (**??**) to determine the weak solutions for these two special categories.

2.4 Similarity Solutions

It may be natural to ask what other types of special solutions exist for the PDE. Similarity solutions can be thought of as solutions of the form

$$u(x,t) = x^a t f(\zeta) \tag{2.10a}$$

$$\sigma(x,t) = x^b t g(\zeta) \tag{2.10b}$$

where $\zeta \stackrel{\text{def}}{=} x^p t$, and a, b, p are real. After substituting these solutions into (**??**) we find that certain choices of a, b and p will not satisfy the PDE. In

[2]Non-classical solutions to a PDE are not differentiable and thus do not satisfy the PDE in its differential form. They are more often referred to as weak solutions and require a careful analysis of the system in order to demonstrate their existence and stability.

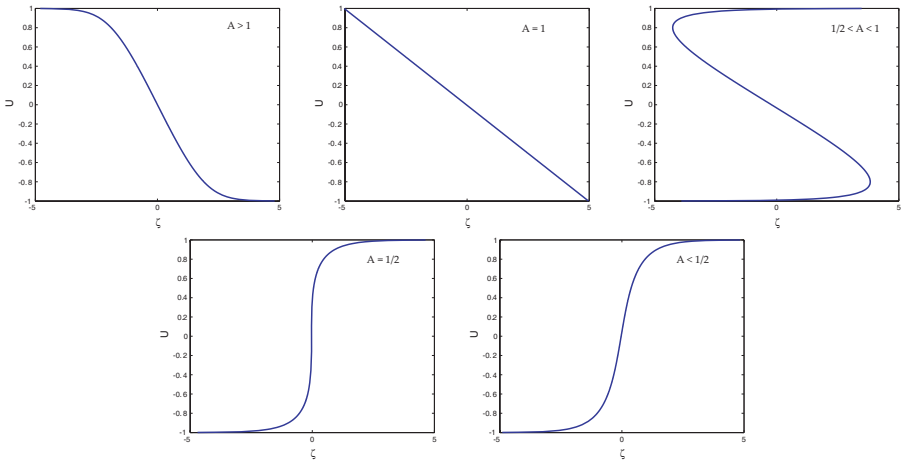


Figure 2.2: Implicit solutions to (**??**) for various values of A. (a) $A > 1$, (b) $A = 1$, (c) $1/2 < A < 1$, (d) $A = 1/2$, (e) $A < 1/2$.

fact, we found that the only possible similarity solutions for (**??**) have the form

$$u(x,t) = xf(t) \tag{2.11a}$$
$$\sigma(x,t) = g(t). \tag{2.11b}$$

Plugging these solutions into **??** gives

$$xf' + xf^2 = 0 \tag{2.12a}$$
$$g' = (g + A)f' - g. \tag{2.12b}$$

Solving (**??**) for f and g gives

$$f(t) = \frac{1}{t + C_1} \tag{2.13a}$$
$$g(t) = \exp\left(\frac{-t^2 - tC_1 + 1}{t + C_1}\right)\left(C_2 - A\int\left(\frac{\exp\left(\frac{t^2+tC_1-1}{t+C_1}\right)}{(t + C_1)^2}\right)dt\right) \tag{2.13b}$$

where C_1 and C_2 are integration constants. There is no closed form solution for $g(t)$.

Normally, these similarity solutions are used to understand the dynamics of non-classical solutions to a PDE. As will be demonstrated in the next

chapter, there do exist weak solutions to (**??**), however we were unable to find a good use for the similarity solutions to contribute to our analysis.

Another type of solution we tested were purely linear solutions that are invariant over time. We let $u(x,t) = mx$ where $m \in \mathbb{R}$. Then (**??**a) forces $\sigma(x,t) = m^2x^2/2 + C$ where C is an integration constant. Substituting these solutions into (**??**b) and simplifying gives

$$(m^3 + m^2)\frac{x^2}{2} + C - m(C + A) = 0.$$

Since m, C, and A are all constants, then we need to eliminate the x^2 term for this equation to be valid. By setting $m = 0$ or $m = -1$, the coefficient $(m^3 + m^2)$ diminishes. This is surprising because it shows that linear solutions are only possible with these two slopes. Substituting $m = -1$ forces the condition $A = -2C$. Thus, given a value for A, there is a unique translation of σ that will satisfy the PDE. For traveling wave solutions we found that $S = (U^2 - 1)/2$, in which case the term $C = -1/2$. This forces $A = 1$, which is precisely the value of A that we found to exhibit a linear traveling wave solution in the last section.

Chapter 3

Regularization

3.1 Introduction of Viscous Term

To understand the behavior of (**??**) when $A \leq 1$, we shall modify (**??**) so that its solutions are guaranteed to be smooth, and unique. Our model has non-analytic traveling wave solutions for bad values of A because we have not directly accounted for fluid viscosity. As shown in the introductory chapter, we have accounted for the polymeric viscosity μ and the relaxation time λ. Fluid viscosity will prevent an actual fluid from experiencing non-analytic behavior. To account for fluid viscosity in our system, we add a diffusion term to (**??**a). The governing equations become

$$u_t + uu_x = \sigma_x + \varepsilon u_{xx} \tag{3.1a}$$

$$\lambda(\sigma_t + u\sigma_x - \sigma u_x) = \mu u_x - \sigma, \tag{3.1b}$$

where ε is a positive constant. This new system has smooth and unique solutions because of the diffusion term. Further, solutions to (**??**) approach the behavior of solutions to (**??**) as $\varepsilon \to 0$.

We shall now non-dimensionalize this system in the same fashion as Section 1.1. If we change to the moving coordinate system $(\zeta, \tau) = (x - \gamma t, t)$ and translate $u(x,t)$ vertically by γ units so that $u^*(x,t) = u(\zeta,\tau) + \gamma$ and $\sigma^*(x,t) = \overline{\sigma}(\zeta,\tau)$, we find that $u^*_{xx}(x,t) = u_{\zeta\zeta}(\zeta,\tau)$. Substituting $u^*(x,t)$ and $\sigma^*(x,t)$ into (**??**) and simplifying gives

$$u_\tau + uu_\zeta = \sigma_\zeta + \varepsilon u_{\zeta\zeta} \tag{3.2a}$$

$$\lambda(\sigma_\tau + u\sigma_\zeta - \sigma u_\zeta) = \mu u_\zeta - \sigma. \tag{3.2b}$$

Since u and σ are assumed to be solutions to (**??**), then this shows that

$u^*(x,t)$ and $\sigma^*(x,t)$ are solutions to (**??**) as well. We may therefore assume without loss of generality that $u_L = -u_R$.

Now, non-dimensionalize the variables u, σ, x, t, λ and μ as they were in Section 1.1. Let $\varepsilon = \bar{\varepsilon}/(\mu\lambda)$ so that $\bar{\varepsilon}$ is dimensionless. Substituting these expressions into (**??**) gives

$$\overline{u}_{\bar{t}} + \overline{u}\overline{u}_{\bar{x}} = \overline{\sigma}_{\bar{x}} + \overline{\varepsilon u}_{\bar{x}\bar{x}} \tag{3.3a}$$

$$\overline{\sigma}_{\bar{t}} + \overline{u}\overline{\sigma}_{\bar{x}} = (\overline{\sigma} + \overline{A})\overline{u}_{\bar{x}} - \overline{\sigma} \tag{3.3b}$$

It is important to note that $\bar{\varepsilon} \to 0$ as $\varepsilon \to 0$.

3.2 Modified ODE

We will now perform the same traveling wave analysis on (**??**) as we did for (**??**) in Chapter 2. The traveling wave solutions are of the form $u(x,t) = U(\zeta)$ and $\sigma(x,t) = S(\zeta)$, where $\zeta = x - ct$. Substituting these solutions into (**??**) and simplifying yields

$$\varepsilon U' = \frac{U^2 - 1}{2} - S \tag{3.4a}$$

$$\varepsilon U S' = (S + A)\left(\frac{U^2 - 1}{2} - S\right) - \varepsilon S. \tag{3.4b}$$

The boundary conditions force $c = 0$. The singularities of this system will no longer occur as a result of a particular choice of A. When $U = 0$ there is a potential problem, but it will be dealt with by the end of this chapter. Recall that the old traveling wave solutions satisfied $S = (U^2 + 1)/2$, which need not be the case here. For simplicity of notation, we define

$$F \stackrel{\text{def}}{=} (U^2 - 1)/2 - S.$$

3.3 Phase Plane Analysis

In this section we shall conduct a phase plane analysis of the system (**??**). We rewrite it below in more concise notation.

$$U' = \frac{F}{\varepsilon} \tag{3.5a}$$

$$S' = \frac{(S + A)F - \varepsilon S}{U\varepsilon} \tag{3.5b}$$

The U-nullcline can easily be described as $F = 0$. When this condition is satisfied, (**??**b) shows that $S' = 0$ only if $S = 0$. Therefore, the only equilibrium points in the phase plane are $(1, 0)$ and $(-1, 0)$. The condition of the U-nullcline forces the exact same relationship between U and S that solved (**??**). There are two S-nullclines:

$$S_{\pm} = \frac{1}{2}\left(K - (A + \varepsilon) \pm \sqrt{(K - (A + \varepsilon))^2 + 4AK}\right), \tag{3.6}$$

where $K \stackrel{\text{def}}{=} (U^2 - 1)/2$. Observe that

$$\lim_{\varepsilon \to 0} S_+ = K \tag{3.7a}$$

$$\lim_{\varepsilon \to 0} S_- = -A. \tag{3.7b}$$

The limit (**??**a) implies that $F \to 0$. Thus the higher S-nullcline, denoted S_+, converges to the U-nullcline (which shall be denoted as S_U) as $\varepsilon \to 0$.

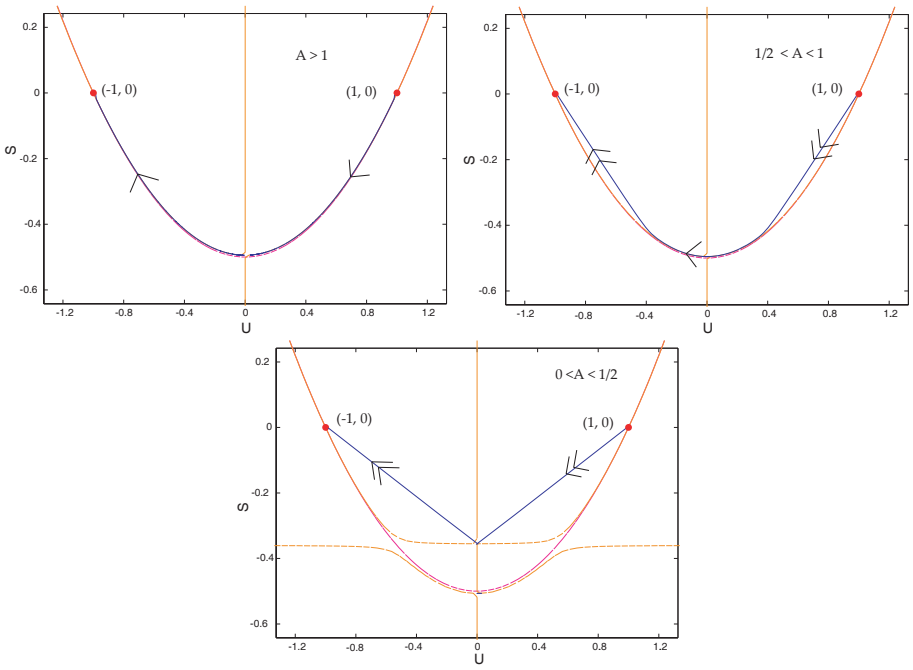


Figure 3.1: Phase portraits of the viscous system for various values of A where $\varepsilon = 0.001$. (a) $A > 1$, (b) $1/2 < A < 1$, (c) $A < 1/2$.

Figure **??** shows three different phase planes for the system (**??**) when ε is very small. In each graph, the curves S_+ and S_U are indistinguish-

able since they converge when ε is small. Notice the qualitative difference among the phase portraits for the three values of A. When $A > 1$, the solutions remains very close to S_U. When $1/2 < A < 1$, the solution travels linearly from $(1,0)$ to S_U, then travels close to the curve S_U, and finally travels linearly to $(-1,0)$. When $A < 1/2$, the solution travels linearly from $(1,0)$ to $(0,-A)$, and then linearly from $(0,-A)$ to $(-1,0)$, without ever approaching S_U. Each solutions trajectory is symmetric about the line $U = 0$.

3.3.1 Analysis of the Equilibrium Points

To better understand why these solutions look qualitatively different, we begin by studying their behavior near the equilibrium points. We denote the information corresponding to the initial equilibrium point, $(1,0)$, with a subscript L, and the terminal equilibrium point, $(-1,0)$, with a subscript R. The Jacobian matrices for this system, at each equilibrium point, are

$$J_L = \begin{bmatrix} \varepsilon & -\frac{1}{\varepsilon} \\ \frac{A}{\varepsilon} & -\frac{(A+\varepsilon)}{\varepsilon} \end{bmatrix} \qquad J_R = \begin{bmatrix} -\frac{1}{\varepsilon} & -\frac{1}{\varepsilon} \\ \frac{A}{\varepsilon} & \frac{(A+\varepsilon)}{\varepsilon} \end{bmatrix}. \tag{3.8}$$

The eigenvalues of these matrices are

$$\lambda_L^{\pm} = \frac{1}{2\varepsilon}\left(Q \pm \sqrt{Q^2+4\varepsilon}\right) \qquad \lambda_R^{\pm} = -\frac{1}{2\varepsilon}\left(Q \pm \sqrt{Q^2+4\varepsilon}\right), \tag{3.9}$$

where $Q \stackrel{\text{def}}{=} 1 - A - \varepsilon$. These eigenvalues are of opposite parity for all $\varepsilon > 0$. Therefore, each equilibrium point must be a saddle point. Since our solutions should begin at $(1,0)$ and end at $(-1,0)$, we will only concern ourselves with the unstable manifold[1] of $(1,0)$ and the stable manifold of $(-1,0)$. The associated eigenvectors of these eigenvalues of interest are

$$\vec{v}_L^{+} = \left[1, \frac{2-Q-\sqrt{Q^2+4\varepsilon}}{2}\right] \tag{3.10a}$$

$$\vec{v}_R^{-} = \left[1, -\frac{2-Q-\sqrt{Q^2+4\varepsilon}}{2}\right]. \tag{3.10b}$$

[1] Associated with a saddle equilibrium point are two curves, defined approximately by the eigenvectors, passing through that point. Trajectories traveling along one of the curves is attracted to the equilibrium point, corresponding to the positive eigenvalue, while the other repelled, corresponding to the negative eigenvalue. We call these two curves the stable and unstable manifolds respectively.

The limit of this eigendata as $\varepsilon \to 0$ is

$$\lim_{\varepsilon\to 0} \lambda_L^+ = \lim_{\varepsilon\to 0} \frac{Q + |Q|}{2\varepsilon} \tag{3.11a}$$

$$\lim_{\varepsilon\to 0} \lambda_R^- = \lim_{\varepsilon\to 0} -\frac{Q + |Q|}{2\varepsilon} \tag{3.11b}$$

$$\lim_{\varepsilon\to 0} \vec{v}_L^+ = \lim_{\varepsilon\to 0} \left[1, \frac{2 - (Q + |Q|)}{2}\right] \tag{3.11c}$$

$$\lim_{\varepsilon\to 0} \vec{v}_R^- = \lim_{\varepsilon\to 0} \left[1, -\frac{2 - (Q + |Q|)}{2}\right]. \tag{3.11d}$$

Each of these limits depend on the value of

$$\lim_{\varepsilon\to 0}(Q + |Q|) = \lim_{\varepsilon\to 0}(1 - A + |1 - A|).$$

Recall that we are doing this analysis to understand the behavior of the traveling wave solutions when $A \leq 1$. This condition forces

$$\lim_{\varepsilon\to 0}(Q + |Q|) = 2(1 - A).$$

Substituting this quantity into the above eigendata gives

$$\lim_{\varepsilon\to 0} \lambda_L^+ = \infty \tag{3.12a}$$

$$\lim_{\varepsilon\to 0} \lambda_R^- = -\infty \tag{3.12b}$$

$$\lim_{\varepsilon\to 0} \vec{v}_L^+ = [1, A] \tag{3.12c}$$

$$\lim_{\varepsilon\to 0} \vec{v}_R^- = [1, -A] \tag{3.12d}$$

This eigendata implies that, along the unstable manifold of $(1, 0)$ and the stable manifold of $(-1, 0)$, the solution travels infinitely fast in the phase plane within the locality of the equilibrium point, as $\varepsilon \to 0$.

3.3.2 Symmetry

From Figure **??** it appears that the solutions of (**??**) are symmetric about the line $U = 0$. If we are able to prove this, then it will only be necessary to study the phase plane in the region $U > 0$.

Let $T : \mathbb{R}^2 \to \mathbb{R}^2$ be a function that maps any point in the phase plane to its corresponding symmetric point about the line $U = 0$. One can easily

verify that $T(U,S) = (-U,S)$. Now make the variable transformation $V = -U$ and $W = S$, and reverse time by setting $\tau = -\zeta$.

Rewriting (**??**) in terms of this new variable transformation yields

$$V' = \frac{F}{\varepsilon} \tag{3.13a}$$

$$W' = \frac{(W+A)F - \varepsilon W}{V\varepsilon}, \tag{3.13b}$$

which is precisely the same governing equation as (**??**). So solutions to this system are symmetric about the line $U = 0$ and inverse in time. This proves the symmetry of solution to (**??**) about the line $U = 0$.

3.3.3 Trajectory Approximation Analysis

Let us focus our attention on the initial equilibrium point, $(1,0)$. The analysis from the Jacobian matrix showed that the solution travels very quickly along a linear path, starting from $(1,0)$. One natural question to ask is how far the solution will move away from the U-nullcline. The slope of S_U at $(1,0)$ is

$$\left.\frac{dS_U}{dU}\right|_{U=1} = U|_{U=1} = 1. \tag{3.14}$$

The local slope of the unstable manifold at $(1,0)$ is A. Figure **??** shows what happens when $A < 1$. The unstable manifold will initially travel away from the U-nullcline. We can set ε small enough so that the solution will continue its linear trajectory to a point outside of an ε neighborhood of the curve S_U.

Now we consider the solution's behavior in a region outside of an ε neighborhood of S_U. In this region, $|F| \gg \varepsilon$. With this approximation we can make the following simplification to (**??**), assuming that $|U| \gg \varepsilon$.

$$U' = \frac{F}{\varepsilon} \tag{3.15a}$$

$$S' \approx \frac{(S+A)F}{\varepsilon U} = \frac{S+A}{U}U'. \tag{3.15b}$$

From the second equation we find,

$$\frac{dS}{dU} = \frac{S'}{U'} \approx \frac{S+A}{U}. \tag{3.16}$$

This simple ODE can be solved analytically for S in terms of U as

$$S = MU - A, \tag{3.17}$$

where M is an integration constant. So we expect approximately linear behavior of the solution trajectory in a region far from S_U, when ε is small. If we assume that line passes through $(1,0)$, this forces $M = A$. Therefore this line will have precisely the same slope as the unstable manifold! This explains the linear behavior of the solutions in the phase plane when $A < 1$, as seen in Figure **??**.

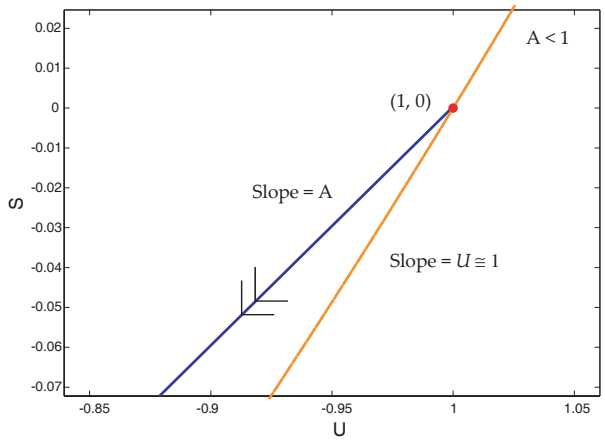


Figure 3.2: Behavior of solution far from the U-nullcline for when $A < 1$

We define the particular solution to **??** as $S_l \stackrel{\text{def}}{=} A(U-1)$. As $\varepsilon \to 0$, the speed at which the solution travels away from $(1,0)$ approaches infinity. From (**??**), it is clear that the solution will always travel quickly and linearly whenever $F \gg \varepsilon$. It then makes sense that linear pieces of the solution in the phase plane correspond to shocks, since $U(\zeta)$ and $S(\zeta)$ will appear to change instantaneously as $\varepsilon \to 0$.

In Figure **??**, it appears that once the solution re-approaches S_U, it remains close to it. In the next section we shall prove this rigorously. For now, given that S_l is the only piece of the solution corresponding to a shock, it is possible to calculate the height of the shock. The shock height is simply the difference between the two points at which S_l and S_U intersect. Setting $S_l = S_U$ gives

$$A(U-1) = (U-1)(U+1)/2.$$

By solving this equation we find that the two points of intersection are

$$\big(2A-1, 2A^2-2A\big) \quad \text{and} \quad (1,0).$$

Taking the componentwise difference between these two points gives the shock heights for U and S, respectively:

$$h_U = 2 - 2A \tag{3.18a}$$

$$h_S = 2A^2 - 2A. \tag{3.18b}$$

When $A < 1/2$, the second point at which S_l intersects S_U is beyond the line of symmetry, $U = 0$. In this case, the solution travels linearly to the point $(0, -A)$, and from there it travels linearly to the point $(-1, 0)$. The solution never slows down and so the entire shock ranges from $(1, 0)$ to $(-1, 0)$. Taking the componentwise difference between these points gives the shock heights:

$$h_U = 2 \tag{3.19a}$$

$$h_S = 0. \tag{3.19b}$$

From Figure **??** we observe that the solution passes through the point $(0, -1/2)$ when $1/2 < A < 1$. However, when $A < 1/2$ the solution will passes through $(0, -A)$. This will be proven rigorously in the next section. Figure **??** illustrates qualitative structure of the two classes of weak traveling wave solutions.

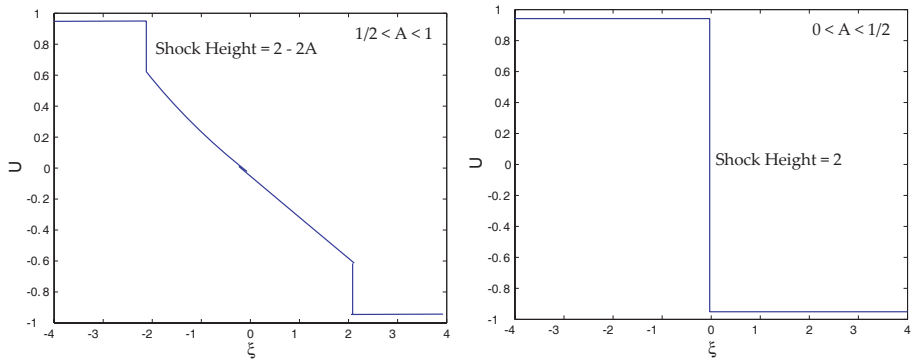


Figure 3.3: Qualitative structure of weak traveling wave solutions for two cases: (a) $1/2 < A < 1$, (b) $A < 1/2$

3.4 Trapping Region Arguments

In this section, we use several trapping region arguments in the phase plane to rigorously prove a few propositions of the previous section. Due to the

symmetry of the solutions, it will only be necessary to study the region in the phase plane corresponding to $U > 0$.

We begin by assuming that $1/2 < A < 1$. It will first be necessary to show that solutions are trapped above S_+ (see (**??**)). It suffices to show that $S_U < S_+$ when $U \in (-1, 1)$. Taking the difference between the two curves gives

$$S_+ - S_u = -\frac{1}{2}\left(K + A + \varepsilon - \sqrt{(K + A + \varepsilon)^2 - 2\varepsilon K}\right). \tag{3.20}$$

Recall that $K = (U^2 - 1)/2$. Notice that $K < 0$ when $U \in (-1, 1)$. The minimal value for K is $-1/2$, which implies that $K + A + \varepsilon > 0$ since $A > 1/2$. It follows that (**??**) must be positive which verifies that $S_+ > S_U$ when $U \in (-1, 1)$.

For all point lying above the curve S_U we know from (**??**) that $U' < 0$. Along S_+, the vector field has no vertical component. Therefore, any point existing above S_+ is trapped above this curve. So S_+ provides a lower boundary for solutions, as shown in Figure **??**.

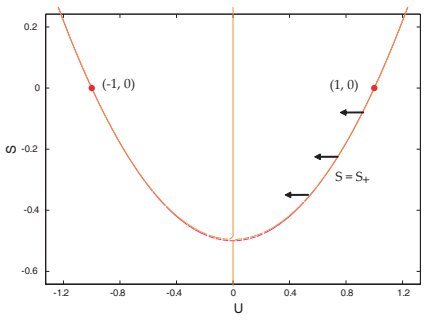


Figure 3.4: The curve S_+ provides a lower left bound for solution trajectories.

In the phase plane analysis of the previous section, we observed that the solution travels linearly from (1,0) to S_U, after which time it remains close to S_U. We will prove the latter property by constructing an appropriate trapping region. We have already constructed the bottom of this trapping region, S_+. We shall now construct the top.

Consider the curve defined as $S_\varepsilon \stackrel{\text{def}}{=} S_u + \varepsilon$. We first demonstrate that

$S_\varepsilon > S_+$ when $U \in (-1,1)$. Taking their difference and simplifying yields

$$\begin{aligned} S_\varepsilon - S_+ &= \varepsilon + \frac{1}{2}\left[K+\varepsilon+A-\sqrt{(K+\varepsilon+A)^2-4K\varepsilon}\right] \\ &> \varepsilon + \frac{1}{2}\left[K+\varepsilon+A-\sqrt{(K+\varepsilon+A)^2}\right] \\ &= \varepsilon \\ &> 0, \end{aligned}$$

as desired.

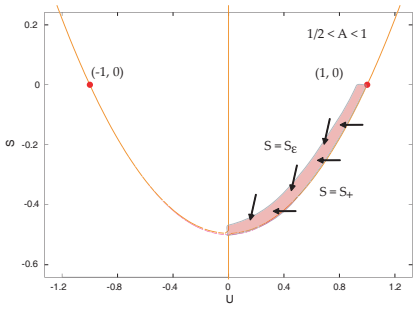


Figure 3.5: Trapping region between the curves S_+ and $S_{\sqrt{\varepsilon}}$

Since ε is small, then $\sqrt{\varepsilon} > \varepsilon$ Therefore, the curve $S_{\sqrt{\varepsilon}} \stackrel{\text{def}}{=} S_u + \sqrt{\varepsilon} > S_\varepsilon > S_+$ when $U \in (-1,1)$. The curve $S_{\sqrt{\varepsilon}}$ shall define the top of the trapping region, shown in Figure **??**. We have already shown that solutions must be trapped above S_+, so what remains to be shown is that they are trapped below $S_{\sqrt{\varepsilon}}$. We do this by demonstrating that the vector field along $S_{\sqrt{\varepsilon}}$ flows into the trapping region. That is, the slope of the vector field along the curve $S_{\sqrt{\varepsilon}}$ exceeds the slope of the curve itself. This can be expressed more formally as $dS/dU|_{S=S_{\sqrt{\varepsilon}}} > dS_{\sqrt{\varepsilon}}/dU$. Simplifying this inequality yields

$$U^2 < (S_{\sqrt{\varepsilon}} + A) + \sqrt{\varepsilon} S_{\sqrt{\varepsilon}}.$$

Letting $\varepsilon \to 0$ makes this inequality as restrictive as possible. Further simplification yields

$$0 < U < \sqrt{2A-1}, \tag{3.21}$$

as long as $A > 1/2$.

This trapping region must be valid for the non-linear part of the solution trajectory. That is, we would like it to be valid for $U \in (0, 2A-1)$. The above inequality shows that it is valid for $(0, \sqrt{2A-1})$. Observe that

$\sqrt{2A-1} > 2A-1$ as long as $1/2 < A < 1$, which is we have assumed. The vertical width of this trapping region depends on ε, so we can make it arbitrarily small by letting $\varepsilon \to 0$. This trapping region shows that once the solution re-approaches S_U, the solution is forced to remain close to S_U until $U = 0$.

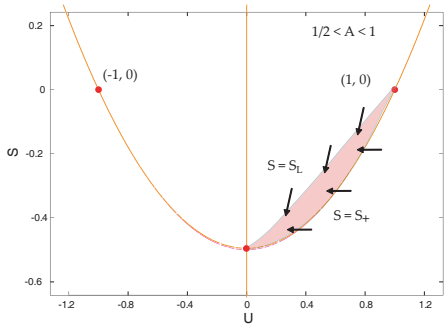


Figure 3.6: Trapping region between S_+ and the line $S_L = (U-1)/2$

Now we shall construct a trapping region that demonstrates that the solution trajectory passes through the point $(0, -1/2)$. As a lower bound we use S_+ once again. For the upper bound, we use the line passing through points $(1,0)$ and $(0,-1/2)$, as seen in Figure **??**. We define this line by $S_L \stackrel{\text{def}}{=} (U-1)/2$. To show that solutions are trapped below S_L, we need to verify the inequality $dS/dU|_{S=S_L} > dS_L/dU$. Simplifying this inequality gives

$$U > \frac{2\varepsilon}{(2A-1)}. \tag{3.22}$$

When ε small enough so that $0 < \varepsilon \ll 2A-1$, then this inequality simplifies to $U > O(\varepsilon)$.[2] So the trapping region is valid for $U \in (O(\varepsilon), 1)$, as we would like. Since ε can be made arbitrarily small, this trapping region shows that the solution trajectory must pass through the point $(0, -1/2)$. Our analysis has not been successful in demonstrating that the solution trajectory reaches the point $(0, -1/2)$ in finite time when $1/2 < A < 0$.

[2] In this case the symbol $O(\varepsilon)$ is meant to denote some term that is proportional to ε, and thus goes to 0 as $\varepsilon \to 0$.

Chapter 4

Characteristic Analysis

4.1 Characteristic Variable Transformation

Recall our original PDE system derived in Chapter 1:

$$u_t + uu_x = \sigma_x \tag{4.1a}$$
$$\sigma_t + u\sigma_x = (\sigma + A)u_x - \sigma. \tag{4.1b}$$

The characteristics of this PDE system are curves in (x,t)-space that show how information is being propagated. The functions u and σ may vary along these characteristic curves. In a hyperbolic system, such as (**??**), the speed at which information travels, and hence influences other points, must be finite. To determine the characteristic curves of (**??**) we will rewrite the system into its matrix-vector form, $\vec{q}_t + \mathbf{B}\vec{q}_x = \vec{b}$, and diagonalize. For our system we have

$$\vec{q} \stackrel{\text{def}}{=} \begin{bmatrix} u \\ \sigma \end{bmatrix} \qquad \mathbf{B} \stackrel{\text{def}}{=} \begin{bmatrix} u & -1 \\ -(A+\sigma) & u \end{bmatrix} \qquad \vec{b} \stackrel{\text{def}}{=} \begin{bmatrix} 0 \\ -\sigma \end{bmatrix}. \tag{4.2}$$

The eigenvalues of $\mathbf{B}$ and their associated left eigenvectors (i.e. the eigenvectors of $\mathbf{B}^T$) are

$$\lambda_\pm = u \pm \sqrt{A+\sigma} \tag{4.3a}$$
$$\vec{v}_\pm = \left[\sqrt{A+\sigma}, \mp 1\right]. \tag{4.3b}$$

Multiplying both sides of the PDE by $\vec{v}_+$ gives

$$\vec{v}_+(\vec{q}_t + \mathbf{B}\vec{q}_x) = \vec{v}_+\vec{b} \tag{4.4a}$$
$$\vec{v}_+(\vec{q}_t + \lambda_+\vec{q}_x) = \vec{v}_+\vec{b}. \tag{4.4b}$$

Dividing (**??**b) by $\sqrt{A+\sigma}$ gives

$$\begin{aligned}\left(u_t - \frac{\sigma_t}{\sqrt{A+\sigma}}\right) + (u+\sqrt{A+\sigma})\left(u_x - \frac{\sigma_x}{\sqrt{A+\sigma}}\right) &= \frac{\sigma}{\sqrt{A+\sigma}} \\ (u-2\sqrt{A+\sigma})_t + (u+\sqrt{A+\sigma})(u-2\sqrt{A+\sigma})_x &= \frac{\sigma}{\sqrt{A+\sigma}}.\end{aligned}$$

Similarly, following the same process with $\vec{v}_-$ gives

$$(u+2\sqrt{A+\sigma})_t + (u-\sqrt{A+\sigma})(u+2\sqrt{A+\sigma})_x = -\frac{\sigma}{\sqrt{A+\sigma}}.$$

The PDE written in this alternative form is rather useful. These equation look similar to a system of transport equations with a source term. To see this we define

$$L \stackrel{\text{def}}{=} u + 2\sqrt{A+\sigma} \tag{4.5a}$$

$$R \stackrel{\text{def}}{=} u - 2\sqrt{A+\sigma}, \tag{4.5b}$$

which is the proper change of variables for the characteristic analysis. Now the PDE system becomes

$$L_t + (u+\sqrt{A+\sigma})L_x = \frac{\sigma}{\sqrt{A+\sigma}} \tag{4.6a}$$

$$R_t + (u+\sqrt{A+\sigma})R_x = -\frac{\sigma}{\sqrt{A+\sigma}}. \tag{4.6b}$$

We know how to write L and R in terms of u and σ, but notice how it is also possible to write u and σ in terms of L and R. From the definitions we find that

$$u = \frac{L+R}{2} \tag{4.7a}$$

$$\sigma = \left(\frac{L-R}{4}\right)^2 - A. \tag{4.7b}$$

Rewriting the entire system completely in terms of L and R gives

$$L_t + \left(\frac{3R+L}{4}\right)L_x = \frac{(L-R)}{4} - \frac{4A}{(L-R)} \tag{4.8a}$$

$$R_t + \left(\frac{3L+R}{4}\right)R_x = -\frac{(L-R)}{4} + \frac{4A}{(L-R)}. \tag{4.8b}$$

The initial conditions for L and R can simply be transformed based on their definitions in terms of u and σ. The boundary conditions are

$$L(-\infty, t) = 1 - 2\sqrt{A} \qquad L(\infty, t) = -1 - 2\sqrt{A} \tag{4.9a}$$

$$R(-\infty, t) = 1 + 2\sqrt{A} \qquad R(\infty, t) = -1 + 2\sqrt{A}. \tag{4.9b}$$

We can study the characteristics in this transformed PDE system and understand how information propagates.

4.2 Characteristic Analysis

In any hyperbolic partial differential equation there is a certain way that information propagates in the system. Characteristic curves are curves in the (x, t)-plane, starting at $(x_0, 0)$, that move continually along a path that describes where information is being propagated to (see Figure **??**). In many cases, solutions to the system will be invariant along the characteristics. However, because of the source terms[1] in (**??**), the solutions will not necessarily remain constant along the characteristics. The transformation of the previous section allows us to analyze the characteristics of the system. Since there are two equations, and two distinct wave speeds,[2] we will need to define two families of characteristic curves. Namely,

$$C_\xi = \{(x_\xi(\xi), t_\xi(\xi)) \mid \xi \geq 0\} \qquad C_\eta = \{(x_\eta(\eta), t_\eta(\eta)) \mid \eta \geq 0\},$$

where ξ and η are the parameters for each family of characteristic curves. We can think of the functions L and R as functions of the parameters ξ and η:

$$L(x_\xi, t_\xi), \quad L(x_\eta, t_\eta), \quad R(x_\xi, t_\xi), \quad R(x_\eta, t_\eta).$$

The eigenvalues, expressed in (**??**a), describe the speed of propagation for the two characteristic curves. The characteristic speed of propagation is given by dx_ξ/dt_ξ and dx_η/dt_η, respectively. Therefore, these functions of ξ and η must satisfy the following set of differential equations.

$$\frac{dx_\xi}{dt_\xi} = \lambda_+ = \frac{3R + L}{4} \tag{4.10a}$$

$$\frac{dx_\eta}{dt_\eta} = \lambda_- = \frac{3L + R}{4} \tag{4.10b}$$

[1] In this context, the source terms refer to those terms in (**??**) that do not contain an x or t derivative

[2] The eigenvalues in (**??**a) are defined to be the two distinct wave speeds in the system.

Observe that we have the freedom to set $dt_\xi/d\xi$ and $dt_\eta/d\eta$ arbitrarily. We found that it is easiest to let

$$\frac{dt_\xi}{d\xi} = 1 \tag{4.11a}$$

$$\frac{dt_\eta}{d\eta} = 1. \tag{4.11b}$$

As mentioned earlier, L and R will not be constant along the characteristics. However, we can easily calculate their rate of change based on the dynamics of x_ξ, t_ξ, x_η and t_η. Using (**??**) we have

$$\frac{\partial L}{\partial \xi} = L_t \frac{dt_\xi}{d\xi} + L_x \frac{dx_\xi}{d\xi} = L_t + \left(\frac{3R+L}{4}\right) L_x = \left(\omega - \frac{A}{\omega}\right) \tag{4.12a}$$

$$\frac{\partial R}{\partial \eta} = R_t \frac{dt_\eta}{d\eta} + R_x \frac{dx_\eta}{d\eta} = R_t + \left(\frac{3L+R}{4}\right) R_x = -\left(\omega - \frac{A}{\omega}\right), \tag{4.12b}$$

where $\omega \stackrel{\text{def}}{=} (L-R)/4$.

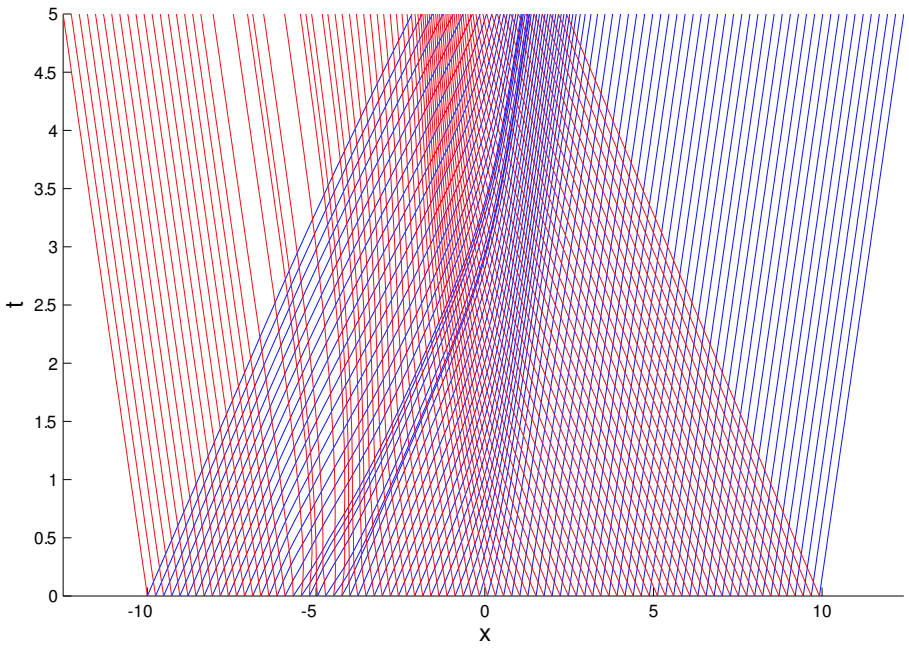


Figure 4.1: An example of a characteristic diagram for our system. Time is represented by the vertical axis and x on the horizontal axis.

Each individual characteristic starts at $t = 0$ and some unique value $x = x_0$, as seen in Figure **??**. From this we can uniquely determine the values of L and R, depending on how the initial conditions are defined. This characteristic information will lead us to design an alternative numerical scheme that solves for the characteristics and the solutions, L and R, at the same time.

Let us now make a few observations about the characteristics themselves. Suppose that L and R take on their boundary values, as seen in (**??**). We demonstrate here that the values along each of their respective characteristics will not change locally. On the left boundary we have $\omega = [(1 + 2\sqrt{A}) - (1 - 2\sqrt{A})]/4 = \sqrt{A}$, and on the right boundary we have $\omega = [(-1 + 2\sqrt{A}) - (-1 - 2\sqrt{A})]/4 = \sqrt{A}$ as well. So in either case we have

$$\frac{\partial L}{\partial \xi} = \left(\sqrt{A} - \frac{A}{\sqrt{A}}\right) = 0 \tag{4.13a}$$

$$\frac{\partial R}{\partial \eta} = -\left(\sqrt{A} - \frac{A}{\sqrt{A}}\right) = 0, \tag{4.13b}$$

as desired. This fact will be useful when we develop the method of characteristics in the next chapter.

Chapter 5

Numerical Algorithms

A helpful tool in our analysis of (**??**) has been our numerical approximations. A graphical user interface program, with many powerful features, uses each of the numerical schemes that will be described in this chapter. The inner-workings of the program itself will be described in Chapter 7.

5.1 Lax-Wendroff Method

Let us first describe some notation. We define the space and time intervals as $[0, T]$ and $[X_L, X_R]$[1], respectively. Both intervals are discretized into step sizes of Δx and Δt, respectively. Let x_i denote the ith spatial point in this discretization, starting from X_L, where $i = 1, 2, ...I$. Let t_n be the nth time point in the discretization starting from zero, where $n = 1, 2, ...N$. For shorthand we shall denote $u_i^n \approx u(x_i, t_n)$ and $\sigma_i^n \approx \sigma(x_i, t_n)$.

The program is designed to step forward through time and update the entire spatial interval at each step so as to simulate the solution. We use a Taylor approximation to approximate the value of the solution using known values from previous time steps:

$$u(x_i, t_{n+1}) = u(x_i, t_n) + \Delta t u_t(x_i, t_n) + \frac{\Delta t^2}{2} u_{tt}(x_i, t_n) + O(\Delta t^3) \tag{5.1a}$$

$$\sigma(x_i, t_{n+1}) = \sigma(x_i, t_n) + \Delta t \sigma_t(x_i, t_n) + \frac{\Delta t^2}{2} \sigma_{tt}(x_i, t_n) + O(\Delta t^3). \tag{5.1b}$$

[1] Although we would like our spatial interval to be infinite, this is impossible for a computer program to simulate. Thus we define X_L and X_R to be far enough away from the center of the solution so that asymptotic boundary conditions can be approximated. In fact, it suffices to simply make the interval larger than the domain of dependence for the segment of the solution that does not satisfy a boundary condition.

The t-derivatives of u and σ, can be expressed completely in terms of x-derivatives using (**??**). We modify (**??**) slightly so that the system is split into two easier subsystems[2]. The numerical algorithm will solve the two subsystems in sequence at each time step. The algorithm will first account for the effect of

$$u_t = \sigma_x - uu_x \tag{5.2a}$$
$$\sigma_t = (A + \sigma)u_x - u\sigma_x, \tag{5.2b}$$

and then accounts for the effect of

$$\sigma_t^* = -\sigma^*. \tag{5.3}$$

To calculate the effect of (**??**), we use the Taylor approximation given in (**??**). To calculate the effect of (**??**) we multiplying the result of the first substep by $e^{-\Delta t}$. This will be made more clear in a moment.

To complete the second order Taylor approximation in terms of only x-derivatives, we must calculate u_{tt} and σ_{tt}. We therefore differentiate both sides of (**??**) with respect to t and after repeated application of the chain rule and some further simplification, we obtain

$$u_{tt} = (A + \sigma + u^2)u_{xx} - 2u\sigma_{xx} - u_x\sigma_x + 2u(u_x)^2 \tag{5.4a}$$
$$\sigma_{tt} = (A + \sigma + u^2)\sigma_{xx} - 2u(A + \sigma)u_{xx} - (\sigma_x)^2. \tag{5.4b}$$

Now that we are able to express all t-derivatives completely in terms of x-derivatives, we may simplify (**??**) into an explicit method where the x-derivatives are calculated using centered first order finite difference approximations. Define the quantities

$$\theta_i^n \approx u_t(x_i, t_n)\Delta x \qquad \pi_i^n \approx \sigma_t(x_i, t_n)\Delta x$$
$$\Theta_i^n \approx u_{tt}(x_i, t_n)\Delta x^2/2 \qquad \Pi_i^n \approx \sigma_{tt}(x_i, t_n)\Delta x^2/2.$$

Observe that each of these quantities can be expressed in terms of the elements of the set

$$\{u_{i-1}^n, u_i^n, u_{i+1}^n, \sigma_{i-1}^n, \sigma_i^n, \sigma_{i+1}^n\}.$$

[2]In [**?**] this is referred to as the *fractional step method*. LeVeque shows that the method converges to the solution as $\Delta x, \Delta t \to 0$

Now our numerical method can be computed by the following steps:

$$u_i^{n'} = u_i^n + \frac{\Delta t}{\Delta x}\theta_i^n + \frac{\Delta t^2}{\Delta x^2}\Theta_i^n \tag{5.5a}$$

$$\sigma_i^{n'} = \sigma_i^n + \frac{\Delta t}{\Delta x}\pi_i^n + \frac{\Delta t^2}{\Delta x^2}\Pi_i^n \tag{5.5b}$$

$$u_i^{n+1} = u_i^{n'} \tag{5.5c}$$

$$\sigma_i^{n+1} = \sigma_i^{n'} e^{-\Delta t}. \tag{5.5d}$$

To calculate the solution values at the spatial boundaries we introduce ghost cells,[3] which are defined beyond the spatial boundaries. We set the value in these ghost cells to be the boundary values given for (**??**). This is justified because we're assuming asymptotically constant boundary conditions on both variables. The ghost cells allow us to use a centered finite difference approximation to approximate the x-derivatives on the spatial boundary.

The above method is not TVD[4]. However, we can always refine the grid so that it passes the CFL stability conditions[5]. The eigendata in (**??**) tells us the rate at which information travels in this system. To pass the CFL condition we need

$$\Delta x > \Delta t \max_{x\in[X_L,X_R]} \left| u(x,t) \pm \sqrt{A + \sigma(x,t)} \right| \tag{5.6}$$

As expected, there is minor oscillation near a shock solution that cannot be resolved regardless of the choice of Δx. The method passed the test for traveling wave solutions of all speeds and, based on many simulations, the traveling wave solution appear to be numerically stable. This will be explored further in the next chapter.

[3]Ghost cells are artificial spatial grid cells that are defined on the left and right spatial boundary. Since this model assumes asymptotically constant boundary conditions, we hold the ghost cells constant. See [**?**] for a more complete explanation of ghost cells.

[4]Total Variation Diminishing. Total variation is defined as TV = $\int_{X_L}^{X_R} |u_x| + |\sigma_x|\, dx$. The total variation of a numerical solution should diminish over time in order for the method to be considered TVD. For more details, see [**?**].

[5]The CFL condition is a condition that some numerical methods must pass in order to ensure that the numerical domain of dependence contains the analytic domain of dependence. In this numerical method, it does not make sense for a point to influence points that are outside of its range of influence. If the CFL condition were not passed it may result in the numerical method being highly unstable. See [**?**] for further details.

5.2 Method of Characteristics

The method of characteristics is a numerical method for computing solutions that utilizes the characteristic information of the solutions. To develop the algorithm, we draw from the results of the previous chapter. We define the space and time intervals as $[0, T]$ and $[X_L, X_R]$, respectively. Both intervals are discretized into step sizes of Δx and Δt, respectively. Let x_i denote the ith spatial point in this discretization, starting from X_L, where $i = 1, 2, ...I$. Let t_n be the nth time point in the discretization starting from zero, where $n = 1, 2, ...N$. For all times after the initial time, the spatial grid need not be uniformly spaced. In fact, at each time step, we shall redefine the grid. Let $x_\xi(i, n)$ represent the x-coordinate of the ξ characteristic originating from x_i, at time t_n. Likewise, let $x_\eta(i, n)$ represent the x-coordinate of the η characteristic originating from x_i, at time t_n.

We will keep track L along the ξ characteristic and R along the η characteristic. We do this because we can easily determine how they change along these characteristic by (**??**). When it is necessary to find R on the ξ characteristic or L on the η characteristic, they will be determined by interpolation. Since it is likely that the characteristics will travel together in a small region, we will keep track of the boundary values far outside of the domain of dependence. For clarity of notation, we will let $R_\xi(i, n)$ denote the value of R along the ξ characteristic originating from x_i, at time t_n. $R_\eta(i, n)$, $L_\xi(i, n)$ and $L_\eta(i, n)$ are denoted analogously. The pseudocode for the algorithm is shown below.

for $n = 1$ through $n = N$

$$\omega_\xi(i) = \frac{R_\xi(i,n) - L_\xi(i,n)}{4}$$
$$\omega_\eta(i) = \frac{R_\eta(i,n) - L_\eta(i,n)}{4}$$
$$\lambda_\xi(i) = \frac{3R_\xi(i,n) + L_\xi(i,n)}{4}$$
$$\lambda_\eta(i) = \frac{3L_\eta(i,n) + R_\eta(i,n)}{4}$$

$$x_\xi(i, n+1) = x_\xi(i, n) + \lambda_\xi(i)\Delta t$$
$$x_\eta(i, n+1) = x_\eta(i, n) + \lambda_\eta(i)\Delta t$$

$$L_\xi{}'(i, n) = \left(\omega_\xi(i) - \frac{A}{\omega_\xi(i)}\right)$$

$$R'_\eta(i,n) = -\left(\omega_\eta(i) - \tfrac{A}{\omega_\eta(i)}\right)$$

$$L_{\tilde{\xi}}(i,n+1) = L_{\tilde{\xi}}(i,n) + L'_{\tilde{\xi}}(i,n)\Delta t$$
$$R_\eta(i,n+1) = R_\eta(i,n) + R'_\eta(i,n)\Delta t$$

$$L_\eta(i,n+1) = \text{Interpolate } \{x_{\tilde{\xi}}(i,n+1), L_{\tilde{\xi}}(i,n+1)\} \text{ onto } x_\eta(i,n+1)$$
$$R_{\tilde{\xi}}(i,n+1) = \text{Interpolate } \{x_\eta(i,n+1), R_\eta(i,n+1)\} \text{ onto } x_{\tilde{\xi}}(i,n+1)$$

end loop

for $n = 1$ through $n = N$

$$L_i^n = \text{Interpolate } \{x_{\tilde{\xi}}(i,n), L_{\tilde{\xi}}(i,n)\} \text{ onto } x_\eta(i,1) \text{ for all } n$$
$$R_i^n = \text{Interpolate } \{x_\eta(i,n), R_\eta(i,n)\} \text{ onto } x_{\tilde{\xi}}(i,1) \text{ for all } n$$

end loop

The boundary conditions are incorporated into the interpolation steps. Because the $\tilde{\xi}$ characteristics moves to the right relative to the η characteristics, we know $x_\eta(i,n) < x_{\tilde{\xi}}(i,n)$ for all i and n. Thus for each time step n, we consider the interval

$$[x_\eta(1,n), x_{\tilde{\xi}}(I,n)]. \tag{5.7}$$

In particular, we calculate the length of (**??**) and multiply it by 2 which we set to the quantity B_n. Then, at the interpolation of L_η, we define the additional points $(x_\eta(1,n) - B_n, u_L + \sqrt{A})$ and $(x_{\tilde{\xi}}(I,n) + B_n, u_R + \sqrt{A})$. Likewise, at the interpolation of $R_{\tilde{\xi}}$, we define the additional points $(x_\eta(1,n) - B_n, u_L - \sqrt{A})$ and $(x_{\tilde{\xi}}(I,n) + B_n, u_R - \sqrt{A})$. We can define these points in this way because we learned in the previous chapter that L and R remain constant along the characteristics when they are both equal to the boundary conditions. In particular, this was shown in equation (**??**). It is assumed that the user will define an interval that is large enough so that, for the given initial condition, the asymptotic boundary conditions are met in the interval $[X_L, X_R]$. If that is the case, then because the points being added are far away from the domain of dependence, they will remain equal to

the boundary values along their characteristics. Thus it is safe for us to assume that, at any given time, the points being added take on the boundary values.

Using these numerical schemes, we verified the apparent global convergence of initial data to traveling wave solutions, which will be discussed in the next chapter.

Chapter 6

Stability and Convergence

6.1 The Global Convergence Conjecture

Using the numerical methods described in the previous chapter, we observed that all initial, defined as a perturbation to the traveling wave solution with compact support, seem to converge to some translation of the traveling wave solution. In particular, the solution will converge to the unique translation of the traveling wave solution such that the area between the perturbed solution and the said traveling wave solution, over the entire spatial domain, is equal to zero. The conjecture can be stated more formally as follows.

Conjecture: Let $u(x,t)$ and $\sigma(x,t)$ be solutions to the boundary valued problem (**??**) on the domain $x \in (-\infty, \infty)$ and $t \in [0, \infty)$. Let $u_w(x,t)$ and $\sigma_w(x,t)$ be traveling wave solutions (i.e. solutions to (**??**)) such that $\int_{\mathbb{R}} [u(x,0) - u_w(x,0)]\ dx = 0$. Then

$$\lim_{t\to\infty} \int_{\mathbb{R}} \left(|u(x,t) - u_w(x,t)| + |\sigma(x,t) - \sigma_w(x,t)|\right)\ dx = 0 \tag{6.1}$$

While we have been unsuccessful in proving this conjecture, we are able to show convergence and stability in a few special cases.

6.2 Perturbation Analysis

We now analyze the stability of these traveling wave solutions under slight perturbations with compact support. Let U and S represent traveling wave solutions to the system. Consider a solution to the system of the form

$u(x,t) = U(x) + \varepsilon(x,t)$ and $\sigma(x,t) = S(x) + \delta(x,t)$ where $\varepsilon(x,t)$ and $\delta(x,t)$ are defined to be zero everywhere except for when $x \in \Omega$ where $\Omega \subset \mathbb{R}$ is closed and bounded. Plugging $u(x,t)$ and $\sigma(x,t)$ into (**??**) and simplifying gives

$$\varepsilon_t + U\varepsilon_x + U'\varepsilon + \varepsilon\varepsilon_x = \delta_x \tag{6.2a}$$

$$\delta_t + U\delta_x + S'\varepsilon + \varepsilon\delta_x - U'\delta - S\varepsilon_x - \varepsilon_x\delta = A\varepsilon_x - \delta. \tag{6.2b}$$

We can simplify (**??**a) using the product rule to get

$$\varepsilon_t + (U\varepsilon)_x + \left(\frac{\varepsilon^2}{2}\right)_x = \delta_x, \tag{6.3}$$

which after integrating both sides with respect to x and rearranging some terms, becomes

$$\frac{\partial}{\partial t}\int_{\mathbb{R}} \varepsilon\, dx = \left[\delta - U\varepsilon - \frac{\varepsilon^2}{2}\right]_{-\infty}^{\infty} = 0. \tag{6.4}$$

Recall that ε and δ have compact support. Equation (**??**) shows that there is no change in the area of the perturbation, $\varepsilon(x,t)$, over time, regardless of its magnitude (see Figure **??**).

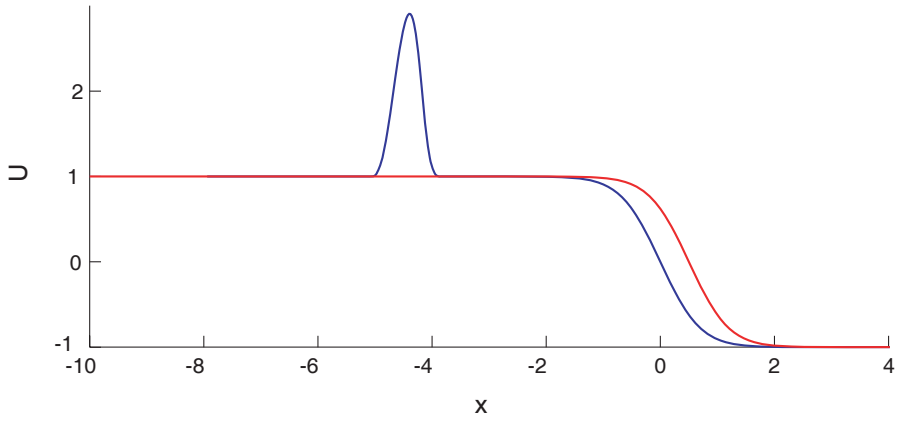


Figure 6.1: Example of a perturbation we have studied. The area of ε here is zero, and will remain zero for all time.

Now let us make the simplifying assumption that the perturbations,

and their derivatives, are extremely small. We could then simplify (**??**) to

$$\varepsilon_t + U\varepsilon_x + U'\varepsilon \approx \delta_x \tag{6.5a}$$
$$\delta_t + U\delta_x + S'\varepsilon - U'\delta - S\varepsilon_x \approx A\varepsilon_x - \delta. \tag{6.5b}$$

We make the second simplifying assumption that $U \approx u_0$ and $S \approx 0$, where u_0 is a constant, when $x \in \Omega$. This is possible if Ω is far away from the center of the wave so that the asymptotic boundary conditions force $u(x,t)$ and $\sigma(x,t)$ to be nearly constant. Now (**??**) can be simplified to

$$\varepsilon_t + u_0\varepsilon_x \approx \delta_x \tag{6.6a}$$
$$\delta_t + u_0\delta_x \approx A\varepsilon_x - \delta. \tag{6.6b}$$

After integrating on both sides with respect to x over the entire real line we find

$$\frac{\partial}{\partial t}\int_{\mathbb{R}} \varepsilon\, dx \approx 0 \tag{6.7a}$$
$$\frac{\partial}{\partial t}\int_{\mathbb{R}} \delta\, dx \approx -\int_{\mathbb{R}} \delta\, dx. \tag{6.7b}$$

We already know that (**??**a) is a strict equality from (**??**). However, notice that (**??**b) implies that the area of of the perturbation $\delta(x,t)$ is exponentially decaying over time. However, this is not sufficient to show convergence since the perturbations could be converging to anything with zero area. It would be more useful to see what happens to the integrals of ε^2 and δ^2 over time. We can analyze this by multiplying (**??**a) by ε and (**??**b) by δ and then integrating both equations with respect to x. After some simplification we obtain

$$\frac{\partial}{\partial t}\int_{\mathbb{R}} \varepsilon^2\, dx \approx -2\int_{\mathbb{R}} \delta\varepsilon_x\, dx \tag{6.8a}$$
$$\frac{\partial}{\partial t}\int_{\mathbb{R}} \delta^2\, dx \approx 2A\int_{\mathbb{R}} \delta\varepsilon_x\, dx - 2\int_{\mathbb{R}} \delta^2\, dx. \tag{6.8b}$$

The quantity $\delta\varepsilon_x$ makes this system rather difficult to analyze. However, there is one extremely important fact to take note of. The function ε^2 will diverge the fastest when $\delta \propto -\varepsilon_x$. However, when this is the case, notice that δ^2 will diminish as quickly as possible. In the reverse case, when $\delta \propto \varepsilon_x$, then ε^2 will diminish as quickly as possible but δ^2 may or may not diminish, depending on its current value. Notice that if ε^2 is small enough, then it would follow that ε_x is small, meaning that δ^2 will decay exponentially

over time. Therefore, if one can show from the equations that ε_x is approximately proportional to δ, then it would easily follow from the above argument that the traveling wave solutions are locally stable when u and σ are approximately constant.

6.2.1 Plane Wave Analysis

Using plane waves, we can show perturbations in the constant region of u and σ tend to diminish. Suppose that the functions $\varepsilon(x,t)$ and $\delta(x,t)$ take on the following forms

$$\varepsilon(x,t) = a(t)e^{ikx} \qquad \delta(x,t) = b(t)e^{ikx},$$

where k is the wave number of the plane wave and, $\varepsilon(t)$ and $\delta(t)$ are the amplitudes of the plane waves over time. Substituting these solutions into (**??**) and simplifying yields

$$a_t(t) = -iu_0ka(t) + ikb(t) \tag{6.9a}$$
$$b_t(t) = iAka(t) - (1 + iu_0k)b(t). \tag{6.9b}$$

To show that these amplitudes decay over time, we need only show that the real part of the eigenvalues of this system are negative. We calculate these eigenvalues to be

$$\lambda_\pm = -\frac{1}{2} \pm \left(\frac{1}{2}\sqrt{1 - 4Ak^2} - iu_0k\right).$$

Clearly, $\mathrm{Re}[\lambda_-] < 0$. However, for $\mathrm{Re}[\lambda_+] < 0$, we need $\sqrt{1 - 4Ak^2} < 1$, which is equivalent to the condition that $k^2 > 0$. This conditions must always hold since we have assumed that k is real. Therefore, the real part of both eigenvalues must always be negative. This result verifies the linear stability of (**??**) in regions where U and S are approximately constant.

Unfortunately, it is only in this simple linearized case that we are able to perform this sort of analysis. More complicated arguments are necessary to analyze the full system. In any case, this set of arguments sets a good foundation.

6.3 Measures and Energy Arguments

Another way to show convergence is to construct an energy for the system that diminishes over time. For simplicity of notation, we will define L_w and

R_w to be traveling wave solutions to (**??**). The measure

$$M(L,R) \stackrel{\text{def}}{=} (L-L_w)^2 + (R-R_w)^2, \tag{6.10}$$

satisfies $M(L_w,R_w) = 0$ and $M(L,R) \geq 0$ for all $x \in \mathbb{R}$. In particular, $M(L,R)$ must be non-zero somewhere if $(L,R) \neq (L_w,R_w)$. We shall define an energy function, $E(t)$, as follows:

$$E(t) \stackrel{\text{def}}{=} \int_{\mathbb{R}} M(L,R)\,dx. \tag{6.11}$$

To show convergence to a traveling wave solution, (L_w,R_w), we must to show that $\lim_{t\to\infty} E(t) = 0$. It will suffice to show that $E_t(t) < 0$ for all $t \in \mathbb{R}^+$. This inequality has been verified numerically (See Section 7.2.2).

$$\frac{d}{dt}E(t) = \int_{\mathbb{R}} \frac{d}{dt}M(L,R)\,dx \tag{6.12a}$$

So we calculate

$$\frac{d}{dt}M(L,R) = 2(L-L_w)(L-L_w)_t + 2(L-L_w)(L-L_w)_t \tag{6.13a}$$

$$= 2L_t(L-Lw) + 2R_t(R-R_w) \tag{6.13b}$$

Unfortunately, we were not able to simplify this quantity to anything useful, even when many assumptions were made about L and R.

The measure above was special because it appears as though its energy decrease monotonically given any initial perturbation (See Figure **??**). We have tested measures that do not necessarily decrease monotonically, instead they are bounded below some monotonically decreasing function (See Figure **??**). For example, we have tested the measure

$$M(\varepsilon,\delta) = (\delta - \varepsilon\sqrt{\sigma_w + A})^2 + (\delta + \varepsilon\sqrt{\sigma_w + A})^2.$$

The time derivative of the energy of this measure can be simplified to $E_t(t) = Q_1 + Q_2$, where

$$Q_1 = \int_{-\infty}^{\infty} \left[\left(\frac{3}{2}(u_w)_x - 1 \right) \delta^2 + \frac{(u_w)_x \varepsilon^2}{4}(u_w^2 + 1) \right] dx \tag{6.14a}$$

$$Q_2 = \int_{-\infty}^{\infty} \left[A\delta\varepsilon_x - u_w(u_w)_x\delta(\delta+\varepsilon) \right] dx. \tag{6.14b}$$

We can easily show that Q_1 is always negative. We must therefore show that Q_2 is smaller in absolute value than Q_1, or that Q_2 is negative. We

observe in our simulations that this is true under certain conditions, but we have been unsuccessful in proving anything. One may notice that Q_1 also contains a $\delta\varepsilon_x$ term, which we are not sure how to treat. Much of the analysis that we have done in this section relies on knowing the behavior of this particular product.

Chapter 7

The VISCO Software Package for Numerical Simulations

In this chapter we discuss the inner workings of the VISCO software package that was used to do much of the numerical simulations and numerical analysis of this PDE system. We begin with a general discussion of the user interface and then move on to explain some of the deeper functionality and how it can be used to analyze this PDE system.

7.1 User Interface

7.1.1 Buttons and Bars

To begin VISCO simply type the command `visco` at the command line prompt in Matlab. This will open a window similar to the one depicted in Figure **??**. All the buttons, text fields and checkboxes may seem a bit overwhelming at first, but they will all be explained in this section. Initially, all the fields are set to their default settings and the user can click the *Run* button in the top left corner of the screen to run the first simulation. Several things will happen after this button is clicked. Text at the very bottom of the window will change to indicate the status and the progress of the program. The status window is the text displayed at the bottom left corner of the screen. The text will indicate at what point in the simulation VISCO is currently at, such as *Calculating Initial Conditions* and *Running Simulation*. This feature is helpful because it can help the user determine if VISCO is frozen or if certain calculations or processes are taking a very long time. The *Running Simulation* step is the main part of the program which uses

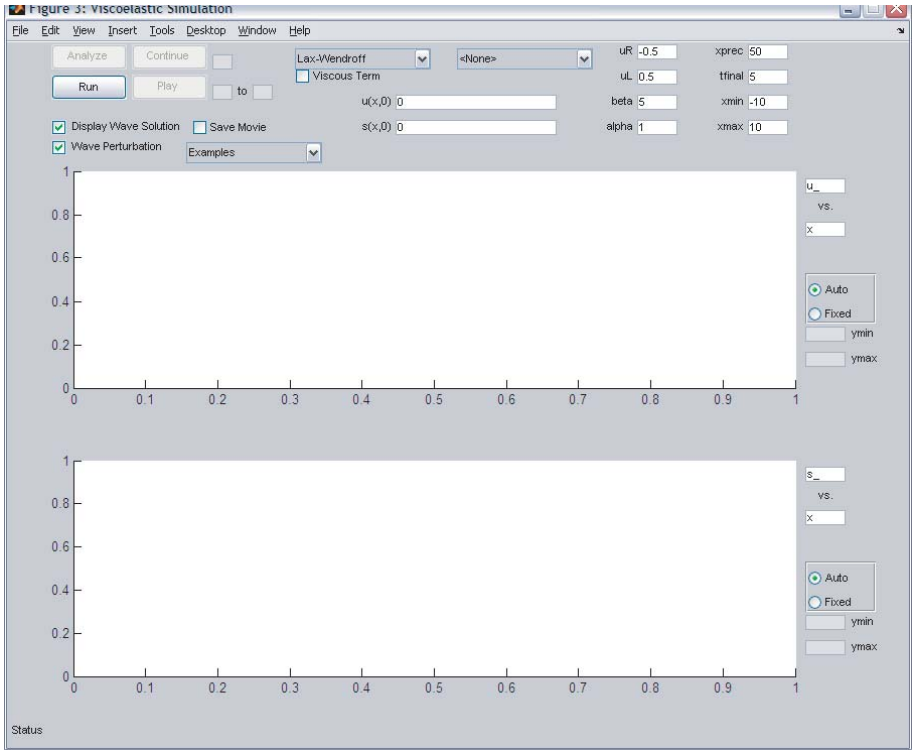


Figure 7.1: Basic screen shot of the startup screen.

the numerical algorithms explained in Chapter 5. This step normally takes the longest, and to keep track of its progress we display the percent to completion in the progress bar, located in the bottom right corner of the screen. When the simulation is complete the progress bar displays *Simulation Complete*. While the simulation is running, the status bar conveniently gives the estimated time for calculating the solution which is determined based on how long the first 200 time steps took.

Once this simulation is complete, several things will happen. The status bar reads *Playing Solution* and the two large set of axes in the window are animated and play the solution that was just calculated. The default solution is a simple traveling wave solution with $c = 0$ and hence does not move. In the default simulation, the solution was solved out to $t = 5$ and so the animation will continue to play until *Time* = 5. The *Time* is displayed on the title of each axis respectively. If the user wishes to pause the animation they may click anywhere on either of the axes.

Once the animation is complete, several more things will happen. The other three buttons in the figure, which shall be explained in a bit, become active. Furthermore, the status bar indicates the amount of memory that is currently being used. This is a courtesy to the user who should be conscience of memory allocation to VISCO. Memory is increased if either the solution grid is refined, the solution space is expanded, or the solution time is increased. Memory is being used after each simulation because the solution is stored so that the user may replay the animation. VISCO already takes many memory conserving measures, such as saving only the minimum number of time frames that are necessary to play a smooth animation. However, as a consequence, the user does not have control over the playback speed.

To playback the simulated solution, simply click the *Play* button. This feature always plays back the most recently simulated solution. There are two text boxes next to the play button which allow the user to specify the time window on which the solution is played. For example if the solution was simulated in the time domain $t \in [0, 5]$ and the user only wants to playback the part corresponding to $t \in [3, 4]$, then simply enter 3 into the first text box and 4 into the second. VISCO catches bad user input and displays an appropriate error message in the status window. All error messages are displayed with red text to draw the users attention to the problem. The default color of text in the status bar is black.

In some cases, the user may have mistakenly defined a time domain that was too small. Instead of recalculating the piece of the solution that has already been solved, the user has the option of calculating the solution in a larger time domain by appending to the current solution. By clicking the *Continue* button, VISCO uses the last time frame of the previous simulation as the initial conditions for the new simulation, while saving the old simulation. The new simulation runs for the amount of time specified in the text box next to the button. Once the simulation is complete, only the new part of the solution is played. To play back the entire solution the user must click the *Play* button.

There is one final button in the window, labeled *Analyze*, which shall be discussed in further detail later on.

7.1.2 Grid and Parameter Settings

The series of small text boxes in the top right corner of the window specify all of the parameters and grid settings. The purpose of most of these variables have been discussed in previous chapters, but we shall quickly

review some of them here.

- **uR:** Specifies the value of u_R as described in Chapter 1. More specifically, $u_R = \lim_{x\to\infty} u(x,t)$. In Chapters 2 through 6 we assumed $u_R = -1$.
- **uL:** Specifies the value of u_L as described in Chapter 1. More specifically, $u_L = \lim_{x\to-\infty} u(x,t)$. In Chapters 2 through 6 we have assumed that $u_L = 1$
- **beta:** Specifies the parameter value $\beta = 1/\lambda$, where λ has the same meaning as in (**??**). In Chapters 2 through 6 we assumed that $\beta = 1$ after non-dimensionalizing.
- **alpha:** Specifies the parameter value $\alpha = \mu/\lambda$ where μ and λ carry the same meaning as in (**??**). In Chapters 2 through 6 we have assumed that $\alpha^2 = A$.
- **xprec:** Defines the precision of the grid on which the solution will be calculated. In particular, *xprec* indicates the number of data points that are saved per unit of space (i.e., *xprec* = $1/\Delta x$). The precision for the time domain is calculated automatically by VISCO. The program chooses the smallest possible precision that will yield a stable solution using the particular algorithm. This is done to conserve memory and to ensure that the CFL conditions are satisfied. [Note: Increasing *xprec* will increase memory usage proportionally]
- **tfinal:** Specifies the size of the time domain. Essentially, the time domain on which the solution will be simulated is $t \in [0, \mathit{tfinal}]$. [Note: Increasing *tfinal* will increase memory usage proportionally]
- **xmin/xmax:** The spatial domain on which the solution will be solved is $x \in [\mathit{xmin}, \mathit{xmax}]$.

7.1.3 Initial Conditions

One of the more complicated aspects of running VISCO is defining initial conditions properly. Initial conditions can be defined in two ways: in terms of $u(x,0)$ and $\sigma(x,0)$, or in terms of perturbations to the traveling wave solution for the given parameter settings, that is $\varepsilon(x,0)$ and $\delta(x,0)$, as discussed in Chapter 6. To toggle between these two modes the user may use

the checkbox in the upper left corner of the screen labeled *Wave Perturbation*.

The user specifies the initial conditions in the boxes near the top of the window labeled $u(x,t)$ and $\sigma(x,t)$ or $\varepsilon(x,t)$ and $\delta(x,t)$, depending on the current mode. There are some important subtleties to be aware of when specifying the initial conditions, and several very useful features.

In the text box, the user enters an expression as if it were a Matlab command. The binary operator symbols have the same significance. The user may also enter any built in Matlab function such as `sin()` and `cos()` and built in Matlab variables such as `pi`. There are certain other functions that the user may enter which will be explained later. The letter `x` represents the independent spatial variable and may be entered as such into the expression. The letters `a` and `b` represent the parameters α and β, respectively. The letter `c` represents the quantity $(u_L + u_R)/2$, and the letter `d` is given by the quantity $|u_L - u_R|$. In Chapters 2 through 6 we have assumed that $c = 0$ and $d = 2$. To use the boundary conditions, the user may enter the variables `uL` and `uR` to represent u_L and u_R, respectively.

VISCO specific functions are very helpful in constructing initial conditions. We give a brief description of them here.

- **waveu**(`uR,uL,a,b,center`): This function outputs the traveling wave solution, $U(\zeta)$, given the parameter settings specified by the arguments. As demonstrated in previous chapters, there exists a unique traveling wave solution once u_L, u_R, α^2 and β have been specified. Here `a` represents α^2 and not α. The last argument, labeled `center`, specifies the particular translation of the traveling wave solution. In particular, `center` specifies the initial location of the center of the traveling wave solution with respect to the x-axis.

- **waves**(`uR,uL,a,b,center`): This function has the same purpose as *waveu* except it pertains to $S(\zeta)$. That is, the traveling wave solution for $\sigma(x,t)$. Again, this can be uniquely determined based on u_L, u_R, α^2 and β. The argument labeled `center` specifies the initial location of the center of the traveling wave solution with respect to the x-axis.

- **blob**(`start, finish`): This function outputs the hump of a sine wave so that it can transition smoothly with a surrounding flat function. The width of the hump is determined by the input variables `start` and `finish`. The hump function is defined to be zero everywhere except within the interval [`start, finish`]. The height of the hump is

set to be of unit height by default, however this height may be adjusted by multiplying the function by any scalar.

VISCO makes it possible to define a piecewise function in the following way. Specify an expression immediately followed by `$start,finish$`. Doing this will set everything outside the interval [`start,finish`] to zero, regardless of the expression that came before it. Then to define another piece of the piecewise function, simply add a second expression after the dollar sign, followed by another interval of the form `$start2,finish2$` which defines the interval on which the second expression is defined. In general the input should have the form

```
expression1 $start1,finish1$ expression2 $start2,finish2$...
```

In this way the user may define a piecewise function. Note that it is possible for the intervals to overlap, what happens in this case is that the functions are simply added. If there is a syntax error when specifying the initial conditions such as too few arguments in a function or not enough dollar signs etc., an error message will be displayed in the status bar.

7.1.4 Axis Settings

Figure 7.2: This is a close up view of the section of the startup screen where the user may adjust the axes settings.

One of the more powerful features of VISCO is the ability to adjust the variables that are being tested on each axis. On the right hand side of each axis there are two activated text boxes, two radio buttons and two inactive text boxes as shown in Figure **??**. The active text boxes display what variable is being represented on the x and y axes of each graph. There is a wide range of possibilities for what can be entered into these boxes. By default, the top axes displays $u(x,t)$ against x and the bottom axis displays $\sigma(x,t)$ against x. Here are some more options.

- `u_` = $u(x,t)$
- `s_` = $\sigma(x,t)$
- `x` = x
- `t` = t
- `e_` = $\varepsilon(x,t)$
- `d_` = $\delta(x,t)$
- `a` = α
- `b` = β
- `uL` = u_L
- `uR` = u_R
- `c` = c (Recall that $c = (u_L + u_R)/2$)
- `d` = d (Recall that $d = |u_L - u_R|$)

What's more amazing is that any function of any combination of these variables can be used as one of the axis settings (e.g. `c*u_ + a*s_*d_`). Further, any built-in Matlab function can be applied to any of these variables, and several functions are provided by VISCO for the users convenience. These include the following.

- **dt**(`function`) = takes the t derivative of `function` which must be a function in terms of x and t (e.g. `dt(u_ + 3*e_)`).
- **dtt**(`function`) = takes the second t derivative of `function`.

- **dx**(`function`) = takes the x derivative of `function`.
- **dxx**(`function`) = takes the second x derivative of `function`.

The user may also adjust the viewing window for the graph that is being plotted. By default, the radio button *auto* is selected which means that the axis settings will be set as tightly as possible. To change the axis settings, select the *fixed* radio button. Notice that the text boxes *ymax* and *ymin* then become activated. It is assumed that the user would like to view the entire x-axis, so this option is only relevant for adjusting the viewing window of the y-axis. Enter the lower bound of the y-axis in *ymin* and the upper bound in *ymax*. To return to the auto axes mode, select the *auto* radio button and VISCO will disregard the values entered in *ymin* and *ymax*.

7.1.5 Analyze Button

As mentioned in the above, there is one button that remains to be discussed, and it deserves its own subsection. After a simulation is complete and the solution has been stored, the user may press the *Analyze* button. This action will open up a new window called *Analysis*, as seen in Figure **??**. The main features of this window are two text boxes, a button, a pull-down menu, and a large set of axes. The pull-down menu sets the mode for the window of which there are two. Namely, *Measure* mode and *Characteristic* mode. We shall begin by describing *Measure* mode.

Measure mode, is used to graph one or two different measures of the solution over time, where measure has the same meaning as in Chapter 6. The user may enter two different measures in the two text boxes to compare them on the same set of axes. To define a measure, the user has access to the same pre-defined variables described in the previous subsection. In addition, the user may access the following quantities.

- `uw_` = the traveling wave solution for $u(x,t)$.
- `sw_` = the traveling wave solution for $\sigma(x,t)$
- `L` = the function $L(x,t)$ as described in Chapter 4.
- `R` = the function $R(x,t)$ as described in Chapter 4.
- `Lw` = the traveling wave solution for $L(x,t)$.
- `Rw` = the traveling wave solution for $R(x,t)$.

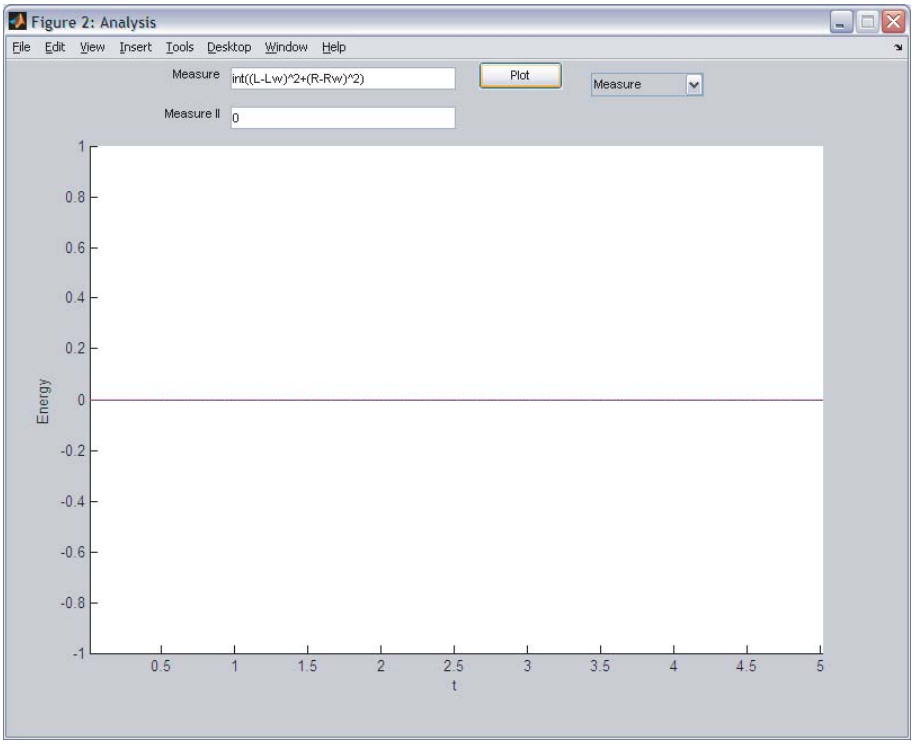


Figure 7.3: Basic screen shot of the analysis screen in *Measure* mode.

- **int**(`function`) = takes the integral of `function` with respect to x (e.g. `int(`$L^2 + R^2$`)`).

The measure is the part defined inside the integrand while the energy is the integral of the measure. In the text boxes the user must enter the energy. So the command should read `int(measure)` where `measure` is the measure that the user specifies. The user is not required to enter two energies since the second text box is allowed to be left blank. After entering the measures correctly, click the *Plot* button and the energies will be plotted on the large set of axes. The energy specified in the upper text box is the blue curve while the energy specified by the lower text box is the red curve. In general this plotting tool is most useful in checking whether energies converge to zero over time.

To switch to characteristic mode, select *Characteristics* in the popup menu. After doing this, several things will happen as shown in Figure **??**. The label on the first text box is changed to *Refinement*. The second text box dis-

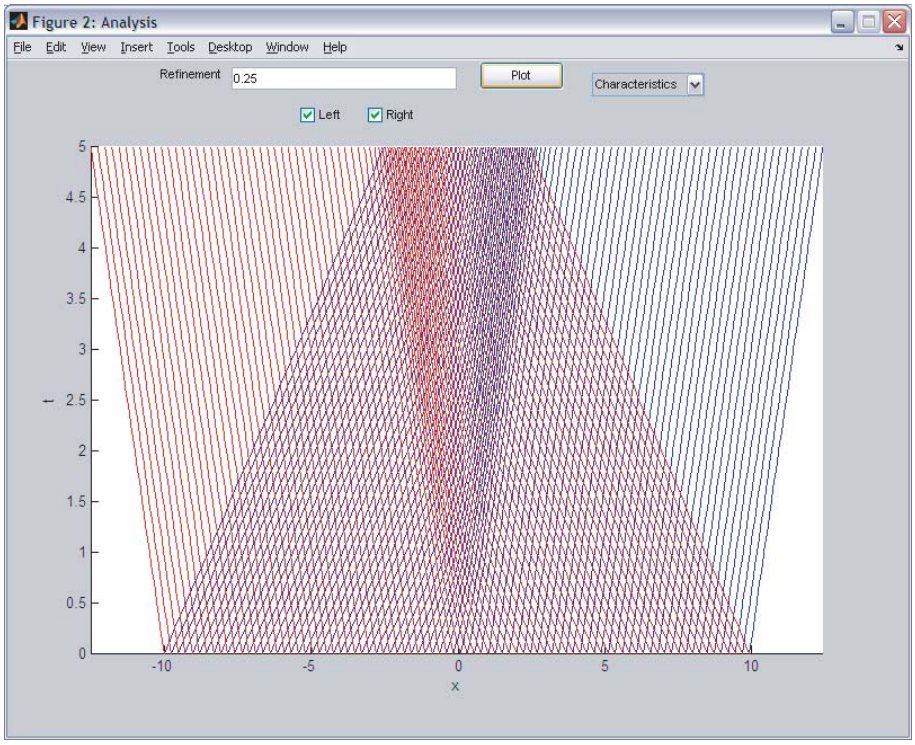


Figure 7.4: Basic screen shot of the analysis screen in *Characteristic* mode.

appears and is replaced with two checkboxes labeled *Left* and *Right*. The purpose of this mode is to graph the characteristic curves of the solution as described in Chapter 4. The *refinement* is how spaced apart each sample characteristic will be. So a smaller refinement implies that more characteristic curves will be plotted. As mentioned in Chapter 4, there are two characteristic curves emerging from each initial condition which we referred to as left and right characteristics. When the checkbox *Left* is checked, the left moving characteristics will be displayed in red. When the checkbox *Right* is checked, the right moving characteristics will be displayed in blue on the same set of axes. When the user reaches a desired setting, click the *Plot* button and watch the magic! The characteristic curves for all initial conditions between *xmin* and *xmax* will be calculated and plotted right before your eyes. This is one of the more beautiful aspects of VISCO. While it is not put into good use in this thesis, it is certainly worth mentioning.

7.1.6 Other Settings and Options

There are a few more settings and options in the main window that have not yet been discussed. At the top center of the window there are two popup menus, each of which specify a numerical algorithm that will be used to simulate a solution. The option <*None*> is available in the second menu in case the user only wishes to run only one simulation at a time. The purpose of having two choices here is to compare two different numerical algorithms on the same set of axes. Two of the three numerical algorithms available in the menu are *Lax-Wendroff* and *Characteristic*. These are the same two algorithms that were described in Chapter 5. The third algorithm, labeled *Finite Volume*, is a finite volume method written by Bob Guy. In general, Guy's algorithm is first-order accurate and tends to smooth out shocks in order to prevent the oscillations observed from using the Lax-Wendroff scheme. If the user decides to plot two different algorithms, the first algorithm is plotted using solid line(s) while the second algorithm is plotted with dotted line(s).

There is a third pull down menu in the window labeled *Examples*. There are several pre-defined example simulations that the user can select and then run. When selecting an option from this menu, all of the parameters, grid settings, initial conditions etc. are adjusted automatically so that so user need only click the *Run* button. These particular examples give a general overview of the types of perturbations that were simulated in order to convince us of the global convergence conjecture explained in Chapter 6.

Finally, there are three checkboxes that have yet to be explained. The one labeled *visco* in the top center of the screen tells VISCO whether or not to simulate viscous solutions. This is basically a toggle between having VISCO simulate solutions to (**??**) as opposed to (**??**). When the checkbox is active, a text box will appear next to the text box where the user needs to specify the desired value for ε, where in this context, ε takes on the same meaning as it did in Chapter 3. The checkbox labeled *Save Movie* will save the most recent playback to a .avi movie file called `visco01.avi`. VISCO does not update the name of the file to avoid overwriting files, so the user must manually copy or rename the movie file once a desirable movie has been created. The *Display Wave Solution* check box allows the user to decide whether the traveling wave solutions should be displayed on the same set of axes as the simulated solution. In some cases it is desirable to compare the simulated solution to the corresponding traveling wave solution, other times it is not. The particular translation of the traveling wave solution is chosen so that the initial conditions will converge to it as $t \to \infty$, assuming

the global convergence conjecture is true.

7.2 Contribution to Analysis

In this section we explain how VISCO has contributed to our analysis of this problem. There are several important problems that will be discussed separately.

7.2.1 Traveling Wave Solution Analysis

One of the most important uses for VISCO was to numerically verify the shape and behavior of the traveling wave solutions. In a way, this was also a good test for the numerical scheme, since at least for $A > 1$ we are certain of the shape of the traveling wave solutions. When entering a traveling wave solution as the initial conditions with boundaries $u_L = 1$, $u_R = -1$, and $A > 1$, we find that the solution is invariant over time. This is verified for all three available numerical schemes. When translating u_L and u_R up by one unit, that is $u_L = 2$, $u_R = 1$, we find that the solution now translates to the right with a speed of one unit per time, as expected.

The challenge of any numerical scheme is to properly deal with shocks. While the Lax-Wendroff method is very accurate, it is not TVD and thus oscillations will naturally develop near shocks in the solution as seen in Figure **??**a. The finite volume method is TVD and will not develop such oscillations but instead smooth out the shock as seen in Figure **??**b. So there is a small degree of trade-off between accuracy and oscillation around the shocks. In this way, the two methods are good at reinforcing each other since the accuracy of one can be checked against the other.

When the traveling waves that contain shocks are used as the initial condition for a simulation, they will change slightly over time, however, the user must be careful to check for numerical inaccuracies. In Figure **??**, a pure shock solution was used as the initial condition, corresponding to $0 < A < 1/2$. Ideally, we would have wanted this solution to remain a shock for all time, however something different happens with each method. We expect the Lax-Wendroff method to produce the oscillations seen in Figure **??**a. Like-wise, we expect the finite volume method to smooth out the shock somewhat, as seen in Figure **??**b. So in fact, this small change in form observed by both numerical schemes does not prove that the pure shock is not a traveling wave, instead we interpret this as expected numerical inaccuracy. One may observe that although the finite volume method causes

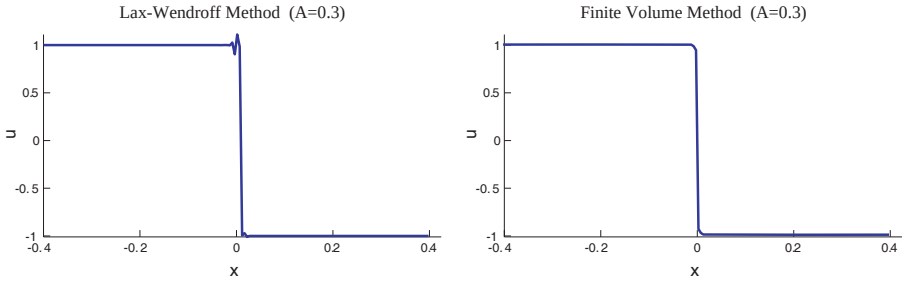


Figure 7.5: These graphs depict typical behavior of our two numerical methods near shocks. For this example $A = 0.3$. (a) Lax-Wendroff Method: Notice the small oscillations near the shock. (b) Finite Volume Method: Notice how the sharp corners have been rounded off.

the solution to change initially, eventually it will converge to a smooth curve, which is very close in appearance to the original shock. A good test for numerical the numerical inaccuracy of this method is to see what happens as *xprec* is made larger. If the smooth convergent solution appears to converge to the shock as *xprec* is made very large, then this is a good indication of numerical inaccuracy and convincing evidence that the actual solution should indeed be a shock.

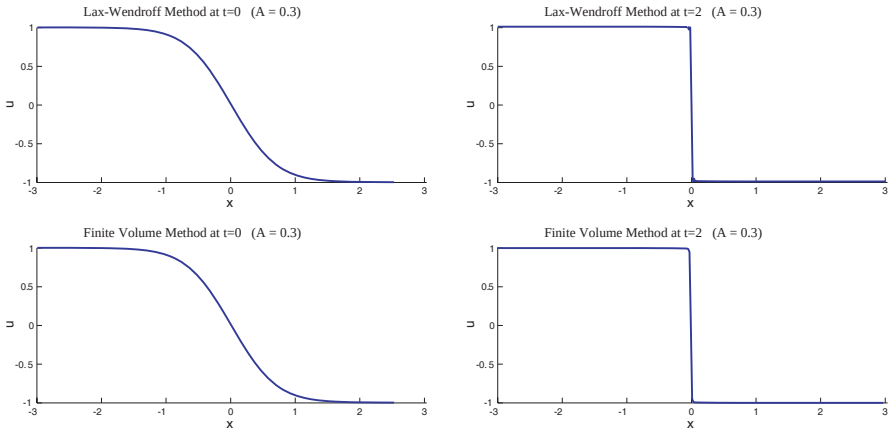


Figure 7.6: These graphs show the result of letting each numerical scheme take smooth initial data and letting it run for two seconds. We have set $A = 0.3$. Observe that the initial data converges to a shock under both methods. The slight numerical inaccuracies are an expected result.

By the global convergence conjecture, we expect any set of initial conditions to converge to the weak traveling wave solution for the given boundary conditions and value of A. If we assume this conjecture is true, this gives a good way to test for the shape of the actual weak traveling wave solution. There are many weak traveling wave solutions that could potentially satisfy (**??**) in some way, however we are mainly concerned with the one that is being converged to. To test that the initial conditions converge to a shock when $A < 1/2$, we set the parameter values suitably and set the initial conditions to a curve that resembles the traveling wave for when $A > 1$. Figure **??** shows the before and after shots of this convergence for both numerical schemes. By setting a very high precision, we can convince ourselves that the solution being converged to is in fact a shock.

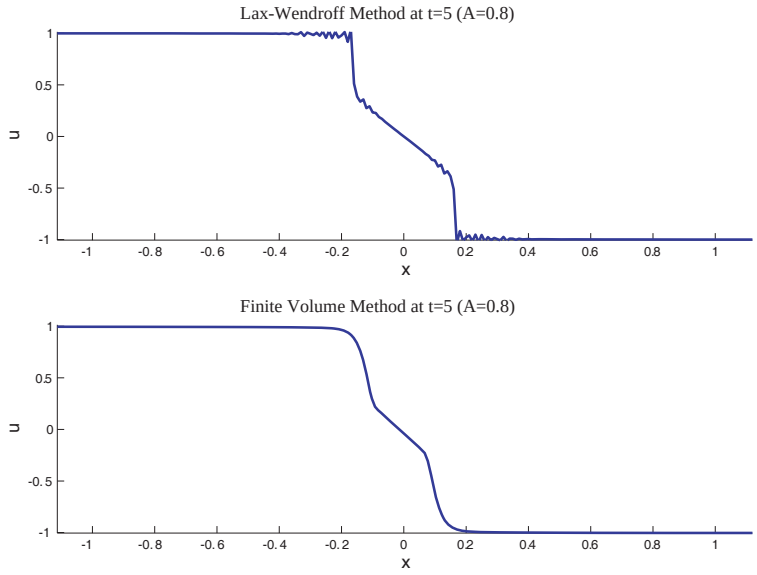


Figure 7.7: These graphs show the result of letting each numerical scheme take smooth initial data and letting it run for two five seconds. The smooth initial conditions we used in this example are the same as in Figure **??**. We have set $A = 0.8$. Observe that the initial data appears to converge to the double shock solution we describe in Chapter 3.

Figure **??** shows the same analysis for the case when $1/2 < A < 1$. Observe that it is difficult to resolve the two smaller shocks, even for very high spatial precisions. The smooth part of the solution in between the shocks is seen to converge to the middle piece of the implicit formula displayed in

equation (**??**) which is depicted in Figure **??**c. By setting the spatial precision high enough and zooming in by the correct magnitude, we were able to convince ourselves that the shock heights in each case are consistent with our predictions in (**??**). By systematically working past the numerical inaccuracies of the two algorithms, we were able to verify the shape of the weak traveling wave solutions depicted in Figure **??**. Therefore, VISCO was very useful in this part of the analysis.

7.2.2 Analysis of Global Convergence

The second major contribution of VISCO to our analysis was the investigation of our global convergence conjecture. Certainly this conjecture was not invented by looking at the equations but rather by observing thousands of simulations in VISCO. The global convergence conjecture is most easily tested using the measure and energy arguments described in Section 6.3. For the majority of test cases we tried, the value of A was set to be greater than 1 so that we didn't need to concern ourselves with the numerical inaccuracies of either algorithm around shocks. Further, when $A < 1/2$ we run into trouble with the definitions of L and R since they can take on imaginary values at some points.

The way we test for convergence is to use the *Analyze* feature to define a measure and test that it goes to zero over time. That is, we define the initial conditions to be some perturbation of a traveling wave solution and then see that its measure goes to zero over time. While it is evident from the animation that the area between of the perturbation diminishes, the graph of the energy over time gives a more objective view of this.

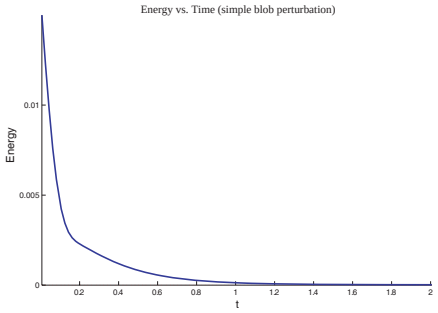


Figure 7.8: This is a graph of the energy of the measure $M(\varepsilon, \delta) = \varepsilon^2 + \delta^2$ against time for a simple *blob* perturbation similar to the one depicted in Figure **??**. Notice that it decays monotonically to zero.

The most basic measure to try is $M(\varepsilon, \delta) = \varepsilon^2 + \delta^2$ whose energy is clearly non-zero unless the perturbed solution has completely converged to the traveling wave solution. The most fundamental perturbation is the *blob* function, described in the previous section. Figure **??** shows the graph of the energy of this measure over time for a simple *blob* perturbation. One nice aspect of this decay in energy is that it is monotonic. If this property were true for all perturbations then we would only need to show analytically that the time derivative of the energy is negative. However, Figure **??** shows the graph of the energy over time when the initial perturbation is sinusoidal rather than a simple *blob*. This disproves the assumption of monotonic decrease in energy. One may observe however that the energy still appears to be bounded below some exponentially decaying function over time. Unfortunately, establishing this bound analytically is much harder than simply showing that the time derivative is always negative.

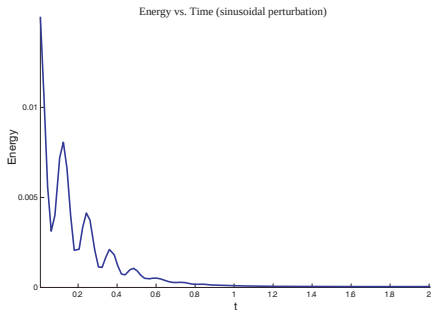


Figure 7.9: This is a graph of the energy of the measure $M(\varepsilon, \delta) = \varepsilon^2 + \delta^2$ against time with a sinusoidal perturbation. In this picture we see that the energy does not decrease monotonically however it does appear to be bounded by an exponentially decaying function.

One special measure that was mentioned in Section 6.3 was

$$M(L, R) = (L - L_w)^2 + (R - R_w)^2.$$

When given sinusoidal initial conditions, the energy of this measure appears to decrease monotonically, as seen in Figure **??**. In fact, we were unsuccessful in finding a set of initial conditions for which the energy of this measure did not decrease monotonically. Based on these empirical observations, we set out to prove that the time derivative of this energy is always negative. Unfortunately, we were unsuccessful in finding an analytical demonstration of this hypothesis.

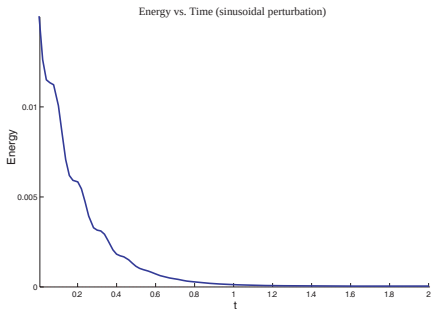


Figure 7.10: This is a graph of the energy of the measure $M(L, R) = (L - L_w)^2 + (R - R_w)$ against time with a sinusoidal perturbation. In this case the energy decreases monotonically which is a desirable property.

7.2.3 Analysis of Plane Waves

The final piece of analysis that was verified by VISCO was the behavior of plane wave perturbations. In this case, we defined plane wave perturbations as bounded sinusoidal perturbations[1], as seen in Figure **??**. Recall that our analysis of linear stability showed that these plane waves have the eigenvalues

$$\lambda_{\pm} = -\frac{1}{2} \pm \left(\frac{1}{2}\sqrt{1 - 4Ak^2} - iu_0 k \right).$$

The real part of these eigenvalues determine the decay rate of the plane waves while the imaginary part should determine the rate of oscillation. Since these eigenvalues were calculated for the linearized system and in a region where u and σ were assumed to be constant, we can only take these eigenvalues to be an approximation to what will actually happen. However, for small perturbations in constant regions we have two major expectations: (1) the decay rate should increase as the wave number, k, and A increase, (2) the amplitude of the plane waves should oscillate under certain conditions, and this rate of oscillation should depend proportionally on the wave number. Both of these expectations were verified when the perturbation were relatively small and defined in a region where u and σ were relatively constant.

We did not expect this general behavior to continue outside of such regions and for larger perturbations. After trying several simulations we

[1] We were able to bound the sinusoidal solutions using the piecewise function feature in VISCO.

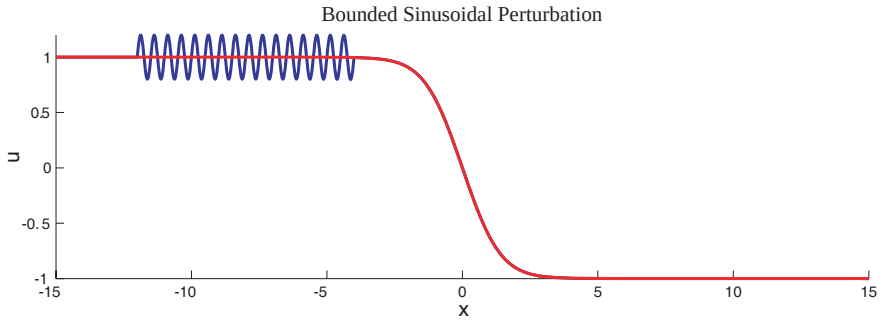


Figure 7.11: This graph depicts the sort of perturbation that was studied to investigate the behavior of plane waves in this system. Notice that the perturbation is bounded within a region in which u and σ are approximately constant

found the behavior to be true for all types of plane waves. While the shapes of the plane waves may have become slightly distorted over time, the amplitudes decayed an oscillated almost exactly as we would have expected for smaller perturbations.

Chapter 8

Future Directions

While a great deal of important work has been accomplished in this thesis, there is much that remains. There are several different ways in which the full three-dimensional viscoelastic model, shown in [**?**], could have been reduced to one dimension. This thesis only studies one such model, however one may wish to study the behavior of other possible one-dimensional simplifications[1]. A slightly more challenging problem is to study the two-dimensional model.

Some of the work presented in Chapter 3 was left unresolved. For example, observe the phase portrait in Figure **??**b. We are able to give a justification for the shape of this trajectory but are unable to prove that it reaches the point $(0, -1/2)$ in finite time. This is a very interesting dynamical systems problem and could be studied more carefully.

Proving the global convergence conjecture would make for a very interesting thesis topic in and of itself. This paper presents a few simple approaches that were tried in attempt to prove the conjecture. However, we have only provided evidence for the conjecture in very special cases. It might be interesting to generalize these methods or try new methods entirely.

The VISCO software program has the potential to be portable, however, no attempt has been made to have it solve other PDE systems. An excellent project might be to work with the shell of VISCO and make it capable of simulating a variety of hyperbolic PDE systems, depending on user input. This would take some work but I am confident that it can be done.

[1] Two additional one-dimensional simplifications are shown in [**?**].