AN ABSTRACT OF THE THESIS OF

Natalia A. McClellan for the degree of Master of Science in Mathematics
presented on June 5, 2014.
Title: Nonlinear Finite Difference Schemes for the Klausmeier System.

Abstract approved: _____

Malgorzata Peszyńska

In this work we consider a model for pattern-producing vegetation in semi-arid regions of the world proposed by Klausmeier. It is a coupled nonlinear diffusion-advection evolutionary PDE system describing vegetation density and water amount. The model was studied extensively by J. Sherratt who took into consideration available field data for the parameter values and considered complexity of the regions of stability. In this thesis we study numerical approximations to the Klausmeier system.

First we consider a simplified zero-dimensional nonlinear ODE system similar to the Klausmeier model. The parameter-dependent equilibria of the system change qualitative behavior as the rainfall varies with respect to the plant loss and plant dispersion coefficients. Three parameter sets are chosen to be descriptive of the system's different states. We consider several Finite Difference schemes to approximate the solutions to this ODE system. These include explicit and various implicit combinations of discretization for the nonlinear term. Initial conditions are chosen to be small random perturbations around the equilibria states for each of the parameter sets. Suitable restrictions on the choice of the time step are considered and catalogued.

Next we work with the original PDE system to discern asymptotic pattern formation. Using the same range of methods as in the ODE case, we discretize the system in space and time and apply periodic boundary conditions along with the realistic parameter values. Conclusive evidence of banded vegetation patterns is obtained.

Nonlinear Finite Difference Schemes for the Klausmeier System

by
Natalia A. McClellan

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented June 5, 2014
Commencement June 2014

Master of Science thesis of Natalia A. McClellan presented on June 5, 2014

APPROVED:

Major Professor, representing Mathematics

Chair of the Department of Mathematics

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Natalia A. McClellan, Author

ACKNOWLEDGEMENTS

I would like to first acknowledge my advisor Dr. Peszynska for investing her time into my development as a mathematician. Her encouragement kept me focused and her classes taught me enough modelling for several lifetimes.

I would like to thank my colleague and dear friend Adriana Mendoza for caring for me and feeding me in times of emergencies. Her rock strength and spirited independence inspired awe in me. She doodled funny comics when times were at their most grey.

I give special thanks to Forrest Parker, a great and devoted friend who always helped me and never ate the apricot jam I gifted him, and to Kial-Ann Rasmussen, who helped me more than she knows. I also want to thank Mathew Huerta-Enochian, Evan Hedlund, and Ben Livingston for being kind and gentle friends when I needed them most. The "induction dance" will be forever imprinted in my memory... I never want to go clam digging again, but another turn on a tandem bike would do me good.

I also wish to acknowledge Dr. Dick and the department of Mathematics' gift of graduate teaching assistantship, which not only helped finance my degree, but also taught me how to be a teacher. One of the greatest parts of my experience at OSU was the opportunity to work in a classroom and build the skills for my future professional career.

The dedication and perseverance in pursuit of my Master's arises from the care and encouragement of my darling husband, who waited for me for two years. This sacrifice for higher education (made by both of us) has made our relationship stronger; we truly can overcome any obstacle.

My parents Aleksandr and Olga Glaznevi raised me to value education and that deserves a special mention. I love my family very much and recognize their great contribution in helping me attain my degree.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

Figure

LIST OF FIGURES (Continued)

Figure

LIST OF FIGURES (Continued)

Figure

LIST OF TABLES

Table

To my parents and my husband for their love, constant support, and encouragement.

Nonlinear Finite Difference Schemes for the Klausmeier System

1 Introduction

In this work we consider a model for pattern-producing vegetation in semi-arid regions of the world originally proposed by Klausmeier in [1] and extensively studied by Sherratt in [4]-[8].

The Klausmeier Model for vegetation dynamics in semi-deserted areas is based on the water-redistribution hypothesis, centering on the idea that rainwater in dry regions is little infiltrated into the ground. Instead, water mostly runs off downhill towards the next patch of vegetation. The soil in such regions of the world as Australia, Africa, and Southwestern North America is prone to non-locality of water uptake due to the semiarid environment [8].

Since ecological laboratory experiments are extremely costly and field studies are frequently insufficient for thorough data collection due to their short duration, mathematical modeling of such systems is of vital importance to keeping semiarid environments from turning into full-blown deserts. The hysteresis curves in paper [8] show that the mean vegetation density could be regained if the level of precipitation does not fall below the limit on the scale of the rainfall parameter A. If the levels of rainfall decrease sufficiently enough to overturn the system, the vegetation could not be recovered with the following return of the rainfall and the land will become devoid of plant life.

However, the analysis of the Klausmeier system need not be so drastic. Paper [8] also detects a sensitive drop in the vegetation productivity U when the rainfall parameter is varied closer to its upper limit for patterns. This is an important result for land management strategies, since both the grazing and timber uses of the dry regions are difficult to plan for. Once the level of rainfall gets closer to the one that produces an abrupt drop in vegetation density, the activity in the region should be reduced in order for the land to recover faster once precipitation is back to its expected levels.

The Klausmeier model which we develop in detail in Chapter 2

$$\frac{\partial u}{\partial t} = wu^2 - Bu + \frac{\partial^2 u}{\partial x^2}, \tag{1.1a}$$

$$\frac{\partial w}{\partial t} = A - w - wu^2 + v\frac{\partial w}{\partial x}, \tag{1.1b}$$

is a coupled system of nonlinear diffusion-advection partial differential equations (PDEs). Here u is a plant density function and w is a water density function. We discuss the meaning of parameters A, B, v in depth in Chapter 2.

Sherratt studied this model extensively and took into consideration filed data for the parameter values and considered the complexity of the regions of stability. However, his analysis was based on several simplifying assumptions which we make clear in Chapter 2. He obtained evidence of long-time pattern formation for the case when some parameters are constants and under assumption of periodic boundary conditions.

We start our work with a zero-dimensional system of ODEs and test our numerical schemes of that system first. We also observe the dependence of solutions on the choice of the parameter sets for the simplified nonlinear ODE system. The parameter-dependent equilibria of the system change qualitative behavior as the rainfall varies with respect to the plant loss and plant dispersion coefficients. Three parameter sets are chosen to be descriptive of the system's different states.

Next we consider several Finite Difference schemes for the ODE system. These include explicit and various implicit combinations of discretization for the nonlinear term. Initial conditions are chosen to be small random perturbations around the equilibria states for each of the parameter sets. Comparison of results to the well-known numerical solver ODE45 allows for the estimation of accuracy and stability of the solutions for the system. Suitable restrictions on the choice of the time step are considered and catalogued.

Finally, we work with the full Klausmeier PDE system to discern asymptotic trends of pattern solutions of the one-dimensional Klausmeier system. Using the same range of methods as in the ODE case, we discretize the system in space and time and apply periodic boundary conditions along with realistic parameter values. Conclusive evidence of banded vegetation patterns is attained.

The outline of this thesis is as follows. In Chapter 2 we introduce the Klausmeier model for plant density in semi-arid environments. We also provide a detailed review of the

work [4]-[8] by J. Sherratt which inspired this thesis. In Chapters 3 and 4 we consider the zero-dimensional variant of the Klausmeier model, and provide analysis of equilibria of the associated planar ODE system as well as numerical illustrations of the behavior predicted by the analysis. We also develop our numerical schemes for the evolution problem which is the first step towards numerical solution of the full Klausmeier model. The work on the ODE system helps us to understand sensitivities and potential difficulties of the numerical solvers before we get to the PDE case. In Chapter 5 we consider the full Klausmeier model. We define numerical discretization and analyze, to the extent possible, numerical stability of the sub-components of the full system. Finally, in Chapter 6 we provide numerical simulation results, which confirm formation of patterns similar to those predicted by Sherratt. We close in Chapter 7 with the summary and outlook to the future work.

2 The Klausmeier System of Equations

As the levels of rainfall fluctuate annually, arid landscapes form into wave-like patterns called bands (in the presence of gentle slopes). Such stripes of vegetation run parallel to the contours of the hills and in documented instances grow upslope with time. Typical wavelengths are about 1km for trees and shrubs, with shorter wavelengths for grasses [7]. The research suggests that the plants migrate uphill with each generation by reasoning that the higher moisture levels are available upslope of the bands, facilitating vegetation growth [8]. The evidence of these patterns is extremely strong and there exist documented reports of their migration that date as far back as the 1950s and '60s in Africa [7].

Some cases of the stationary banded patterns are also present; however, it is difficult to trace the data far enough to make serious general assumptions. It is also uncertain in such circumstances whether or not the length of time during which the cases were observed is sufficient enough to draw conclusions; however, some field studies show the existence of stationary patterns over the time scale of decades. This might be explained by the gradual loss of soil nutrients and the subsequent hardening of the ground that is not accounted for in the Klausmeier model.

The model makes a few other conceptual assumptions [5]:

- constant rainfall, with water lost via both evaporation and uptake by plants,
- per capita growth is proportional to water density and biomass,
- plant loss is assumed to occur at a constant per capita rate, which may include herbivory,
- water is not separated into surface and subsurface components,
- water flows downhill while plant dispersal is modeled by linear diffusion,
- the seed dispersal in the downslope direction due to transport in run-off is neglected,
- pattern solutions move uphill at a constant speed.

The original Klausmeier PDE system is nondimensionalized in the following way:

$$\frac{\partial u}{\partial t} = \overbrace{wu^2}^{\text{plant growth}} - \overbrace{Bu}^{\text{plant loss}} + \overbrace{\frac{\partial^2 u}{\partial x^2}}^{\text{plant dispersal}}, \tag{2.1a}$$

$$\frac{\partial w}{\partial t} = \underbrace{A}_{\text{rainfall}} - \underbrace{w}_{\text{evaporation}} - \underbrace{wu^2}_{\text{uptake by plants}} + \underbrace{v\frac{\partial w}{\partial x}}_{\text{flow downhill}}, \tag{2.1b}$$

where $u(x,t)$ is plant density, $w(x,t)$ is water density, t is time, and x is a one-dimensional space variable running in the uphill direction. The non-dimensional parameters A, B, and v are interpreted for simplicity as corresponding to the rainfall, plant loss, and slope-gradient respectively. The range for parameter estimates for trees is such that

$$A \in [0.08, 0.2], B = 0.045,$$

and for grass

$$A \in [0.9, 2.8], B = 0.45, \tag{2.2}$$

with

$$v = 182.5 \tag{2.3}$$

according to [8]. Inspecting v, we note that the extremely large value is not due to the steepness of the hills on which vegetation bands are observed (the slopes are quite gentle, just a few percent). The slope gradient v is large because the plant diffusion coefficient is small compared to the advection rate of water, and it is the relative values of these quantities that determine this parameter [6]. The nonlinear term wu^2 relates that the infiltration rate is proportional to vegetation density [8].

Since the pattern solutions of the system in (2.1) are periodic traveling waves, they are transformed by the use of the Ansatz

$$z = x - ct \tag{2.4}$$

into the system

$$\frac{d^2U}{dz^2} + c\frac{dU}{dz} + WU^2 - BU = 0, \tag{2.5a}$$

$$(v+c)\frac{dW}{dz} + A - W - WU^2 = 0, \tag{2.5b}$$

which depends solely on the traveling wave coordinate z (where c is postitive migration speed of vegetation in the uphill direction and $u(x,t) = U(z), w(x,t) = W(z)$). The transformation of the derivatives is done with the chain rule

$$\begin{aligned}\frac{\partial u(x,t)}{\partial x} &= \frac{\partial U(z)}{\partial x} = \frac{\partial U(x-ct)}{\partial x} = U'(x-ct)\frac{\partial (x-ct)}{\partial x} = U'(x-ct) = U'(z) \\ &= \frac{dU}{dz}.\end{aligned} \tag{2.6}$$

Similarly we can conclude that

$$\frac{\partial w(x,t)}{\partial x} = \frac{dW}{dz}. \tag{2.7}$$

Using the chain rule once more produces

$$\begin{aligned}\frac{\partial u(x,t)}{\partial t} &= \frac{\partial U(z)}{\partial t} = \frac{\partial U(x-ct)}{\partial t} = U'(x-ct)\frac{\partial (x-ct)}{\partial t} = U'(x-ct)\cdot(-c) = \\ &= -cU'(z) = -c\frac{dU(z)}{dz} = -c\frac{dU}{dz}.\end{aligned} \tag{2.8}$$

Analogously,

$$\frac{\partial w(x,t)}{\partial t} = -c\frac{dW}{dz}. \tag{2.9}$$

The second derivative is calculated through (2.6) and the equality of the mixed partials:

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial}{\partial x}\Big(\frac{\partial u}{\partial x}\Big) = \frac{\partial}{\partial x}\Big(\frac{\partial U}{\partial z}\Big) = \frac{\partial}{\partial z}\Big(\frac{\partial u}{\partial x}\Big) = \frac{\partial}{\partial z}\Big(\frac{\partial U}{\partial z}\Big) = \frac{\partial^2 U}{\partial z^2}. \tag{2.10}$$

Substituting (2.8) and (2.10) into (2.1a), we get

$$-c\frac{dU}{dz} = WU^2 - BU + \frac{\partial^2 U}{\partial z^2},$$

which is equivalent to (2.5a).

Substituting (2.7) and (2.9) into (2.1b), we get

$$-c\frac{dW}{dz} = A - W - WU^2 + v\frac{\partial W}{\partial z},$$

which is equivalent to (2.5b).

2.1 Literature Review

In studying the Klausmeier system (2.1), J. Sherratt analyzes the dependence of the migration speed c in the uphill direction on the mean annual rainfall A. The A-c space considers boundaries of the regions in which patterns occur; J. Sherratt studies stability of solutions for certain values of these parameters.

Below is the collection of regions considered the papers [4] - [8] (notation $f = O_s(g)$ is used in place of $f = O(g)$ and $f \neq o(g)$):

1. $v^{1/2} << c << v$ and $A^2 = O_s(vc)$; [4]
2. $c = O_s(v), A = O_s(1)$ as $v \to \infty$; [5]
3. $c = O_s(v^{1/2}), A = O_s(v^{1/4})$; [6]
4. The following parameter ranges are considered in [7]:
   (a) $1/v << c << 1$,
   (b) $c = O_s(1/v), A = O_s(1)$,
   (c) $c = o(1/v)$ as $v \to \infty$,
   (d) $vc >> A^2 >> 1/vc$ and $A^2 = O_s(1/vc)$;
5. In [8], either $c = O_s(1)$ and three cases for A are considered:
   (a) $A = O_s(v^{1/2})$,
   (b) $v^{-1/2} << A << v^{1/2}$,
   (c) $A = O_s(v^{-1/2})$,

   or the extended case $1 << c << v^{1/2}$ where $v \to \infty$.

The paper [4] establishes thresholds for pattern formation. It was discovered that leading order equations change close to one boundary of the parameter region, giving a homoclinic solution. Biomass productivity is affected with the changes in the mean annual rainfall levels. Productivity varies between the upper limit and the lower limit on A for pattern existence. This implies that semi-deserted regions do not display wave-like patterns if precipitation is over

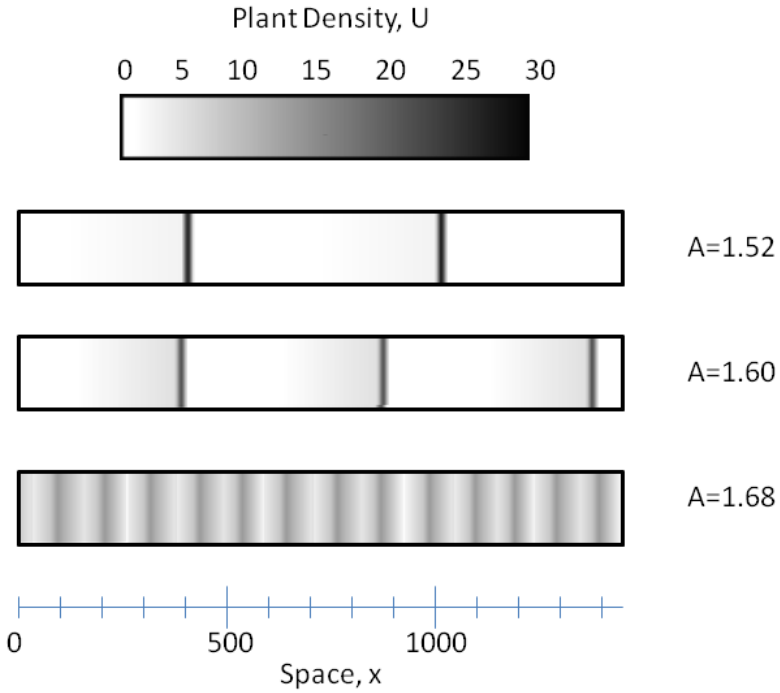

FIGURE 2.1: Transition from diffuse patterns of low amplitude to localized patterns with higher plant density. Illustration is based on Figure 8 from [6].

the limit, suggesting that plants form grouped patches in order to survive drought periods. Under the limit annual rainfall infers the inactive vegetation state.

The paper [5] focuses on the region where homoclinic solution and Hopf bifurcation loci cross. The A-c parameter space involves propagation speed of patterns close to its maximum possible value. Wave-like pattern solutions of the Klausmeier system are obtained. For a fixed migration speed, pattern wavelength depends sensitively on rainfall.

The paper [6] concentrates on studying the region at the intersection point of the two loci of the homoclinic solutions. As the rainfall parameter A decreases, a sharp transition from diffuse patterns of relatively low amplitude to much more localized patterns with higher plant density occurs. The transition is marked by an increase in pattern wavelength, which is illustrated in Figure 2.1.

The paper [7] obtains a complete analytical description of the periodic travelling waves with $c << 1/v$ for large values of the slope gradient v. Investigation of pattern stability brought

on the conclusion that very slowly moving patterns are unstable.

The final paper [8] centers on analytical understanding of the existence and form of pattern solutions (PTWs) as v tends to infinity. The formulas for maximum and minimum rainfall levels for which the patterns exist are derived. The author obtains evidence of hysteresis with semi-deserts turning into full-blown deserts in the absence of rainfall. The threshold for A is dependent on parameters B, v.

Due to the fact that many semiarid regions with banded vegetation are used for grazing and/or timber, the paper [8] carries important information for land management strategies. As the rainfall parameter decreases, the mean plant density experiences an abrupt change in its productivity. This allows for adjustments in land uses (which could protect productivity outputs) in anticipation of change.

Other discoveries made in connection to the minimum rainfall level emphasize its significance in triggering a rapid change in the pattern form. In a certain parameter regime, changes in grazing intensity will not affect either the migration speed of patterns or their wavelength though it will affect the vegetation density in the bands.

Investigation into the wave-like patterns over a long period of time uncovers the periodic structure of solutions.

Simplicity of the model means that one cannot expect a quantitative agreement between the Klausmeier formulas and any given field data. The general conclusion from Sherratt's work [4]-[8] and the underlying mathematical analysis depend fundamentally on the way in which the migration speed c scales with v. The parametric trends implied by the model can be expected to be paralleled in the field, at least qualitatively.

In general, it is clear that the complexity and heterogeneity of real environments complicate appearance of any patterns. For instance, the amount of rainfall can vary from decade to decade, interrupting and/or changing any emerging patterns. It may be difficult to trace the evidence of banded vegetation due to environmental conditions and lack of long-term data.

In this thesis, we study the behavior of solutions to the Klausmeier model without using the special form of traveling wave solutions. A direct numerical simulation allows us to study arbitrary rainfall patterns (in particular, one can use real time series data for simulations). Thus, more realistic and more complex patterns can be studied.

3 Phase Plane Analysis for an ODE System

In this chapter we study a zero-dimensional counterpart of the Klausmeier system (2.1). This means that we represent it as a system of ordinary differential equations (ODEs). Such an analysis helps understand sensitivities of a numerical model that will be applied to (2.1). In addition, it helps to understand the interaction of nonlinear terms with diffusion and the presence of equilibria.

We model the effect of diffusion $-\frac{\partial^2 u}{\partial x^2}$ by a positive definite term Du. Such a term is expected to have a stabilizing effect on the system. We also neglect the advection term $v\frac{\partial u}{\partial x}$, which is expected to have a neutral effect on stability.

3.1 Introduction

In analyzing the phase plane also known as the vector field plane of an ODE, we consider a general autonomous planar ODE system (where the right hand side does not depend on the independent variable t):

$$u' = f(u). \tag{3.1}$$

Here $u : \mathbb{R} \to \mathbb{R}^N$, and $f : \mathbb{R}^N \to \mathbb{R}^N$.

We find the equilibrium solutions $u_j \in \mathbb{R}^N$ for which $f(u_j) = 0$, where $j = 1, 2, ..., n, n+1$.

Consider first the scalar case when $N = 1$. The intervals (a, b) between the equilibria are consecutive pairs coming from the sets of entries $a \in \{-\infty, u_1, u_2, ..., u_n, u_{n+1}\}$ and $b \in \{u_1, u_2, ..., u_n, u_{n+1}, \infty\}$.

We can analyze whether $f(u)$ is positive or negative in each interval (u_i, u_{i+1}). This helps to determine whether the solution u is increasing or decreasing on a certain (a, b). Picking initial conditions from each interval will establish the behavior of u as $t \to \infty$.

When $f(u) > 0$ for an initial condition $u_o \in (a, b)$, u is increasing and $\lim_{t\to-\infty} u(t) \to a$, $\lim_{t\to\infty} u(t) \to b$.

When $f(u) < 0$ for an initial condition $u_o \in (a, b)$, u is decreasing and $\lim_{t\to-\infty} u(t) \to b$, $\lim_{t\to\infty} u(t) \to a$.

Plotting u against t for each initial condition will create a myriad of lines that either tend toward equilibria (in this case equilibria are called nodal sinks) or away from it (called nodal sources). A saddle corresponds to the case when initial conditions from both sides of the critical values do not lead to just one type of equilibrium, but to one of a mixed type (a sink from one side, a source from the other). For the autonomous equation (3.1) the phase plane is horizontal-shift invariant because the values of the slopes u' do not vary along each horizontal line.

Now we consider a planar system (3.1) of first order ODE's where $N = 2$. We get

$$u' = f(u, w) \tag{3.2a}$$

$$w' = g(u, w). \tag{3.2b}$$

A good tool for the analysis of the system (3.2) is the nullcline. The u-nullcline is obtained by setting $f(u, w) = 0$; similarly, w-nullcline is calculated through $g(u, w) = 0$. The u-component of the u-w vector field is zero (which produces vertical lines) along the u-nullcline, (analogously for w). The intersections of nullclines are the equilibrium points [3].

Estimating the sign of the eigenvalues of the Jacobian

$$J = \begin{bmatrix} f_u & f_w \\ g_u & g_w \end{bmatrix} \tag{3.3}$$

at each equilibrium point for nonlinear system (3.2) helps to ascertain the type of the equilibrium. When both eigenvalues are negative, we have a sink; when both are positive, a source; when the signs of the eigenvalues differ, we have a saddle [3]. When either eigenvalue vanishes, we have a degenerate node.

If the ODE system involves parameters, certain bifurcations are possible and equilibrium points as well as their type might change. Section 3.2 deals with those issues.

3.2 Finding Equilibria Based on Coefficient Values

The original Klausmeier PDE system (2.1) is transformed into an ODE system

$$u' = wu^2 - Bu - Du, \tag{3.4a}$$

$$w' = A - w - wu^2. \tag{3.4b}$$

where we introduce the Du term which models the diffusion. Notice that the advection is neglected here.

Assumption 3.2.1. *The coefficient values are such that*

$$A > 0, B > 0, D \geq 0, \widetilde{B} = B + D > 0. \tag{3.5}$$

The first two are consistent with their meaning assigned in the Klausmeier model [4] - [8]. The third assumption follows from identification of D as the zero-dimension diffusion operator. The last assumption is helpful in further calculations.

Remark 3.2.1. *The solutions u and w to the Klausmeier system* (2.1) *should satisfy*

$$u \geq 0, \quad w > 0 \tag{3.6}$$

due to their realistic meaning (plant density and water density respectively).

Now we describe equilibria of (3.4). These depend on the parameters A, B, D.

Setting $u' = 0$ and $w' = 0$, we solve the system

$$wu^2 - (B + D)u = 0, \tag{3.7a}$$

$$A - w - wu^2 = 0. \tag{3.7b}$$

Working with (3.7a) and applying (3.5), we get

$$wu^2 - \widetilde{B}u = 0,$$

which implies that either

$$u = 0 \tag{3.8}$$

or

$$u = \frac{\widetilde{B}}{w}. \tag{3.9}$$

Working with (3.7b), we consider two cases. First, if (3.8) holds, then using equation (3.7b), we obtain

$$w = A.$$

So

$$E = (0, A) \tag{3.10}$$

is the first equilibrium point. This means that we always have at least this one equilibrium. In Section 3.3.1 we show that it is a sink independently of the values of the coefficients from (3.5).

Next, if (3.9) holds, then using equation (3.7a), we get

$$A - w - w\left(\frac{\widetilde{B}}{w}\right)^2 = 0$$

or, equivalently,

$$w^2 - Aw + \widetilde{B}^2 = 0. \tag{3.11}$$

By analyzing at the equation (3.11) we can uncover the relationship between the coefficients that produce zero, one, or two additional equilibrium points.

Case 1: If the discriminant of (3.11) is < 0, then $A^2 < 4\widetilde{B}^2$, so $A < 2\widetilde{B}$.

There are no real solutions w; thus, it is impossible to have an additional equilibrium.

Case 2: If the discriminant of (3.11) is $= 0$, then $A^2 = 4\widetilde{B}^2$, thus $A = 2\widetilde{B}$.

We see that solution u is according to (3.9) and $w = \frac{A}{2}$ due to (3.9); it follows that the equilibrium point is at

$$E_1 = \left(\frac{2\widetilde{B}}{A}, \frac{A}{2}\right) = \left(1, \frac{A}{2}\right) = (1, \widetilde{B}). \tag{3.12}$$

Case 3: If the discriminant of (3.11) is > 0, then $A^2 > 4\widetilde{B}^2$, so

$$A > 2\widetilde{B}. \tag{3.13}$$

Then

$$w_{1,2} = \frac{A \pm \sqrt{A^2 - 4\widetilde{B}^2}}{2}. \tag{3.14}$$

and, according to (3.9),

$$u_{1,2} = \frac{2\widetilde{B}}{A \pm \sqrt{A^2 - 4\widetilde{B}^2}}. \tag{3.15}$$

Thus, we can have two equilibria of the form

$$E_{2,3} = \left(\frac{2\widetilde{B}}{A \pm \sqrt{A^2 - 4\widetilde{B}^2}}, \frac{A \pm \sqrt{A^2 - 4\widetilde{B}^2}}{2}\right). \tag{3.16}$$

The preceding calculations are summarized in the following result.

Lemma 3.2.1. *We always have either only one equilibrium* $\{E\}$*, two equilibria* $\{E, E_1\}$*, or three equilibria* $\{E, E_2, E_3\}$ *given by* (3.10)*,* (3.12)*, and* (3.16) *respectively.*

3.3 Type of Equilibria

Now we investigate the Jacobian of the system at equilibrium points.

The general form of the Jacobian of (3.4) is as follows:

$$J = \begin{bmatrix} 2wu - \widetilde{B} & u^2 \\ -2wu & -1 - u^2 \end{bmatrix}. \tag{3.17}$$

In what follows we will investigate the type of each equilibrium. We will also illustrate the analysis with the well-known public domain computer tool `PPLANE`. The tool plots the vector field given by (3.2) and helps to find equilibria. This is especially useful for the nonlinear systems with complicated right hand sides and a multitude of coefficients.

3.3.1 Equilibrium $E = (0, A)$

The Jacobian (3.17) is now

$$J\big|_E = \begin{bmatrix} -\widetilde{B} & 0 \\ 0 & -1 \end{bmatrix}, \tag{3.18}$$

whose determinant is never zero due to (3.5) and whose eigenvalues are $-\widetilde{B}$ and -1. Since these are negative by (3.5), we have the following.

Lemma 3.3.1. *E is an equilibrium of type sink independently of the values A and $\widetilde{B}$.*

This behavior is illustrated in Figure 3.1.

3.3.2 Equilibrium $E_1 = (1, \widetilde{B})$ where $A = 2\widetilde{B}$

The Jacobian (3.17) transforms into

$$J\big|_{E_1} = \begin{bmatrix} 2\widetilde{B} - \widetilde{B} & 1 \\ -2\widetilde{B} & -1 - 1 \end{bmatrix} = \begin{bmatrix} \widetilde{B} & 1 \\ -2\widetilde{B} & -2 \end{bmatrix}, \tag{3.19}$$

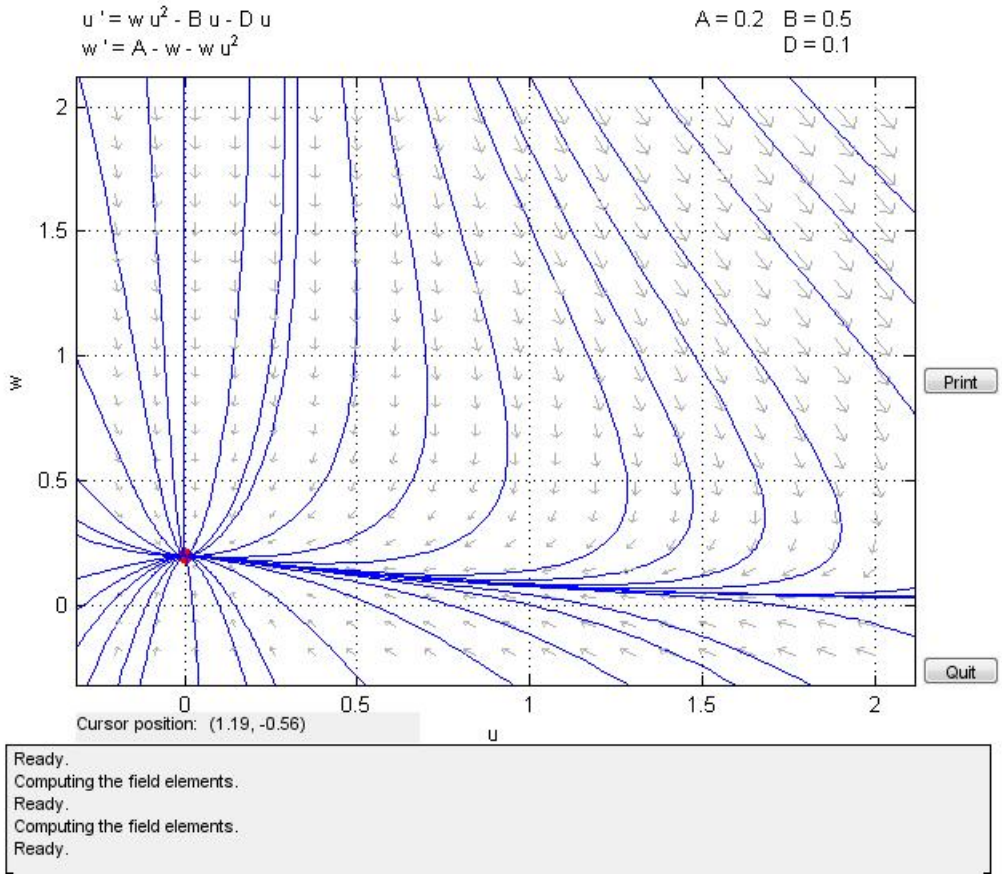

FIGURE 3.1: `PPLANE` illustration of the sink behavior of equilibrium E for parameter values $A = 0.2, B = 0.5, D = 0.1$.

We thus have

$$\det(J)|_{E_1} = -2\widetilde{B} - (-2\widetilde{B}) = 0 = \lambda_1\lambda_2. \tag{3.20}$$

We see that the equilibrium E_1 is producing a singular Jacobian for any $\widetilde{B} = \frac{A}{2}$ due to (3.20).

We also know that

$$\text{trace}(J)|_{E_1} = -2 - \widetilde{B} = \lambda_1 + \lambda_2. \tag{3.21}$$

To find the eigenvalues from (3.20) and (3.21) formulas, we solve the system

$$\begin{aligned} \lambda_1 + \lambda_2 &= -2 - \widetilde{B}, \\ \lambda_1\lambda_2 &= 0. \end{aligned}$$

It is clear that one of the eigenvalues is zero and the other is equal to $-2 - \widetilde{B}$. By (3.5), the second eigenvalue is negative.

Lemma 3.3.2. *E_1 is a degenerate node.*

The behavior is shown in Figure 3.2.

3.3.3 Equilibria $E_{2,3}$ where $A > 2\widetilde{B}$

In preparation to find the Jacobian (3.17) at $E_{2,3}$, we use (3.14) and (3.15) in simplification

$$2uw = 2 \cdot \frac{2\widetilde{B}(A \pm \sqrt{A^2 - 4\widetilde{B}^2})}{2(A \pm \sqrt{A^2 - 4\widetilde{B}^2})} = 2\widetilde{B}.$$

The u^2 term does not simplify nicely. Instead, we notice that (3.17) becomes

$$J\Big|_{E_{2,3}} = \begin{bmatrix} 2\widetilde{B} - \widetilde{B} & u^2 \\ -2\widetilde{B} & -1 - u^2 \end{bmatrix} = \begin{bmatrix} \widetilde{B} & u^2 \\ -2\widetilde{B} & -1 - u^2 \end{bmatrix}. \tag{3.22}$$

Furthermore,

$$\det(J)|_{E_{2,3}} = \widetilde{B}(-1 - u^2) + 2u^2\widetilde{B} = -\widetilde{B} + \widetilde{B}u^2 = \widetilde{B}(u^2 - 1) = \lambda_1\lambda_2, \tag{3.23}$$

$$\text{trace}(J)|_{E_{2,3}} = -1 - \widetilde{B} - u^2 = \lambda_1 + \lambda_2. \tag{3.24}$$

Proposition 3.3.1. *Jacobian in* (3.22) *is never singular when* (3.13) *holds.*

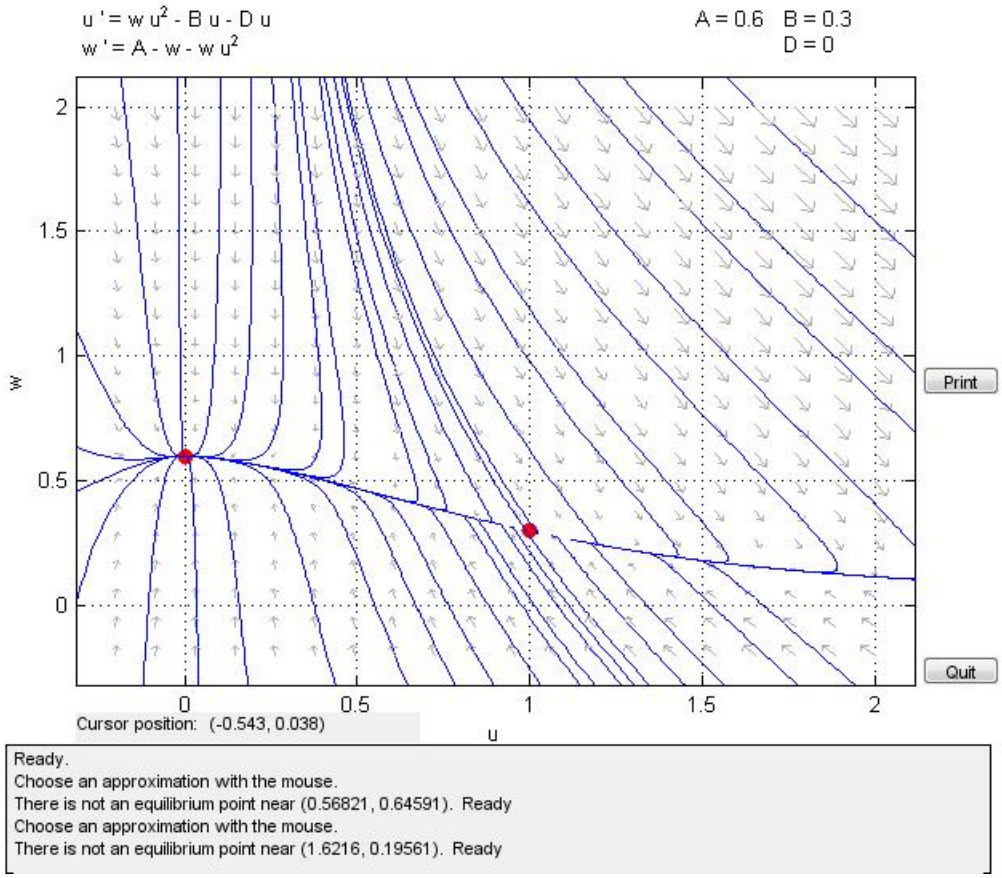

FIGURE 3.2: `PPLANE` illustration of the behavior of equilibria E and E_1 for parameter values $A = 0.6, B = 0.3, D = 0$.

Analyzing the possibility of the singular Jacobian, we set (3.23) equal to zero and, using (3.5) deduce that

$$u^2 - 1 = 0$$

holds. Now, knowing (3.6) and (3.15), we observe that in terms of the coefficients this means that

$$\frac{2\widetilde{B}}{A \pm \sqrt{A^2 - 4\widetilde{B}^2}} = 1$$

Accounting for (3.13) allows us to see that

$$A - 2\widetilde{B} = \sqrt{A^2 - 4\widetilde{B}^2}$$

and so

$$4\widetilde{B}(2\widetilde{B} - A) = 0,$$

which is impossible for the range of the coefficients we have set in (3.5) and (3.13). Thus, the Jacobian (3.22) is never singular and

$$u^2 - 1 \neq 0. \tag{3.25}$$

Now we want to estimate the eigenvalues of (3.22). We set up the following system using (3.23) and (3.24):

$$\lambda_1 + \lambda_2 = -1 - \widetilde{B} - u^2, \tag{3.26}$$

$$\lambda_1\lambda_2 = \widetilde{B}(u^2 - 1). \tag{3.27}$$

By (3.5) it is evident that

$$\lambda_1 + \lambda_2 < 0. \tag{3.28}$$

To analyze the product of the eigenvalues, we consider the following three cases.

Case 1: $\lambda_1\lambda_2 = 0$.

This means that

$$\widetilde{B}(u^2 - 1) = 0, \tag{3.29}$$

which we know to be false due to (3.5) and (3.25).

Case 2: $\lambda_1\lambda_2 < 0$. WLOG this implies that $\lambda_1 < 0 < \lambda_2$ and $|\lambda_1| > \lambda_2$.

By (3.27) we have

$$\widetilde{B}(u^2 - 1) \quad > \quad 0, \tag{3.30}$$

which is only true if $u > 1$ due to (3.5) and (3.6) restrictions. This means that by (3.15), the u_1 coordinate of E_2 is such that

$$u_1 = \frac{2\widetilde{B}}{A - \sqrt{A^2 - 4\widetilde{B}^2}} > 1.$$

Solving the above inequality further, we have

$$A - 2\widetilde{B} < \sqrt{A^2 - 4\widetilde{B}^2},$$
$$-\widetilde{B}(A - 2\widetilde{B}) < 0,$$

which is always true for our restrictions (3.5) and (3.6).

It follows that E_2 has eigenvalues of opposite signs.

Now we consider E_3 and its u_2-coordinate

$$u_2 = \frac{2\widetilde{B}}{A + \sqrt{A^2 - 4\widetilde{B}^2}} > 1 \tag{3.31}$$

which produces

$$A - 2\widetilde{B} < -\sqrt{A^2 - 4\widetilde{B}^2} \tag{3.32}$$

and is never true independently of the parameter values.

This implies that the eigenvalues of E_3 are of the same sign.

Case 3: $\lambda_1\lambda_2 > 0$. WLOG we have $\lambda_1 \leq \lambda_2 < 0$.

Using (3.27), we get

$$\widetilde{B}(u^2 - 1) < 0,$$

which by (3.6) implies that

$$u < 1.$$

Considering u_1 of E_2 first, we have

$$u_1 = \frac{2\widetilde{B}}{A - \sqrt{A^2 - 4\widetilde{B}^2}} < 1.$$

This means that

$$\begin{aligned} \widetilde{B}(2\widetilde{B} - A) &> 0, \\ A - 2\widetilde{B} &< 0, \end{aligned} \tag{3.33}$$

which is inconsistent with our restrictions (3.13).

The result is supported by the evidence in Case 2.

Now we analyze E_3 with its u_2-coordinate. We have

$$\begin{aligned} u_2 = \frac{2\widetilde{B}}{A + \sqrt{A^2 - 4\widetilde{B}^2}} &< 1, \\ 2\widetilde{B} - A &< \sqrt{A^2 - 4\widetilde{B}^2}, \end{aligned}$$

which is always true for (3.13).

This leads to the conclusion that E_3 produces two negative eigenvalues.

Lemma 3.3.3. *Equilibrium E_2 is of saddle type since λ's are of different signs, and equilibrium E_3 is of type sink because λ's are both negative.*

The behavior is illustrated in Figure 3.3.

3.4 Summary of Equilibria Analysis

Now we illustrate the results shown above as concerns the occurrence and type of equilibria depending on the values A, B, D. We choose three sets of these parameters

$$A = 0.2, B = 0.5, D = 0.1, \tag{3.34a}$$

$$A = 0.6, B = 0.3, D = 0, \tag{3.34b}$$

$$A = 0.7, B = 0.3, D = 0, \tag{3.34c}$$

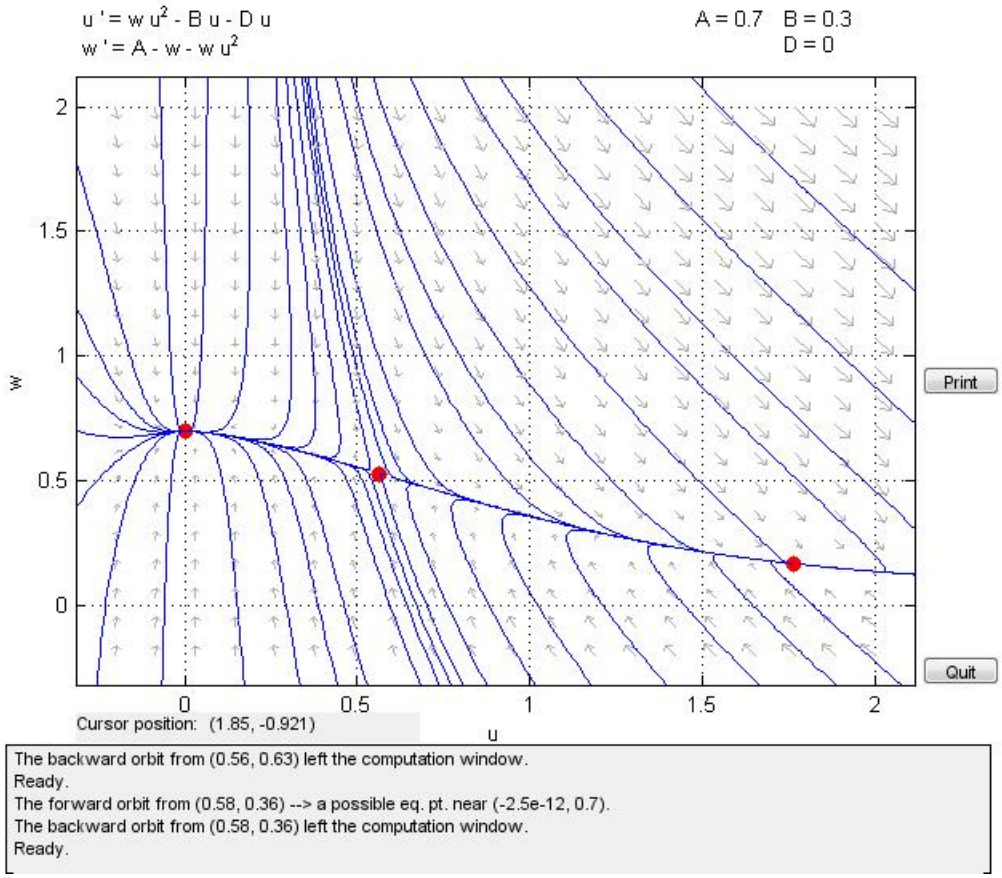

FIGURE 3.3: `PPLANE` illustration of the behavior of equilibria E, E_2, E_3 for parameter values $A = 0.7, B = 0.3, D = 0$.

to show the existence of each type of equilibria that were predicted in Lemmas 3.2.1, 3.3.1, 3.3.2, and 3.3.3. The PPLANE illustrations of the vector field of solutions to (3.4) for each parameter set are shown in Figures 3.1, 3.2, 3.3.

Table 3.1 displays results of equilibrium points' calculations and their type for each of the parameter sets (3.34). These parameters are also used in our numerical calculations for the ODE system.

TABLE 3.1: Equilibria Classification (where $\widetilde{B} = B + D$)

Equilibrium		E	E_1
Coordinates in the u-w plane		$(0, A)$	$(1, \widetilde{B})$
Range of Parameter Values $A, B \in \mathbb{R}^+$, and $D \in \mathbb{R}^+ \cup \{0\}$		Exists $\forall A, B, D$; unique for $A < 2\widetilde{B}$	Exists only for $A = 2\widetilde{B}$
Parameter Values Used	A	0.2	0.6
	B	0.5	0.3
	D	0.1	0
Analytical Value		(0, 0.2)	(1, 0.3)
Numerical Value (`PPLANE`)		(0, 0.2)	$\approx (1.0001, 0.29997)$
Type		Sink	Degenerate Node

Continued...

Equilibrium		E_2	E_3
Coordinates in the u-w plane		$E_{2,3} = \left(\frac{2\widetilde{B}}{A \pm \sqrt{A^2 - 4\widetilde{B}^2}}, \frac{A \pm \sqrt{A^2 - 4\widetilde{B}^2}}{2} \right)$	
Range of Parameter Values $A, B \in \mathbb{R}^+$, and $D \in \mathbb{R}^+ \cup \{0\}$		Both exist for $A > 2\widetilde{B}$ only	
Parameter Values Used	A	0.7	
	B	0.3	
	D	0	
Analytical Value		$\approx (0.56574, 0.53027)$	$\approx (1.76759, 0.16972)$
Numerical Value (`PPLANE`)		$\approx (0.56574, 0.53028)$	$\approx (1.76760, 0.16972)$
Type		Saddle	Sink

4 Numerical Methods for an ODE system

In this Chapter we continue working with the ODE counterpart (3.4) of the system (2.1). We use the Finite Difference approach to solve the system numerically and analyze the stability and accuracy of results.

4.1 Finite Difference Approach

In solving numerically a general first-order autonomous ordinary differential equation of the form

$$u'(t) = f(u(t)), t \in (t_0, T) \tag{4.1}$$

with an initial condition

$$u(t_0) = \alpha, \tag{4.2}$$

it is common practice to use a finite difference scheme. Finite difference methods are built on the idea of approximating u on the given interval by evaluating it at the finite number of grid points t_n for $n = 1, ..., M$ (where the time interval is divided into M+1 subintervals of equal length). Hereafter we will denote $u(t_n)$ by u^n, $u(t_{n+1})$ by u^{n+1}, etc. The boundary conditions are t_0 and $t_{M+1} = T$. The size of the uniform grid is denoted by $k = t_{n+1} - t_n$ where $n = 0, ..., M$.

The standard discretization of the derivative on the left hand side of (4.1) is based on the definition of the derivative as the limiting value with $k \to 0$ [2]. This one-sided scheme (Forward Euler) will transform (4.1) into

$$\frac{u^{n+1} - u^n}{k} = f(u^n). \tag{4.3}$$

The accuracy order of the scheme is determined through the calculation of the local truncation error (LTE). In (4.3), LTE is calculated by inserting the true solution into the difference equation and then applying Taylor series expansion to cancel out common factors [2]:

$$\begin{aligned} \tau^n &= \frac{u(t_{n+1}) - u(t_n)}{k} - f(u(t_n)) \\ &= \frac{1}{k}\left(u(t_n) + ku'(t_n) + \frac{1}{2}k^2u''(t_n) + O(k^3) - u(t_n)\right) - u'(t_n) \end{aligned}$$

$$= \frac{1}{2}ku''(t_n) + O(k^3) = O(k), \tag{4.4}$$

with the assumption that u is smooth enough ($u \in C^2$). Since the local truncation error is $O(k)$, Forward Euler is first-order accurate.

A good example of the standard second order accurate scheme is the centered approximation to the first derivative. Its application to (4.1) produces

$$\frac{u^{n+1} - u^{n-1}}{2k} = f(u^n). \tag{4.5}$$

In dealing with the system of m ODE's (where $m \in \mathbb{Z}^+$), we can express it through (4.1), where $u(t) \in \mathbb{R}^m$. It is important to keep the accuracy of derivative approximations consistent among the equations, since the system is only as good as its least accurate member.

To guarantee that the global error of the approximation will exhibit the same rate of convergence as the local truncation error, we need the *absolute stability* condition [2]. For a linear homogeneous system, we can express (4.1) as

$$u' = Au, \tag{4.6}$$

where A is an $m \times m$ coefficient matrix. A necessary condition for absolute stability is that $k\lambda$ be in the stability region for each eigenvalue λ of A. Stability region $R(z)$ is determined by

$$u^{n+1} = R(z)u^n, \tag{4.7}$$

where $R(z)$ (which is usually a polynomial for an explicit method or a rational function for an implicit method) is some function of $z = k\lambda$.

For example, in the case of Forward Euler, considering $f(u^n) = \lambda u^n$ for (4.3), $R(z) = 1+z$. We want to bind the absolute value of $R(z)$ by 1 in order to ensure absolute stability. It follows that the region of absolute stability is the disk in the complex z plane of radius 1 centered at -1, described by the equation $|1 + k\lambda| \leq 1$ [2].

For nonlinear systems, we would want $k\lambda$ to be in the stability region for each eigenvalue λ of the Jacobian matrix $f'(u)$ [2]. The time step k is chosen in accordance with the possible range of the eigenvalues.

Unfortunately, an explicit method like Forward Euler has a bounded region of stability that extends only out to $\mathrm{Re}(\lambda) = -2$; this implies that the size of the time step is severely

limited by the eigenvalue with the largest magnitude, i.e. $k \approx \frac{2}{|\lambda_{max}|}$ [2]. If the solution is very smooth and does not vary significantly as time goes on, Forward Euler will waste computational effort, when in reality we would like to be able to take larger time steps.

It follows that explicit methods (producing bounded stability regions) are inefficient in solving problems with a large range of the eigenvalues. In the case when the "stiffness ratio"

$$\frac{\max|\lambda|}{\min|\lambda|} \tag{4.8}$$

is large, we would like to pick a numerical method with a large stability region that extends far into the left half-plane.

An ODE method is said to be *A-stable* if its region of absolute stability includes $z \in \mathbb{C}$ such that $\mathrm{Re}(z) \leq 0$ [2]. Implicit methods like Backward Euler

$$\frac{u^{n+1} - u^n}{k} = f(u^{n+1}) \tag{4.9}$$

and Trapezoidal

$$\frac{u^{n+1} - u^n}{k} = \frac{1}{2}(f(u^n) + f(u^{n+1})) \tag{4.10}$$

fit the description. In fact, both methods encompass the entire left half-plane in their stability regions (as a consequence, any time step is allowed provided that the real part of the eigenvalues is negative).

Although the Trapezoidal method is A-stable, it does not damp the oscillations in rapidly transient solutions (the computed solution keeps revolving without damping about the true solution that rapidly decays). This effect on the quality of solutions is described by the notion of *L-stability*, which requires a bit more than an A-stability. In order to have the oscillations damp quickly (in a single time step), the true decay of the transient requires that, in addition to A-stability, $\lim_{z\to\infty} |R(z)| = 0$ [2]. Backward Euler is a well-known L-stable method.

Implicit method like the Backward Euler is considered in general to be better than an explicit one like Forward Euler (here we assume that we would rather under predict the solution than over predict it, but in general the decision depends on the meaning behind the system). Backward Euler does have drawbacks in the case when the right hand side of (4.9) contains nonlinear terms. Then the numerical solution will transform into a Newton's method,

whose fully implicit structure significantly differs from other methods. The inconvenience of the Newton's method is counterbalanced by the enhanced stability (in theory) of the system.

In a case of a nonlinear system, absolute stability is not a guaranteed property. It may be difficult to assess the behavior of computed solutions in terms of stability since it strongly depends on a particular form of $f(t, u(t))$. To experimentally determine the behavior of the solution, we can compute the value (at each time step) of the 2-norm of the solutions for the ODE system and determine if it decreases with time. If the norm approaches zero as $t \to \infty$, the method is stable. A smaller time step should be chosen if the method does not definitively display stable results.

4.2 Management of the Nonlinear Term

In this Chapter we discretize the system (3.4) with the first order forward difference in time

$$\frac{u^{n+1} - u^n}{k} = (wu^2)|_{t^{n+1}} - (B + D)u|_{t^*}, \tag{4.11a}$$

$$\frac{w^{n+1} - w^n}{k} = A - w|_{t^*} - (wu^2)|_{t^{n+1}}, \tag{4.11b}$$

where in Section 4.5 we use $t^* = t^n$ and in Sections 4.6, 4.7, 4.8 we use $t^* = t^{n+1}$.

Since the equations have a nonlinear term $(wu^2)|_{t^{n+1}}$, we treat it either implicitly (with Newton's method or through linearization), or explicitly (with forward difference in time), or with a combination of both (via Sequential method also called Semi-implicit). Table 4.1 gives a brief description of these particular methods.

The linearization of the implicit nonlinear terms is done in following way:

$$\begin{aligned} f(u^{n+1}, w^{n+1}) &= w^{n+1}(u^{n+1})^2 \\ &\approx f(u^n, w^n) + f_u(u^n, w^n)(u^{n+1} - u^n) + f_w(u^n, w^n)(w^{n+1} - w^n) \\ &= w^n(u^n)^2 + 2u^n w^n(u^{n+1} - u^n) + (u^n)^2(w^{n+1} - w^n). \end{aligned} \tag{4.12}$$

TABLE 4.1: Management of the Nonlinear Term (wu^2) at t_{n+1}.

Method	Discretization Type	Overview
Forward Euler	$w^n(u^n)^2$	Fully explicit
Sequential (Time Lagging)	$w^{n+1}(u^n)^2$	Implicit-explicit
Linearized Backward Euler	$w^{n+1}(u^{n+1})^2$ linearized about u^n, w^n	see (4.12)
Newton's Method	$w^{n+1}(u^{n+1})^2$	Fully Implicit

4.3 Initial Conditions

In Figures 4.1 - 4.15 supplied for the ODE methods in Chapter 4 we use initial conditions with a small random perturbation around the equilibria

$$u_0(j) = Ru_j, \quad w_0(j) = Rw_j, \tag{4.13}$$

where R is a random (uniformly distributed) value from the interval $(0.9, 1.1)$. We also use two additional initial conditions

$$u_0 = 0.5, \quad w_0 = 3, \tag{4.14}$$

$$u_0 = 1.5, \quad w_0 = 2 \tag{4.15}$$

for visual impact.

4.4 ODE45 Solver

We illustrate the behavior of solutions to (3.4) using a well-known MATLAB solver ODE45. This Dormand-Prince solver is a very accurate explicit general solver that uses a combination of fourth and fifth order Runge-Kutta formulas.

Using the different values of parameter sets outlined in (3.34), we plot the results in Figures 4.1, 4.2, 4.3. In the absence of a known solution, ODE45 will help us establish the reasonability of each method's results as well as determine the general behavior of solutions for particular values of the parameter set.

ODE45 automatically chooses the size of the time step. In the interest of improving accuracy and stability of the results, we will refine the time step so that $k = 5 \cdot 10^{-7}$, which

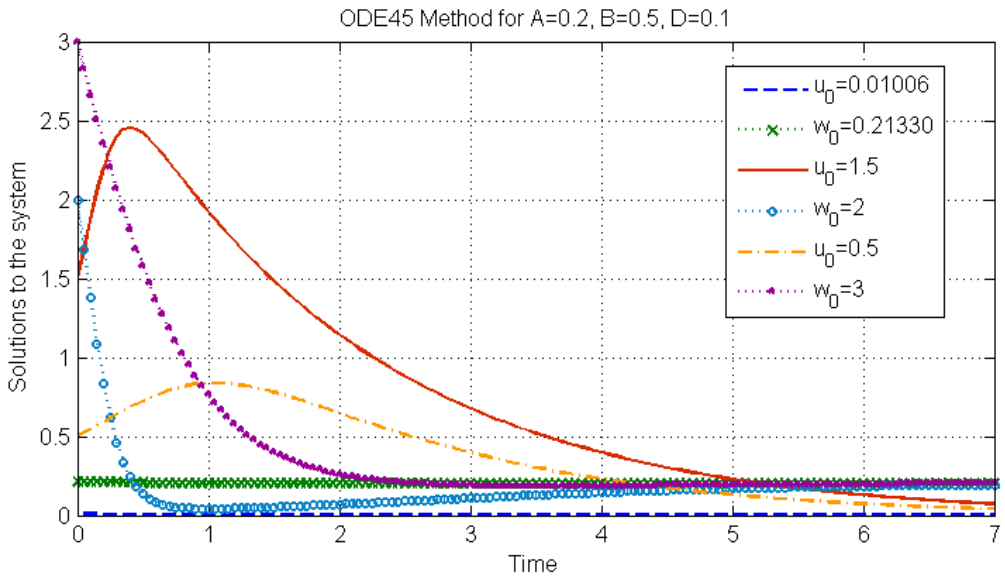

FIGURE 4.1: ODE 45 method for the set of parameters (3.34a). Expected equilibrium E is given in Table 3.1. Initial conditions are displayed in consecutive pairs according to Chapter 4.3.

will always be at least 4 orders of magnitude higher than other methods'.

We will use $k = 0.05$ in other ODE methods unless specified otherwise.

4.5 Forward Euler Method

In this section discretization of the system (3.4) is handled with the forward difference on time derivatives and explicit treatment of their right hand sides. We get

$$\frac{u^{n+1} - u^n}{k} = w^n(u^n)^2 - (B + D)u^n, \tag{4.16a}$$

$$\frac{w^{n+1} - w^n}{k} = A - w^n - w^n(u^n)^2. \tag{4.16b}$$

Leaving u^{n+1}, w^{n+1} terms on the left and moving everything to the right transforms (4.16) into

$$u^{n+1} = (1 - (B + D)k)u^n + kw^n(u^n)^2, \tag{4.17a}$$

$$w^{n+1} = kA + (1 - k)w^n - kw^n(u^n)^2. \tag{4.17b}$$

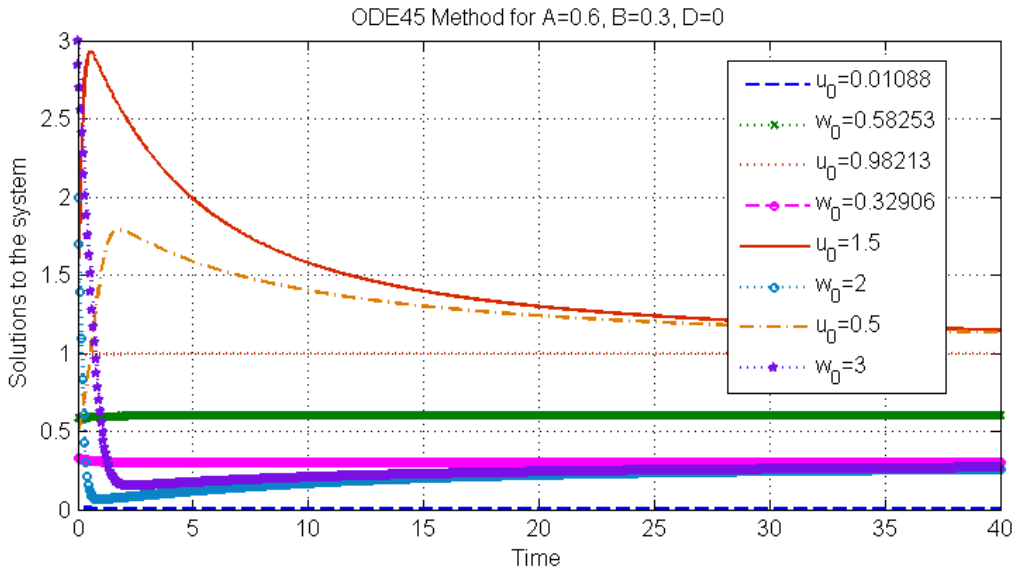

FIGURE 4.2: ODE 45 method for the set of parameters (3.34b). Expected equilibria E, E_1 are given in Table 3.1. Initial conditions are displayed above in consecutive pairs according to Chapter 4.3.

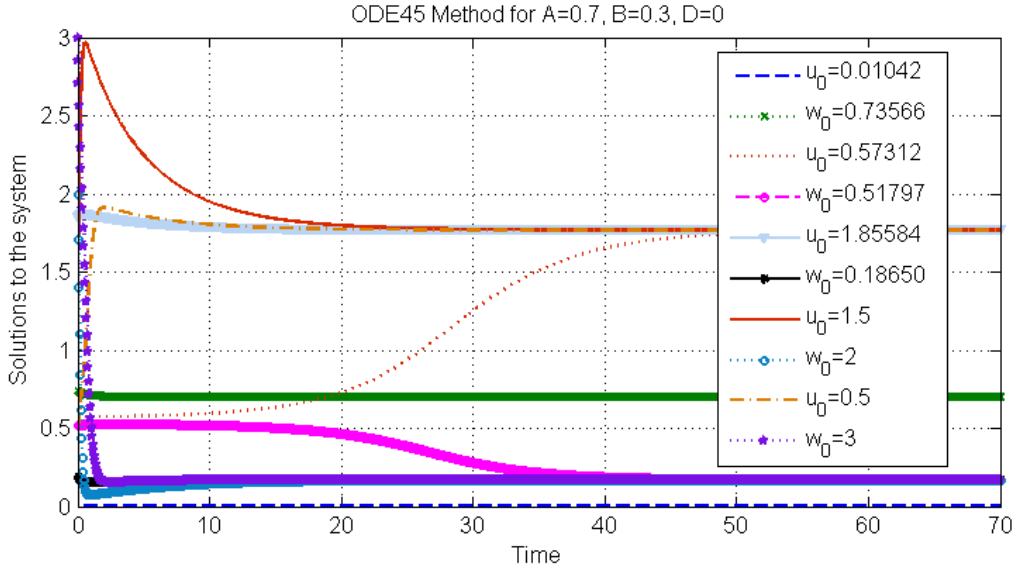

FIGURE 4.3: ODE 45 method for the set of parameters (3.34c). Expected equilibria E, E_2, E_3 are given in Table 3.1. Initial conditions are displayed above in consecutive pairs according to Chapter 4.3.

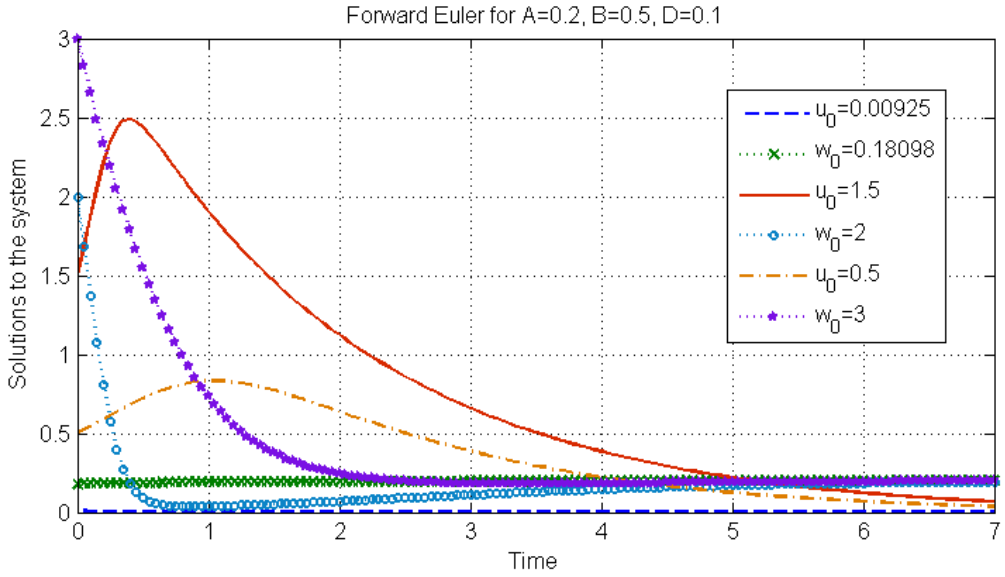

FIGURE 4.4: Forward Euler method applied as shown in Section 4.5 for the set of parameters (3.34a). Expected equilibrium E is given in Table 3.1. Initial conditions are displayed in consecutive pairs according to Chapter 4.3.

The explicit terms on the right hand side of the equation determine the value of the solution at the future time step.

The plots of solutions of (4.16) for parameter values discussed in (3.34) are shown in Figures 4.4, 4.5, 4.6. The results are respectively similar to those obtained through the ODE45 method in Figures 4.1, 4.2, 4.3.

4.6 Sequential Method

In this section we discretize the nonlinear term in an implicit-explicit way described in Table 4.1. Time-lagging helps smooth out the nonlinear term of our ODE system (3.4). Applying the forward difference to the time derivative, we get

$$\frac{u^{n+1}-u^n}{k} = w^{n+1}(u^n)^2 - (B+D)u^{n+1}, \tag{4.18a}$$

$$\frac{w^{n+1}-w^n}{k} = A - w^{n+1} - w^{n+1}(u^n)^2. \tag{4.18b}$$

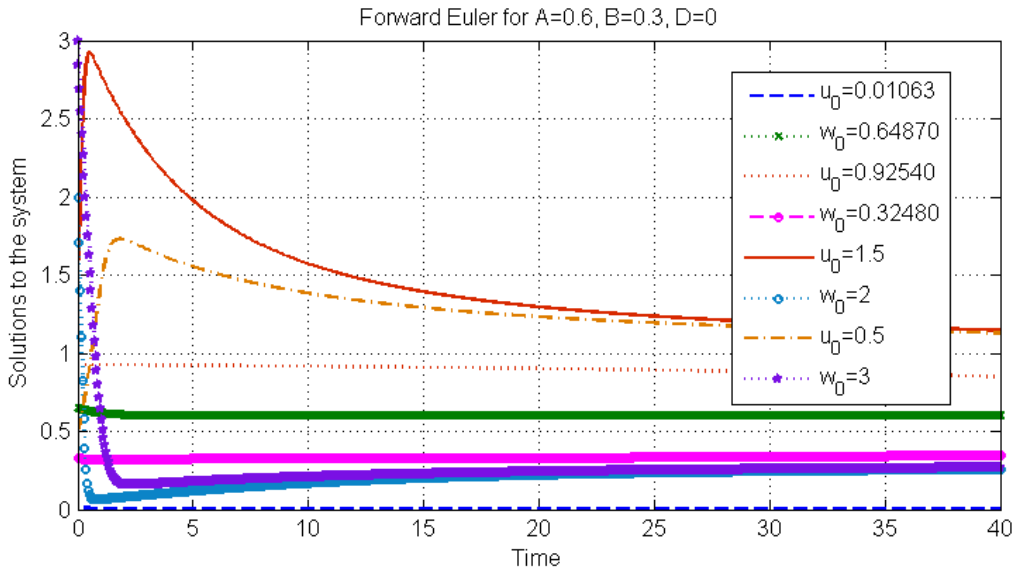

FIGURE 4.5: Forward Euler method applied as shown in Section 4.5 for the set of parameters (3.34b). Expected equilibria E, E_1 are given in Table 3.1. Initial conditions are displayed above in consecutive pairs according to Chapter 4.3.

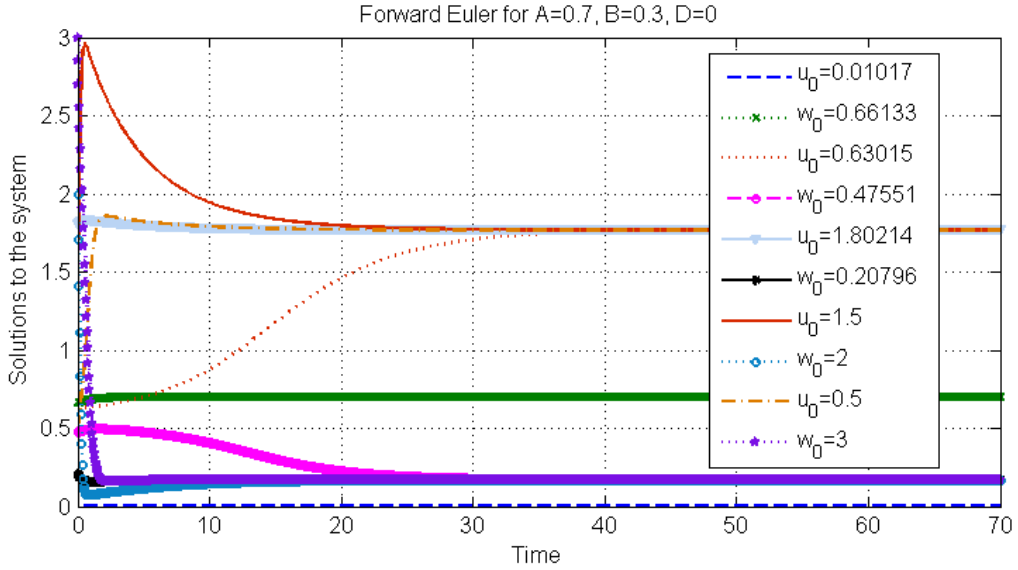

FIGURE 4.6: Forward Euler method applied as shown in Section 4.5 for the set of parameters (3.34c). Expected equilibria E, E_2, E_3 are given in Table 3.1. Initial conditions are displayed above in consecutive pairs according to Chapter 4.3.

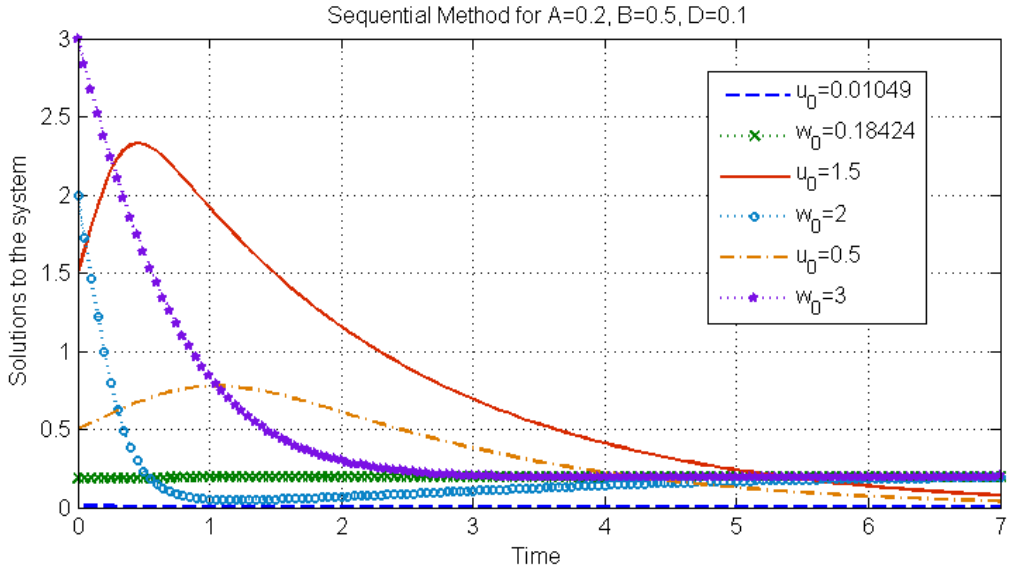

FIGURE 4.7: Sequential method applied as shown in Section 4.6 for the set of parameters (3.34a). Expected equilibrium E is given in Table 3.1. Initial conditions are displayed in consecutive pairs according to Chapter 4.3.

Rearranging the terms, we obtain

$$(1 + k(B + D))u^{n+1} - k(u^n)^2 w^{n+1} = u^n, \tag{4.19a}$$

$$0 \cdot u^{n+1} + (1 + k + k(u^n)^2)w^{n+1} = w^n + kA. \tag{4.19b}$$

In matrix form, problem (4.18) is posed as a linear system

$$\begin{bmatrix} 1 + k(B + D) & -k(u^n)^2 \\ 0 & 1 + k + k(u^n)^2 \end{bmatrix} \cdot \begin{bmatrix} u^{n+1} \\ w^{n+1} \end{bmatrix} = \begin{bmatrix} u^n \\ w^n + kA \end{bmatrix}. \tag{4.20}$$

Making use of MATLAB's backslash operator, we can solve the system easily.

Solutions to the system (4.18) are shown in Figures 4.7, 4.8, 4.9 for the corresponding sets of parameters in (3.34). We see that Sequential method solutions are similar to those of the ODE45 method (in Figures 4.1, 4.2, 4.3) and the Forward Euler method (in Figures 4.4, 4.5, 4.6) respectively.

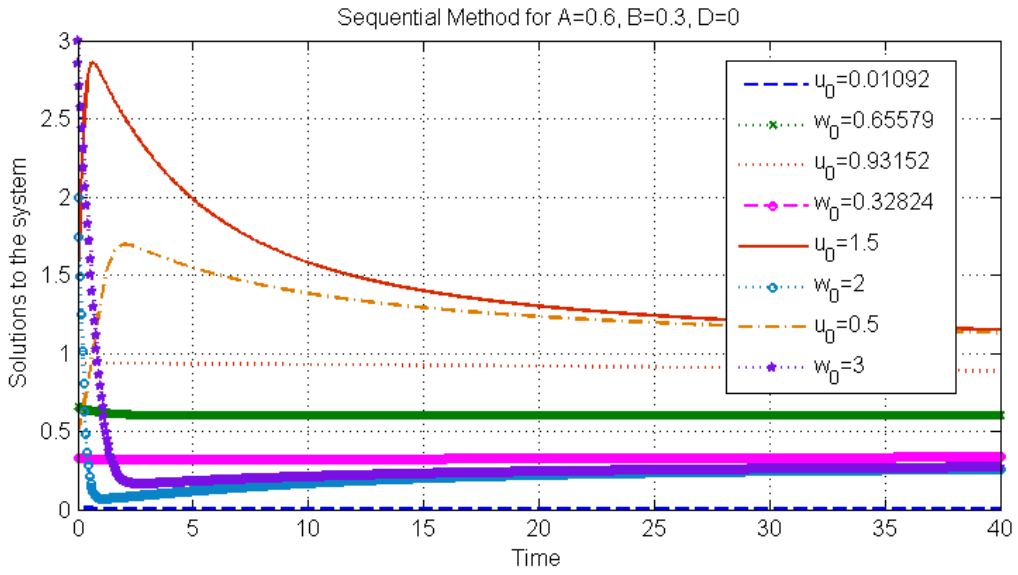

FIGURE 4.8: Sequential method applied as shown in Section 4.6 for the set of parameters (3.34b). Expected equilibria E, E_1 are given in Table 3.1. Initial conditions are displayed above in consecutive pairs according to Chapter 4.3.

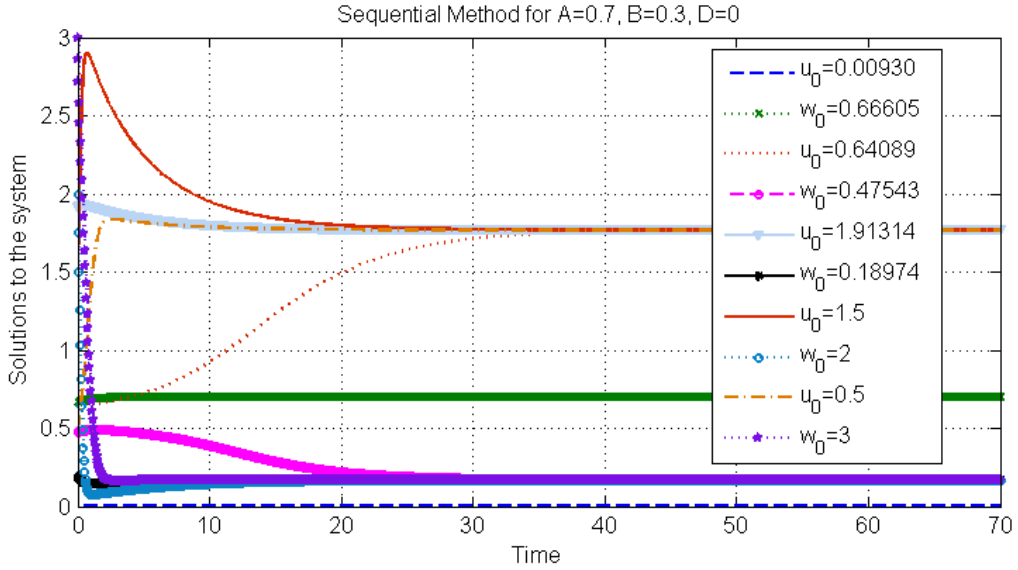

FIGURE 4.9: Sequential method applied as shown in Section 4.6 for the set of parameters (3.34c). Expected equilibria E, E_2, E_3 are given in Table 3.1. Initial conditions are displayed above in consecutive pairs according to Chapter 4.3.

4.7 Linearized Backward Euler Method

In the case when the nonlinear term is discretized implicitly, we can either linearize the system, or apply Newton's method (as will be shown in Section 5.7). The discretization is as follows:

$$\frac{u^{n+1}-u^n}{k} = w^{n+1}(u^{n+1})^2 - (B+D)u^{n+1}, \tag{4.21a}$$

$$\frac{w^{n+1}-w^n}{k} = A - w^{n+1} - w^{n+1}(u^{n+1})^2. \tag{4.21b}$$

In this section we will linearize according to (4.12) in order to obtain

$$\begin{aligned} u^{n+1} &= u^n - k(B+D)u^{n+1} \\ &\quad + k[w^n(u^n)^2 + 2u^n w^n(u^{n+1}-u^n) + (u^n)^2(w^{n+1}-w^n)] \end{aligned} \tag{4.22a}$$

$$\begin{aligned} w^{n+1} &= w^n + kA - kw^{n+1} \\ &\quad - k[w^n(u^n)^2 + 2u^n w^n(u^{n+1}-u^n) + (u^n)^2(w^{n+1}-w^n)] \end{aligned} \tag{4.22b}$$

Re-arranging the terms to isolate the future time solutions yields

$$[1 + k(B+D) - 2ku^n w^n]u^{n+1} + [-k(u^n)^2]w^{n+1} = u^n - 2kw^n(u^n)^2, \tag{4.23a}$$

$$[2ku^n w^n]u^{n+1} + [1 + k + k(u^n)^2]w^{n+1} = kA + w^n + 2kw^n(u^n)^2. \tag{4.23b}$$

Then, the linear system could be represented as

$$\begin{bmatrix} 1 + k(B+D) - 2ku^n w^n & -k(u^n)^2 \\ 2ku^n w^n & 1 + k + k(u^n)^2 \end{bmatrix} \cdot \begin{bmatrix} u^{n+1} \\ w^{n+1} \end{bmatrix} = \begin{bmatrix} u^n - 2kw^n(u^n)^2 \\ kA + w^n + 2kw^n(u^n)^2 \end{bmatrix}. \tag{4.24}$$

The plots for the linearized system in (4.22) are shown in Figures 4.10, 4.11, 4.12 corresponding to the sets of parameters (3.34). The results are respectively similar to those obtained through the ODE45 method (in Figures 4.1, 4.2, 4.3), Forward Euler method (in Figures 4.4, 4.5, 4.6), and Sequential method (in Figures 4.7, 4.8, 4.9).

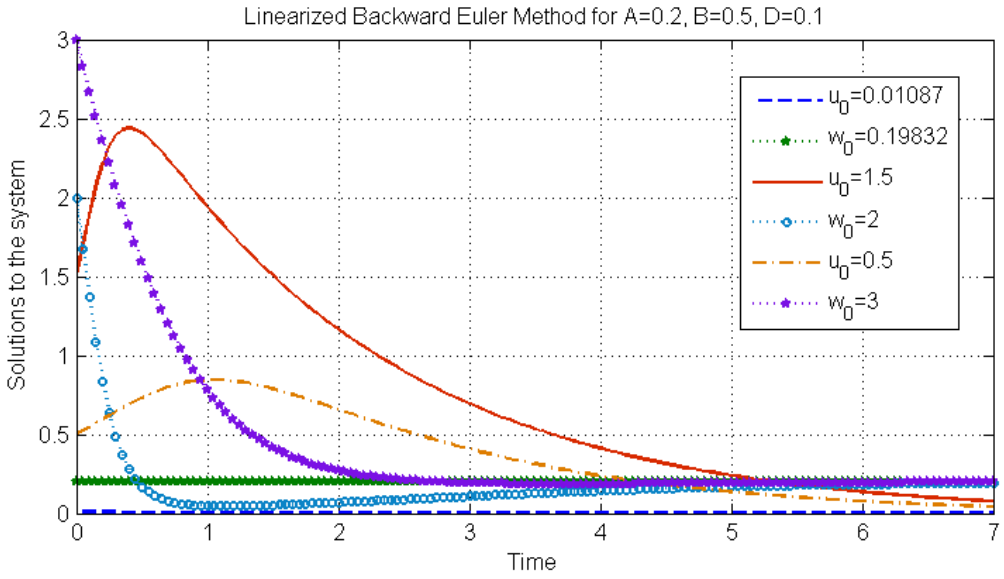

FIGURE 4.10: Linearized Backward Euler method applied as shown in Section 4.7 for the set of parameters (3.34a). Expected equilibrium E is given in Table 3.1. Initial conditions are displayed in consecutive pairs according to Chapter 4.3.

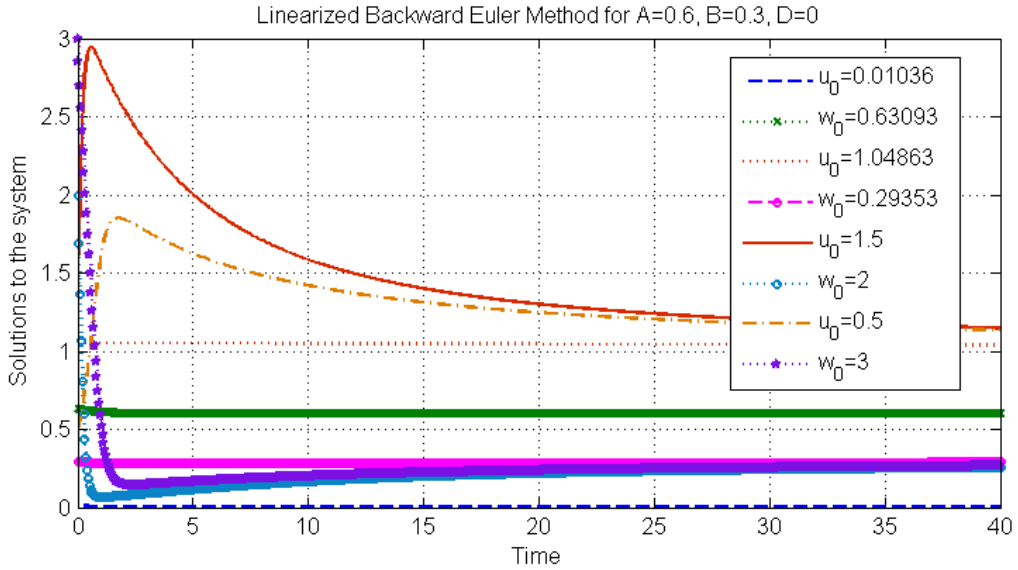

FIGURE 4.11: Linearized Backward Euler method applied as shown in Section 4.7 for the set of parameters (3.34b). Expected equilibria E, E_1 are given in Table 3.1. Initial conditions are displayed above in consecutive pairs according to Chapter 4.3.

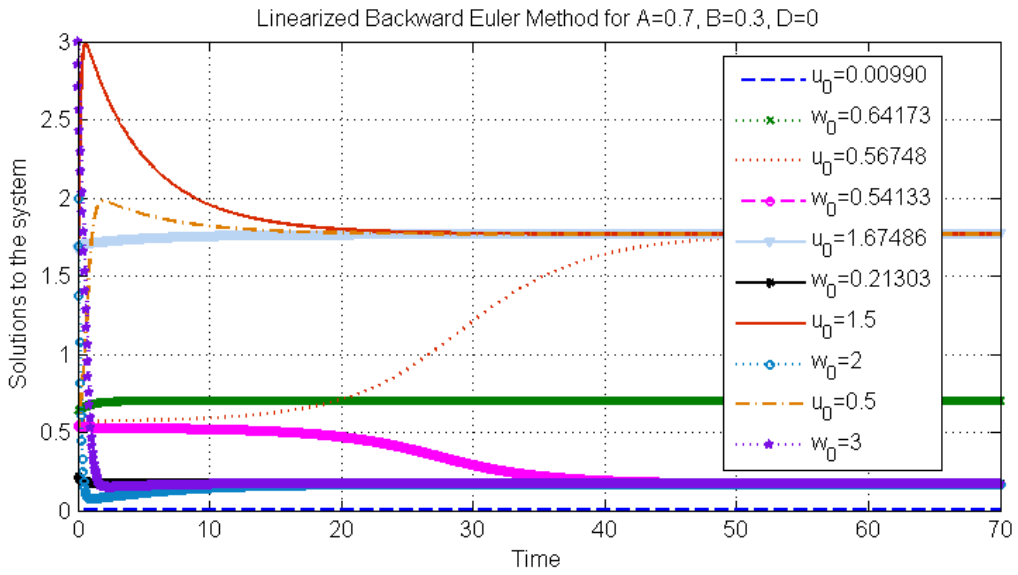

FIGURE 4.12: Linearized Backward Euler applied as shown in Section 4.7 for the set of parameters (3.34c). Expected equilibria E, E_2, E_3 are given in Table 3.1. Initial conditions are displayed above in consecutive pairs according to Chapter 4.3.

4.8 Newton's Method

Working with equations (4.21a), (4.21b) (where the nonlinear term is treated implicitly), we transfer all the terms to their respective left hand sides, naming the first residual $\psi(u^{n+1}, w^{n+1})$ and the second residual $\phi(u^{n+1}, w^{n+1})$. The equations are

$$\psi(u^{n+1}, w^{n+1}) = u^{n+1} - [u^n + kw^{n+1}(u^{n+1})^2 - k(B+D)u^{n+1}] = 0, \tag{4.25a}$$

$$\phi(u^{n+1}, w^{n+1}) = w^{n+1} - [w^n + kA - kw^{n+1} - kw^{n+1}(u^{n+1})^2] = 0. \tag{4.25b}$$

For the code we will need the Jacobian Determinant of ψ and ϕ, where the terms u^n and w^n are being considered as constants (since they are evaluated at the current time step):

$$\begin{aligned} Jac(\psi, \phi) &= \begin{bmatrix} \dfrac{\partial \psi}{\partial u^{n+1}} & \dfrac{\partial \psi}{\partial w^{n+1}} \\ \dfrac{\partial \phi}{\partial u^{n+1}} & \dfrac{\partial \phi}{\partial w^{n+1}} \end{bmatrix} \\ &= \begin{bmatrix} 1 + k(B+D) - 2kw^{n+1}u^{n+1} & -k(u^{n+1})^2 \\ 2kw^{n+1}u^{n+1} & 1 + k + k(u^{n+1})^2 \end{bmatrix}. \end{aligned} \tag{4.26}$$

Newton's method computes the value of the solution in the future by taking a guess

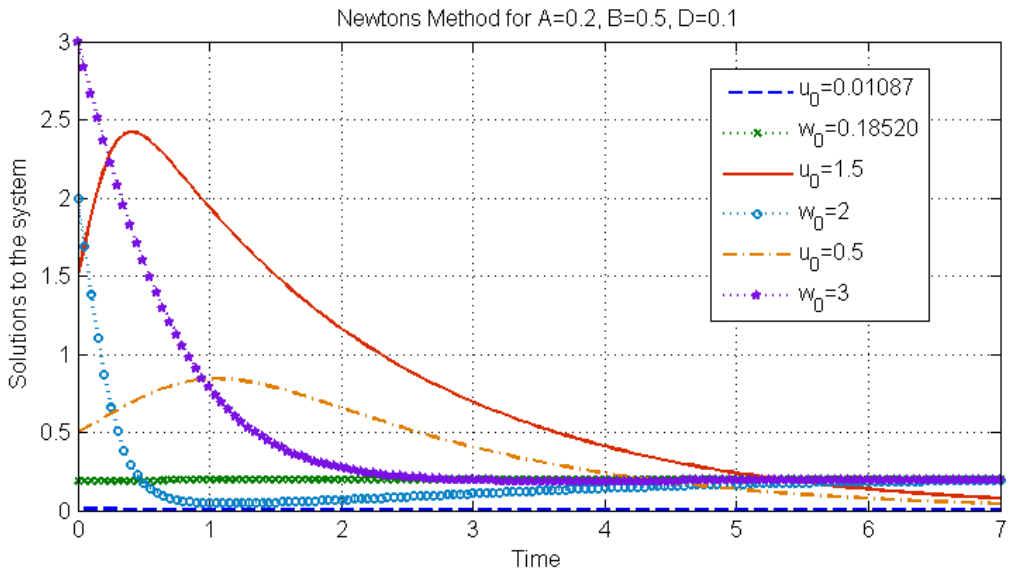

FIGURE 4.13: Newton's method applied as shown in Section 4.8 for the set of parameters (3.34a). Expected equilibrium E is given in Table 3.1. Initial conditions are displayed in consecutive pairs according to Chapter 4.3.

(that would hopefully lead to convergence) and following along the line tangent to the residual at that point to obtain the horizontal-axis intercept. The intercept becomes the new guess in the next time step. The calculations are

$$u^{n+1} = u^n - \text{Jac}^{-1}\big|_{u^n} \cdot \text{Residual}\big|_{u^n}. \tag{4.27}$$

We then rename the new solution u to be the old one, and continue to iterate until the norm of the residual is less than a certain tolerance level (for us, 10^{-12}). Provided that the initial guess is good and the solution converges, the norm of the residual should be approaching zero. Another threat is singularity of the Jacobian (we check how close it comes to zero at each time step, and if the value is "sufficiently" close, the coded error message should appear). The code for Newton's Method is provided in Appendix A.

The plots for the Newton's method for the system in (4.25) are shown for each of the parameter sets (3.34) in Figures 4.13, 4.14, 4.15. The graphs are respecively similar to ODE45 Figures 4.1, 4.2, 4.3, Forward Euler Figures 4.4, 4.5, 4.6, Sequential Figures 4.7, 4.8, 4.9, and Backward Euler method's Figures 4.10, 4.11, 4.12.

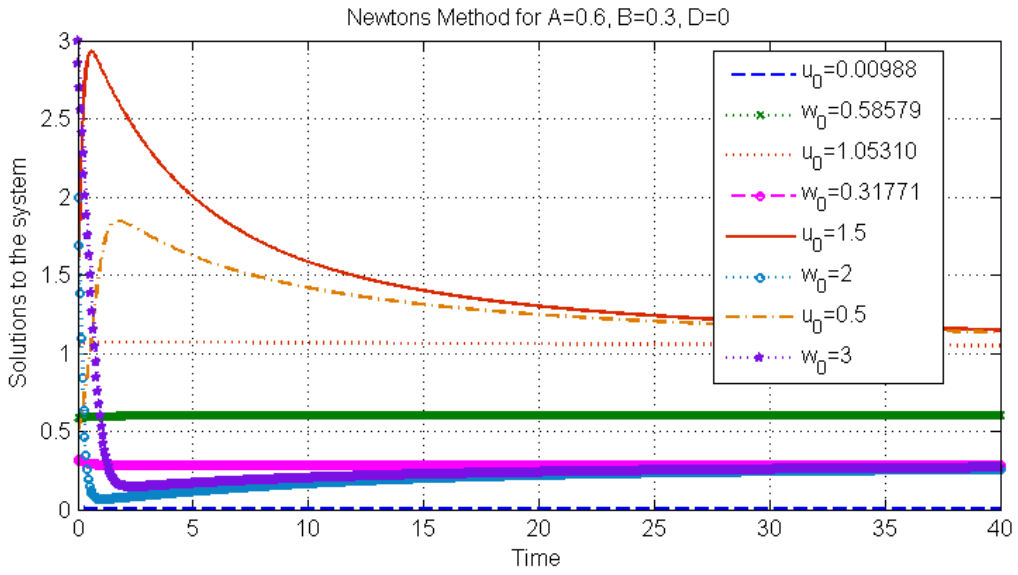

FIGURE 4.14: Newton's method applied as shown in Section 4.8 for the set of parameters (3.34b). Expected equilibria E, E_1 are given in Table 3.1. Initial conditions are displayed above in consecutive pairs according to Chapter 4.3.

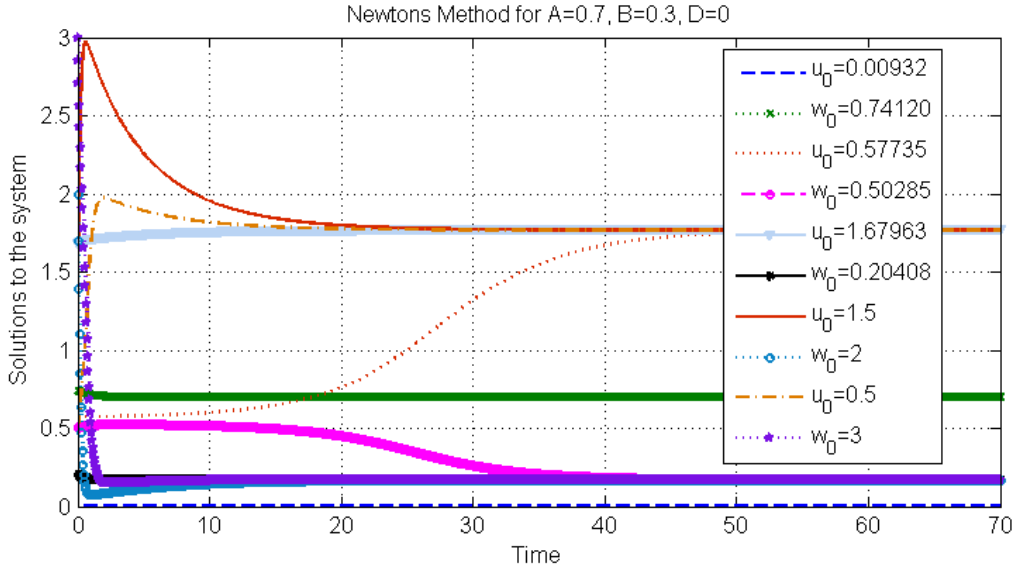

FIGURE 4.15: Newton's method applied as shown in Section 4.8 for the set of parameters (3.34c). Expected equilibria E, E_2, E_3 are given in Table 3.1. Initial conditions are displayed above in consecutive pairs according to Chapter 4.3.

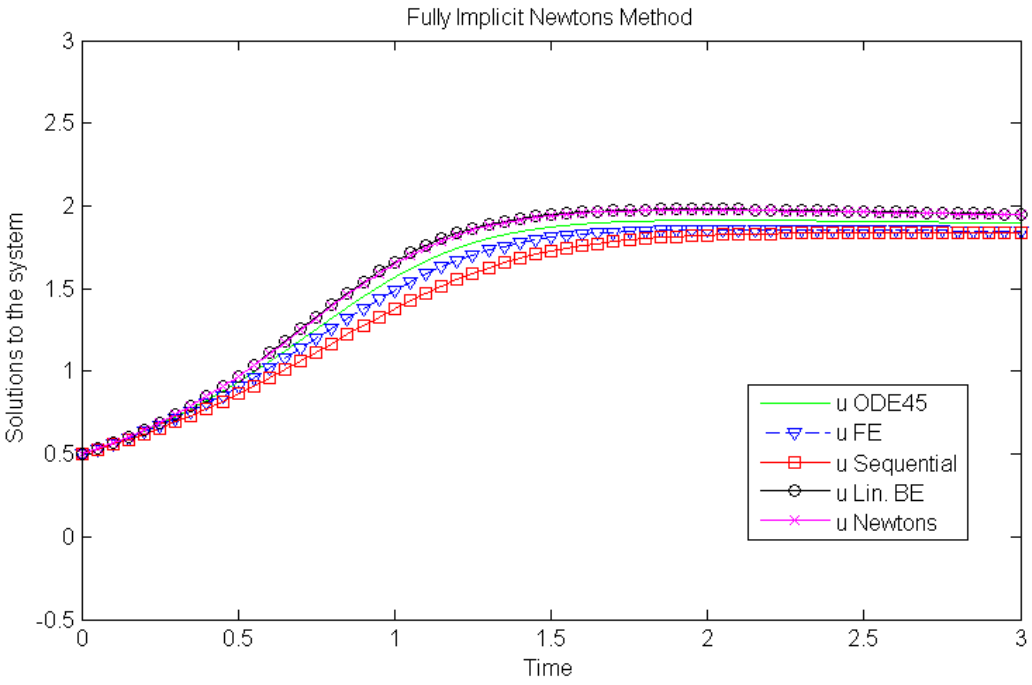

FIGURE 4.16: Vegetation U (for the set of parameters (3.34c) and $k = 0.05$).

4.9 Discussion of Numerical Results

To visually compare the way each method solves the ODE system (3.4), each solution is plotted on the same graph. Solution u is offered in Figure 4.16 and solution w in Figure 4.18. Both figures use the set of parameters (3.34c) and initial conditions $u_0 = 0.5, w_0 = 3$ with time stepping size $k = 0.05$.

Since we don't know the true solution to the system (3.4) but need an estimation of the error, we will consider the differences between ODE45 and other methods. Notice that Linearized Backward Euler and Newton's methods over predict and Forward Euler and Sequential methods under predict the solution of ODE45 as is visible in Figures 4.16 and 4.18.

Tables 4.2, 4.4 show the 2-norm of the errors the methods display over the time interval [0,3] for different magnitudes of the time steps. The results for $k = 0.05$ correlate to the visual evidence in Figures 4.16, 4.18: Sequential method is the farthest from ODE45, and Forward Euler method is the closest. However, as we decrease k to 0.005, Newton's method becomes better than Forward Euler. Alternatively, increasing k to 0.5 moves Forward Euler ahead of the rest, and places Linearized Backward Euler last. Sequential Method reacts especially

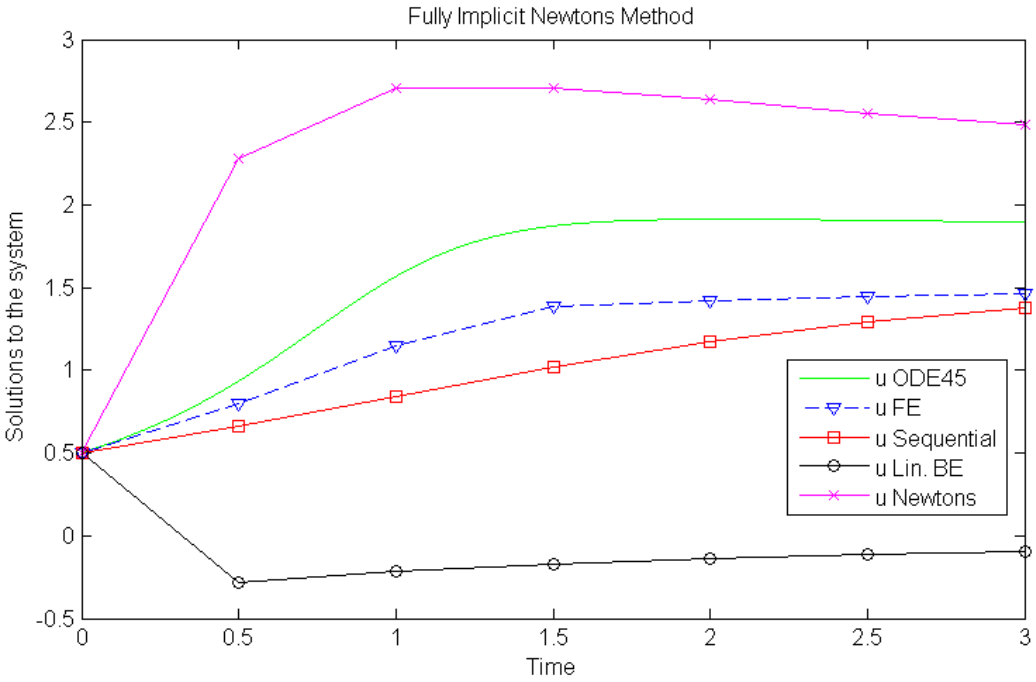

FIGURE 4.17: Vegetation U (for the set of parameters (3.34c) and $k = 0.5$).

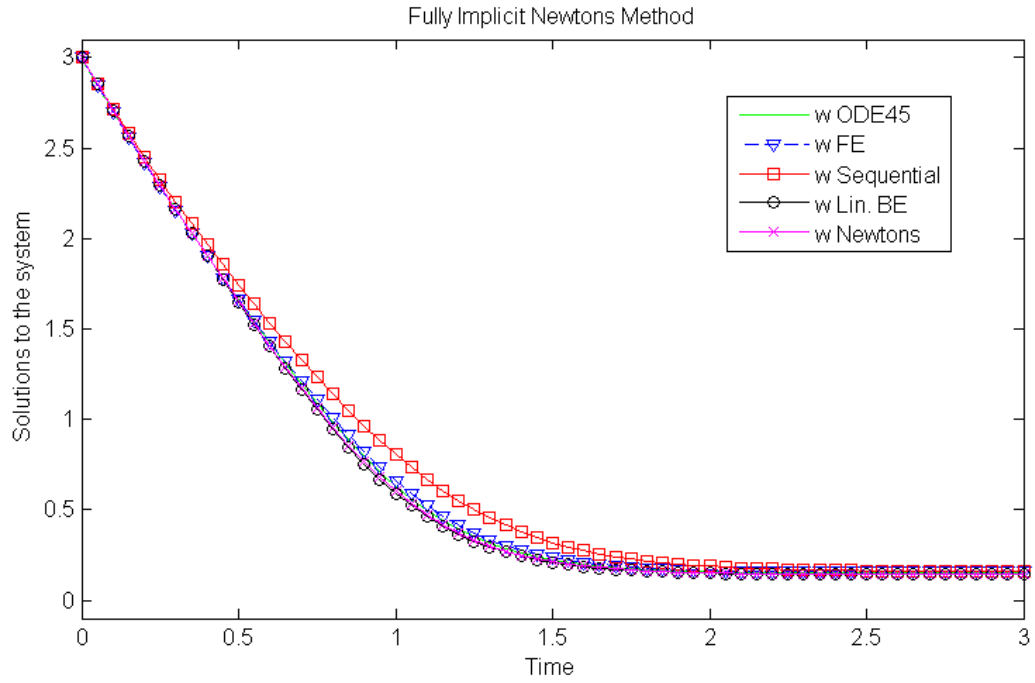

FIGURE 4.18: Water W (for the set of parameters (3.34c) and $k = 0.05$).

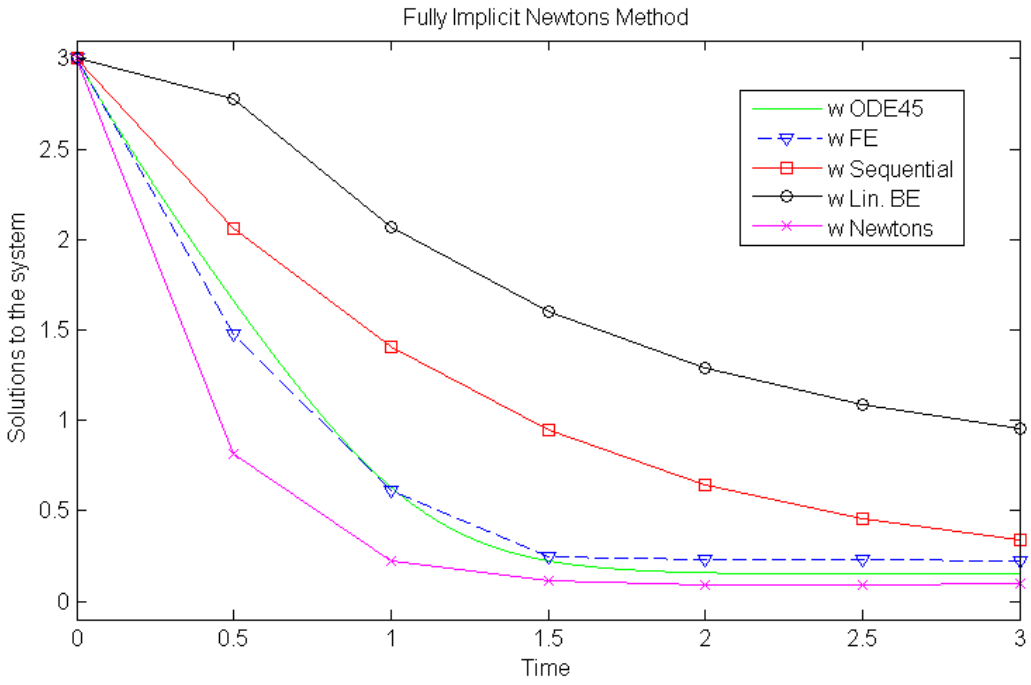

FIGURE 4.19: Water W (for the set of parameters (3.34c) and $k = 0.5$).

badly to the largest time step as is seen in Figure 4.17. The method does not seem to capture the general behavior of the solution when $k = 0.5$.

Each method is consistent due to the decrease in error as $k \to 0$. Since ODE45 method is not actually a true solution to the system (3.4), we cannot expect to see a nearly perfect convergence order. Recall that the infinity-norm only considers the largest error and ignores the rest; thus, it may carry less information than the 2-norm. In fact, Tables 4.3 and 4.5 are of order $O(1)$, and Tables 4.2 and 4.4 are of order less than $O(1)$.

TABLE 4.2: The 2-norm of errors in U (in comparison to the ODE45 solution) for different time stepping over the time $T = 3$, initial conditions (4.14), and parameter set (3.34c).

Method	Error for k=0.5	Error for k=0.05	Error for k=0.005
FE	0.7301	0.0937	0.0100
Sequential	1.1236	0.1948	0.0216
Linearized BE	3.2482	0.1151	0.0096
Newton's	1.5978	0.1026	0.0095

TABLE 4.3: The l_∞-norm of errors in U (in comparison to the ODE45 solution) for different time stepping over the time $T = 3$, initial conditions (4.14), and parameter set (3.34c).

Method	Error for k=0.5	Error for k=0.05	Error for k=0.005
FE	0.3492	0.0165	$5.6564 \cdot 10^{-4}$
Sequential	0.6038	0.0430	$15.6757 \cdot 10^{-4}$
Linearized BE	1.4522	0.0218	$5.5842 \cdot 10^{-4}$
Newton's	0.9556	0.0189	$5.4979 \cdot 10^{-4}$

TABLE 4.4: The 2-norm of errors in W (in comparison to the ODE45 solution) for different time stepping over the time $T = 3$, initial conditions (4.14), and parameter set (3.34c).

Method	Error for k=0.5	Error for k=0.05	Error for k=0.005
FE	0.1628	0.0204	0.0025
Sequential	0.9047	0.1526	0.0169
Linearized BE	2.0006	0.0341	0.0023
Newton's	0.6733	0.0264	0.0022

TABLE 4.5: The l_∞-norm of errors in W (in comparison to the ODE45 solution) for different time stepping over the time $T = 3$, initial conditions (4.14), and parameter set (3.34c).

Method	Error for k=0.5	Error for k=0.05	Error for k=0.005
FE	0.1314	0.0057	$2.1959 \cdot 10^{-4}$
Sequential	0.5437	0.0386	$14.1234 \cdot 10^{-4}$
Linearized BE	1.0116	0.0099	$2.0819 \cdot 10^{-4}$
Newton's	0.5980	0.0077	$2.0179 \cdot 10^{-4}$

In summary, Linearized Backward Euler method is the most sensitive to changes in the size of k. Forward Euler method is consistently close to solutions of ODE45, reacting rather mildly (in comparison to others) to changes of k.

Figures (4.17), (4.19) display the range of results for a larger time step $k = 0.5$.

To assess the stability of the ODE system numerically, we check the value of the norm $\|(u, w)\|_2$. The print-outs of each implicit method's norm $\|(u, w)\|_2$ up to $T = 30$ are provided

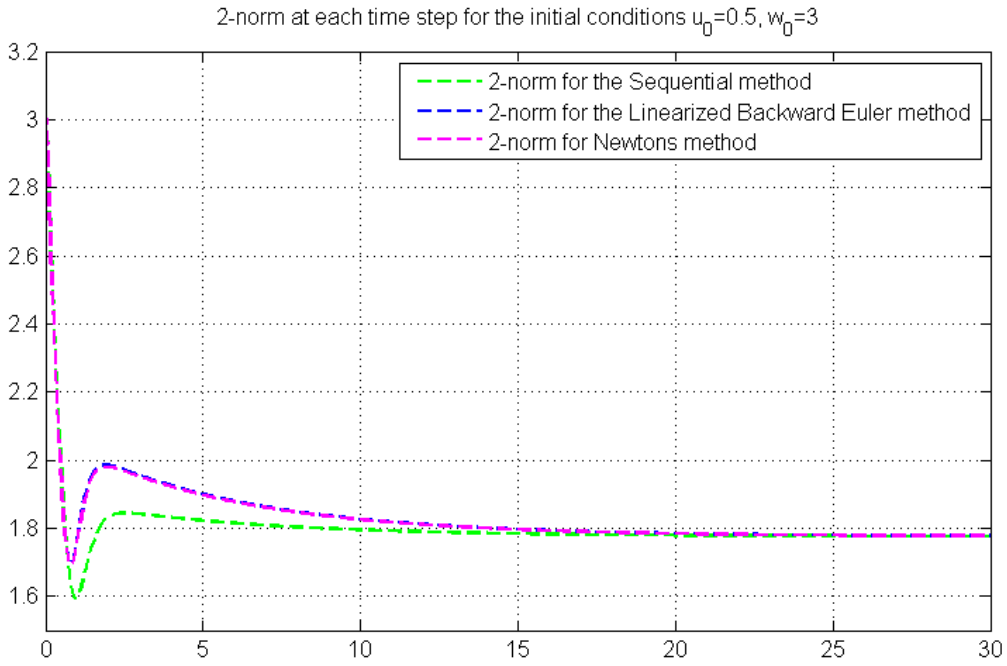

FIGURE 4.20: The 2-norm of the implicit solutions where $k = 0.05$ (for the set of parameters (3.34c)).

in Figure 4.20 (with initial conditions (4.14)). At $T \approx 0.75$, the norms of the Newton's and Linearized Backward Euler methods are at their lowest; Sequential method, however, continues to decrease up to the time $T = 1$. Following further in time, the norms for all the above mentioned methods increase and reach a local maximum at $T = 2$. It appears that $\lim_{t \to +\infty} \|(u, w)\|_2 = 1.7^{+}$. Since the norm decreases and then increases, we do not expect for numerical solution to decrease. The norms for initial conditions (4.15) display similar limit behavior at T increases (results are in Figure 4.21). This does not produce notable results for stability.

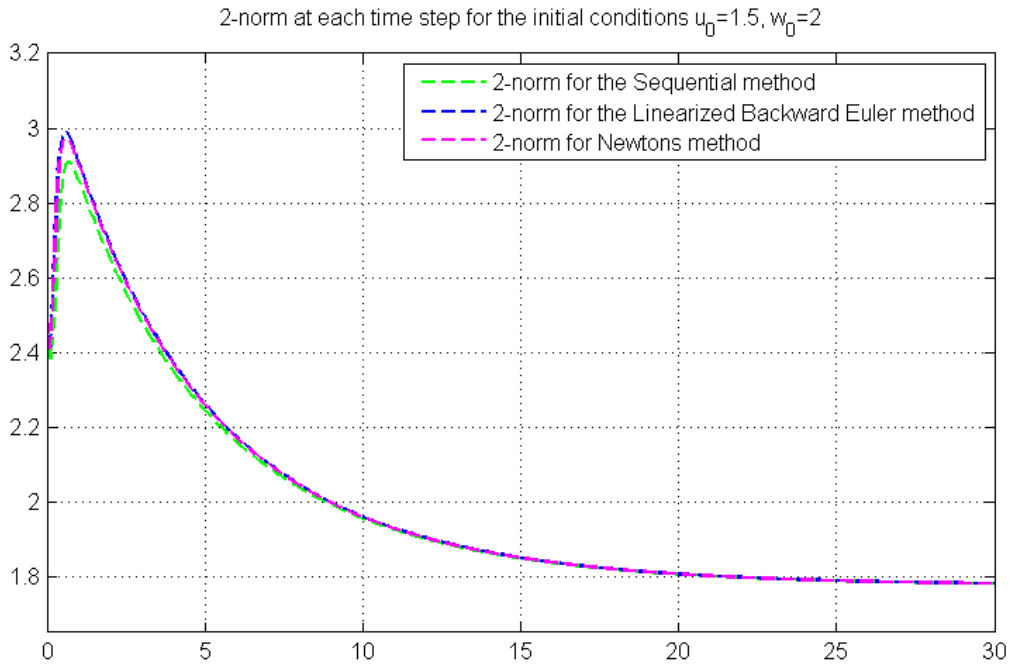

FIGURE 4.21: The 2-norm of the implicit solutions where $k = 0.05$ (for the set of parameters (3.34c)).

5 PDE Methods in 1-D

In this Chapter we work with the full Klausmeier model. We formulate a Finite Difference scheme to approximate its solutions, and consider several variants of handling the nonlinear terms similarly to what was done in Chapter 4 for the ODE system. We develop stability analysis for the simplified version of the system and determine constraints on the time step without which the proposed scheme would be unstable.

We start with a general introduction on numerical schemes and stability for linear advection-diffusion-reaction PDEs. Then we describe the scheme for the Klausmeier model and give details on the nonlinear solver. In Chapter 6 we present numerical results obtained with that scheme.

5.1 Introduction

In studying time-dependent PDEs, we can begin with the diffusion equation (also called the heat equation)

$$u_t = Du_{xx}, \quad x \in (a,b), \quad t > 0 \tag{5.1}$$

where the diffusion coefficient $D > 0$. Suitable initial and boundary conditions must be supplied to complete the model:

$$u(x,0) = f(x), \quad x \in (a,b), \tag{5.2a}$$

$$u(a,t) = g_0(t), \quad u(b,t) = g_1(t), \quad t > 0. \tag{5.2b}$$

The boundary conditions in (5.2b) are of Dirichlet type. Other boundary conditions that could be used would be of Neumann or Robin type.

The Klausmeier system uses periodic boundary conditions as follows:

$$u(a,t) = u(b,t), \quad u'(a,t) = u'(b,t), \quad t > 0, \tag{5.3}$$

where we will be using the first condition in practice through coding. The second condition of derivative equality on the boundary is implied by the meaning of periodic boundary conditions. These boundary conditions would replace (5.2b).

For simplicity of notation we will set

$$(a,b) = (0,1), \quad D = 1, \quad g_0 \equiv const, \quad g_1 \equiv const. \tag{5.4}$$

To solve the equation (5.1) numerically, we discretize in space and time with a uniform grid. We use grid spacing $h = \frac{1}{1+M}$ and $k = \frac{T}{N}$, where M is the number of subintervals and T is the final stopping time. Grid points (x_j, t_n) are

$$x_j = jh, \quad j = 0, \ldots, M+1 \tag{5.5}$$

$$t_n = nk, \quad n = 0, \ldots, N. \tag{5.6}$$

Numerical approximation to the solution at discrete grid points is $u(x_j, t_n) \approx u_j^n$. To discretize (5.1), we must approximate the derivatives u_t and u_{xx}. We discretize u_t with a forward difference in time and u_{xx} with a centered difference in space, transforming (5.1) into

$$\frac{u_j^{n+1} - u_j^n}{k} = \frac{1}{h}(u_{j-1}^n - 2u_j^n + u_{j+1}^n); \quad j = 1, \ldots, M; \quad n = 1, \ldots, N. \tag{5.7}$$

In addition, boundary conditions (5.2b) produce

$$u_0 = g_0, \quad u_{M+1} = g_1, \tag{5.8}$$

and initial condition (5.2a) becomes

$$u_j^0 = f(x, j), \quad j = 0, \ldots, M+1. \tag{5.9}$$

Much like in the ODE case, Local Truncation Error is calculated by use of the Taylor series expansion about (x_j, t_n) [10]:

$$\begin{aligned}
\tau_j^n &= \tau(x_j, t_n) = \frac{1}{k}\left[u(x_j, t_{n+1}) - u(x_j, t_n)\right] \\
&- \frac{1}{h^2}\left[u(x_{j-1}, t_n) - 2u(x_j, t_n) + u(x_{j+1}, t_n)\right] \\
&= \frac{1}{k}\left[k u_t(x_j, t_n) + O(k^2)\right] \\
&- \frac{1}{h^2}[u(x_j, t_n) - h u_x(x_j, t_n) + \frac{h^2}{2} u_{xx}(x_j, t_n) - \frac{h^3}{6} u_{xxx}(x_j, t_n) + O(h^4) \\
&- 2u(x_j, t_n) \\
&+ u(x_j, t_n) + h u_x(x_j, t_n) + \frac{h^2}{2} u_{xx}(x_j, t_n) + \frac{h^3}{6} u_{xxx}(x_j, t_n) + O(h^4)] \\
&= u_t(x_j, t_n) - u_{xx}(x_j, t_n) + O(k + h^2) \\
&= O(k + h^2).
\end{aligned} \tag{5.10}$$

We see that if all the derivatives in (5.10) are bounded, then the LTE is $O(k + h^2)$, so the method in (5.7) is first order accurate in time and second order accurate in space. The method is thus *consistent* due to the fact that $\lim_{k,h \to 0} \tau(x, t) = 0$.

Consistency along with some form of stability should provide *convergence* of the method (so that the global accuracy agrees with the order of LTE). This is known as Lax Equivalence Theorem, which is given in [2] by Theorem 9.2.

The scheme (5.7) is forward in time and is known as Forward Euler, which is only conditionally stable. It requires a restriction on the time step

$$k \leq \frac{h^2}{2}. \tag{5.11}$$

Other schemes can be formulated. In particular, Backward Euler scheme would be unconditionally stable. That scheme is

$$\frac{u_j^{n+1} - u_j^n}{k} = \frac{1}{h}(u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1}), \quad j = 1, \ldots, M; \quad n = 1, \ldots, N. \tag{5.12}$$

and it requires solution of a linear system.

We now turn to an advection equation of the form

$$u_t = -vu_x, \quad x \in (a,b), \quad t > 0, \tag{5.13}$$

where the coefficient v is a positive constant. Using forward difference in time and upwind difference in space, equation (5.13) is discretized as

$$\frac{u_j^{n+1} - u_j^n}{k} = -\frac{v}{h}(u_j^n - u_{j-1}^n); \quad j = 1, \ldots, M; \quad n = 1, \ldots, N. \tag{5.14}$$

The boundary conditions for (5.14) described in (5.2b) are discretized according to (5.8). The initial condition for (5.14) stated in (5.2a) is discretized as in (5.9).

By the Taylor expansion, LTE of (5.14) is

$$\begin{aligned}
\tau_j^n &= \tau(x_j, t_n) = \frac{1}{k}\left[u(x_j, t_{n+1}) - u(x_j, t_n)\right] + \frac{v}{h}\left[u(x_j, t_n) - u(x_{j-1}, t_n)\right] \\
&= \frac{1}{k}\left[ku_t(x_j, t_n) + O(k^2)\right] \\
&+ \frac{v}{h}[hu_x(x_j, t_n) - \frac{h^2}{2}u_{xx}(x_j, t_n) + O(h^3)] \\
&= u_t + vu_x + O(k+h) \\
&= O(k+h).
\end{aligned} \tag{5.15}$$

If the derivatives in (5.15) are bounded, the local truncation error is $O(k+h)$. We see that the method in (5.14) is first order accurate in time and first order accurate in space. The method is consistent since $\lim_{k,h\to 0} \tau(x,t) = 0$.

The scheme (5.14) is known to be only conditionally stable. It requires a restriction on the time step,

$$c = \frac{vk}{h} \leq 1. \tag{5.16}$$

The constant c from equation (5.16) is called a Courant-Friedrichs-Lewy constant.

To obtain stability-inducing constraints on the time step size, one can apply either a Von-Neumann technique or a Method of Lines. The first applies to problems on infinite intervals or those with periodic boundary conditions. Von-Neumann analysis, when applicable, is considerably easier than Method of Lines, even though their findings are equivalent.

Since we will be using periodic boundary conditions of the type (5.3) in solving the linear version of the Klausmeier PDE system in section 5.2, Von-Neumann stability analysis is the best way to assess the constraints on the size of the time step in relation to the spatial step.

In solving the PDE system, the discretizations are only as accurate as the least-accurate methods. Stability of the system is accessed through intersection of the stability constraints for each equation.

5.2 Von Neumann Stability Analysis of the Linear PDE System

The original Klausmeier PDE system is given by (2.1). We now want to propose a discretization for the system and analyze its consistency and stability. Since we will be using standard discretization for the spatial and time derivatives, consistency follows directly from what we discussed in Section 5.1. However, Klausmeier system is nonlinear and coupled, and stability analysis cannot be done directly using either Von-Neumann or Method of Lines techniques developed for linear scalar problems.

Thus, in this Section we first discuss stability constraints for a simplified version of the Klausmeier system, which is uncoupled and does not have nonlinear terms or constant source terms.

In order to assess the size of the time step k, we can apply the von Neumann stability analysis to the linear system of PDE's and use that knowledge as the stability constraints for the nonlinear system. At the very least, the time step should abide the constraint on k that we will find in section (5.2), although in practice we will bind k by a smaller number still to ensure stability.

Discretizing in space and time, neglecting the nonlinearity and treating advection explicitly, but diffusion implicitly in time, we have:

$$\frac{u_j^{n+1} - u_j^n}{k} = -Bu_j^{n+1} + D\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{h^2}, \quad j = 0, \ldots, M+1, \tag{5.17a}$$

$$\frac{w_j^{n+1} - w_j^n}{k} = A - w_j^{n+1} + v\frac{w_{j+1}^n - w_j^n}{h}, \quad j = 0, \ldots, M+1. \tag{5.17b}$$

Note that our treatment of advection is downwind, not upwind, as in Section 5.1, because the direction of the velocity is to the left.

We note that the equations (5.17a), (5.17b) at $j = 0$ require the values of unknowns for $j = -1$. Due to the periodic boundary condition

$$u_0^n = u_{M+1}^n \tag{5.18}$$

at any n, we replace u_{-1} with u_M, etc. We proceed similarly for (5.17b).

Equation (5.17a) is an approximation to a reaction-diffusion equation, modeling diffusion with coefficient D and decay with rate B. Equation (5.17b) approximates a reaction-advection equation with decay rate 1, a constant source term A, and velocity $-v$.

To determine the stability constraints of the equation (5.17a), we re-shuffle the terms and multiply through by k:

$$\begin{aligned}
&\left(\frac{-Dk}{h^2}\right) u_{j-1}^{n+1} + \left(1 + Bk + \frac{2Dk}{h^2}\right) u_j^{n+1} + \left(\frac{-Dk}{h^2}\right) u_{j+1}^{n+1} = u_j^n, \;\; j = 0, \ldots, M+1, \\
&(-c_2)\, u_{j-1}^{n+1} + (c_1 + 2c_2)\, u_j^{n+1} + (-c_2)\, u_{j+1}^{n+1} = u_j^n + k w_j^n (u_j^n)^2, \;\; j = 0, \ldots, M+1,
\end{aligned} \tag{5.19}$$

where due to (3.5)

$$c_1 = 1 + kB > 1, \tag{5.20}$$

$$c_2 = \frac{Dk}{h^2} \geq 0, \tag{5.21}$$

$$c_1 + 2c_2 > 1. \tag{5.22}$$

Now we use the usual Von-Neumann Ansatz

$$u_j^n = e^{ijh\xi}, \tag{5.23}$$

$$u_j^{n+1} = g_u(\xi) e^{ijh\xi}, \tag{5.24}$$

where $g_u(\xi)$ is the amplification factor whose absolute value we hope to bind above by 1 in order to guarantee stability (we recall $i = \sqrt{-1}$).

Inserting expressions (5.23) and (5.24) into (5.19) gives

$$g_u(\xi)[(c_1 + 2c_2)e^{ijh\xi} - c_2 e^{i(j-1)h\xi} - c_2 e^{i(j+1)h\xi}] = e^{ijh\xi},$$

which simplifies to

$$g_u(\xi)[(c_1 + 2c_2) - c_2 e^{-ih\xi} - c_2 e^{ih\xi}] = 1,$$

where by use of the trigonometric identity $\frac{e^{-ih\xi} + e^{ih\xi}}{2} = \cos(h\xi)$ we obtain

$$g_u(\xi)[c_1 + 2c_2 - 2c_2\cos(h\xi)] = 1.$$

Therefore,

$$g_u(\xi) = \frac{1}{c_1 + 2c_2(1 - \cos(h\xi))}.$$

The last step is safely achieved due to the (5.20) and (5.21) implying that the denominator is positive (in fact, it is bounded below by 1) and, therefore, is nonzero.

Setting $|g_u| \leq 1$ will reveal which constraints (if there are any) should be placed on the time step size in relation to the spatial step.

Hence we have the double inequality

$$-1 \leq \frac{1}{c_1 + 2c_2(1 - \cos(h\xi))} \leq 1. \tag{5.25}$$

Both sides of (5.25) are always satisfied for any h and k, because

$$0 \quad \leq \quad 1 - \cos(h\xi) \leq 2 \tag{5.26}$$

implies

$$c_1 \quad \leq \quad c_1 + 2c_2(1 - \cos(h\xi)) \leq c_1 + 4c_2.$$

This means that by (5.20) the denominator of (5.25) is a positive number greater than one.

The calculations above provide the proof of the following result.

Lemma 5.2.1. *Equation* (5.17a) *is unconditionally stable.*

Now we analyze stability of (5.17b). We must set $A = 0$, since otherwise there is a source term to this equation which makes the solution w grow as it should; thus, there is no stability to this equation understood in the traditional Von-Neumann sense. But with $A = 0$ we expect the stability to hold.

Multiplication by k transforms the equation (5.17b) into

$$(1 + k)w_j^{n+1} = \left(1 - \frac{vk}{h}\right)w_j^n + \frac{vk}{h}w_{j+1}^n, \tag{5.27}$$

$$(1 + k)w_j^{n+1} = (1 - c)w_j^n + cw_{j+1}^n, \tag{5.28}$$

where the CFL constant c abides equation (5.16).

With the Ansatz analogous to (5.23), we proceed with the Von-Neumann stability analysis. We find

$$(1+k)g_w(\xi)e^{ijh\xi} = (1-c)e^{ijh\xi} + ce^{i(j+1)h\xi},$$

which simplifies to

$$(1+k)g_w(\xi) = 1 + c(e^{ih\xi} - 1).$$

The use of Euler's Formula gives

$$\begin{aligned}(1+k)g_w(\xi) &= 1 + c(\cos(h\xi) + i\sin(h\xi) - 1) \\ &= 1 - c(1-\cos(h\xi)) + ic\sin(h\xi). \end{aligned} \tag{5.29}$$

Applying the square of the norm to both sides of (5.29), we have

$$\begin{aligned}(1+k)^2\|g_w(\xi)\|^2 &= [1-c(1-\cos(h\xi))]^2 + [c\sin(h\xi)]^2 \\ &= 1 - 2c(1-\cos(h\xi)) + c^2[1 - 2\cos(h\xi) + \cos^2(h\xi)] + c^2\sin^2(h\xi) \\ &= 1 - 2c + 2c^2 + 2c(1-c)\cos(h\xi) \\ &= 1 + 2c(c-1)(1-\cos(h\xi)). \end{aligned} \tag{5.30}$$

If we bound the right hand side of equation (5.30) above by $(1+k)^2$, we obtain that $\|g_w(\xi)\| \leq 1$. Since $k > 0$, we bound the right hand side by 1 to require

$$1 + 2c(c-1)(1-\cos(h\xi)) \leq 1 < (1+k)^2,$$

which gives

$$(c-1)(1-\cos(h\xi)) \leq 0. \tag{5.31}$$

We have natural bounds on the second term of the product as indicated in (5.26). Thus, to preserve (5.31), we must have $c \leq 1$. By (5.16), we see that this is equivalent to

$$k \leq \frac{h}{v}. \tag{5.32}$$

The restriction (5.32) of the time step in relation to the space step is called the Courant-Friedrichs-Lewy stability criterion. It must be satisfied in order for (5.17b) to be stable.

In summary, we have the following result.

Lemma 5.2.2. *Equation* (5.17b) *is conditionally stable only with restriction in* (5.32).

Remark 5.2.1. *To satisfy the stability constraint for the entire linear uncoupled system* (5.17)*, we must pick the time step k in accordance with* (5.32)*. This discretization is $O(k+h^2)$ accurate in u and $O(k+h)$ accurate in w. However, once the system is coupled, the order of convergence for both variables together is expected to be at most $O(k+h)$.*

5.3 PDE System's Initial and Boundary Conditions for the Klausmeier System

For the Klausmeier system treated by Sherratt, periodic boundary conditions are assumed in [9]. We will use the following initial conditions for any of the equilibria in Chapter (3) of the form (u^*, w^*):

$$u_i = R_i u^*, \tag{5.33a}$$

$$v_i = S_i w^*, \tag{5.33b}$$

where R_i and S_i are chosen randomly from a uniform distribution between 0.9 and 1.1.

5.3.1 Periodic Boundary Conditions

In solving the equations (5.1) and (5.13) numerically, periodic boundary conditions on the first and last nodes are coded in the following way:

$$u^n_{-1} = u^n_M, \;\; u^n_{M+2} = u^n_1, \tag{5.34a}$$

$$w^n_{-1} = w^n_M, \;\; w^n_{M+2} = w^n_1 \tag{5.34b}$$

for any time step n.

5.4 PDE System with the Nonlinear Term Treated Explicitly

The original Klausmeier PDE system is nondimensionalized according to (2.1). We explained above how to approximate the derivatives appearing in this system. It remains to handle the nonlinear coupling terms wu^2. We will handle them in one of the ways described in Table 4.1. In this Section we show how to solve the system corresponding to the explicit

treatment of the nonlinearity. The following sections deal with other ways described in that Table.

Discretizing in space and time as explained in (5.17) and handling nonlinear term wu^2 explicitly gives

$$\frac{u_i^{n+1}-u_i^n}{k} = w_i^n(u_i^n)^2 - Bu_i^{n+1} + D\frac{u_{i+1}^{n+1}-2u_i^{n+1}+u_{i-1}^{n+1}}{h^2}, \tag{5.35a}$$

$$\frac{w_i^{n+1}-w_i^n}{k} = A - w_i^{n+1} - w_i^n(u_i^n)^2 + v\frac{w_{i+1}^n - w_i^n}{h}. \tag{5.35b}$$

We consider other variants than the fully explicit treatment in the sequel.

Now we describe how the algebraic system (5.35) will be solved.

Letting

$$M = \frac{1}{h^2}\begin{bmatrix} 2 & -1 & & & -1 & \\ -1 & 2 & -1 & & & \\ & & & \ddots & & \\ & & & -1 & 2 & -1 \\ & -1 & & & -1 & 2 \end{bmatrix}, \tag{5.36}$$

we see that (5.35) is

$$\underbrace{[I(1+Bk)+DkM]}_{S_1} u^{n+1} = u^n + (kw_i^n(u_i^n)^2)\big|_{i=0}^{M+1}, \tag{5.37a}$$

$$w_i^{n+1} = \frac{w_i^n + kA + kv\dfrac{w_{i+1}^n - w_i^n}{h} - kw_i^n(u_i^n)^2}{1+k}, \quad i = 0,\ldots,M+1. \tag{5.37b}$$

The right hand side of (5.37b) determines the value on the left, and (5.37a) is solved with a backslash operator in MATLAB using the tridiagonal matrix

$$S_1 = \begin{bmatrix} \frac{2Dk}{h^2}+1+Bk & -\frac{Dk}{h^2} & & & -\frac{Dk}{h^2} & \\ -\frac{Dk}{h^2} & \frac{2Dk}{h^2}+1+Bk & & & & \\ & & & \ddots & & \\ & & & -\frac{Dk}{h^2} & \frac{2Dk}{h^2}+1+Bk & -\frac{Dk}{h^2} \\ & -\frac{Dk}{h^2} & & & -\frac{Dk}{h^2} & \frac{2Dk}{h^2}+1+Bk \end{bmatrix}. \tag{5.38}$$

5.5 PDE System with the Sequential Nonlinear Term $w_i^{n+1}(u_i^n)^2$

Now instead of treating the nonlinear term fully explicitly, we want to make it more up-to-date with the time step t_{n+1}. Still we want to keep the system linear to be solved at each time step, so we time-lag the quadratic part of the term wu^2. This type of treatment is called Semi-implicit or Sequential.

We transform the system (5.17) into

$$\frac{u_i^{n+1}-u_i^n}{k} = w_i^{n+1}(u_i^n)^2 - Bu_i^{n+1} + D\frac{u_{i+1}^{n+1}-2u_i^{n+1}+u_{i-1}^{n+1}}{h^2}, \tag{5.39a}$$

$$\frac{w_i^{n+1}-w_i^n}{k} = A - w_i^{n+1} - w_i^{n+1}(u_i^n)^2 + v\frac{w_{i+1}^n - w_i^n}{h}. \tag{5.39b}$$

To get the system (5.39) to the matrix-vector form, we work with (5.39a). We get

$$\underbrace{[I(1+Bk)+DkM]}_{S_1}u^{n+1} + \underbrace{[-Ik(u_i^n)^2]\big|_{i=0}^{M+1}}_{S_2}w^{n+1} = u^n, \tag{5.40}$$

where S_1 is according to (5.38) and

$$S_2 = \begin{bmatrix} -k(u_0^n)^2 & & & \\ & -k(u_1^n)^2 & & \\ & & \ddots & \\ & & & -k(u_{M+1}^n)^2 \end{bmatrix}. \tag{5.41}$$

Working with (5.39b), we obtain

$$(1+k)w_i^{n+1} = w_i^n + kA + kv\frac{w_{i+1}^n - w_i^n}{h} - kw_i^{n+1}(u_i^n)^2,$$

where we have to diagonalize the coefficient on vector w^{n+1} to have consistent matrix dimensions. Namely, we have

$$\underbrace{[I(1+k+k(u_i^n)^2)]\big|_{i=0}^{M+1}}_{S_3}w^{n+1} = w^n + kA\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix} + kv\left(\frac{w_{i+1}^n - w_i^n}{h}\right)\Big|_{i=0}^{M+1}, \tag{5.42}$$

where $w^n_{M+2} = w^n_1$ and

$$S_3 = \begin{bmatrix} 1+k+k(u_0^n)^2 & & & \\ & 1+k+k(u_1^n)^2 & & \\ & & \ddots & \\ & & & 1+k+k(u_{M+1}^n)^2 \end{bmatrix}. \tag{5.43}$$

Combining (5.40) and (5.42) results in the matrix representation for the linear system

$$\begin{bmatrix} S_1 & S_2 \\ O & S_3 \end{bmatrix} \cdot \begin{bmatrix} u^{n+1} \\ w^{n+1} \end{bmatrix} = \begin{bmatrix} u^n \\ w^n + kA \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + kv \left(\frac{w^n_{i+1} - w^n_i}{h} \right) \Big|_{i=0}^{M+1} \end{bmatrix}. \tag{5.44}$$

5.6 PDE System with the Nonlinear Term Linearized about the Previous Time Step

We discretize equations (2.1a), (2.1b) in the following way:

$$\frac{u_i^{n+1} - u_i^n}{k} = w_i^{n+1}(u_i^{n+1})^2 - Bu_i^{n+1} + D\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{h^2}, \tag{5.45a}$$

$$\frac{w_i^{n+1} - w_i^n}{k} = A - w_i^{n+1} - w_i^{n+1}(u_i^{n+1})^2 + v\frac{w_{i+1}^n - w_i^n}{h}. \tag{5.45b}$$

Linearizing the nonlinear term from (5.45a) according to (4.12) evaluated at the space step i gives

$$\begin{aligned} [1 + Bk - 2ku_i^n w_i^n]\big|_{i=0}^{M+1} u^{n+1} + [-k(u_i^n)^2]\big|_{i=0}^{M+1} w^{n+1} = u^n - (2kw_i^n(u_i^n)^2)\big|_{i=0}^{M+1} \\ + \frac{Dk}{h^2}(u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1})\big|_{i=0}^{M+1}, \end{aligned} \tag{5.46}$$

where we apply periodic boundary conditions as discussed in (5.34).

Using matrices M from (5.36), S_1 from (5.38), and S_2 from (5.41) and diagonalizing vectors to match matrix dimensions converts (5.45a) into

$$\underbrace{[I(1+Bk)+DkM}_{S_1}\underbrace{-I(2ku_i^n w_i^n)\big|_{i=0}^{M+1}]}_{P_1}u^{n+1}+\underbrace{[-Ik(u_i^n)^2\big|_{i=0}^{M+1}]}_{S_2}w^{n+1}$$
$$= u^n + (2kw_i^n(u_i^n)^2)\big|_{i=0}^{M+1}, \tag{5.47}$$

where

$$P_1 = \begin{bmatrix} 2ku_0^n w_0^n & & & \\ & 2ku_1^n w_1^n & & \\ & & \ddots & \\ & & & 2ku_{M+1}^n w_{M+1}^n \end{bmatrix}. \tag{5.48}$$

Linearizing (5.45b) gives

$$\begin{aligned} [(2ku_i^n w_i^n)\big|_{i=0}^{M+1}]u^{n+1} &+ [(1+k+k(u_i^n)^2)\big|_{i=0}^{M+1}]w^{n+1} \\ &= kA + w^n + (2kw_i^n(u_i^n)^2)\big|_{i=0}^{M+1} + k\left(\frac{w_{i+1}^n - w_i^n}{h}\right)\Big|_{i=0}^{M+1}, \end{aligned} \tag{5.49}$$

which to accommodate matrix sizes transforms into

$$\underbrace{[I(2ku_i^n w_i^n)\big|_{i=0}^{M+1}]}_{P_1}u^{n+1}+\underbrace{[I(1+k+k(u_i^n)^2)\big|_{i=0}^{M+1}]}_{S_3}w^{n+1}$$
$$= kA\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix} + w^n + (2kw_i^n(u_i^n)^2)\big|_{i=0}^{M+1} + k\left(\frac{w_{i+1}^n - w_i^n}{h}\right)\Big|_{i=0}^{M+1}. \tag{5.50}$$

The matrix form of the linear system is

$$\begin{bmatrix} S_1 - P_1 & S_2 \\ P_1 & S_3 \end{bmatrix}\begin{bmatrix} u^{n+1} \\ w^{n+1} \end{bmatrix} = \begin{bmatrix} u^n - (2kw_i^n(u_i^n)^2)\big|_{i=0}^{M+1} \\ w^n + kA\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix} + k\left(\frac{w_{i+1}^n - w_i^n}{h}\right)\Big|_{i=0}^{M+1} + (2kw_i^n(u_i^n)^2)\big|_{i=0}^{M+1} \end{bmatrix}. \tag{5.51}$$

5.7 PDE System with Newton's Method Applied to the Fully Implicit Nonlinear Term

Referring to the set of equations (5.45), where the nonlinear term is discretized implicitly in time, we can transfer all the terms to the left side of each equation and multiply through by k. We get

$$\begin{aligned}\psi(u^{n+1},w^{n+1}) &= u^{n+1}-u^n-\left(kw_i^{n+1}(u_i^{n+1})^2\right)\Big|_{i=0}^{M+1}+Bku^{n+1}\\ &\quad-\frac{Dk}{h^2}\left(u_{i+1}^{n+1}-2u_i^{n+1}+u_{i-1}^{n+1}\right)\Big|_{i=0}^{M+1}=0, \qquad (5.52a)\\ \phi(u^{n+1},w^{n+1}) &= w^{n+1}-w^n-kA+kw^{n+1}+\left(kw_i^{n+1}(u_i^{n+1})^2\right)\Big|_{i=0}^{M+1}\\ &\quad-\frac{vk}{h}(w_{i+1}^n-w_i^n)\Big|_{i=0}^{M+1}=0. \qquad (5.52b)\end{aligned}$$

Here the vectors $\psi(u^{n+1},w^{n+1})$ and $\phi(u^{n+1},w^{n+1})$ are the residuals of the nonlinear system to be solved for u^{n+1},w^{n+1}.

The block-diagonal Jacobian matrix for the system above is

$$Jac(\psi,\phi)=\begin{bmatrix}\dfrac{\partial\psi}{\partial u} & \dfrac{\partial\psi}{\partial w}\\ \dfrac{\partial\phi}{\partial u} & \dfrac{\partial\phi}{\partial w}\end{bmatrix}=\begin{bmatrix}Jac_{\psi u} & Jac_{\psi w}\\ Jac_{\phi u} & Jac_{\phi w}\end{bmatrix}. \qquad (5.53)$$

To calculate the individual parts of the Jacobian, we use the following notation. We let $\{d^-,\ d,\ d^+\}$ denote the subdiagonal d^-, the diagonal d, and the superdiagonal d^+ components of a tridiagonal matrix.

For example, the i^{th} row of $Jac_{\psi u}$ is $[0,\cdots,0,\ \dfrac{\partial\psi}{\partial u_{i-1}},\ \dfrac{\partial\psi}{\partial u_i},\ \dfrac{\partial\psi}{\partial u_{i+1}},\ 0,\cdots,0]$. This means we have

$$\begin{aligned}Jac_{\psi u} &= \left\{\frac{\partial\psi}{\partial u_{i-1}},\ \frac{\partial\psi}{\partial u_i},\ \frac{\partial\psi}{\partial u_{i+1}}\right\}\\ &= \left\{-\frac{Dk}{h^2},\ 1-\left(2kw_i^{n+1}(u_i^{n+1})\right)\Big|_{i=0}^{M+1}+Bk+2\frac{Dk}{h^2},\ -\frac{Dk}{h^2}\right\}.\end{aligned}$$

Here for convenience we let

$$\omega(i)=1-\left(2kw_i^{n+1}(u_i^{n+1})\right)\Big|_{i=0}^{M+1}+Bk+2\frac{Dk}{h^2}.$$

Prior to the application of periodic boundary conditions $Jac_{\psi u}$ is tridiagonal. After (5.34) is

applied, the matrix acquires the extra terms in the first row and the last row as follows:

$$Jac_{\psi u} = \begin{bmatrix} \omega(0) & -\frac{Dk}{h^2} & & & -\frac{Dk}{h^2} \\ -\frac{Dk}{h^2} & \omega(1) & -\frac{Dk}{h^2} & & \\ & & \ddots & & \\ & & -\frac{Dk}{h^2} & \omega(M) & -\frac{Dk}{h^2} \\ & -\frac{Dk}{h^2} & & -\frac{Dk}{h^2} & \omega(M+1) \end{bmatrix}. \tag{5.54}$$

Now working with

$$\begin{aligned} Jac_{\psi w} &= \left\{ \frac{\partial \psi}{\partial w_{i-1}}, \ \frac{\partial \psi}{\partial w_i}, \ \frac{\partial \psi}{\partial w_{i+1}} \right\} \\ &= \left\{ 0, \ -(k(u_i^{n+1})^2)\big|_{i=0}^{M+1}, \ 0 \right\}, \end{aligned}$$

which produces the matrix

$$Jac_{\psi w} = \begin{bmatrix} -k(u_0^{n+1})^2 & & & \\ & -k(u_1^{n+1})^2 & & \\ & & \ddots & \\ & & & -k(u_{M+1}^{n+1})^2 \end{bmatrix}. \tag{5.55}$$

Next, we look at

$$\begin{aligned} Jac_{\phi u} &= \left\{ \frac{\partial \phi}{\partial u_{i-1}}, \ \frac{\partial \phi}{\partial u_i}, \ \frac{\partial \phi}{\partial u_{i+1}} \right\} \\ &= \left\{ 0, \ (2kw_i^{n+1}u_i^{n+1})\big|_{i=0}^{M+1}, \ 0 \right\}, \end{aligned}$$

which gives us the matrix

$$Jac_{\phi u} = \begin{bmatrix} 2kw_0^{n+1}u_0^{n+1} & & & \\ & 2kw_1^{n+1}u_1^{n+1} & & \\ & & \ddots & \\ & & & 2kw_{M+1}^{n+1}u_{M+1}^{n+1} \end{bmatrix}. \tag{5.56}$$

Finally,

$$\begin{aligned} Jac_{\phi w} &= \left\{ \frac{\partial \phi}{\partial w_{i-1}}, \ \frac{\partial \phi}{\partial w_i}, \ \frac{\partial \phi}{\partial w_{i+1}} \right\} \\ &= \left\{ 0, \ (1 + k + k(u_i^{n+1})^2)\big|_{i=0}^{M+1}, \ 0 \right\}, \end{aligned}$$

which gives

$$Jac_{\phi w} = \begin{bmatrix} 1+k+k(u_0^{n+1})^2 & & & \\ & 1+k+k(u_1^{n+1})^2 & & \\ & & \ddots & \\ & & & 1+k+k(u_{M+1}^{n+1})^2 \end{bmatrix}. \quad (5.57)$$

The Jacobian block-matrix $Jac(\psi, \phi)$ is sparse.

The code for the Newton's method is provided in Appendix B. Note that the method is consistent with the description provided in Section 4.8 and equation (4.27).

6 Numerical Results of Newton's Method for the PDE System

In this Chapter we present numerical experiment results for the Newton's method described in Section 5.7. We expect other methods from Chapter 5 to give similar results, and although we have implemented them, full reports and comparisons are outside of the present scope.

We vary the parameters A and D as well as the spatial step size in order to produce a large range of graphical evidence for asymptotic behavior of the discretized one-dimensional PDE system (5.45). Taking into consideration Sherratt's parameter values of A, B, and v in paper [8] for pattern existence as seen in (2.2) and (2.3), we look for similar periodic numerical results. Table 6.1 records all the specific parameter values chosen for each of the experiments presented in the Figures to follow.

The initial condition for w is chosen to be 0.5. We use the initial condition for solution u to be a small random perturbation around 5 of the magnitude being at most 0.1. To achieve this, we divide the spatial interval into s equal subintervals, which produces $s-1$ inner nodes. The value at those nodes is a random uniformly distributed number from the interval (-1,1). We use linear interpolation scaled by 0.1 to obtain the values of the initial function u ranging (at the nodes) from -0.1 to 0.1. Adding 5 to the output of the function simply raises the graph. The number of fluctuations in the initial condition is referred to as frequency. The coding for the initial conditions with frequency of $s = 10$ can be found in Appendix B.

To illustrate how the choice of frequency affects the behavior of solutions, we show the solution of the system at the first time step in Figure 6.1. A frequency of 40 is chosen to illustrate the sharpness of the oscillatory behavior. For visual comparison we show Figure 6.2 with a frequency of 10. Notice that the initial condition function for solution u does not exceed 5.1 and does not fall below 4.9 in both Figures, which is consistent with previous explanations.

The results for each set of parameter values are recorded in Table 6.1. For each Figure we choose to use $B = 0.45$ according to (2.2) and for the spatial interval (a, b), total time T, the size of the time step Δt, and the size of the spatial step Δx we use

$$a = 0, \;\; b = 100, \;\; T = 100, \;\; \Delta t = 10^{-3}, \;\; \Delta x = 1. \tag{6.1}$$

For the tolerance level set at 10^{-8}, Newton's Method converges after 1-2 iterations at each

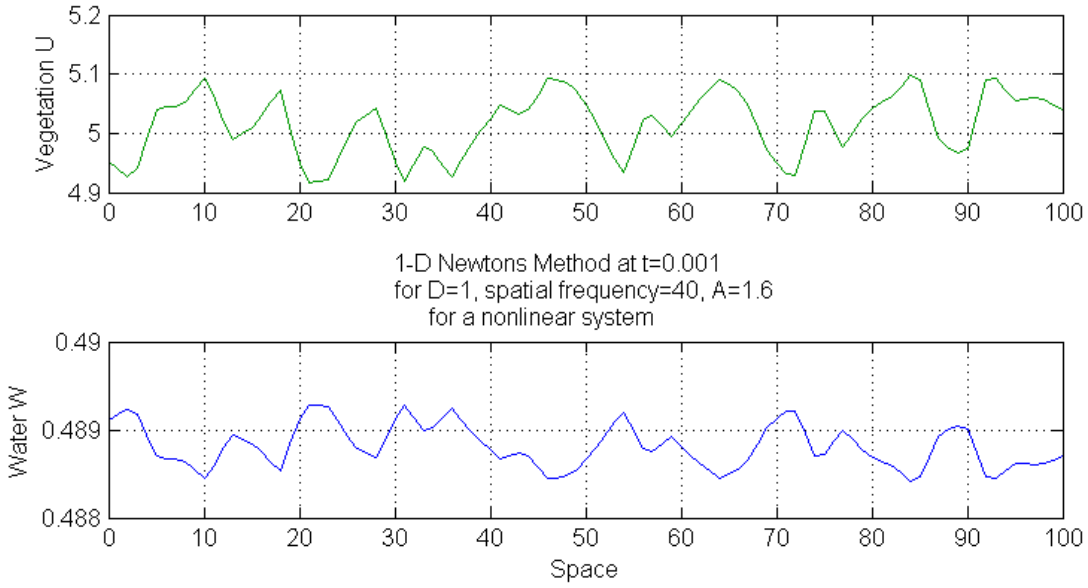

FIGURE 6.1: Newton's Method solutions for the parameter values displayed in the center of the Figure.

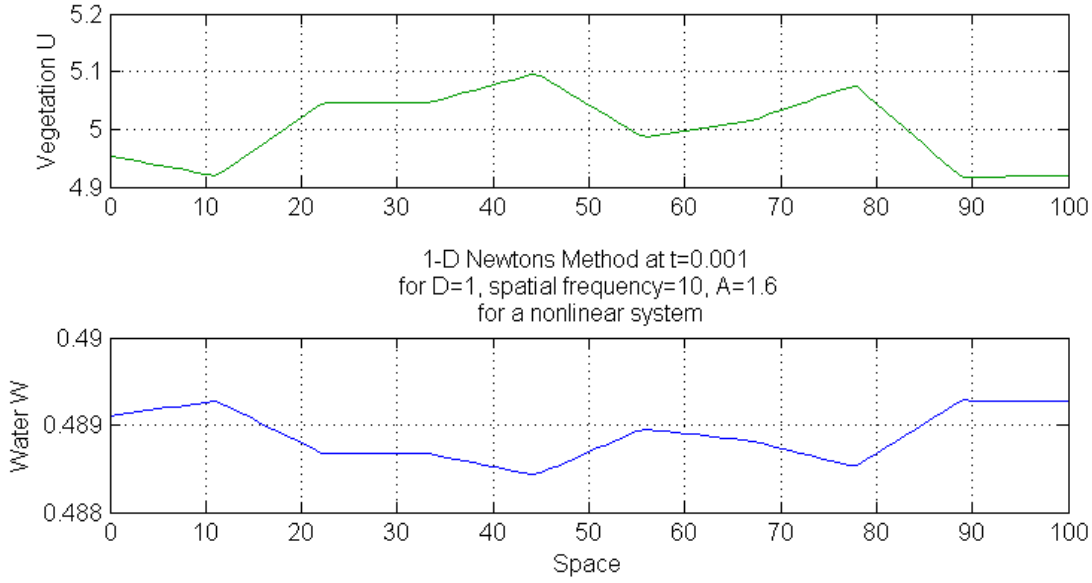

FIGURE 6.2: Newton's Method solutions for the parameter values displayed in the center of the Figure.

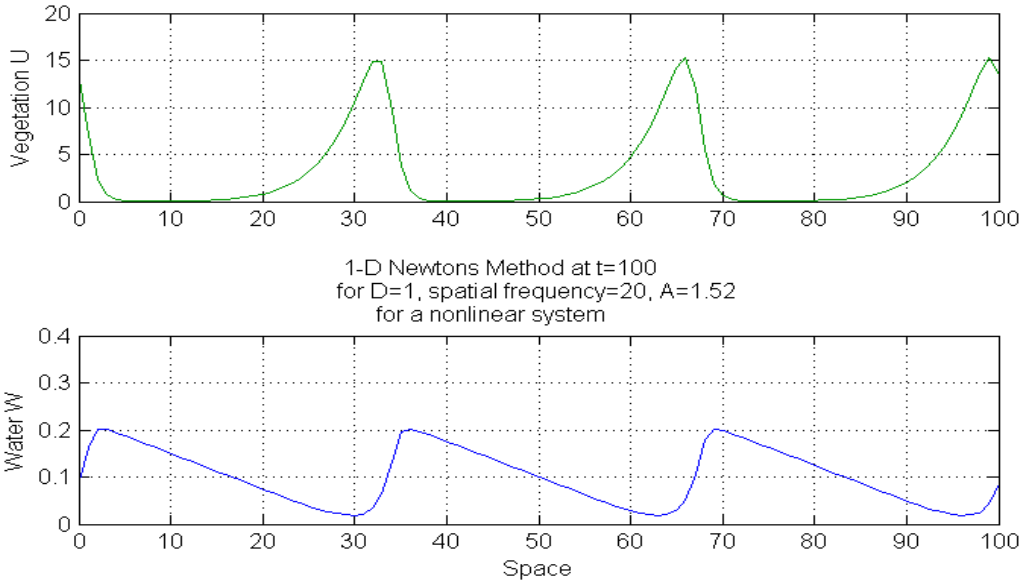

FIGURE 6.3: Newton's Method solutions for the parameter values displayed in the center of the Figure.

time step. We choose Δt to be very small to make sure that the experiments run smoothly.

Sherratt uses $T = 2000$ in his simulations to ensure the decay of the transients. For practical reasons we choose a smaller time that allows us to visually ascertain the asymptotic trends from [8]. For $T < 100$ the results are not as strongly pronounced. The average running time of the code for the Newton's Method is approximately 5.5 hours. The shortest time of 2.5 hours was registered for the linear cases in Figures 6.11 - 6.13. The longest time of 8.5 hours was taken by the code with numerical results displayed in Figure 6.8.

We vary the rainfall parameter A within its limits for vegetation stripe production, recording the details for parameter choices in the Table 6.1. Figures 6.3, 6.4, and 6.5 show the impact of the mean annual rainfall on the stripe formations. The expected increase in the number of peaks as the rainfall parameter A grows is more visible in transition from $A = 1.6$ to $A = 1.68$. We notice that as A increases, the amplitude of the waves increases as well, contrary to what is anticipated. It seems that solutions are still evolving, which is confirmed by a different amplitude of each wave as well its length. We might need to run the simulations longer in order to obtain uniformity of the wave-like patterns.

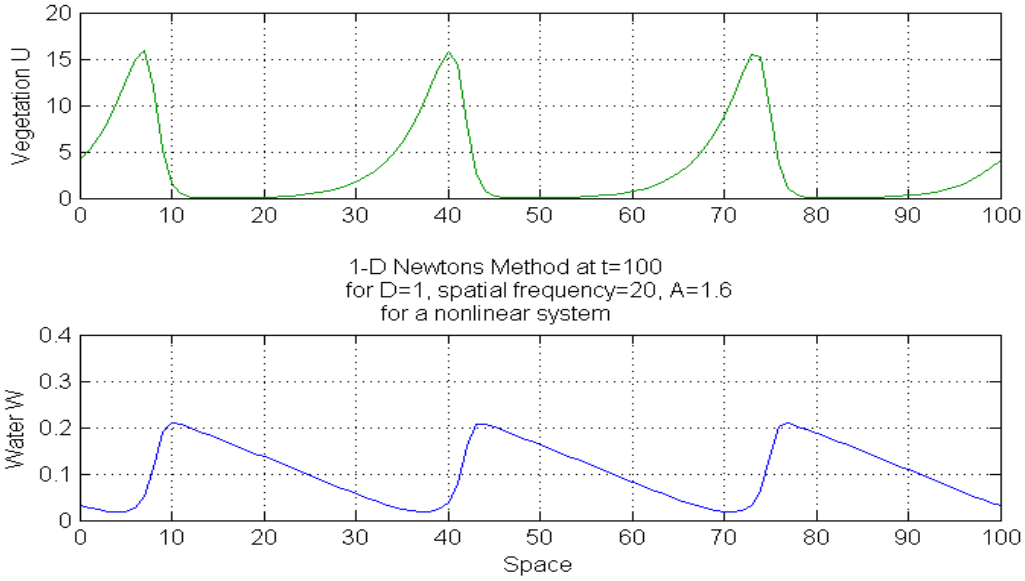

FIGURE 6.4: Newton's Method solutions for the parameter values displayed in the center of the Figure.

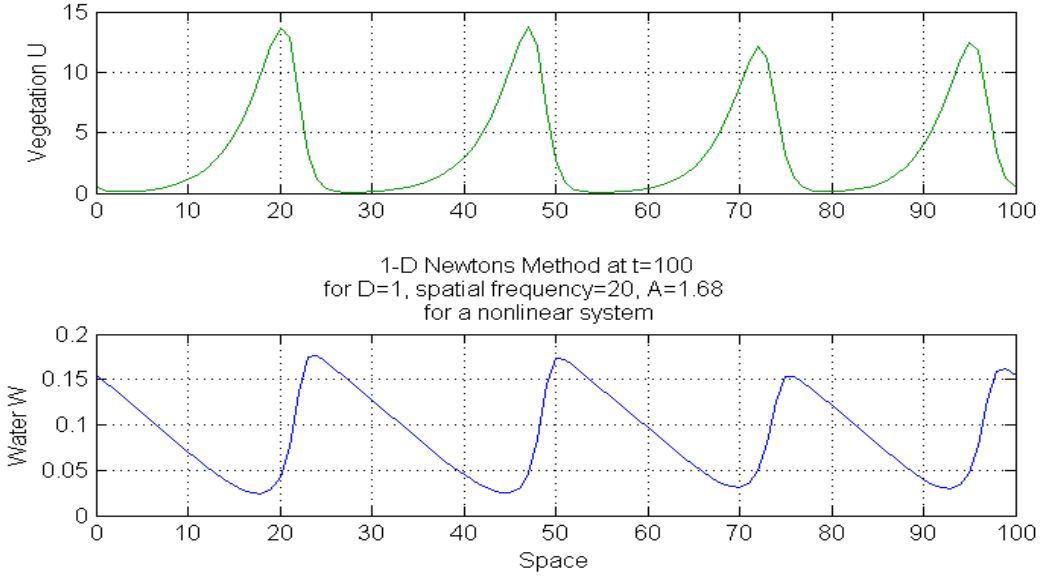

FIGURE 6.5: Newton's Method solutions for the parameter values displayed in the center of the Figure.

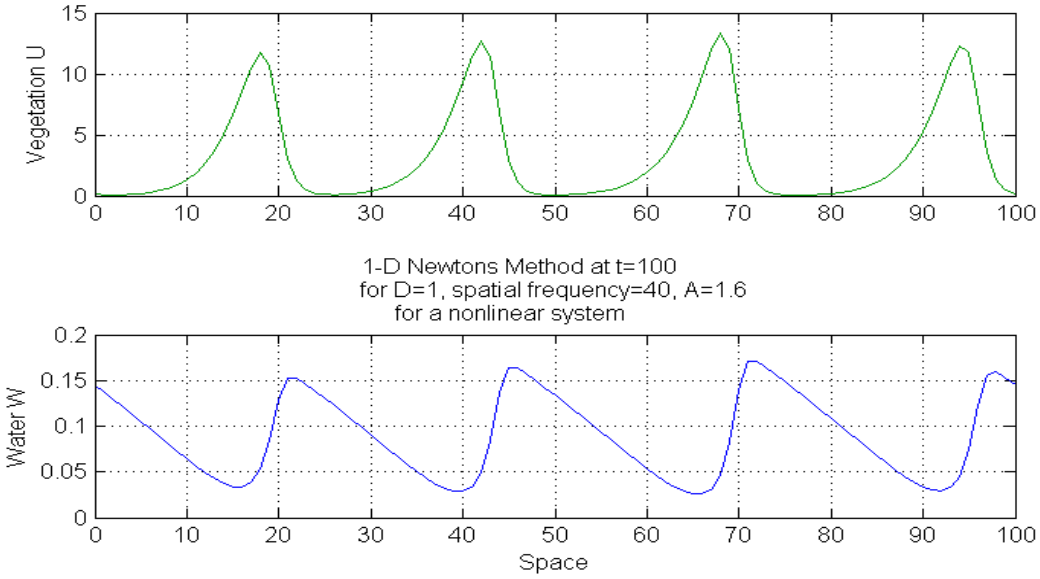

FIGURE 6.6: Newton's Method solutions for the parameter values displayed in the center of the Figure.

Then we consider a smaller diffusion coefficient D to discern its bearing on the behavior of the Klausmeier system. Figures 6.8, 6.7, 6.6 show that as D decreases, the maximum value of solution u increases. The peaks appear to be much sharper and the waves closer together (as is especially strongly marked by the transition from $D = 0.5$ to $D = 0.1$). We can still detect the asymptotic trends, although it is harder to see for the smallest diffusion value.

We illustrate the effect of different spatial frequencies on the minimum wavelength in solution u through Figures 6.6, 6.4, and 6.9. We see that as frequency increases, wavelength decreases. The expectation of the impact of frequency on the number of peaks in the solution is refuted by the collected evidence. It seems that A and D have more bearing on the number of peaks in solution u. The system appears to be very delicate and intricately dependent on its parameter values.

We also consider a case for a higher velocity v (compared to what is indicated in 2.3). Sherratt [4]-[8] studies the behavior of the system as $v \to \infty$ and produces numerical results for $v = 200$. We use this value for v in Figure 6.10. It appears that the slight increase in the slope gradient does not affect the system in a significant way.

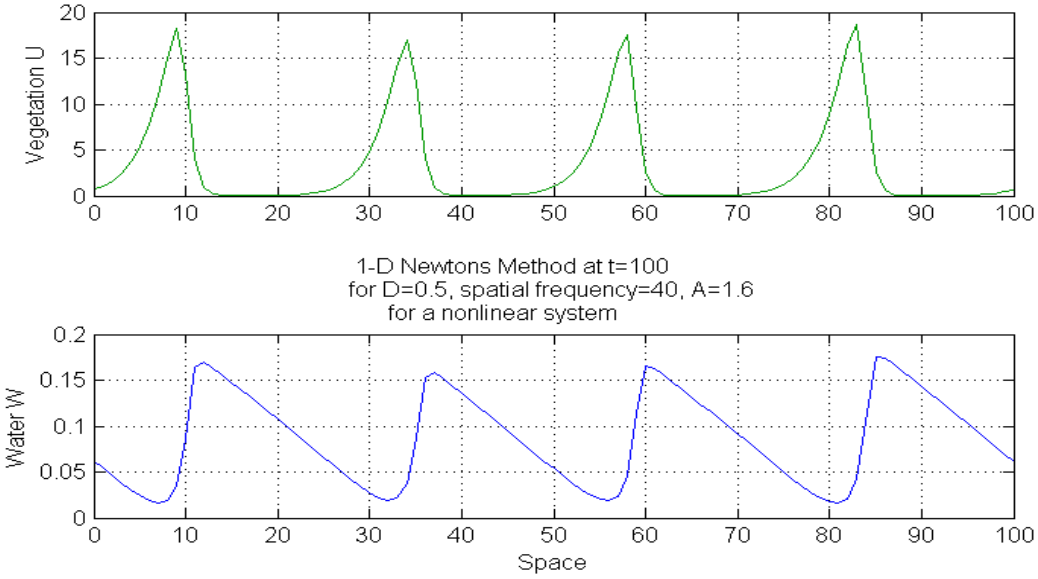

FIGURE 6.7: Newton's Method solutions for the parameter values displayed in the center of the Figure.

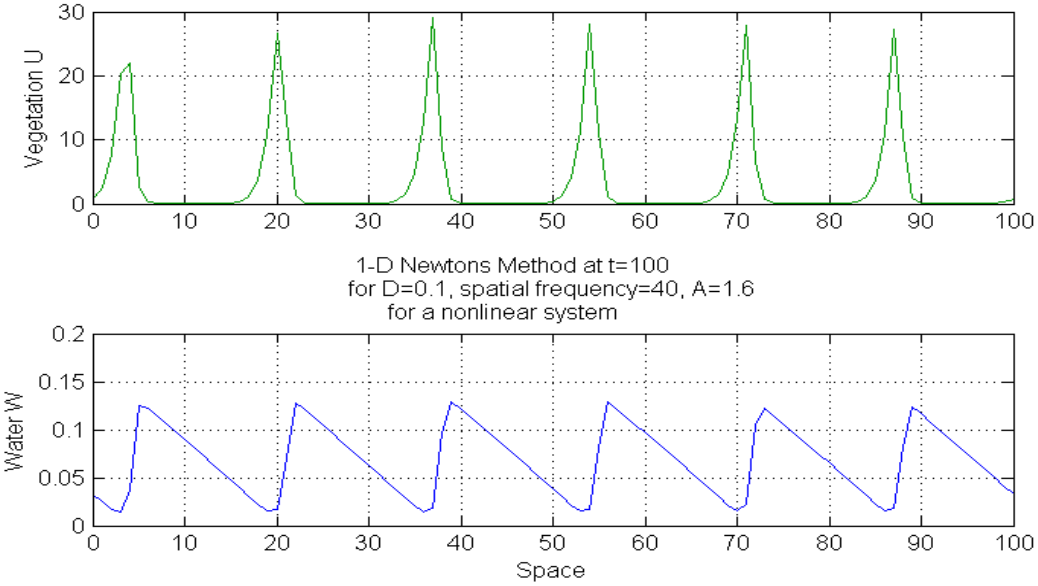

FIGURE 6.8: Newton's Method solutions for the parameter values displayed in the center of the Figure.

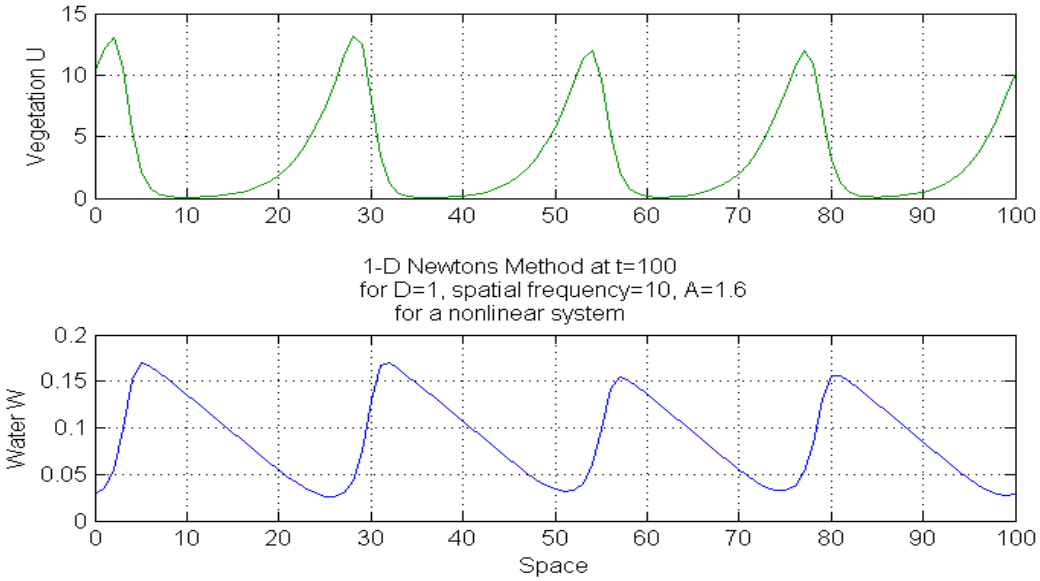

FIGURE 6.9: Newton's Method solutions for the parameter values displayed in the center of the Figure.

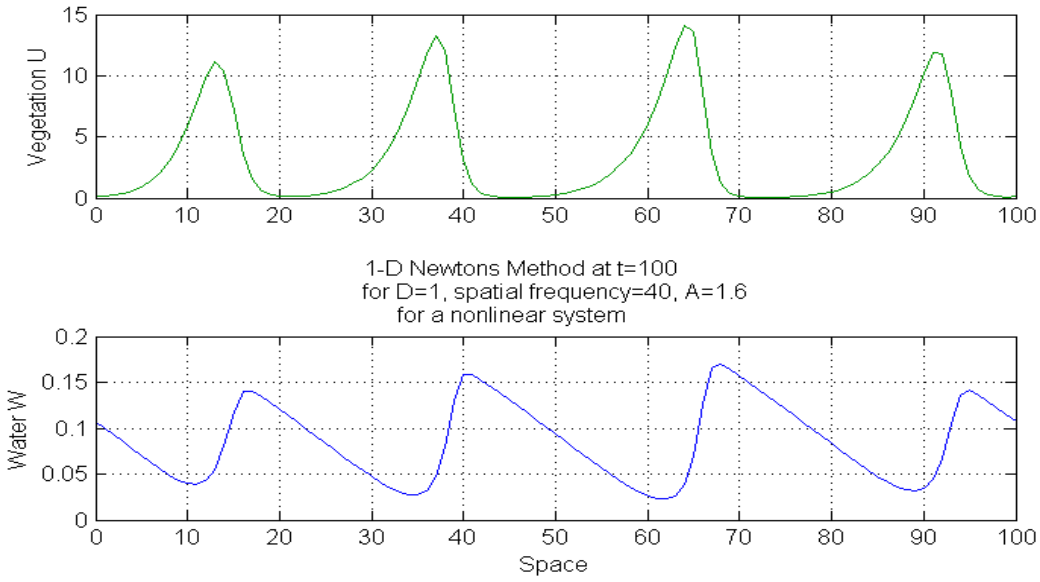

FIGURE 6.10: Newton's Method solutions for the parameter values displayed in the center of the Figure. Here we use velocity $v = 200$.

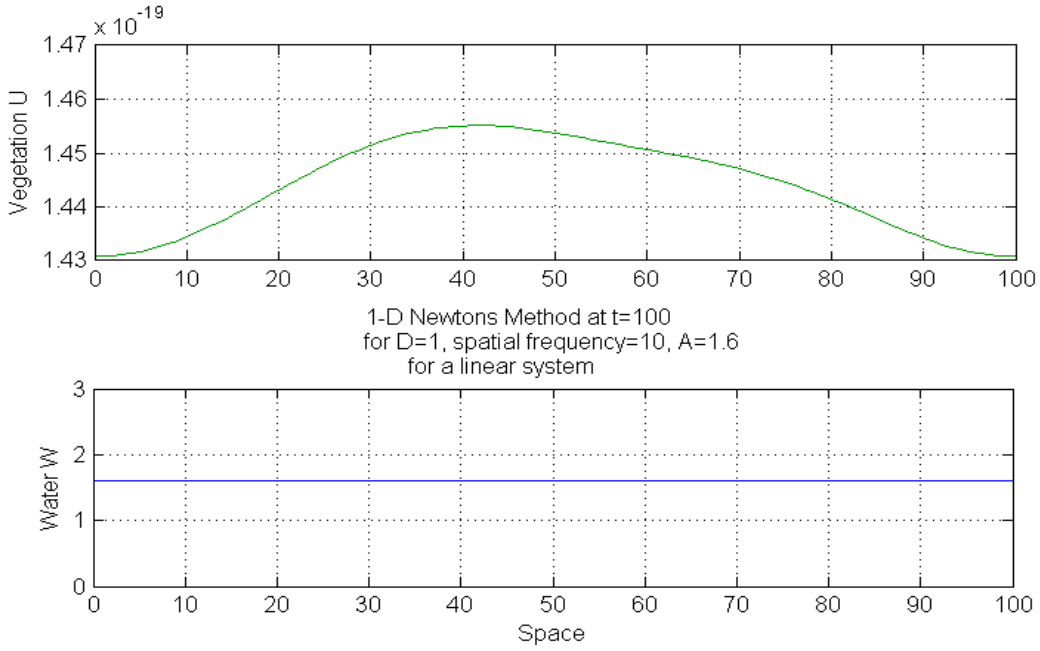

FIGURE 6.11: Newton's Method solutions for the parameter values displayed in the center of the Figure.

Numerical results in Figures 6.3 - 6.10 show extremely strong confirmation of the periodic wave character of the problem.

Finally, we compare the results of the nonlinear cases to the linear ones when the wu^2 term is eliminated from the system (5.45). This lets us determine the strength of its influence on the solutions of the system. Figures 6.11, 6.12, 6.13 represent the linear cases of the PDE system (5.45) for different spatial frequencies.

It appears that for the linear case of (1.1) (where coupling is removed), solution $u \approx 0$ due to the combination of decay and diffusion terms and solution $w \approx 1.6$. The constant result for w can be confirmed by solving $w' = A - w$, where the initial condition is consistent with its description earlier. We see that $w = 1.6 - 1.1e^{-t}$. For large t, the limit for w is 1.6 as is visible in Figures 6.11, 6.12, 6.13.

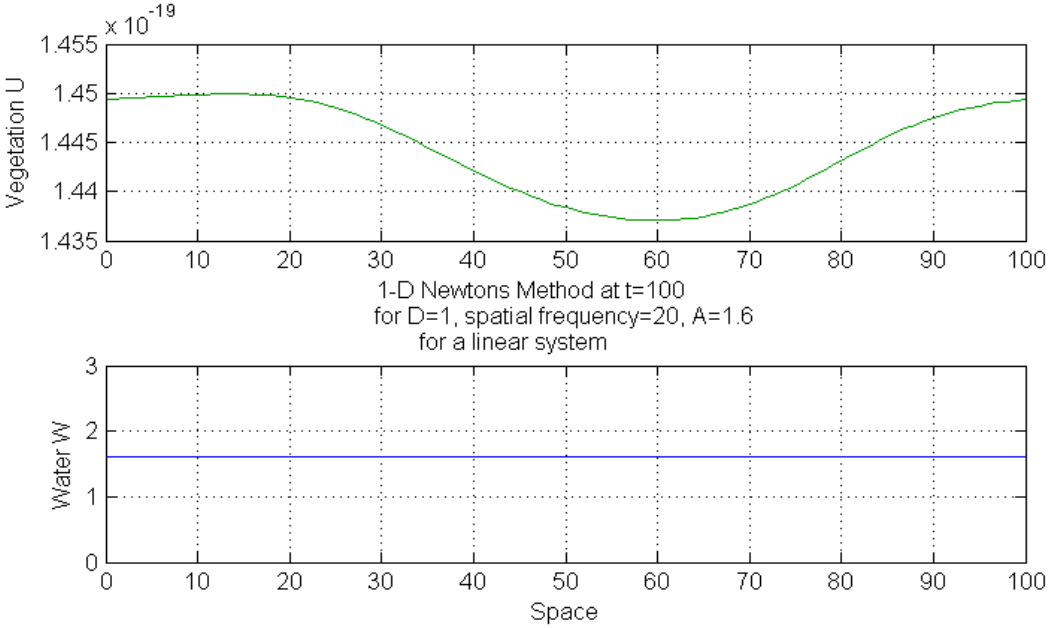

FIGURE 6.12: Newton's Method solutions for the parameter values displayed in the center of the Figure.

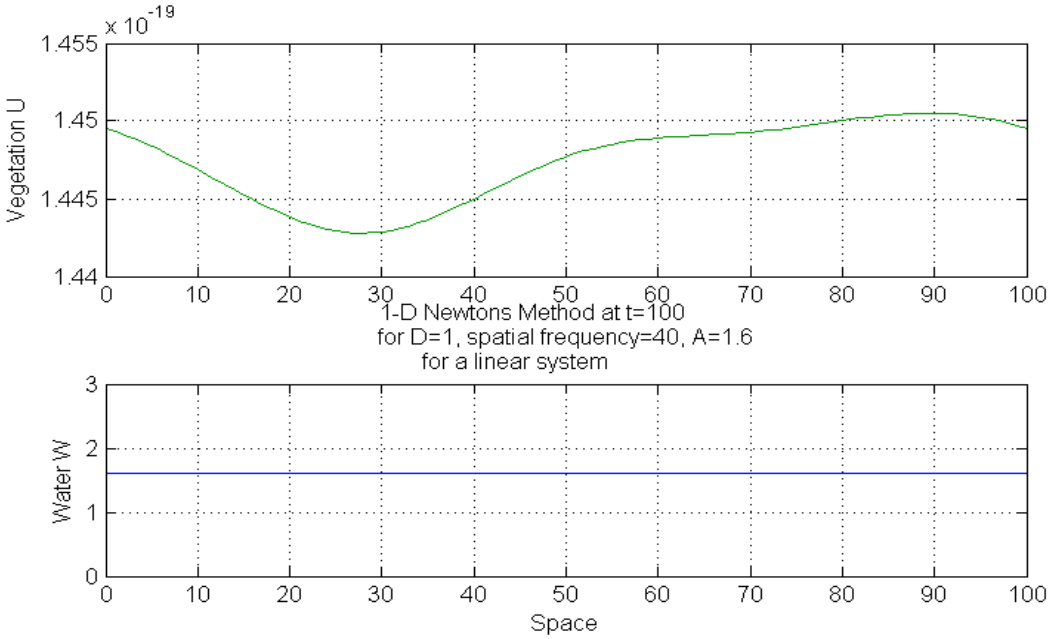

FIGURE 6.13: Newton's Method solutions for the parameter values displayed in the center of the Figure.

TABLE 6.1: Data Sets for the Numerical Results for Solution U via Newton's Method for the PDE System.

Fig.	Presence of wu^2 Term	A	D	Spatial Frequency	v	Number of Peaks	Max Value	Min Wavelength
6.3	✓	1.52	1	20	182.5	3	15	27.1
6.4	✓	1.60	1	20	182.5	3	15.9	27.1
6.5	✓	1.68	1	20	182.5	4	16.3	18.9
6.6	✓	1.60	1	40	182.5	4	13.8	19.6
6.7	✓	1.60	0.5	40	182.5	4	18.5	19.6
6.8	✓	1.60	0.1	40	182.5	6	29.5	16.1
6.9	✓	1.60	1	10	182.5	4	13.6	23.4
6.10	✓	1.60	1	40	200	4	13.9	19.6
6.11	X	1.60	1	10	182.5	–	–	–
6.12	X	1.60	1	20	182.5	–	–	–
6.13	X	1.60	1	40	182.5	–	–	–

7 Summary and Outlook

In this work we developed methods of numerical approximation of solutions to the Klausmeier model which can be easily extended to the case where Sherratt's assumptions are lifted. To do so, we had to handle the nonlinear coupled nature of the system as well as the presence of the competing mechanisms of vegetation and water transport.

Numerical simulations can give good approximations only when the underlying method is stable and accurate. To the extent possible we provided stability analysis of the numerical schemes we employed. The biggest challenge was handling the nonlinear term. We proposed several ways of including that term in the system which range from an explicit treatment through semi-implicit to fully implicit. Since the PDE system was expected to develop patterns and strong gradients, the proper accounting for the coupling was crucial.

We studied the effect of various parameters on the behavior of the system in analyzing the number of peaks, wavelength, and amplitude of the waves. The system showed to be extremely sensitive to changes in its parameters.

In studying the Klausmeier system, we would have liked to continue in further exploring the potential of the system.

First, we would study the basins of attraction for the Phase Plane analysis of the ODE counterpart of the Klausmeier system (3.4) and the effect that the choice of the initial conditions has on solutions' equilibria attraction.

We would have liked to use the Wavetrain and Auto packages that Sherratt applied to produce the numerical results in papers [4] - [8].

Then we would consider the 2-dimensional nonlinear PDE. This would involve running numerical simulations for the fully implicit case to detect expected wave-like patterns and, if possible, obtaining the asymptotic behavior.

Next, we would express the rainfall parameter A (responsible for the mean annual precipitation) to be a function of space and time. This would enhance the system and make it a more interesting study.

We would also consider the slope gradient parameter v to be a function of space, which would add complexity to the system. In this case we would be able to study how vegetation

stripes form on multiple hills of varying slopes (which is a more likely scenario for the natural world). We expect for the sign of v to shift from a positive to a negative, because the direction in which the plants climb changes as we move up and down the hill (forward in space).

Currently the model does not encompass the root systems ability to absorb water and plant-soluble nutrients. We would like to add a constant term into the system that would be responsible for carrying the information on qualitative assessment of the soil type such as its density, porosity, ability to retain nutrients, etc.

The Klausmeier system has an abundant potential and proves to be a fruitful study.

Bibliography

1. Christopher A. Klausmeier. Regular and Irregular Patterns in Semiarid Vegetation. *Science*, 284(5421):1826–1828, Jun 11 1999.

2. R. J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. SIAM, 2007.

3. G. R. Hall P. Blanchard, R. L. Devaney. *Differential Equations*. Thomson Brooks/Cole, 2006.

4. Jonathan A. Sherratt. Pattern Solutions of the Klausmeier model for banded vegetation in semiarid environments I. *Nonlinearity*, 23:2657–2675, 2010.

5. Jonathan A. Sherratt. Pattern Solutions of the Klausmeier model for banded vegetation in semiarid environments II: Patterns with the largest possible propagation speeds. *Proceedings of The Royal Society A Mathematical Physical and Engineering Sciences*, 476:3272–3294, 2011.

6. Jonathan A. Sherratt. Pattern Solutions of the Klausmeier model for banded vegetation in semiarid environments III: The transition between homoclinic solutions. *Physica D: Nonlinear Phenomena*, 242(1):30–41, 2013.

7. Jonathan A. Sherratt. Pattern Solutions of the Klausmeier model for banded vegetation in semiarid environments IV: Slowly moving patterns and their stability. *SIAM Journal on Applied Mathematics*, 73(1):330–350, 2013.

8. Jonathan A. Sherratt. Pattern Solutions of the Klausmeier model for banded vegetation in semiarid environments V: The transition from patterns to desert. *SIAM Journal on Applied Mathematics*, 73(4):1347–1367, 2013.

9. Jonathan A. Sherratt and Gabriel J. Lord. Nonlinear dynamics, pattern bifurcations in a model for vegetation stripes in semi-arid environments. *Theoretical Population Biology*, 71(1):1–11, 2007.

10. Guo Chun Wen. *Approximate Methods and Numerical Analysis for Elliptic Complex Equations*. Gordon and Breach Science Publishers, 1999.

APPENDICES

A Code for the Newton's Method for the ODE system

```
% _*Newton's Method*_
% completely implicit on the nonlinear term $wn^2$
maxnewtoniter=20;
tol=1e-8;
udel=zeros(2,1);
res=zeros(2,1);
% choosing a set of parameters
A=0.2; B=0.5; D=0.1;
% A=0.6; B=0.3; D=0;
% A=0.7; B=0.3; D=0;
k=.05; T=7;
tsteps=0:k:T;
% 2 x 2 Jacobian matrix
Jac = zeros(2,2);

u=0*tsteps;
w=0*tsteps;
% initial conditions
u0=[0.01,1.5,0.5];
w0=[0.2,2,3];
u_0=zeros(length(u0));
w_0=zeros(length(u0));
% pertubation of the first initial condition
for s=1:length(u0)
u_0(s) = u0(s).*0.1.*(2*rand+9);
w_0(s) = w0(s).*0.1.*(2*rand+9);
u_0(2)=1.5; w_0(2)=2;
u_0(3)=0.5; w_0(3)=3;
```

```
u(1)=u_0(s);
w(1)=w_0(s);
for n=1:(length(tsteps)-1)
    %un is old, wn is old; we want unew, wnew
     u(n+1)=u(n);
     w(n+1)=w(n);

    %start Newton loop
      iter=0;
      while iter<maxnewtoniter

        %compute residual
        res1 = u(n+1)- u(n)-k*w(n+1).*u(n+1).^2+k*(B+D)*u(n+1);
        res2 = w(n+1)-w(n)-k*(A-w(n+1)-w(n+1).*u(n+1).^2);

        res=[res1;res2];

        %check if converges
        if norm(res)<tol, break;
        end

        Jac=[1+k*(B+D)-2*k*w(n+1)*u(n+1), -k*u(n+1)^2;
            2*k*w(n+1)*u(n+1),       1+k+k*u(n+1)^2];

        %check for singular Jacobian
        if abs(det(Jac))<tol*1e-4, error('singular Jacobian');
        end

        %solve the linear system
```

```
        udel=Jac\res;

        %correct the guess
        u(n+1)=u(n+1)-udel(1);
        w(n+1)=w(n+1)-udel(2);

        iter=iter+1;
        %unew is the new time step, wnew is the new time step
      end

      %print number of iterations at each time step
      fprintf('At n=%d number of Newton iters=%d\n',n,iter);
end

%plot
plot(tsteps,u,'--',tsteps,w,'--','LineWidth',1.75)
hold on
grid on
title(sprintf('Newtons Method for A=%g, B=%g, D=%g',A,B,D))
xlabel('Time')
ylabel('Solutions to the system')
end
legend(sprintf('u_0=%0.5f \n',u_0(1)),sprintf('w_0=%0.5f \n',w_0(1)),...
    sprintf('u_0=%g \n',u_0(2)),sprintf('w_0=%g \n',w_0(2)),...
    sprintf('u_0=%g \n',u_0(3)),sprintf('w_0=%g \n',w_0(3)))
```

B Code for the Newton's Method for the PDE system

```
% _*Newton's Method*_
% completely implicit on the nonlinear term $$wn^2$
function [courant] = Newton_1_d (M,aa,bb,T,dt)
%setting up a sample case
if nargin < 1,
    M = 100;
    aa = 0; bb = 100;
    T = 100;
    dt = 1e-3;
end
clf;
rand('seed',0);

maxnewtoniter=15;
tol=1e-8;

dx = (bb-aa)/M;           %% use uniform grid
x = (aa:dx:bb)';
n = floor(T/dt);
tsteps = dt:dt:T;

%set up parameter values
D=1; B=0.45; vi=182.5; a=1.60;
%mycof is responsible for the nonlinear term's presence
mycof = 1;

% 2 x 2 matrix
resu=zeros(length(x),1);
```

```
resw=zeros(length(x),1);

uprev=initfunu(x);
wprev=initfunw(x);

for nt = 1:n
    t = nt*dt;
    %uprev is old, wprev is old; we want unew, wnew by Newton iteration
     uguess=uprev;
     wguess=wprev;

    %start Newton loop
      iter=0;
      while iter<maxnewtoniter

        %compute residuals
        for j=2:(length(x)-1)
resu(j) = uguess(j)- uprev(j)-dt*mycof*wguess(j).*uguess(j).^2+...
    dt.*(B).*uguess(j)+D*dt./(dx.^2).*(2.*uguess(j)-uguess(j-1)-uguess(j+1));
resw(j) = wguess(j)- wprev(j)-dt*(a-wguess(j)-...
    mycof*wguess(j).*uguess(j).^2)...
    -vi.*dt./dx.*(wprev(j+1)-wprev(j));
        end

         j=1;
resu(j) = uguess(j)- uprev(j)-dt*mycof*wguess(j).*uguess(j).^2+...
   dt.*(B).*uguess(j)+D*dt./(dx.^2).*(2.*uguess(j)-uguess(length(x)-1)-...
   uguess(j+1));
resw(j) = wguess(j)- wprev(j)-...
   dt*(a-wguess(j)-mycof*wguess(j).*uguess(j).^2)...
```

```
   -vi.*dt./dx.*(wprev(j+1)-wprev(j));

         j=length(x);
resu(j) = uguess(j)- uprev(j)-dt*mycof*wguess(j).*uguess(j).^2+...
    dt.*(B).*uguess(j)+...
    D*dt./(dx.^2).*(2.*uguess(j)-uguess(j-1)-uguess(2));
resw(j) = wguess(j)- wprev(j)-...
    dt*(a-wguess(j)-mycof*wguess(j).*uguess(j).^2)...
    -vi.*dt./dx.*(wprev(2)-wprev(j));

  %combine the residues into a column vector
        res=[resu;resw];

        %setting up the Jacobian matrices
        Jacuu = sparse(length(x),length(x));
        for j=1:length(x)
            Jacuu(j,j)=1+B*dt+2.*D.*dt./(dx.^2)-...
                2.*dt.*mycof*wguess(j).*(uguess(j));
        end
        for j=2:length(x)
            Jacuu(j,j-1)=-D.*dt./(dx.^2);
        end
        for j=1:(length(x)-1)
            Jacuu(j,j+1)=-D.*dt./(dx.^2);
        end

        % fix the matrices for periodic boundary conditions
        j=1;
        Jacuu(j,length(x)-1)=-D.*dt./(dx.^2);
        j=length(x);
```

```
Jacuu(j,2)=-D.*dt./(dx.^2);

Jacuw = sparse(length(x),length(x));
for j=1:length(x)
    Jacuw(j,j)=-dt.*mycof*(uguess(j)).^2;
end

Jacwu = sparse(length(x),length(x));
for j=1:length(x)
    Jacwu(j,j)=2.*dt.*mycof*wguess(j).*(uguess(j));
end

Jacww = sparse(length(x),length(x));
for j=1:length(x)
    Jacww(j,j)=1+dt+mycof*dt.*(uguess(j)).^2;
end

%compiling the general Jacobian
Jac=[Jacuu, Jacuw;
     Jacwu, Jacww];

%check for singular Jacobian
if abs(det(Jac))<tol*1e-4, error('singular Jacobian');
end

%solve for "correction"
corr=Jac\res;
corru=corr(1:length(x));
corrw=corr((length(x)+1):(2*length(x)));
uguess=uguess-corru;
```

```
        wguess=wguess-corrw;

        iter=iter+1;
        %unew is the new time step, wnew is the new time step
      end
      unew=uguess; wnew=wguess;
      uprev=unew; wprev=wnew;

   %print the number of Newton iterations at each time step
   fprintf('At the time step number %d number of Newton iters=%d\n',nt,iter);

%plot the system
subplot(2,1,1);plot(x,unew,'Color',[0 0.6 0]);
ylabel('Vegetation U')
grid on
subplot(2,1,2);plot(x,wnew,'b-')

%show chosen frequency
nf=10;
title(sprintf('1-D Newtons Method at t=%g \n for D=%g,
spatial frequency=%g, A=%g \n for a nonlinear system',t,D,nf,a))
xlabel('Space')
ylabel('Water W')
grid on
sprintf ('solution at t=%g',t);pause(0.05)
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%initial condition set-up
function v = initfunu(x)
```

```
%small pertubations around 5 of the max size 0.1
    nf = 10;
    xf = linspace(min(x),max(x),nf);
    rf = 2*rand(nf,1)-1;
    v = 5.*ones(size(x))+0.1.*interp1(xf,rf,x);

function v = initfunw(x)
v = 0.5.*ones(size(x));
```