

DEVELOPMENT OF STRIAE DISTENSAE - A MECHANOCHEMICAL MODEL

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

Aneesh Shankar Hariharan

Certificate of Approval:

Amnon J Meir
Professor
Mathematics

Anotida Madzvamuse, Chair
Assistant Professor
Mathematics

Paul G Schmidt
Professor
Mathematics

Narendra Kumar Govil
Professor
Mathematics

Jo Heath
Professor
Mathematics

George.T.Flowers
Interim Dean
Graduate School

DEVELOPMENT OF STRIAE DISTENSAE - A MECHANOCHEMICAL MODEL

Aneesh Shankar Hariharan

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Master of Science

Auburn, Alabama

May 10, 2007

DEVELOPMENT OF STRIAE DISTENSAE - A MECHANOCHEMICAL MODEL

Aneesh Shankar Hariharan

Permission is granted to Auburn University to make copies of this thesis at its discretion, upon the request of individuals or institutions and at their expense. The author reserves all publication rights.

Signature of Author

Date of Graduation

VITA

Aneesh Shankar Hariharan was born on the 26th of October, 1983 in Chennai, India. He completed his schooling at Kalpakkam, Tamilnadu, India in 2001 and then went on to pursue his undergraduate education in Computer Science and Engineering at Pondicherry Engineering College, Pondicherry, India. He was conferred the Bachelor of Technology Degree in May 2005 and joined the graduate program at Auburn University, AL, USA, in the Department of Mathematics and Statistics as a Graduate Teaching Assistant in August 2005.

THESIS ABSTRACT

DEVELOPMENT OF STRIAE DISTENSAE - A MECHANOCHEMICAL MODEL

Aneesh Shankar Hariharan

Master of Science, May 10, 2007

(B.Tech, Pondicherry Engineering College, Pondicherry, India)

101 Typed Pages

Directed by Anotida Madzvamuse

Striae distensae, otherwise known as stretch marks, are common skin lesions found in a variety of clinical settings. They occur frequently during adolescence or pregnancy when there is rapid tissue expansion, but may also occur in severe weight loss and in corticosteroid excess. Despite a considerable volume of investigative research, the pathogenesis of striae distensae remains obscure. The interpretation of histologic samples, the major investigative tool, demonstrates an association between dermal lymphocytic inflammation, elastolysis and a scarring response. In this thesis, we investigate the pathogenesis of striae distensae by addressing the coupling between mechanical forces and dermal pathology. We develop a mathematical model for illustrating the effect of fibroblasts and the extracellular matrix in the formation of striae, use linear stability analytical theory to predict the range of parameter values where instability sets in and confirm them using finite difference schemes. We show how a mathematical approach provides a realistic framework that may account for the formation of the stretch marks.

Keywords: *Striae Distensae, Mechanochemical model, Stretch marks, Diffusion-driven instability, Finite Difference.*

ACKNOWLEDGMENTS

I would like to express my immense gratitude to Dr Anotida Madzvamuse for serving as a role model and constantly encouraging and supporting me throughout this work. He has been instrumental in fostering my interest in the field of Applied Mathematics. I would also like to thank Professor Amnon Meir for being an inspiration and patiently going through the initial versions of the manuscript and suggesting valuable changes. I am very grateful to all my committee members for consenting to mentor this work. I sincerely thank the Department of Mathematics and Statistics, Auburn University for accepting me into its graduate program with a teaching assistantship. My father, Dr Y Hariharan and my mother, Mrs Parimala Hariharan, have been my childhood inspiration for leading an academic life, for which I am eternally grateful. I would also like to thank Dr A Bharathi, for being a friend and mentor. Finally, I would like to thank my best friend and fiance, Sangeetha Srinivasan, who has encouraged me all through this work and whose company has been invaluable to me.

Style manual or journal used Journal of Approximation Theory (together with the style known as “aums”). Bibliography follows van Leunen’s *A Handbook for Scholars*.

Computer software used The document preparation package T_EX (specifically L^AT_EX) together with the departmental style-file `aums.sty`.

TABLE OF CONTENTS

LIST OF FIGURES	x
1 INTRODUCTION	1
1.1 Striae Distensae: An overview and motivation	1
1.2 Ways of expressing the problem mathematically	2
1.3 Derivation of the mathematical models	4
1.3.1 Fibroblast density conservation equation	5
1.3.2 Conservation equation for the ECM density	7
1.3.3 Fibroblast: ECM interaction equation	8
1.4 Non-dimensionalization of the Model Equations	11
2 LINEAR STABILITY ANALYSIS	16
2.1 Mode selection and parameter values	22
3 FINITE DIFFERENCE SCHEMES	31
3.1 Introduction	31
3.1.1 Derivation of finite difference schemes	32
3.1.2 Equation (A): $n(x, t)$	35
3.1.3 Equation (B): $s(x, t)$	40
3.1.4 Equation (C): $v(x, t)$	41
3.1.5 Updating $u(x, t)$	43
3.1.6 Computer implementation	44
4 RESULTS, CONCLUSIONS AND FUTURE DIRECTIONS	46
4.1 Numerical results	46
4.1.1 The 2-SBDF scheme	46
4.1.2 The 2-SAFBDF scheme	47
4.2 Conclusion and future directions	49
BIBLIOGRAPHY	59
A LINEAR STABILITY ANALYSIS: MATLAB	61
B THE THOMAS ALGORITHM	63
B.1 Description of the algorithm	64
B.1.1 The forward elimination phase	64
B.1.2 The backward substitution phase	64

C IMPLEMENTATION OF THE FINITE DIFFERENCE AND TRI-DIAGONAL SOLVER: FORTRAN	65
D GRAPHS OF THE SOLUTIONS OBTAINED: MATLAB	85
E ERROR CONVERGENCE: MATLAB	86
F TRANSIENT SOLUTIONS: MATLAB	88

LIST OF FIGURES

2.1	Dispersion relation for the positive root by fixing all the parameter values other than τ as listed in the text with $b=0.01$, $\gamma = 0.01$ and varying τ : (a) $\tau = 0.01$, (b) $\tau = 0.1$, (c) $\tau = 0.4$, (d) $\tau = 0.5$, (e) $\tau = 1$, (f) $\tau = 16$	25
2.2	Dispersion relation for the positive root by fixing all the parameter values other than τ as shown in the text with $b = 0.12$, $\gamma = 0.01$ and varying τ : (a) $\tau = 0.01$, (b) $\tau = 0.1$, (c) $\tau = 0.6$, (d) $\tau = 0.7$, (e) $\tau = 1$, (f) $\tau = 30$, and (g) $\tau = 50$	27
2.3	Dispersion relation for the positive root by fixing the parameter values other than b with as listed in the text with $\tau = 0.5$, $\gamma = 0.01$ and taking (a) $b = 1$, (b) $b = 0.04$, (c) and $b = 0.001$ to illustrate the effect that a decrease in b implies an increase in corticosteroid excess thereby giving rising to the formation of striae.	30
4.1	Solutions obtained by fixing all the parameters with $\tau = 0.01$ and $b=0.01$ (a) n, s, u solution values, (b)-(d) error graph, (e)-(g) transient solution for n, s and u using the 2-SBDF scheme for equation (A).	51
4.2	Solutions obtained by fixing all the parameters with $\tau = 0.7$ and $b=0.01$ (a) n, s, u solution values, (b)-(d) error graph, (e)-(g) transient solution for n, s and u using the 2-SBDF scheme for equation (A).	52
4.3	Solutions obtained by fixing all the parameters with $\tau = 0.01$ and $b=0.01$ (a) n, s, u solution values, (b)-(d) error graph, (e)-(g) transient solution for n, s and u using the 2-SAFBDF scheme for equation (A).	53
4.4	Solutions obtained by fixing all the parameters with $\tau = 0.7$ and $b=0.01$ (a) n, s, u solution values, (b)-(d) error graph, (e)-(g) transient solution for n, s and u using the 2-SAFBDF scheme for equation (A).	54
4.5	Solutions obtained by fixing all the parameters with $\tau = 0.01$ and $b=0.12$ (a) n, s, u solution values, (b)-(d) error graph, (e)-(g) transient solution for n, s and u using the 2-SAFBDF scheme for equation (A).	55
4.6	Solutions obtained by fixing all the parameters with $\tau = 0.7$ and $b=0.12$ (a) n, s, u solution values, (b)-(d) error graph, (e)-(g) transient solution for n, s and u using the 2-SAFBDF scheme for equation (A).	56

4.7	Solutions obtained by fixing all the parameters with $\tau = 0.5$ and $b=1$ (a) n , s , u solution values, (b)-(d) error graph, (e)-(g) transient solutions for n , s and u using the 2-SAFBDF scheme for Equation (A).	57
4.8	Solutions obtained by fixing all the parameters with $\tau = 0.7$, $b= 0.01$, $L=10$ and $m=1000$ (a) n , s , u solution values, (b)-(d) error graph, (e)-(g) transient solutions for n , s and u using the 2-SAFBDF scheme for Equation (A). . .	58

Development of Striae Distensae - A Mechanochemical Model

Aneesh Shankar Hariharan

May 10, 2007

CHAPTER 1
INTRODUCTION

1.1 Striae Distensae: An overview and motivation

Striae distensae, commonly known as stretch marks is a skin condition that does not cause any significant medical problem but causes sizable distress to those affected. Striae distensae results when the skin is subjected to continuous stretching; increased stress is placed on the connective tissue due to increased size of the various parts of the body. It occurs on the abdomen and the breasts of pregnant women, on the shoulders of body builders, in adolescence, and in individuals who are overweight. Skin elastolysis leads to excessive mast cell degranulation with subsequent damage of collagen and elastin. Prolonged use of oral or topical corticosteroids or Cushing syndrome (increased adrenal cortical activity) leads to the development of striae. Histological findings suggest that in the early stages, inflammatory changes may predominate; edema is present in the dermis. In the later stages, the epidermis becomes thin and flattened with loss of the rete ridges. The dermis has thin, densely packed collagen bundles arranged in a parallel array horizontal to the epidermis at the level of the papillary dermis. Elastic stains show breakage and retraction of the elastic fibers in the reticular dermis. The broken elastic fibers curl at the sides of the striae to form a distinctive pattern. Scanning electron microscopy shows extensive tangles of fine, curled elastic fibers with a random arrangement. This arrangement is in contrast to normal skin, which has thick, elastic fibers with a regular distribution. When viewed by a transmission electron microscope, the ultrastructure of elastic and collagen fibers in striae is similar to that of healthy skin. The factors that lead to the development of striae are

poorly understood. No general consensus exists as to what causes striae. One suggestion is that they develop as a result of stress rupture of the connective tissue framework. It has also been suggested that they develop more easily in skin that has a high proportion of rigid cross-linked collagen, for example, in early adult life. This is evident in the striae caused by pregnancy, lactation, weight lifting, and other stressful activities. Increased adrenal cortical activity has been implicated in the formation of striae, as in the case of Cushing syndrome. Additionally, the cellular and extracellular matrix (ECM) alterations that mediate the clinical phenotype of stretch marks remain poorly understood [1]. There have been several detailed studies involving the effects and even temporary treatments for this problem, but the pathogenesis is still a mere speculation. It has been established, though, that the cause for striae distensae could be predominantly mechanical and in certain cases such as the Cushing's syndrome, it is due to corticosteroid excess. Furthermore, it has also been established that pattern forming processes of striae distensae have lots of similarities to those of wound healing, for which a mechanochemical model has been successfully devised and studied by Maini [2]. This also serves as motivation for devising a mathematical framework for striae distensae.

1.2 Ways of expressing the problem mathematically

Histological samples suggest rupture of the connective tissue framework and the ECM alterations to facilitate the phenotype of striae. The most common cells in the connective tissue are the fibroblasts. A fibroblast is a type of cell that synthesizes and maintains the ECM of many animal tissues. Fibroblasts provide a structural framework (stroma) for many tissues. The main function of fibroblasts is to maintain the structural integrity of connective

tissue by continuously secreting precursors of the ECM. Fibroblasts have a branched cytoplasm surrounding an elliptical, speckled nucleus having 1 or 2 nucleoli. Though disjointed and scattered when they have to cover a large space, fibroblasts when crowded often locally align parallel in clusters. Fibroblasts make collagen, glycosaminoglycans, reticular and elastic fibers, and glycoproteins found in the extracellular matrix. In growing individuals, fibroblasts are dividing and synthesizing the ECM. The composition of the extracellular matrix determines the physical properties of connective tissues. Fibroblasts are morphologically heterogeneous, with diverse appearances depending on their location and activity. Though morphologically inconspicuous, ectopically transplanted fibroblasts can often retain positional memory of the location and tissue context where they had previously resided. Fibroblasts can also migrate slowly over substratum as individual cells. While epithelial cells form the lining of body structures it is fibroblasts and related connective tissues which sculpt the bulk of an organism.

The basic mechanochemical model for striae relies on two of the following experimental results: (i) fibroblasts migrate within a tissue substratum, which is made up of fibrous extracellular matrix [3], and (ii) these cells generate large traction forces during the formative stages of stretch marks [4]. Hence the mechanism we shall develop will model the mechanical interaction between the motile cells and the elastic substratum within which they move. Fibroblast cells move by exerting forces on their surroundings, consisting of ECM and the surface of other cells. They use their cellular protrusions, filopodia, which stretch out from the cells in all directions and pulling everything in its path. The biology of these protrusions is available in [5]. As the fibroblasts move through the ECM, they deform

it by virtue of their traction forces. These deformations induce anisotropy effects which in turn affect the cell motion. The effects of these various coordinations results in spatially organized cell aggregations in striae (refer to [13] and [14] for further details).

A continuum model for striae has three governing equations:

1. The conservation equation for the fibroblast density,
2. The conservation law governing the ECM and
3. The mechanical balance of forces between the cells and the ECM.

This thesis is organized as follows: In Section 1.3, we derive the mathematical models that describe pattern formation in striae. These models are then non-dimensionalized; as illustrated in Section 1.4. In Section 2, we carry out detailed linear stability studies in order to determine ranges for parameter values that will give rise to striae. The predictions of linear stability theory are then verified through finite difference schemes to provide approximate solutions to the model equations. Finally, in Section 4, we present our numerical results, conclusions and highlight future research ideas.

1.3 Derivation of the mathematical models

Let $V \subset \mathbb{R}^3$ be a simply connected bounded volume at time $t \in (0, T)$, $T > 0$ and $S \subset \mathbb{R}^2$ be the surface closing the volume V . Throughout the course of this work, we denote $N(\mathbf{x}, t)$ and $S(\mathbf{x}, t)$ as the fibroblast and ECM densities respectively at position $\mathbf{x} \in V$ at time t . $\mathbf{U}(\mathbf{x}, t)$ denotes the displacement vector of the ECM, that is a material point in the matrix initially at position \mathbf{x} and undergoes a displacement to $\mathbf{x} + \mathbf{U}$. The following equations are thus derived.

1.3.1 Fibroblast density conservation equation

Let $c(\mathbf{x},t)$ be the density of any material. The general conservation equation says the rate of change of amount of material in V equals the rate of flow of material across S into V plus the material created in V . Thus

$$\frac{d}{dt} \int_V c(\mathbf{x},t) dV = - \int_V J \cdot dS + \int_V f dV, \quad (1.1)$$

where J is the flux of material and f represents the source of material which may be a function of $c(\mathbf{x},t)$. Applying divergence theorem to the surface integral and assuming $c(\mathbf{x},t)$ is continuous and assuming V is continuous for all time t , the above equation becomes

$$\int_V \left(\frac{\partial c}{\partial t} + \nabla \cdot J - f(c, \mathbf{x}, t) \right) dV = 0. \quad (1.2)$$

Since the volume is arbitrary at all times, the integral must be 0 and so the conservation equation for $c(\mathbf{x},t)$ is

$$\frac{\partial c}{\partial t} + \nabla \cdot J = f(c, \mathbf{x}, t). \quad (1.3)$$

If classical diffusion is the process then the generalization, for example, is $J = -D\nabla c$ [11]. We can also generalize the above to a case where we have several interacting materials, species or chemicals. We will then have a vector $\mathbf{c}_i(\mathbf{x},t)$ each diffusing with its own diffusion coefficient D_i and interacting according to the vector source term f . For simplicity, we will restrict our studies to a one-dimensional model. The above obtained equation represents a reaction-diffusion system. We will use this conservation equation to model the fibroblast density. Since N represents the fibroblast density, the general form of the conservation

equation is

$$\frac{\partial N}{\partial t} = -\nabla \cdot J + M, \quad (1.4)$$

where J is the flux of cells, that is, the number of fibroblast cells crossing a unit area per unit time and M is cell proliferation. For simplicity, we will assume that there is no cell proliferation. The inclusion of cell proliferation should be trivial. The flux can be expressed as

$$J = J_d + J_c$$

where $J_d = -D \nabla N$ represent the flux due to Fick's law. The diffusion term models random motion, and the flux term models directed orientation [11]. Biologically it is known that fibroblasts move up ridges in the substratum and more specifically, experiments by Guido and Tranquillo [11] show that, within oriented collagen gels, fibroblasts move preferentially in the direction of collagen orientation by pulling themselves along the fibers. Thus the flux term is due to this tendency of fibroblasts to move in the direction of collagen fibers, so that if there is a gradient of collagen, the fibroblasts tend to re-orient so as to move up that gradient. With $U(x, t)$ representing the displacement vector of the ECM, the convective flux contribution J_c is

$$J_c = N \frac{\partial U}{\partial t}. \quad (1.5)$$

Here the velocity of deformation of the matrix is $\frac{\partial U}{\partial t}$ and the amount of cells transported is simply N times the velocity. The convective flux is the most important contributor to the fibroblast transport [11].

Cells also tend to disperse randomly in a homogenous isotropic medium. Classical diffusion contributes a flux term $-D \nabla N$ which models the random motion in which fibroblasts

respond to local variations in fibroblast density and tend to move down the density gradient [19]. Also, in the case of striae, the motility of the fibroblasts is enhanced along the strain lines and hence $D(U_x) = D_0(1 + \gamma U_x)$ where $\gamma > 0$ is a constant. Combining all the results obtained above, we derive the conservation equation for fibroblast density as

$$N_t = (D N_x)_x - (N U_t)_x. \quad (1.6)$$

It can be easily deduced therefore that the equation simplifies to

$$N_t = D_0 N_{xx} + D_0 \gamma (N_{xx} U_x + N_x U_{xx}) - N_x U_t - N U_{tx}. \quad (1.7)$$

1.3.2 Conservation equation for the ECM density

The next governing equation that we model is the conservation equation for the ECM density. Within the extracellular matrix, collagen takes the form of a fibrous network with an elaborate structure including cross links, and consequently there is essentially no random re-orientation of the collagen. Because the fibroblasts degrade and produce collagen, thus reforming the network with collagen oriented in the direction of the fibroblasts, the equation for ECM has an angular flux term. For simplicity, in our system we do not allow any net change in the amount of collagen; the fibroblasts simply remodel the existing network. As a result there is only one term in the ECM equation which involves the density of ECM, the interaction of ECM with fibroblast, and the gradient of interaction with fibroblasts. This term is similar to the flux term in the equation for the fibroblasts and it also contains an additional term which arises because the rate of collagen remodeling depends on the density of fibroblasts doing that re-modeling [19]. The ECM primarily moves by advection and we

mathematically model its conservation. The conservation equation for the matrix material $S(x, t)$ is

$$\frac{\partial S}{\partial t} + \nabla \cdot (SU) = P(N, S, U). \quad (1.8)$$

where matrix flux is taken to be mainly via convection and $P(N, S, U)$ is the rate of secretion of matrix by the cells [11]. As already discussed, secretion and degradation is thought to play a role in certain situations involving fibroblast organization and in wound healing [15]. However, on the time scale of cell motions that we consider, we can neglect this effect and hence we can assume that $P = 0$. Also experiments by [16] indicate $P = 0$ during pattern formation. Hence the conservation equation for striae is taken as

$$S_t = -(SU_t)_x = -S_x U_t - S U_{tx}. \quad (1.9)$$

1.3.3 Fibroblast: ECM interaction equation

The composition of the ECM within which the fibroblasts move is a lot more complex and moreover its constituents change as development proceeds. Its mechanical properties have not been well studied either. In this case though, we are interested only in the mechanical interaction between the fibroblasts and the matrix. Also the mechanical deformations are very small [11], so as a reasonable approximation, we take the composite material of fibroblasts plus matrix to be modeled as a linear, isotropic visco-elastic continuum with stress tensor σ . We assume that the traction forces generated by the cell are in mechanical equilibrium with the elastic restoring forces developed in the matrix and any external forces

present. The mechanical force balance fibroblast-matrix equation is then [17],

$$\nabla \cdot \sigma + KF = 0, \quad (1.10)$$

where F is the external force acting on the matrix (per unit matrix) and σ is the stress tensor. K is a measure of the anchoring force that resists the movement of the ECM relative to the subcutis [11]. This equation, when applied to a spring loaded with a weight, simply says that the applied force is balanced by the elastic forces from the extended spring. Considering the stress tensor σ that has contributions both from the ECM and the cells, we have

$$\sigma = \sigma_{\text{ECM}} + \sigma_{\text{cell}}. \quad (1.11)$$

The expression for a linear visco-elastic material as derived in [17], gives the stress strain constitutive relation as

$$\sigma_{\text{ECM}} = [(\mu_1 + \mu_2)\epsilon_t + E'\epsilon[1 + \nu']], \quad (1.12)$$

where $E' = \frac{E}{1+\nu}$, $\nu' = \frac{\nu}{1-2\nu}$, μ_1 and μ_2 are the shear and bulk viscosities of the ECM, $\epsilon = U_x$ is the strain tensor, and E and ν are the Young's modulus and the Poisson ratio respectively. Denoting $B = E \frac{1-\nu}{(1+\nu)(1-2\nu)}$ and $A = \mu_1 + \mu_2$, we have

$$\sigma_{\text{ECM}} = A\epsilon_t + B\epsilon. \quad (1.13)$$

Next we consider the contribution to the stress tensor from the fibroblast tractions, that is σ_{cell} . The more fibroblasts there are, the greater the traction force. Also the traction force is enhanced along the strain lines, hence the measure of the traction force as a function of

U_x is modeled as $\phi(U_x) = \tau_0(1 + \delta U_x)$ where the dimensionless parameter δ quantifies the increase in traction along the strain lines [11]. If the filopodia, with which the fibroblasts attach to the ECM extend beyond their immediate neighborhood, as they probably do, we need to include a long range diffusion effect. Hence combining all these parameters, we get the stress tensor to be

$$\sigma_{\text{cell}} = \phi S N, \quad (1.14)$$

where ϕ is as defined above and S is replaced by $S + h_0 S_{xx}$ to include a long range effect in the cell traction term [11]. Finally we consider the body force F . The matrix material is attached to a substratum of underlying tissue or the epidermis, like ropes. We model these restraining forces as body forces proportional to the density of the ECM and the displacement of the matrix from its unrestrained position and thus take $F = -SU$. Combining all the above the force balance equation (1.10) reduces to:

$$(A\epsilon_t + B\epsilon + \phi S N) = K S U. \quad (1.15)$$

This equation reduces to

$$\begin{aligned} K S U &= (A U_{xt} + B U_x + \tau_0 (1 + \delta U_x)(S + h_0 S_{xx}) N)_x \\ &= A U_{xxt} + B U_{xx} + \tau_0 (S N)_x + \tau_0 \delta (S N U_x)_x + \tau_0 h_0 (N S_{xx})_x + \tau_0 h_0 \delta (N U_x S_{xx})_x \\ &= A U_{xxt} + B U_{xx} + \tau_0 N S_x + \tau_0 S N_x + \tau_0 \delta N S_x U_x + \tau_0 \delta S N_x U_x + \tau_0 \delta N S U_{xx} \\ &\quad + \tau_0 h_0 N_x S_{xx} + \tau_0 h_0 N S_{xxx} + \tau_0 h_0 \delta N_x U_x S_{xx} + \tau_0 h_0 \delta N U_{xx} S_{xx} \\ &\quad + \tau_0 h_0 \delta N U_x S_{xxx}. \end{aligned} \quad (1.16)$$

1.4 Non-dimensionalization of the Model Equations

Let us define the following new variables and time- and spatial-scales

$$n(\hat{t}, \hat{x}) = N^* N, \quad s(\hat{t}, \hat{x}) = S^* S, \quad u(\hat{t}, \hat{x}) = U^* U, \quad \hat{t} = t^* t \quad \text{and} \quad \hat{x} = x^* x. \quad (1.17)$$

We can therefore compute the following derivatives

$$\frac{\partial t}{\partial \hat{t}} = \frac{1}{t^*}, \quad \frac{\partial x}{\partial \hat{x}} = \frac{1}{x^*},$$

then it follows that

$$\begin{aligned} n_{\hat{t}} &= \frac{N^*}{t^*} N_t, & n_{\hat{x}} &= \frac{N^*}{x^*} N_x, & n_{\hat{x}\hat{x}} &= \frac{N^*}{x^{*2}} N_{xx}, \\ u_{\hat{t}} &= \frac{U^*}{t^*} U_t, & u_{\hat{x}} &= \frac{U^*}{x^*} U_x, & U_{\hat{t}\hat{x}} &= \frac{U^*}{t^* x^*} U_{tx}, & u_{\hat{x}\hat{x}} &= \frac{U^*}{x^{*2}} U_{xx}, \\ u_{\hat{x}\hat{x}\hat{x}} &= \frac{U^*}{x^{*3}} U_{xxx}, & u_{\hat{x}\hat{x}\hat{t}} &= \frac{U^*}{x^{*2} t^*} U_{xxt}, \\ s_{\hat{t}} &= \frac{S^*}{t^*} S_t, & s_{\hat{x}} &= \frac{S^*}{x^*} S_x, & s_{\hat{x}\hat{x}} &= \frac{S^*}{x^{*2}} S_{xx}, & s_{\hat{x}\hat{x}\hat{x}} &= \frac{S^*}{x^{*3}} S_{xxx}. \end{aligned} \quad (1.18)$$

Equation (1.7) can be written as

$$\begin{aligned} n_{\hat{t}} &= \frac{N^*}{t^*} (D_0 N_{xx} + D_0 \gamma (N_{xx} U_x + N_x U_{xx}) - N_x U_t - N U_{tx}) \\ &= \frac{D_0}{t^*} N^* N_{xx} + \frac{D_0 \gamma}{t^*} N^* N_{xx} U_x + \frac{D_0 \gamma}{t^*} N^* N_x U_{xx} - \frac{1}{t^*} N^* N_x U_t - \frac{1}{t^*} N^* N U_{tx} \\ &= \frac{D_0 x^{*2}}{t^*} \frac{N^*}{x^{*2}} N_{xx} + \frac{D_0 \gamma x^{*2}}{t^*} \frac{N^*}{x^{*2}} N_{xx} \frac{x^* U_x}{x^*} + \frac{D_0 \gamma x^*}{t^*} \frac{N^*}{x^*} N_x \frac{x^{*2} U_{xx}}{x^{*2}} \\ &\quad - \frac{N^*}{x^*} N_x \frac{x^*}{t^*} U_t - N^* N \frac{x^*}{t^* x^*} U_{tx}. \end{aligned}$$

Define

$$t^* = \frac{1}{T}, \quad x^* = \frac{1}{L}, \quad d = \frac{D_0 T}{L^2}, \quad U^* = x^* = \frac{1}{L}$$

where T and L are defined time and length scales. Using the definitions in (1.17) - (1.18) we obtain the non-dimensionalized equation

$$(A) \quad n_{\hat{t}} = d n_{\hat{x}\hat{x}} + d\gamma (n_{\hat{x}\hat{x}} u_{\hat{x}} + n_{\hat{x}} u_{\hat{x}\hat{x}}) - (n u_{\hat{t}})_{\hat{x}}. \quad (1.19)$$

Equation (1.9) can be written as

$$\frac{S_t S^* U^*}{x^* t^*} = \frac{S_x S^* U^* U_t}{x^* t^*} - S^* S \frac{U^* U_{tx}}{t^* x^*}. \quad (1.20)$$

The above representation for Equation (1.20) is valid since all we have done is multiply both sides by $\frac{S^* U^*}{t^* x^*} > 0$. Using the definitions (1.17) - (1.18) we obtain the equation

$$(B) \quad s_{\hat{t}} = -s_{\hat{x}} u_{\hat{t}} - s u_{\hat{t}\hat{x}}. \quad (1.21)$$

Similarly equation (1.16) is multiplied by $\frac{U^*}{x^{*2} t^*} > 0$ yields

$$\begin{aligned} K \frac{S}{t^* x^{*2}} U^* U &= A \frac{U^* U_{xxt}}{x^{*2} t^*} + B \frac{U^* U_{xx}}{x^{*2} t^*} + \tau_0 \frac{1}{t^*} \frac{S_x U^*}{x^* x^*} N + \tau_0 \frac{S}{t^*} \frac{N_x U^*}{x^* x^*} + \tau_0 \delta \frac{N}{t^*} \frac{S_x U^* N_x}{x^* x^*} \\ &+ \tau_0 \delta \frac{S}{t^*} \frac{N_x U^* U_x}{x^* x^*} + \tau_0 \delta \frac{S}{t^*} N \frac{U^* U_{xx}}{x^{*2}} + \tau_0 h_0 N_x \frac{S_{xx} U^*}{x^{*2} t^*} + \tau_0 h_0 \frac{N}{t^*} \frac{S_{xxx} U^*}{x^{*2}} \\ &+ \tau_0 h_0 \frac{\delta}{t^*} \frac{N_x U^* U_x}{x^* x^*} S_{xx} + \tau_0 h_0 \frac{N}{t^*} \frac{U^* U_{xx}}{x^{*2}} S_{xx} + \tau_0 h_0 \frac{N}{t^*} \frac{U^* U_x S_{xxx}}{x^* x^*}. \end{aligned}$$

Multiplying both sides by $\frac{1}{S_0}$ where $S^* = \frac{1}{S_0}$ and defining $N^* = \frac{1}{N_0}$ and applying definitions (1.17) - (1.18) we obtain

$$\begin{aligned}
KTL^2su &= \frac{A}{S_0}u_{\hat{x}\hat{x}\hat{t}} + \frac{B}{S_0}Tu_{\hat{x}\hat{x}} + \tau_0N_0Ts_{\hat{x}}n + \tau_0N_0Tsn_{\hat{x}} + \tau_0N_0\delta Tns_{\hat{x}}u_{\hat{x}} + \tau_0N_0\delta Tsn_{\hat{x}}u_{\hat{x}} \\
&+ \tau_0N_0\delta Tsn_{\hat{x}}u_{\hat{x}} + \tau_0N_0Th_0x^{*2}n_{\hat{x}}s_{\hat{x}\hat{x}} + \tau_0N_0Th_0x^{*2}ns_{\hat{x}\hat{x}} + \tau_0N_0Th_0x^{*2}\delta n_{\hat{x}}u_{\hat{x}}s_{\hat{x}\hat{x}} \\
&+ \tau_0N_0Th_0x^{*2}\delta nu_{\hat{x}\hat{x}}s_{\hat{x}\hat{x}} + \tau_0N_0Th_0x^{*2}\delta nu_{\hat{x}}s_{\hat{x}\hat{x}\hat{x}}. \tag{1.22}
\end{aligned}$$

Dividing both sides by KTL^2 and defining

$$a = \frac{A}{K S_0 T L^2}, \quad b = \frac{B}{K S_0 L^2}, \quad \tau = \frac{\tau_0 N_0}{K L^2}, \quad h = \frac{h_0}{L^2}$$

we get the following equation

$$\begin{aligned}
\text{(C)} \quad &a u_{\hat{x}\hat{x}\hat{t}} + b u_{\hat{x}\hat{x}} + \tau (n s_{\hat{x}} + s n_{\hat{x}} + h n_{\hat{x}} s_{\hat{x}\hat{x}} + h n s_{\hat{x}\hat{x}\hat{x}}) \\
&+ \tau \delta (n s_{\hat{x}} u_{\hat{x}} + s n_{\hat{x}} u_{\hat{x}} + s n u_{\hat{x}\hat{x}} + h n_{\hat{x}} u_{\hat{x}} s_{\hat{x}\hat{x}} + h n u_{\hat{x}\hat{x}} s_{\hat{x}\hat{x}} + h n u_{\hat{x}} s_{\hat{x}\hat{x}\hat{x}}) = s u. \tag{1.23}
\end{aligned}$$

We can drop the \hat{s} above all the new variables and also set $\tau = t$ without any loss of generality to obtain the following non-dimensionalized equations

$$n_t = d n_{xx} + d \gamma (n_{xx} u_x + n_x u_{xx}) - (n u_t)_x, \tag{1.24}$$

$$s_t = -s_x u_t - s u_{tx} \tag{1.25}$$

$$\begin{aligned}
&a u_{xxt} + b u_{xx} + \tau (n s_x + s n_x + h n_x s_{xx} + h n s_{xxx}) \\
&+ \tau \delta (n s_x u_x + s n_x u_x + s n u_{xx} + h n_x u_x s_{xx} + h n u_{xx} s_{xx} + h n u_x s_{xxx}) = s u. \tag{1.26}
\end{aligned}$$

These new equations would be our governing equations for the rest of the work. To close the system, we prescribe initial conditions as small random perturbations around the uniform steady state, if it exists. Boundary conditions are taken as zero-flux. Zero-flux boundary conditions imply self-organizing processes. As it can be seen, there are a number of parameter values throughout these system of equations, each having a physical meaning even though they are dimensionless. We set the following parameter values based on experimental results obtained from our collaborators.

$$\left\{ \begin{array}{l} A = 10^8 N s cm^{-2}, \quad D_0 = 10^{-9} cm^2 s^{-1}, \quad N_0 = 10^4 cm^{-3} \\ S_0 = 10^{-1} g cm^{-3}, \quad \delta = 1, \quad h_0 = 10^{-1} cm^2, \quad K = 10^3 N cm^{-4} S_0^{-1}, \\ B = 1, \quad 12, \quad 100 \quad N cm^{-2}, \\ \tau_0 = 0.001, \quad 0.0075, \quad 0.01, \quad 0.045, \quad 0.1, \quad 0.2 \quad N cm^{-2} N_0^{-1} S_0^{-1}, \\ \gamma = 0.01, \quad 0.05. \end{array} \right. \quad (1.27)$$

Also $L = 1$ cm and $T = 10^{6.5}$ s. The values corresponding to the non-dimensional parameters can be shown to be given by:

$$\left\{ \begin{array}{l} d = 10^{-2.5}, \quad a = 10^{-0.5}, \quad \delta = 1, \quad h = 0.1 \\ b = 0.01, \quad 0.12, \quad 1, \\ \tau = 0.01, \quad 0.075, \quad 0.1, \quad 0.45, \quad 1, \quad 2, \\ \gamma = 0.01, \quad 0.05. \end{array} \right. \quad (1.28)$$

The parameters with variable values suggest different systems independent of each other and are in relation to common situations encountered by biologists and dermatologists. For example, increasing values for the parameter τ correspond to an increase in the traction force

exerted by the fibroblasts on the ECM and decreasing b values correspond to a decrease in Young's modulus, which may occur in cases of corticosteroid excess. The rest of the parameters are kept constant since their increase or decrease does not have much physical significance and the studies are assumed to in with relation to varying τ , b and γ , since they would have direct biological consequences which would help us to better understanding the causes of striae distensae. The process of non-dimensionalization has many advantages, the evident one being that the number of relevant parameters has significantly reduced dimensionless groupings, from 10 to 7. This would ease the task of determining the dynamics of the system [11]. The reader is referred to [12] for a detailed record on the advantages of non-dimensionalization. One thing to be noted here is that all the dimensionless parameters, all of which are positive, can be divided into those associated with the fibroblast properties such as τ and those related to the matrix properties, such as b , γ etc. In particular we vary b and τ , since these signify important physical concepts of the Young's modulus and the cell traction force respectively. Therefore, we will concentrate on these parameters in order to get an idea of the restraints that both the fibroblasts and the ECM experience. We also vary γ to see if it has any effect on these parameters and the patterns generated.

Although we have modeled an analytically formidable system, the conceptual framework is pretty clear. We have not included all the effects that might be relevant such as cell proliferation and the effects of matrix secretion. However, it is presumed that if this model works out well, more complex features that dermatologists feel are important can be incorporated. We have also made sure that all the essential features for the generation of stretch marks have been incorporated and hence the model generated will be very realistic.

CHAPTER 2
LINEAR STABILITY ANALYSIS

In a seminal paper, Turing [21] showed mathematically that small random spatial fluctuations in an otherwise well-mixed system of reacting and diffusing chemicals could be driven unstable in the presence of diffusion to give rise to a spatial pre-pattern [11]. This process is now widely known as diffusion-driven instability. Turing's discovery was ground-breaking; not only was it mathematically counter-intuitive but also it gave rise to explaining mathematically the development of form or structure in an organism.

Our biological problem is very much concerned with pattern development and structure. Using Turing's theory in order to model spatial aspects observed during pattern formation of striae, the equation system (1.24) - (1.26) must admit spatially inhomogeneous solutions. Considering their complexity, it is highly unlikely that useful analytical solutions can be found to these systems. However, it is known that much of any pattern formation potential is predicted by linear analysis about the uniform steady state solutions close to the bifurcation points [11]. It is also known that such linear predictions are not infallible and must be backed up by numerical simulations if finite amplitude structures are required [11]. Therefore it is worthwhile, from a practical standpoint, to carry out a detailed linear stability analysis of the governing equations, not only as a first analytical step to indicate spatial pattern forming potentialities but also because striae may involve solutions that effectively fall within the linear regime. The latter case is shown to be effectively those from a nonlinear theory close to bifurcation from uniformity [11]. So, even though this process is quite messy and unduly complicated, if we wish to demonstrate the powerful pattern forming capabilities of striae,

the effort is worthwhile. To investigate solution behavior we first examine the linearization of equations (1.24) - (1.26) about the normalized steady state $\omega_0 = (n_0, s_0, u_0)^T = (1, 1, 0)^T$ looking for solutions

$$\omega_1 = n(t, x) - n_0, \quad \omega_2 = s(t, x) - s_0, \quad \text{and} \quad \omega_3 = u(t, x) - u_0, \quad (2.1)$$

which are proportional to $e^{\lambda t + i k x}$ where k is the wave number and λ is the linear growth factor. Here we assume zero flux boundary conditions for the system of partial differential equations which correspond to the case of a closed cylindrical or spherical surface in three dimensions. Zero-flux boundary conditions imply no external input (a case of self-organizing process) which is the case in striae distensae. Hence we now have

$$n(t, x) = n_0 e^{\lambda t + i k x}, \quad s(t, x) = s_0 e^{\lambda t + i k x}, \quad \text{and} \quad u(t, x) = u_0 e^{\lambda t + i k x}. \quad (2.2)$$

Using the above obtained forms, we compute the following derivatives:

$$\begin{aligned} n_t &= \lambda n(t, x), & s_t &= \lambda s(t, x), & u_t &= \lambda u(t, x), \\ n_x &= i k n(t, x), & s_x &= i k s(t, x), & u_x &= i k u(t, x), \\ n_{xx} &= -k^2 n(t, x), & s_{xx} &= -k^2 s(t, x), & u_{xx} &= -k^2 u(t, x), \\ & & s_{xxx} &= -i k^3 s(t, x), & u_{xxx} &= -i k^3 u(t, x), \\ u_{tx} &= i k \lambda u(t, x), & u_{xxt} &= -k^2 \lambda u(t, x). \end{aligned} \quad (2.3)$$

The above derivatives are identical to the derivatives of the new variables ω_1 , ω_2 and ω_3 .

Let us consider equation (1.24) and linearise about the steady state:

$$n_t = d n_{xx} + d \gamma (n_x u_x)_x - (n u_t)_x,$$

$$(\omega_1 + n_0)_t = d(\omega_1 + n_0)_{xx} + d \gamma ((\omega_1 + n_0)_x (\omega_3 + u_0)_x)_x - ((\omega_1 + n_0) (\omega_3 + u_0)_t)_x \quad \text{and}$$

$$\omega_{1t} = d \omega_{1xx} + d \gamma (\omega_{1x} \omega_{3x})_x - (\omega_1 \omega_{3t})_x - \omega_{3tx}.$$

Substituting the definitions (2.2) and derivatives (2.3) and simplifying where non-linear terms are ignored we have

$$(\lambda + d k^2) n_0 + i \lambda k u_0 = 0. \quad (2.4)$$

Similarly consider the equation (1.25). Linearizing about the steady state we have the following

$$s_t = -(s u_t)_x = -((\omega_2 + 1)(\omega_{3t}))_x = -(\omega_2 \omega_{3t})_x - \omega_{3tx} = \omega_{2t}.$$

Substituting the derivatives and simplifying yields

$$\lambda s_0 + i \lambda k u_0 = 0. \quad (2.5)$$

For equation (1.26), linearizing about the steady state, we obtain the following equation

$$\begin{aligned} s u = & a u_{xxt} + b u_{xx} + \tau (n s_x + s n_x + h n_x s_{xx} + h n s_{xxx}) + \\ & \tau \delta (s n_x u_x + n s_x u_x + n s u_{xx} + h n_x u_x s_{xx} + h n u_{xx} s_{xx} + h n u_{xx} s_{xxx}). \end{aligned}$$

Hence

$$\begin{aligned}
(\omega_2 + s_0)(\omega_3 + u_0) &= a\omega_{3xxt} + b\omega_{3xx} + \tau[(\omega_1 + 1)(\omega_2 + 1)_x + (\omega_2 + 1)(\omega_1 + 1)_x + \\
&\quad h(\omega_1 + 1)_x(\omega_2 + 1)_{xx} + h(\omega_1 + 1)(\omega_2 + 1)_{xxx}] + \\
&\quad \tau\delta[(\omega_2 + 1)(\omega_1 + 1)_x\omega_{3x} + (\omega_1 + 1)(\omega_2 + 1)_x\omega_{3x} + \\
&\quad (\omega_1 + 1)(\omega_2 + 1)(\omega_3 + 1)_{xx} + h(\omega_1 + 1)_x\omega_{3x}(\omega_2 + 1)_{xx} + \\
&\quad h(\omega_1 + 1)\omega_{3xx}(\omega_2 + 1)_{xx} + hn\omega_{3x}(\omega_2 + 1)_{xxx}].
\end{aligned}$$

Ignoring non-linear terms, we have

$$a\omega_{3xxt} + b\omega_{3xx} + \tau[\omega_{2x} + \omega_{1x} + h\omega_{2xxx}] + \tau\delta[\omega_{3xx}] = u_0.$$

Re-arranging terms after substituting the values of the derivatives (2.3) and canceling common terms on both sides, we get

$$i\tau k n_0 + i\tau (k - h k^3) s_0 - (a\lambda k^2 + b k^2 + \tau \delta k^2 + 1) u_0 = 0. \quad (2.6)$$

Combining equations (2.4) - (2.6) in matrix form:

$$\begin{bmatrix}
\lambda + d k^2 & 0 & i \lambda k \\
0 & \lambda & i \lambda k \\
i \tau k & i \tau (k - h k^3) & -(a\lambda k^2 + b k^2 + \tau \delta k^2 + 1)
\end{bmatrix}
\begin{bmatrix}
n_0 \\
s_0 \\
u_0
\end{bmatrix}
=
\begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix}. \quad (2.7)$$

The (3×3) matrix is known as the Jacobian matrix. The characteristic equation for the Jacobian matrix can be shown to be given by

$$-\lambda (\lambda + dk^2) (a \lambda k^2 + bk^2 + \tau \delta k^2 + 1) + \lambda k (\tau k - h \tau k^3) (\lambda + dk^2) + \tau \lambda^2 k^2 = 0$$

Clearly $\lambda = 0$ is a trivial root. The quadratic equation

$$a(k) \lambda^2 + b(k) \lambda + c(k) = 0 \tag{2.8}$$

gives rise to the non trivial roots, where

$$a(k) = a k^2, \tag{2.9}$$

$$b(k) = (a d + \tau h) k^4 + (\tau (\delta - 2) + b) k^2 + 1, \tag{2.10}$$

$$c(k) = d \tau h k^6 + (d \tau (\delta - 1) + b d) k^4 + d k^2. \tag{2.11}$$

The roots of the characteristic equation (2.8) are given by

$$2 a(k) \lambda_{1,2} = -b(k) \pm \sqrt{b(k)^2 - 4a(k)c(k)}. \tag{2.12}$$

Spatially heterogeneous solutions of the linear system are characterized by the dispersion relation $\lambda(k)$ which has $\text{Re}(\lambda(k)) \leq 0$ but which exhibits a range of unstable modes with $\text{Re}(\lambda(k)) > 0$ for some $k \neq 0$. If $k = 0$, then $\lambda = 0$, the only spatially stable homogeneous solution. Although these are stable solutions, for our purposes, we are interested in stable spatially inhomogeneous solutions. At the onset of diffusion-driven instability, the

stable homogeneous solution becomes unstable, instead the spatially inhomogeneous solution becomes stable thereby characterizing the phenotype for pattern formation. For this to occur we require that $\text{Re}(\lambda(k)) > 0$ to be satisfied for some $k \neq 0$. All the solutions with these k 's are then linearly unstable and grow exponentially with time. These unstable heterogeneous solutions will evolve into finite amplitude spatially structured solutions [11]. The key assumption here is that these linearly unstable eigenfunctions which are growing exponentially with time will be eventually bounded by the non linear terms of the reaction diffusion system of equations and ultimately steady state spatially inhomogeneous solution will emerge [11]. We would intuitively expect that if a confined set exists for the kinetics, the same set would also contain the solutions when diffusion is included [20]. So, part of the analysis of this mechanism involves proving the presence of a confined state within the positive quadrant. As it can be seen from the dispersion relation, the only way a solution with $\text{Re}(\lambda(k)) > 0$ can exist is if $b(k) < 0$ or $c(k) < 0$ or both, from Descartes's rule of signs. Since the only negative terms involve τ , it becomes necessary that $\tau > 0$. Biologically too, this makes sense, since the cell traction forces are the only contributors to the aggregative process in the matrix-fibroblast equation and hence $\tau > 0$. Because of the central role of the cell traction, we shall use τ as the bifurcation parameter, while varying other parameters such as b and γ . Experimentally, it is also known that, in vitro, the traction generated by the cell increases with time for a limited period and hence τ would be the ideal parameter to study bifurcation. The expressions for λ , $b(k)$ and $c(k)$, determine the domains in parameter space where spatially inhomogeneous linearly unstable solution exist. They also give the bifurcation surfaces in parameter space, that is, the surface which separates homogeneous from inhomogeneous solutions. In general it is algebraically complicated to determine these

surfaces. However, because of the dimensionality of the parameter space, it would be of little conceptual help in understanding the basic features of forming striae. We could make such kinds of analysis relatively simple by neglecting one or more factors affecting fibroblast motion and ECM deformation. However, as already mentioned previously, we have as such included only the essential factors that would form a realistic model. Simpler models are also capable of generating spatial models but we are interested in the one that will solve the problem of striae to a feasible extent and hence we will not venture much into analyzing the domain surface spaces. With the polynomial complexity of $b(k)$ and $c(k)$ in the dispersion relation, we can look forward to complex linear growth behavior.

2.1 Mode selection and parameter values

In order to carry out a detailed analysis of the mode selection and determination of the parameter values, we plot the zeros of the characteristic equation (2.8). It must be observed that equation (2.8) has two roots given the \pm sign as shown in equation (2.12). Therefore it is important to consider each case separately. However, here is a proof that shows that the negative root is not appropriate for our case study. From previous discussions, it is evident that we are interested in solutions where $\text{Re}(\lambda(k)) > 0$, since these are the ones that exhibit unstable modes and hence generate pattern. From Descartes's rule of sign, we also inferred that $b(k)$ and $c(k)$ cannot be simultaneously positive, since in this case there would be no positive real roots to the dispersion relation. Hence consider the roots of the characteristic equation,

$$2 a(k) \lambda_{1,2} = -b(k) \pm \sqrt{b(k)^2 - 4a(k)c(k)}. \quad (2.13)$$

The terms under the square root are positive. If $b(k)$ is positive, then $-b(k) - \sqrt{b(k)^2 - 4a(k)c(k)}$ is negative and $\text{Re}(\lambda(k)) < 0$ which is of no use to us since it implies that only spatially homogeneous solutions are stable. If $b(k)$ is negative and $c(k)$ is also negative, $-4a(k)c(k)$ is positive and hence we once again arrive at the same conclusion with $\text{Re}(\lambda(k)) < 0$. If $b(k)$ is negative and $c(k)$ is positive, we will use the fact that the characteristic equation yields to the dispersion relation leading to pattern formation only for the solution with the largest $\text{Re}(\lambda(k)) \geq 0$ [13]. In the case when $b(k)$ is negative and $c(k)$ positive, the positive root would most definitely lead to the largest positive value for the $\text{Re}(\lambda(k))$. Hence, the negative root does not serve the purpose for pattern formation in our case and we shall not consider it. The polynomial corresponding to the positive root can be plotted easily in MATLAB and one can then look at possible wave numbers $k_n^2 = n^2 \pi^2$, since the time independent solution of the spatial eigenvalue problem is proportional to $\cos\left(\frac{n\pi x}{L}\right)$ which satisfies zero-flux boundary conditions at $x = 0$ and $x = 1$. The eigenvalue in this case is $k = n\pi$ and $1/k = 1/(n\pi)$ is a measure of the wave like pattern and the eigenvalue k is the wave number. On finite domains there is a discrete set of possible wave numbers [11]. To study instability we look for $\text{Re}(\lambda(k_n^2)) > 0$ for $k_-^2 < k_n^2 < k_+^2$ where k_-^2 and k_+^2 are the roots of the polynomial equations in some finite interval. Note that the order of the polynomials obtained is greater than two and there are possibilities of having more than two real roots to the polynomials. We however rely on MATLAB to consider possible roots that can be isolated, whatever their number may be.

Positive root

Let us consider first the equation

$$2a(k)\lambda_{1,2} = -b(k) + \sqrt{b(k)^2 - 4a(k)c(k)} = 0. \quad (2.14)$$

At the outset, we fix all the parameter values other than τ as follows

$$d = 10^{-2.5}, a = 10^{-0.5}, b = 0.01, \gamma = 0.01, \delta = 1 \text{ and } h = 0.1.$$

It can be shown that the critical value of τ for which $\text{Re}(\lambda(k^2)) > 0$ i.e the region where instability starts occurring, is given by $\tau \approx 0.4291$, the critical bifurcation point (Appendix A). By increasing the value of τ beyond this value we obtain the results illustrated in Table 2.1 and Figure 2.1 respectively. Also we can see that when the value of $\tau \geq 16$, we can isolate wave numbers of the form $n\pi$ but are not able to go beyond $n = 1$. Increasing τ does not seem to have much effect as the upper bound seems to saturate very close to the value of π . Besides τ signifies the cell traction force and it is usually less than 1 in normal skin, around 0.01 [11], so increasing τ would have no meaning physically. We have included very high values for τ in order to see the increase in regions of instability. In order for the system to exhibit spatially inhomogeneous solutions, τ must be greater than 16 but this has no physical meaning. Non-uniform solutions cannot be observed for $\tau \leq 16$.

Clearly, as expected, the isolated values of k do not exponentially blow up and heuristically we can conclude that they are bounded by the square root term. The fact to be noted here is that a bifurcation to instability occurs at $\tau \approx 0.4291$. Increasing τ corresponds

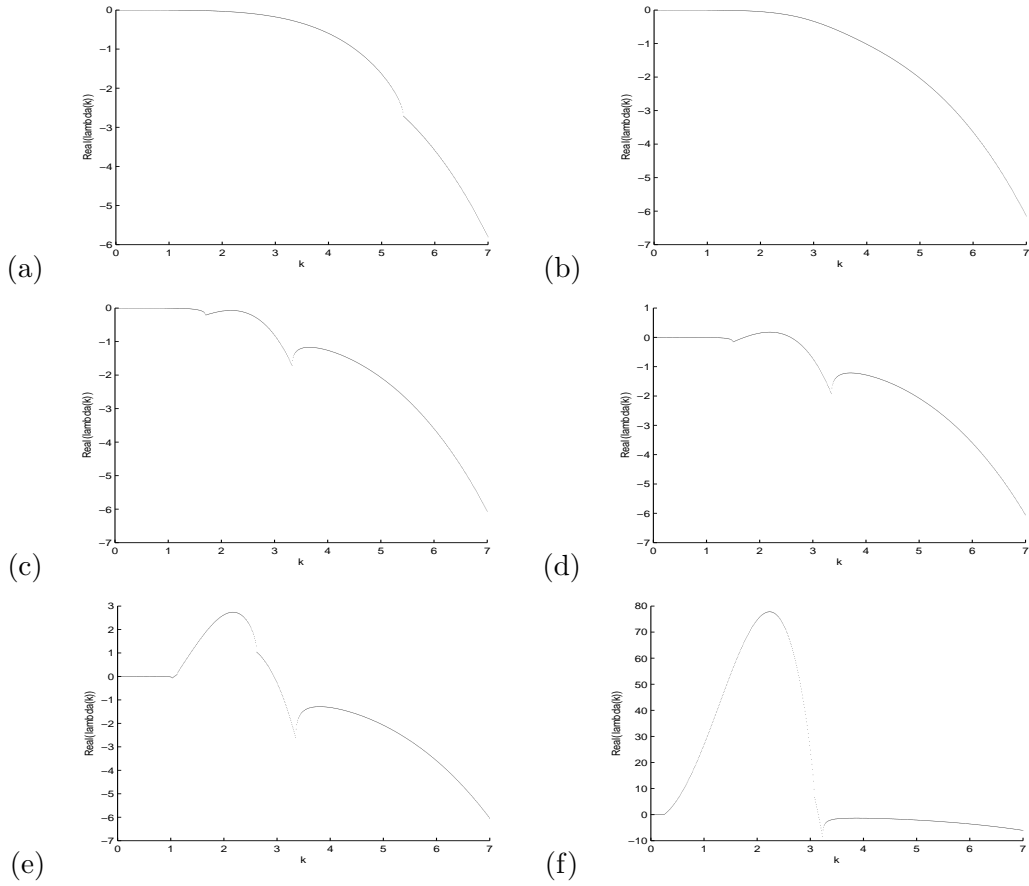


Figure 2.1: Dispersion relation for the positive root by fixing all the parameter values other than τ as listed in the text with $b=0.01$, $\gamma = 0.01$ and varying τ : (a) $\tau = 0.01$, (b) $\tau = 0.1$, (c) $\tau = 0.4$, (d) $\tau = 0.5$, (e) $\tau = 1$, (f) $\tau = 16$.

Value of τ	Region of instability	Wave number of form $n\pi$ isolated
0.01	None	None
0.075	None	None
0.1	None	None
0.4	None	None
0.429	None	None
0.4291	[2.18,2.19]	None
1.00	[1.07,2.94]	None
15.0	[0.26,3.14]	None
16.0	[0.26,3.15]	π
50.0	[0.15,3.15]	π
100.0	[0.11,3.15]	π

Table 2.1: Regions of instability for the linearized system obtained for increasing values of τ and the wave numbers isolated.

to an increase in the traction force exerted by fibroblasts on the ECM, and with all other parameters fixed as above, we find that the threshold value of $\tau \approx 0.4291$.

Now take $b = 0.12$ and keep all the parameters fixed other than τ . Repeat the same exercise as above to find the threshold value of τ for onset of instability. The results are generated and shown in Table 2.2 and Figure 2.2 respectively. The onset of instability for these parameter values occurs at the threshold value of $\tau \approx 0.624$. The value of τ where we can isolate the wave number π is around $\tau \approx 29$ and hence for values of $\tau \leq 29$ non-uniform spatially inhomogeneous solutions cannot be observed (Table 2.2 and Figure 2.2). Increasing b corresponds to increase in Young's modulus of fibroblasts.

Similarly for $b = 1$, we apply the same procedure. In Table 2.3, we vary b , γ and search for a region of τ to isolate wave numbers. It turns out that only π can be isolated as before (see Figure 4.1 for the plot of the dispersion relation). The notable fact here is that a change in the value of γ does not affect the threshold value of τ to a great extent. This indicates

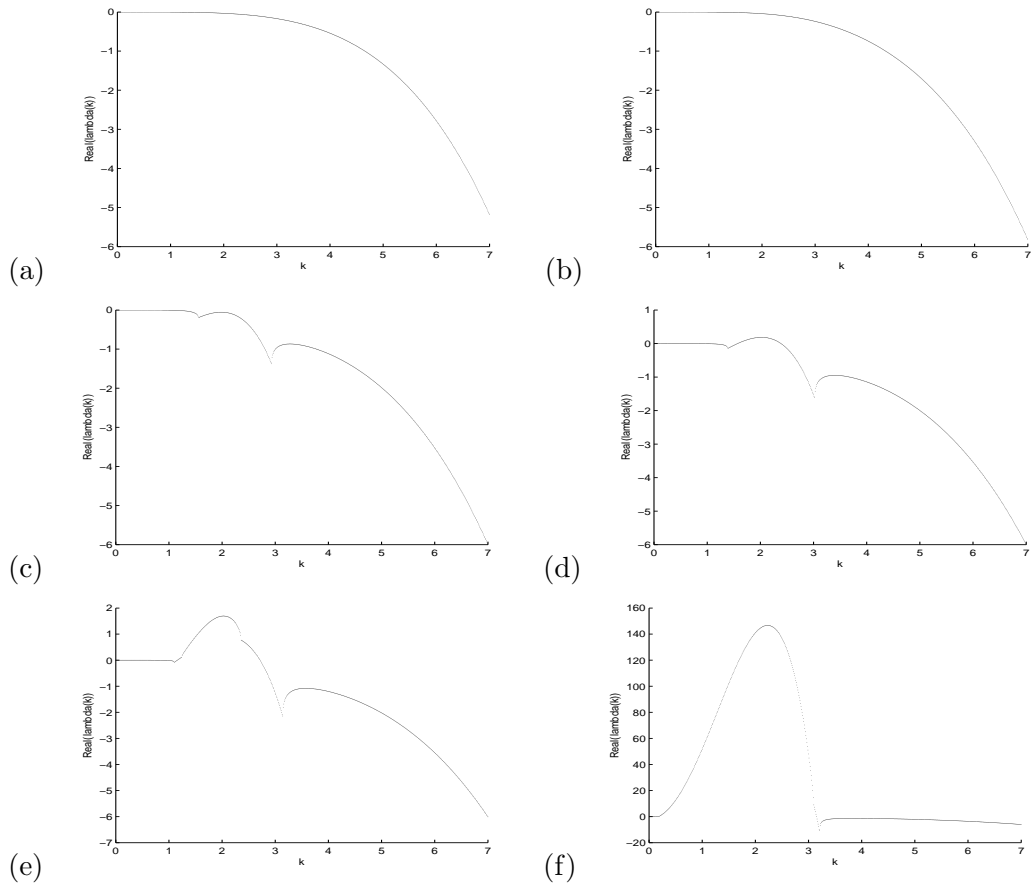


Figure 2.2: Dispersion relation for the positive root by fixing all the parameter values other than τ as shown in the text with $b = 0.12$, $\gamma = 0.01$ and varying τ : (a) $\tau = 0.01$, (b) $\tau = 0.1$, (c) $\tau = 0.6$, (d) $\tau = 0.7$, (e) $\tau = 1$, (f) $\tau = 30$, and (g) $\tau = 50$.

Value of τ	Region of instability	Wave number of form $n\pi$ isolated
0.01	None	None
0.075	None	None
0.1	None	None
0.4	None	None
0.623	None	None
0.624	[1.96,2.03]	None
1.00	[1.16,2.71]	None
29.0	[0.19,3.14]	None
30.0	[0.19,3.15]	π
50.0	[0.15,3.15]	π
100.0	[0.11,3.15]	π

Table 2.2: Regions of instability for the linearised system obtained for increasing values of τ and the wave numbers isolated.

that the factor contributing to the random re-orientation of the fibroblasts does not have a significant effect on the formation of striae, since γ is related to this factor. The values of γ chosen, $\gamma = 0.01$ and $\gamma = 0.05$, are typical values as mentioned in [19]. Combining all the results obtained above, we can confidently say that increasing τ beyond the respective threshold values leads to an increase in the traction force exerted by the fibroblasts on the ECM and hence would produce stable spatially inhomogeneous solutions which in turn produce patterns as observed in striae. These findings will be validated through numerical solutions as illustrated in Chapter 3.

Having dealt with the mechanical aspect for modeling the pathogenesis of striae, we now try to reason out the experimental result of corticosteroid excess leading to striae. For this, we fix $\tau = 0.05$, since we know the onset of instability occurs around this value and take γ to be 0.01. We then vary $b = 1, 0.12, 0.04, 0.01$ as shown in Table 2.4 and Figure 2.3 respectively. Decreasing b from 1, 0.12, 0.05 and to 0.01 leads to a bifurcation to instability

Value of b	Value of γ	Bifurcation Value of τ	Wave number isolation value of τ
0.01	0.01	0.43	30
0.12	0.01	0.63	30
1	0.01	1.87	145
0.01	0.05	0.43	16
0.12	0.05	0.63	30
1	0.05	1.87	147

Table 2.3: Regions of τ when b and γ are varied.

Value of b	Region of instability	Wave number of form $n\pi$ isolated
1	None	None
0.12	None	None
0.04	[1.72,2.58]	None
0.001	[1.68,2.74]	None

Table 2.4: Regions of instability for the linearised system obtained for decreasing values of b and the wave numbers isolated.

at $b = 0.04$ and larger instability at $b = 0.01$. The spatially homogeneous solution is the only stable solution for $b = 1$ and 0.12 . Hence decreasing b leads to instability with the threshold value of $b = 0.05$. This implies a decrease in Young's modulus of the fibroblast cells, which in turn occurs during corticosteroid excess. Now, all that remains is to validate predictions obtained from the linear stability theory through appropriate numerical schemes in order to complete the framework for the pathogenesis of striae.

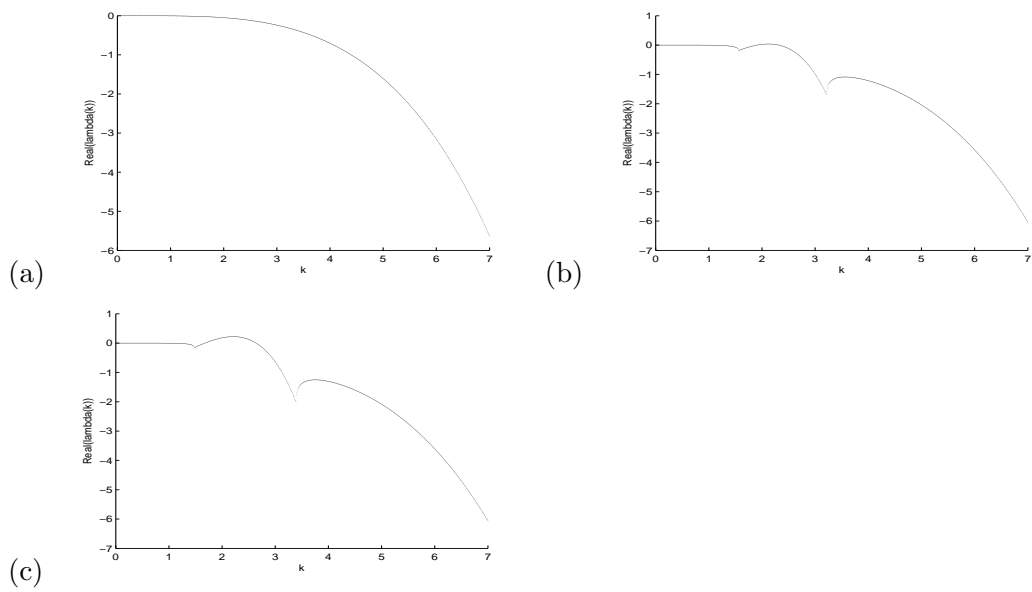


Figure 2.3: Dispersion relation for the positive root by fixing the parameter values other than b with as listed in the text with $\tau = 0.5$, $\gamma = 0.01$ and taking (a) $b = 1$, (b) $b = 0.04$, (c) and $b = 0.001$ to illustrate the effect that a decrease in b implies an increase in corticosteroid excess thereby giving rising to the formation of striae.

CHAPTER 3
FINITE DIFFERENCE SCHEMES

3.1 Introduction

Having shown the existence of confined sets within the positive quadrant and having isolated regions of instability, and also the corresponding set of threshold values for τ thereon, we have now got an indication of what set of parameter values would lead to diffusion-driven instability. Hence the linearly unstable eigenfunctions which are growing exponentially with time will be eventually bounded by the non linear terms of the reaction diffusion system of equations and we can ultimately expect to obtain stable spatially inhomogeneous steady state set of solutions. From Smoller's result, since there exists a confined set in the absence of diffusion, it would also exist in the presence of diffusion. The aim of this chapter is to find out these solutions numerically for different set of parameter values.

Due to the complexity of the model equations derived in (1.24)-(1.26), it is only prudent to look for numerical approximate solutions. We will use finite difference schemes as these are simpler to implement and are known to handle boundary conditions a lot easier than finite elements and spectral methods [10]. In order to efficiently implement numerical methods to the model equations (1.24)-(1.26), it is convenient to first assign v to be the velocity

$$\frac{\partial u}{\partial t} = v \tag{3.1}$$

and therefore the equations can be re-written conveniently as

$$n_t = dn_{xx} + d\gamma (n_{xx}u_x + n_x u_{xx}) - (nv)_x, \quad (3.2)$$

$$s_t = -s_x v - s v_x, \quad (3.3)$$

$$\begin{aligned} & a v_{xx} + b u_{xx} + \tau (n s_x + s n_x + h n_x s_{xx} + h n s_{xxx}) \\ & + \tau \delta (n s_x u_x + s n_x u_x + s n u_{xx} + h n_x u_x s_{xx} + h u_{xx} s_{xx} + h u_x s_{xxx}) = s u. \end{aligned} \quad (3.4)$$

3.1.1 Derivation of finite difference schemes

To implement the finite difference schemes, approximate finite difference stencils to the derivatives have to be derived. These are derived from Taylor series expansions as illustrated below [10]. In all our derivations, it is enough to derive only the following expansions

$$\omega(x + \delta x, t) = \omega(x, t) + \frac{\delta x}{1!}\omega_x(x, t) + \frac{\delta x^2}{2!}\omega_{xx}(x, t) + \frac{\delta x^3}{3!}\omega_{xxx}(x, t) + O(\delta x^4) \quad (3.5)$$

$$\omega(x - \delta x, t) = \omega(x, t) - \frac{\delta x}{1!}\omega_x(x, t) + \frac{\delta x^2}{2!}\omega_{xx}(x, t) - \frac{\delta x^3}{3!}\omega_{xxx}(x, t) + O(\delta x^4) \quad (3.6)$$

$$\omega(x + 2\delta x, t) = \omega(x, t) + 2\frac{\delta x}{1!}\omega_x(x, t) + \frac{(2\delta x)^2}{2!}\omega_{xx}(x, t) + \frac{(2\delta x)^3}{3!}\omega_{xxx}(x, t) + O(\delta x^4) \quad (3.7)$$

$$\omega(x - 2\delta x, t) = \omega(x, t) - 2\frac{\delta x}{1!}\omega_x(x, t) + \frac{(2\delta x)^2}{2!}\omega_{xx}(x, t) - \frac{(2\delta x)^3}{3!}\omega_{xxx}(x, t) + O(\delta x^4) \quad (3.8)$$

that will be used to derive all the approximate finite difference stencils. We first derive the forward, backward and centred finite differences

$$\omega_x = \frac{\omega(x + \delta x, t) - \omega(x, t)}{\delta x} + O(\delta x), \quad (3.9)$$

$$\omega_x = \frac{\omega(x, t) - \omega(x - \delta x, t)}{\delta x} + O(\delta x), \quad (3.10)$$

$$\omega_x = \frac{\omega(x + \delta x, t) - \omega(x - \delta x, t)}{2\delta x} + O(\delta x^2). \quad (3.11)$$

The forward and backward finite differences are only first order accurate while the centred finite difference is second order accurate and hence yields a better accuracy. Similarly, we derive the rest of the necessary finite difference approximations focussing on minimizing the error growth and increasing the order of approximation. It can be shown that

$$\begin{aligned} [\omega(x + \delta x, t) + \omega(x, t)][z(x + \delta x, t) + z(x, t)] &= 4\omega(x, t)z(x, t) + 2\delta x(\omega z_x + z\omega_x) \\ &+ \delta x^2(\omega z_{xx} + \omega_x z_x + z\omega_{xx}) + O(\delta x^3), \end{aligned} \quad (3.12)$$

and

$$\begin{aligned} [\omega(x - \delta x, t) + \omega(x, t)][z(x - \delta x, t) + z(x, t)] &= 4\omega(x, t)z(x, t) - 2\delta x(\omega z_x + z\omega_x) \\ &+ \delta x^2(\omega z_{xx} + \omega_x z_x + z\omega_{xx}) - O(\delta x^3). \end{aligned} \quad (3.13)$$

Subtracting (3.13) from (3.12) we obtain the finite difference approximation to the derivative

$$\begin{aligned} \frac{\partial(\omega z)}{\partial x} = \omega z_x + z\omega_x &= \frac{[\omega(x + \delta x, t) + \omega(x, t)][z(x + \delta x, t) + z(x, t)]}{4\delta x} \\ &- \frac{[\omega(x - \delta x, t) + \omega(x, t)][z(x - \delta x, t) + z(x, t)]}{4\delta x} + O(\delta x^2). \end{aligned} \quad (3.14)$$

The second derivative of ω can be easily obtained as

$$\omega_{xx} = \frac{\omega(x + \delta x, t) - 2\omega(x, t) + \omega(x - \delta x, t)}{(\delta x)^2} + O(\delta x^2). \quad (3.15)$$

It can be shown that a second order accurate finite difference approximation to the third derivative can be derived by observing that

$$\begin{aligned} & \omega(x + 2\delta x, t) - 2\omega(x + \delta x, t) + 2\omega(x - \delta x, t) - \omega(x - 2\delta x, t) \\ &= \left(\frac{8}{3!} - \frac{2}{3!} - \frac{2}{3!} + \frac{8}{3!}\right)\delta x^3\omega_{xxx} + O(\delta x^5), \end{aligned} \quad (3.16)$$

which when re-arranged conveniently yields

$$\omega_{xxx} = \frac{\omega(x + 2\delta x, t) - 2\omega(x + \delta x, t) + 2\omega(x - \delta x, t) - \omega(x - 2\delta x, t)}{2(\delta x)^3} + O(\delta x^2). \quad (3.17)$$

Similar derivations can be obtained for the time-derivatives and in particular, it can be easily shown that a second order finite difference scheme can be derived for the time-derivative of the form

$$\omega_t = \frac{3\omega(x, t + \delta t) - 4\omega(x, t) + \omega(x, t - \delta t)}{2\delta t} + O(\delta t^2). \quad (3.18)$$

Having developed these preliminary results, we are now in a position to apply these finite difference approximations to our model equations. Also note that we will take only those finite difference approximations with at least second order accuracy. It must be noted that taking finite difference approximations of higher order accuracies leads to long computational times due to the restrictions of the time-step Δt and this has not been possible to implement on the machine we were working with. Equally important, we will not use first order finite difference approximations simply because they fail to converge for our model problems.

3.1.2 Equation (A): $n(x, t)$

Consider Equation 3.2

$$n_t = d n_{xx} + d\gamma (n_{xx} u_x + n_x u_{xx}) - (n v)_x.$$

We consider two finite difference schemes for equation (A), both second order accurate in time and space. The first scheme is the well-known IMEX scheme (implicit-explicit [9, 18]): the second order finite backward differentiation formula (2-SBDF) and the second scheme is a modified version of the first scheme. We will denote this scheme: the second order almost fully implicit finite differentiation formula (2-SAFBDF). Implicit-explicit (IMEX) schemes use an implicit scheme to approximate the diffusion term and an explicit scheme to approximate the reaction terms. Such schemes have been shown to be more stable and more accurate than most schemes for reaction-diffusion systems [9, 18]. First order accurate finite difference schemes blow-up for the choice of any time-steps and space-steps around 10^{-2} and 10^{-4} and even lower (results not shown) and hence we will not discuss them in this thesis. These two feasible schemes will be devised, both of which do not allow numerical error magnification. We discuss these two methods in detail for our particular model equation below and compare the results obtained from the two schemes later on. For the derivations below, it is important to introduce appropriate notations. We assume that the one-dimensional mesh connectivity covering the interval $\Omega = [0, 1]$ is a uniform mesh of n elements such that

$$0 = x_0 < x_1 < \cdots < x_{n-1} < x_n = 1$$

where for each i , $i = 1, \dots, n$, $x_i = x_{i-1} + \delta x$, where $\delta x = \frac{1}{n}$. An unstructured mesh can also be used. For the time $t \in [0, T]$, $T > 0$, let δt be a typical time-step such that $t_q = t_{q-1} + \delta t$, with $q = 1, 2, \dots$ and $t_0 = 0$. Therefore we will denote by $\omega_j^q = \omega(x_j, t_q)$ where the point (x_j, t_q) , ($j = 0, 1, \dots, n$ and $q = 0, 1, \dots$) coincides with the finite difference grid points. Also, the subscripts j and superscripts q indicate the space-step and time-step respectively. For example, $\omega(x + \delta x, t - \delta t)$ will be denoted as $\omega_{j+1}^{q-1} = \omega(x_{j+1}, t_{q-1})$.

The second order finite backward differentiation formula: 2-SBDF

This scheme has been referenced from [9, 18]. The 2-SBDF scheme can be shown to be given by (see [9] for specific details)

$$\frac{3n_j^{q+1} - 4n_j^q + n_j^{q-1}}{2\delta t} = d \frac{n_{j+1}^{q+1} - 2n_j^{q+1} + n_{j-1}^{q+1}}{\delta x^2} + 2F(n_j^q, u_j^q, v_j^q) - F(n_j^{q-1}, u_j^{q-1}, v_j^{q-1})$$

where

$$\begin{aligned} F(n_j^q, u_j^q, v_j^q) &= \frac{d\gamma}{2\delta x^3} (n_{j+1}^q - 2n_j^q + n_{j-1}^q)(u_{j+1}^q - u_{j-1}^q) \\ &+ \frac{d\gamma}{2\delta x^3} (n_{j+1}^q - n_{j-1}^q)(u_{j+1}^q - 2u_j^q + u_{j-1}^q) \\ &- \frac{1}{4\delta x} [(n_{j+1}^q + n_j^q)(v_{j+1}^q + v_j^q) - (n_j^q + n_{j-1}^q)(v_j^q + v_{j-1}^q)]. \end{aligned} \quad (3.19)$$

Notice that this scheme involves calculating the values of the variable $n(x, t)$ at time t_{q+1} given the values of the solutions at the previous two time-levels t_q and t_{q-1} , requiring us to keep in memory, solutions at two different time-levels. Define $\lambda_1 = 2d \frac{\delta t}{\delta x^2}$. Re-arranging

terms, we get

$$\begin{aligned}
-\lambda_1 n_{j-1}^{q+1} + (3 + 2\lambda_1) n_j^{q+1} - \lambda_1 n_{j+1}^{q+1} = & 4n_j^q - n_j^{q-1} + 4\delta t F(n_j^q, u_j^q, v_j^q) \\
& - 2\delta t F(n_j^{q-1}, u_j^{q-1}, v_j^{q-1}). \tag{3.20}
\end{aligned}$$

Hence (3.20) can be written as

$$A_j n_{j-1}^{q+1} + B_j n_j^{q+1} + C_j n_{j+1}^{q+1} = D_j \tag{3.21}$$

where $j = 0, 1, \dots, n$ and $q = 0, 1, \dots$ and

$$A_j = -\lambda_1, \quad B_j = (3 + 2\lambda_1), \quad C_j = -\lambda_1,$$

and the right-hand side is defined by

$$D_j = 4n_j^q - n_j^{q-1} + 2\delta t \left(2F(n_j^q, u_j^q, v_j^q) - F(n_j^{q-1}, u_j^{q-1}, v_j^{q-1}) \right)$$

We impose zero-flux boundary conditions on the system. The coefficients A_j, B_j, C_j however, are not functions of the solution values. Equation (3.21) gives rise to a tridiagonal system with the coefficients satisfying the diagonally dominant criteria

$$|B_j| > |A_j| + |C_j|, \quad j = 0, 1, \dots, n.$$

Hence the system is in perfect structure to be solved efficiently using the Thomas algorithm [10]. Therefore when $j = 0$ for the first row of the tridiagonal matrix, C_0 is replaced by

$A_0 + C_0$ and similarly when $j = m$ for the last row of the tridiagonal matrix A_m is replaced by $A_m + C_m$ for zero flux boundary conditions. We next derive the second scheme based on modifying the 2-SBDF scheme such that A_j, B_j, C_j are functions of the previous solution values.

**The second order almost fully implicit finite backward differentiation formula:
2-SAFBDF**

The second scheme is an almost fully implicit scheme. The motive behind this is to construct a scheme that exploits the implicitness from every single term in equation (B) but at the same time, requiring at least second order accuracy both in time and space. This idea was originally proposed and implemented in papers by Madzvamuse [9, 8]. The second order scheme is used to approximate the time derivative (3.18) as shown in the previous section. Similarly the second order spatial derivative is approximated implicitly using a second order finite difference scheme. The remaining derivative terms are then approximated by finite difference schemes such that only at least second or higher order accurate schemes are used. The 2-SAFBDF is therefore derived as

$$\begin{aligned} \frac{3n_j^{q+1} - 4n_j^q + n_j^{q-1}}{2\delta t} &= d\left(\frac{n_{j+1}^{q+1} - 2n_j^{q+1} + n_{j-1}^{q+1}}{\delta x^2} + d\gamma\left(\frac{u_{j+1}^q - u_{j-1}^q}{2\delta x}\right)\left(\frac{n_{j+1}^{q+1} - 2n_j^{q+1} + n_{j-1}^{q+1}}{\delta x^2}\right)\right) \\ &+ d\gamma\left(\frac{u_{j+1}^q - 2u_j^q + u_{j-1}^q}{\delta x^2}\right)\left(\frac{n_{j+1}^{q+1} - n_{j-1}^{q+1}}{2\delta x}\right) \\ &- \frac{1}{4\delta x}\left[(n_{j+1}^{q+1} + n_j^{q+1})(v_{j+1}^q + v_j^q) - (n_j^{q+1} + n_{j-1}^{q+1})(v_j^q + v_{j-1}^q)\right] \quad (3.22) \end{aligned}$$

Observe that fully implicit schemes are used to approximate spatial derivatives as illustrated on the right-hand side. It is important, as observed from numerous experiments,

to first multiply by $(\delta x)^2$ both sides. This is to balance the terms ranging from $O(\delta x)$ to $O(\delta x)^3$. Experience has shown us that any other way results in a highly unbalanced system. Therefore multiplying both sides by $(\delta x)^2$ and re-arranging terms we obtain a tri-diagonal system of the form

$$A_j n_{j-1}^{q+1} + B_j n_j^{q+1} + C_j n_{j+1}^{q+1} = D_j \quad (3.23)$$

where

$$\begin{aligned} A_j &= - \left[d + d\gamma \frac{u_{j+1}^q - u_{j-1}^q}{2\delta x} - d\gamma \frac{u_{j+1}^q - 2u_j^q + u_{j-1}^q}{2\delta x} + \frac{\delta x}{4} (v_j^q + v_{j-1}^q) \right], \\ B_j &= \frac{3\delta x^2}{2\delta t} + 2d + d\gamma \frac{u_{j+1}^q - u_{j-1}^q}{\delta x} + \frac{\delta x}{4} (v_{j+1}^q - v_{j-1}^q), \\ C_j &= - \left[d + d\gamma \frac{u_{j+1}^q - u_{j-1}^q}{2\delta x} + d\gamma \frac{u_{j+1}^q - 2u_j^q + u_{j-1}^q}{2\delta x} - \frac{\delta x}{4} (v_{j+1}^q + v_j^q) \right], \end{aligned}$$

and the right-hand side is given by

$$D_j = (4n_j^q - n_j^{q-1}) \frac{\delta x^2}{2\delta t}. \quad (3.24)$$

Clearly, now A_j , B_j and C_j are functions of the solution values at the previous time t_q . This is the key advantage of this scheme. The Thomas algorithm is then used, just as before, taking into account the boundary conditions by replacing C_0 by $A_0 + C_0$ when $j = 0$ and A_m by $A_m + C_m$ when $j = m$ to give numerical approximate solutions. However, this equation does not exist independently and is influenced at every time step by the other two equations and we impose the same kind of boundary conditions for other equations also. Details and computer implementation of the Thomas algorithm are given in Appendix B.

3.1.3 Equation (B): $s(x, t)$

Let us consider equation (B) that models the ECM given by (3.3)

$$s_t = -s_x u_t - s u_{tx} = -\frac{\partial(sv)}{\partial x}.$$

We use the results derived in Section 3.1.1 to approximate the derivatives as

$$\frac{s_j^{q+1} - s_j^q}{\delta t} = -v_j^q \left(\frac{s_{j+1}^{q+1} - s_j^{q+1}}{\delta x} \right) - s_j^{q+1} \left(\frac{v_{j+1}^q - v_{j-1}^q}{2\delta x} \right) \quad (3.25)$$

if v_j is positive and

$$\frac{s_j^{q+1} - s_j^q}{\delta t} = -v_j^q \left(\frac{s_j^{q+1} - s_{j-1}^{q+1}}{\delta x} \right) - s_j^{q+1} \left(\frac{v_{j+1}^q - v_{j-1}^q}{2\delta x} \right) \quad (3.26)$$

if v_j is negative. The above schemes depend on whether v_j is positive or negative and depending on the case, an equivalent tri-diagonal system is formed. Since the v 's are updated at every stage due to dependencies on the other equations, it is important that if v_j is positive, the solver automatically chooses the finite difference scheme. Re-arranging the finite difference schemes (3.25) and (3.26) results in tri-diagonal systems whose coefficients are given below depending on the sign of v_j .

- If $v_j > 0$ then

$$A_j = 0, \quad B_j = 1 + \lambda_2(v_{j+1} - v_{j-1}) - 2\lambda_2 v_j, \quad C_j = 2\lambda_2 v_j,$$

where $\lambda_2 = \frac{\delta t}{2\delta x}$.

- If $v_j < 0$ then

$$A_j = -2\lambda_2 v_j, \quad B_j = 1 + \lambda_2(v_{j+1} - v_j - 1) + 2\lambda_2 v_j, \quad C_j = 0,$$

for $j = 0, 1, \dots, n$. Note that the right-hand side is $D_j = 0$ for both schemes.

Hence we can use the same tri-diagonal solver as explained in the Appendix B with $C_0 = -2\lambda_2 v(j)$ when $v(j) < 0$ and $j = 0$ and $A_m = 2\lambda_2 v(j)$ when $v(j) > 0$ and $j = m$ to impose the zero-flux boundary conditions.

3.1.4 Equation (C): $v(x, t)$

Consider the finite difference scheme for equation (C) given by (3.4).

$$\begin{aligned} & a v_{xx} + b u_{xxx} + \tau (n s_x + s n_x + h n_x s_{xx} + h n s_{xxx}) \\ & + \tau \delta (n s_x u_x + s n_x u_x + s n u_{xx} + h s n_x u_x s_{xx} + h n u_{xx} s_{xx} + h n u_x s_{xxx}) = s u. \end{aligned} \quad (3.27)$$

Since we are only interested in the solution values of $v(x, t)$ from this equation, a fully implicit scheme is used to approximate the spatial derivative. There is no time-derivative explicit in this equation. Therefore we use a centred finite difference scheme for v_{xx} at time t_{q+1} and compute the rest of the derivatives at the previous time t_q . It is crucially important also to balance the δx terms. This can be achieved by simply multiplying by $(\delta x)^2$ throughout the scheme to obtain a tri-diagonal system whose coefficients are

$$A_j = -a, \quad B_j = 2a, \quad C_j = -a,$$

and whose right-hand side is given by

$$\begin{aligned}
D_j = & b(u_{j+1} - 2u_j + u_{j-1}) \\
& + \frac{\tau\delta x}{4} \left[(s_{j+1} + s_j)(n_{j+1} + n_j) - (s_j + s_{j-1})(n_j + n_{j-1}) \right] \\
& + \frac{\tau h}{2\delta x} (n_{j+1} - n_{j-1})(s_{j+1} - 2s_j + s_{j-1}) \\
& + \frac{\tau h}{2\delta x} n_j (s_{j+2} - 2s_{j+1} + 2s_{j-1} - s_{j-2}) \\
& + \frac{\delta\tau}{8} (u_{j+1} - u_{j-1}) \left[(s_{j+1} + s_j)(n_{j+1} + n_j) - (s_j + s_{j-1})(n_j + n_{j-1}) \right] \\
& + \delta\tau n_i s_j (u_{j+1} - 2u_j + u_{j-1}) \\
& + \frac{\delta\tau h}{4(\delta x)^2} (u_{j+1} - u_{j-1})(n_{j+1} - n_{j-1})(s_{j+1} - 2s_j + s_{j-1}) \\
& + \frac{\delta\tau h}{(\delta x)^2} n_j (u_{j+1} - 2u_j + u_{j-1})(s_{j+1} - 2s_j + s_{j-1}) \\
& + \frac{\delta\tau h}{4(\delta x)^2} n_j (u_{j+1} - u_{j-1})(s_{j+2} - 2s_{j+1} + 2s_{j-1} - s_{j-2}) \\
& - s_i u_j \delta x^2.
\end{aligned}$$

The boundary conditions are implemented by replacing C_j by $A_j + C_j$ when $j = 0$ and A_j by $A_j + C_j$ when $j = m$. Since $|B_j| = |A_j| + |C_j|$, a small perturbation is added to B_j to make the matrix diagonally dominant so that the same Thomas algorithm solver can be used to solve each of the tri-diagonal systems. The algorithm is based on Gaussian forward and backward elimination [10]. It is very efficient because to solve per mesh point, it needs $3(\text{add}) + 3(\text{multiply}) + 2(\text{divide})$ operations.

3.1.5 Updating $u(x, t)$

Once the main equations have been dealt with, we update the value of u_j since the third equation has given solution values for v_j which have been defined to simplify computations. A first order finite difference scheme in time applied to the equation

$$v = \frac{\partial u}{\partial t}$$

gives the updated scheme

$$u_j^{q+1} = u_j^q + \delta t v_j^q, \quad j = 0, 1, \dots, n. \quad (3.28)$$

Initial and boundary conditions

We take initial conditions as small random perturbations around the uniform steady state. At time $t = 0$, we take

$$n_j = 1 + \frac{0.50d0 - \epsilon}{100} \quad s_j = 1, \quad v_j = 0, \quad u_j = 0, \quad \text{with } \epsilon = \text{random.}$$

We prescribe zero-flux boundary conditions, for example, for $u(x, t)$ these are of the form

$$u(-1, t) = u(1, t),$$
$$u(m + 1, t) = u(m - 1, t)$$

for all $t > 0$. Similar boundary conditions are applied to n , s and v .

3.1.6 Computer implementation

We describe the implementation of the algorithm to solve the model equations here.

1. We are interested in global model parameter values (in double precision) given by

$$\left\{ \begin{array}{l} d = 10^{-2.5}, \quad a = 10^{-0.5}, \quad \delta = 1, \quad h = 0.1, \\ b = 0.01, \quad 0.12, \quad 1, \\ \tau = 0.01, \quad 0.075, \quad 0.1, \quad 0.45, \quad 1, \quad 2, \\ \gamma = 0.01, \quad 0.05. \end{array} \right.$$

2. Global solution variables are n , s , v and u . Also dummy global variables are required to store solution values at times t_q and t_{q-1} . We define these as nn , sn , vn and un .
3. We also need to define global numerical parameter values: dt , dx and $length$.
4. Results are stored in the four files: `nfinsol.dat` ($n(x, t)$), `sfinsol.dat` ($s(x, t)$), `ufinsol.dat` ($u(x, t)$) and `x.dat` (x_j mesh points). We also store the solution values at each time step in the files: `wnsol.m`, `wssol.m` and `wusol.m` which correspond to solution values n , s and u . These files are huge arrays of size $(m \times n)$ where $m = \frac{T}{\delta t}$, and $n = \frac{L}{\delta x}$ where T is the final time and $L = 1$ is the length of the computational interval. These results are used in computing the error decay and the transient solutions.
5. The two sets of data are written to their respective opened files, they are formatted to the needed level of accuracy and they are closed once all the iterations are completed.

6. An error control is made by observing the convergence of the errors by calculating $\|e\| = \sqrt{\sum (u_i^q - u_i^{q-1})^2}$ where u_i^q and u_i^{q-1} are two successive numerical solutions at times t_{q-1} and t_q .
7. Each set of data values are plotted using MATLAB and graphs of the solution states at the final time and the transient solution states are obtained. The numerical error graph is also plotted from the files with the iterative solutions (wnsol.m wssol.m and wusol.m)
8. The time-step and the space-steps are then varied and some parameter values are changed (based on the results obtained from linear stability theory) and different sets of results are then presented.

CHAPTER 4

RESULTS, CONCLUSIONS AND FUTURE DIRECTIONS

4.1 Numerical results

4.1.1 The 2-SBDF scheme

We first compile the results obtained by solving the set of equations using the 2-SBDF scheme for equation (A). Set $\delta t = 10^{-4}$ and $\delta x = 0.5 * 10^{-2}$ and fix all the parameter values other than τ as follows

$$d = 10^{-2.5}, a = 10^{-0.5}, b = 0.01, \gamma = 0.01, \delta = 1, h = 0.1.$$

Let us take $\tau = 0.01$ and 0.7 . The corresponding results of the numerical simulations are illustrated in Figures 4.1 - 4.2 respectively. For the above set of parameters, the bifurcation value for instability for τ was approximately 0.43 , as predicted by linear stability analysis. As we can see, when $\tau = 0.01$, we get solutions very close to the homogeneous steady state solution with deviation of the order of 10^{-5} . For $\tau = 0.7$, the scheme converges to a uniform steady state. This is consistent with the results obtained from linear stability analysis, since no wave number is isolated for this value of τ and hence the uniform steady state is actually the solution of the system at the final time. This scheme is stable and the numerical errors and the transient solutions converge. It is also seen that this scheme blows up for values of $\tau > 0.8$ and hence we are not able to illustrate the existence of spatially inhomogeneous solutions for large values of τ . We perform the same experiments with $b = 0.12$ and $b = 1$ and get similar results with the uniform steady state obtained at the final time for different

values of τ above and below the bifurcation values, as predicted by linear stability theory. We also illustrate the results obtained using the 2-SAFBDF scheme next.

4.1.2 The 2-SAFBDF scheme

We test the 2-SAFBDF scheme applied to equation (A) in a similar way to the 2-SBDF scheme. We choose two sets of model parameters values

$$d = 10^{-2.5}, \quad a = 10^{-0.5}, \quad b = 0.01, \quad \gamma = 0.01, \quad \delta = 1, \quad h = 0.1.$$

and

$$d = 10^{-2.5}, \quad a = 10^{-0.5}, \quad b = 0.12, \quad \gamma = 0.01, \quad \delta = 1, \quad h = 0.1.$$

for the same δx and δt as chosen for the 2-SBDF scheme and vary $\tau = 0.01$ and 0.7 . The results are shown in Figures 4.3 - 4.6. As we can see, these numerical results are very similar to the ones presented using the 2-SBDF scheme. For the first set of parameter values chosen, the threshold value of τ was predicted as 0.43 and in the second case the threshold value was predicted as 0.63 . In both the cases we are able to see the homogeneous steady state solution for $\tau = 0.01$. The scheme also converges to the uniform steady state when $\tau = 0.7$ for both values of b and is consistent with the results obtained using linear stability theory, since no wave number is isolated. It was also observed that a decrease in δx and δt yielded a similar pattern of solution behavior for the both of the above schemes, only that it was more resolved. Also the schemes seemed to converge only if the ratio $\frac{\delta t}{(\delta x)^2}$ was less than 1.3 (approximately) else they blew up. If the length L is increased from 1 to

10 and m is changed from 100 to 1000 to increase the number of space steps for the first set of parameter values, we are able to see the onset of spatially inhomogeneous solutions (figure 4.8) especially for n , but they are not prominent enough to be classified as finite amplitude structures. Nevertheless, we are able to see a tendency towards formation of spatial patterns. Hence we conclude that increase in τ leads to an increase to instability beyond the bifurcation values in each case which implies that the cell traction force is also increased. This shows that the pathogenesis of striae is indeed mechanical!

We now turn our attention to the other experimental hypothesis for the pathogenesis of striae, corticosteroid excess. We have linked corticosteroid excess to be characterized by a decrease in Young's modulus of the fibroblast which in turn is characterized by a decrease in parameter b . We shall see how decreasing b responds to the numerical approximations. We take $b = 1$ with τ fixed as 0.5 and the results are illustrated in figure 4.7. Linear stability theory predicted that the threshold value was 0.04 and hence values greater than this should exhibit homogeneous state solutions which is seen clearly. It has also been observed that when $b = 0.01$, the scheme converges to the uniform steady state at the final time solution. Hence we have shown computationally that corticosteroid excess, characterized by decrease in Young's modulus in turn characterized by b , when decreased below the instability bifurcation value of 0.04 yields patterns as seen in striae. We do not see spatially inhomogeneous solutions that isolate a wave number since the code blows up for extremely small values of b , where the wave number π is isolated.

4.2 Conclusion and future directions

The problem of striae distensae was successfully studied and it was derived from a realistic model that an increase in traction force beyond a certain threshold value leads to the formation of stretch marks, accounting for the mechanical aspect of pathogenesis of striae and decreasing the Young's Modulus beyond a certain threshold value may also leads to stretch mark pattern exhibition and this accounts for the experimental evidence suggesting that corticosteroid excess, in certain cases, may cause striae . Detailed linear stability analysis was applied to the proposed model and results were predicted as obtained in Tables 1 - 4 in Chapter 2 in a certain range of parameter values. These results were confirmed using novel finite difference schemes. Linear stability analysis indicates that pattern formation is possible in a certain parameter regime and this was validated by numerical computations. We require that $Re(\lambda(k)) > 0$ for some $k > 0$, but the results for $k=1$ occur for very high values of τ which have no physical significance since the typical value of τ in normal humans is around 0.01. Hence even though we were not able to isolate finite amplitude spatially inhomogeneous structures due to blow up of the schemes, we have been successful in showing that when $k \neq 1$, the uniform steady state is reached by convergent schemes. Two different schemes were tested for the first equation, and both yielded convergent and similar solutions. The main conclusions derived are significant and their biological interpretation may have direct consequences to a daily life problem faced by millions of people.

These results have to be validated by experiments which is important before preventive measures can be taken to resolve the problem of striae distensae. The parameter values arrived at could serve as guidelines for experiments to search for threshold values around

this region. The same model can also be extended to incorporate other features. This work is a stepping stone and we have been successful in predicting values of characteristic parameters that would lead to pattern formation in striae and hopefully these will find their way in fostering future research in this area.

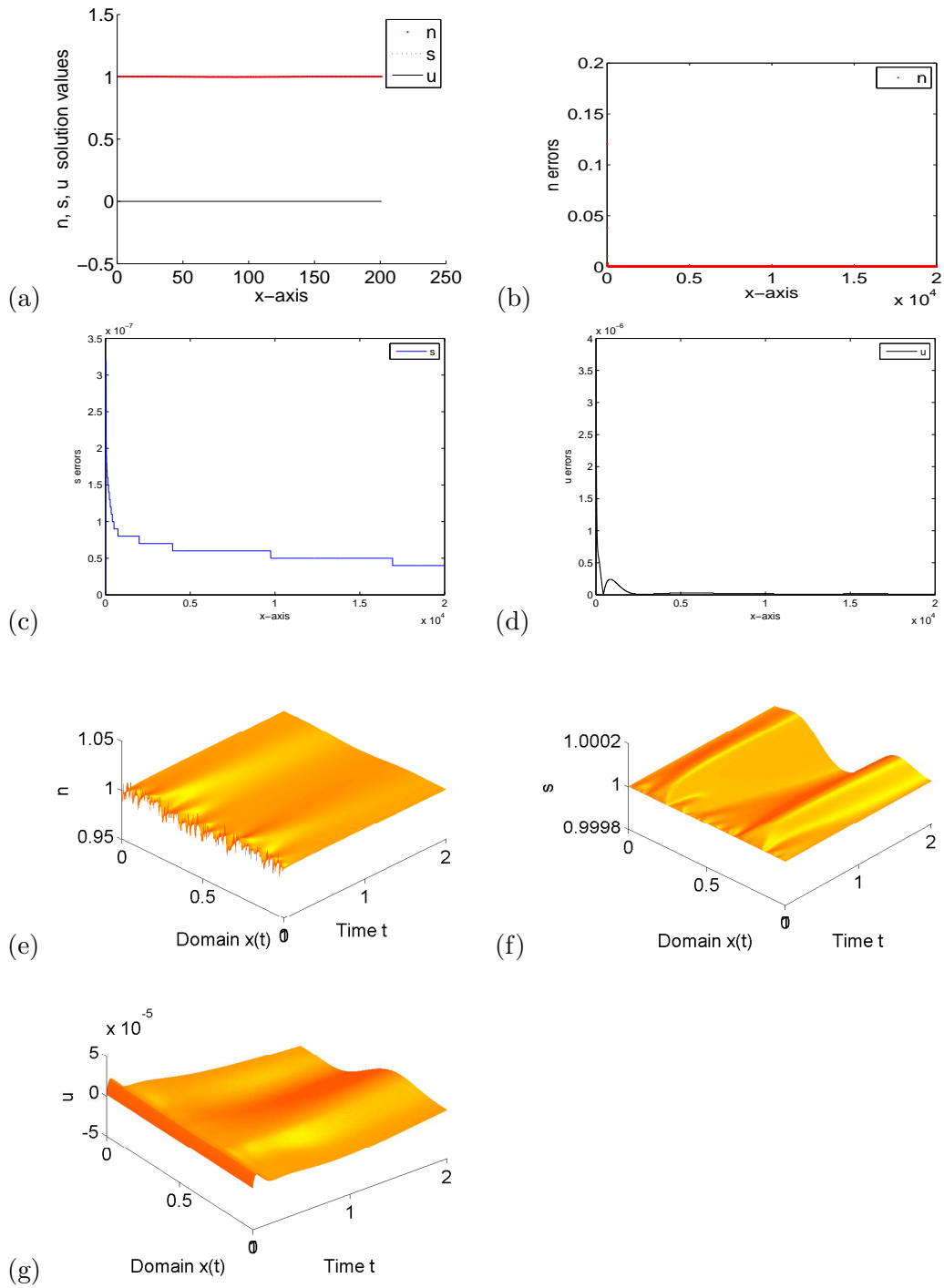


Figure 4.1: Solutions obtained by fixing all the parameters with $\tau = 0.01$ and $b=0.01$ (a) n , s , u solution values, (b)-(d) error graph, (e)-(g) transient solution for n , s and u using the 2-SBDF scheme for equation (A).

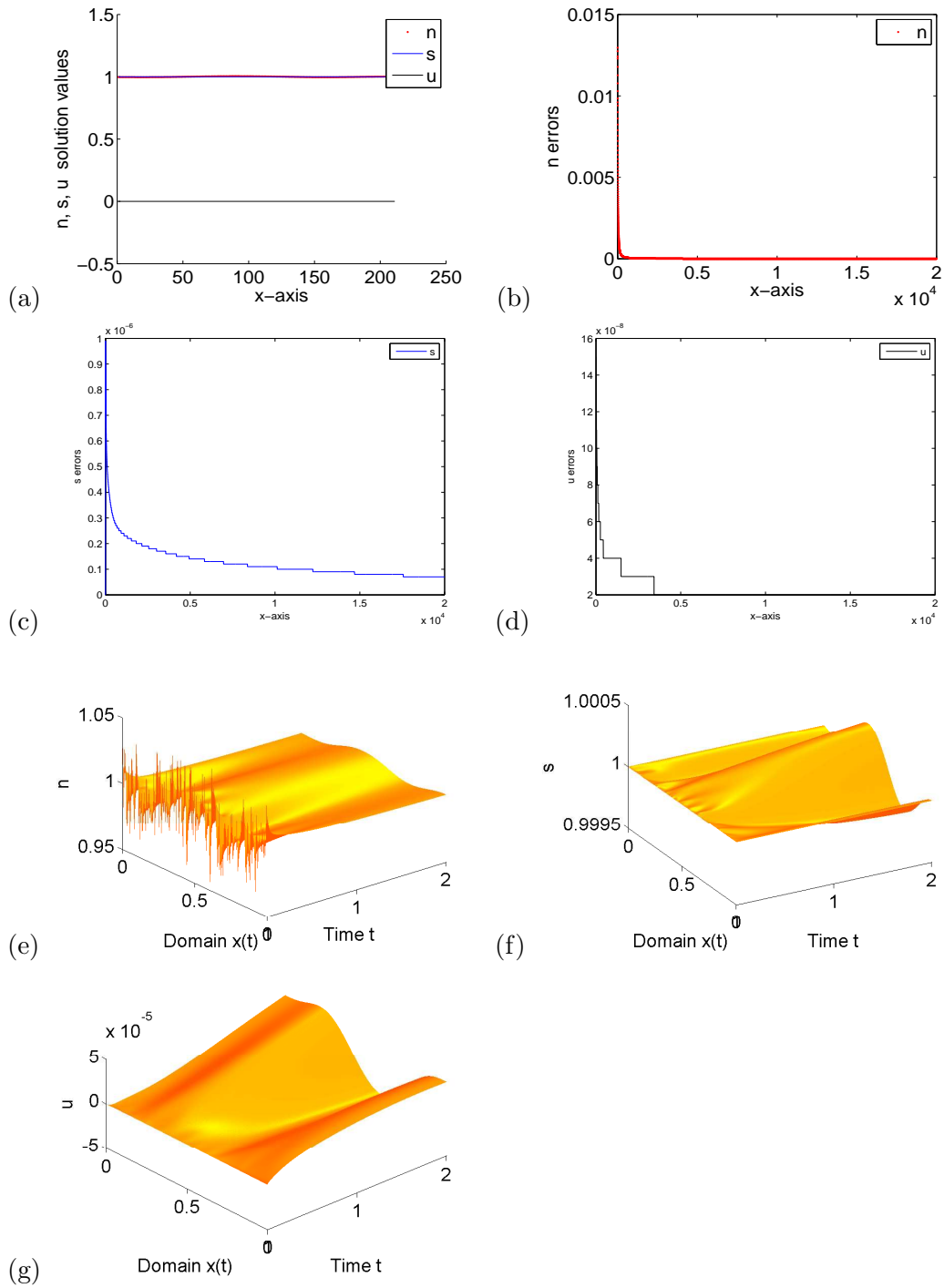


Figure 4.2: Solutions obtained by fixing all the parameters with $\tau = 0.7$ and $b=0.01$ (a) n , s , u solution values, (b)-(d) error graph, (e)-(g) transient solution for n , s and u using the 2-SBDF scheme for equation (A).

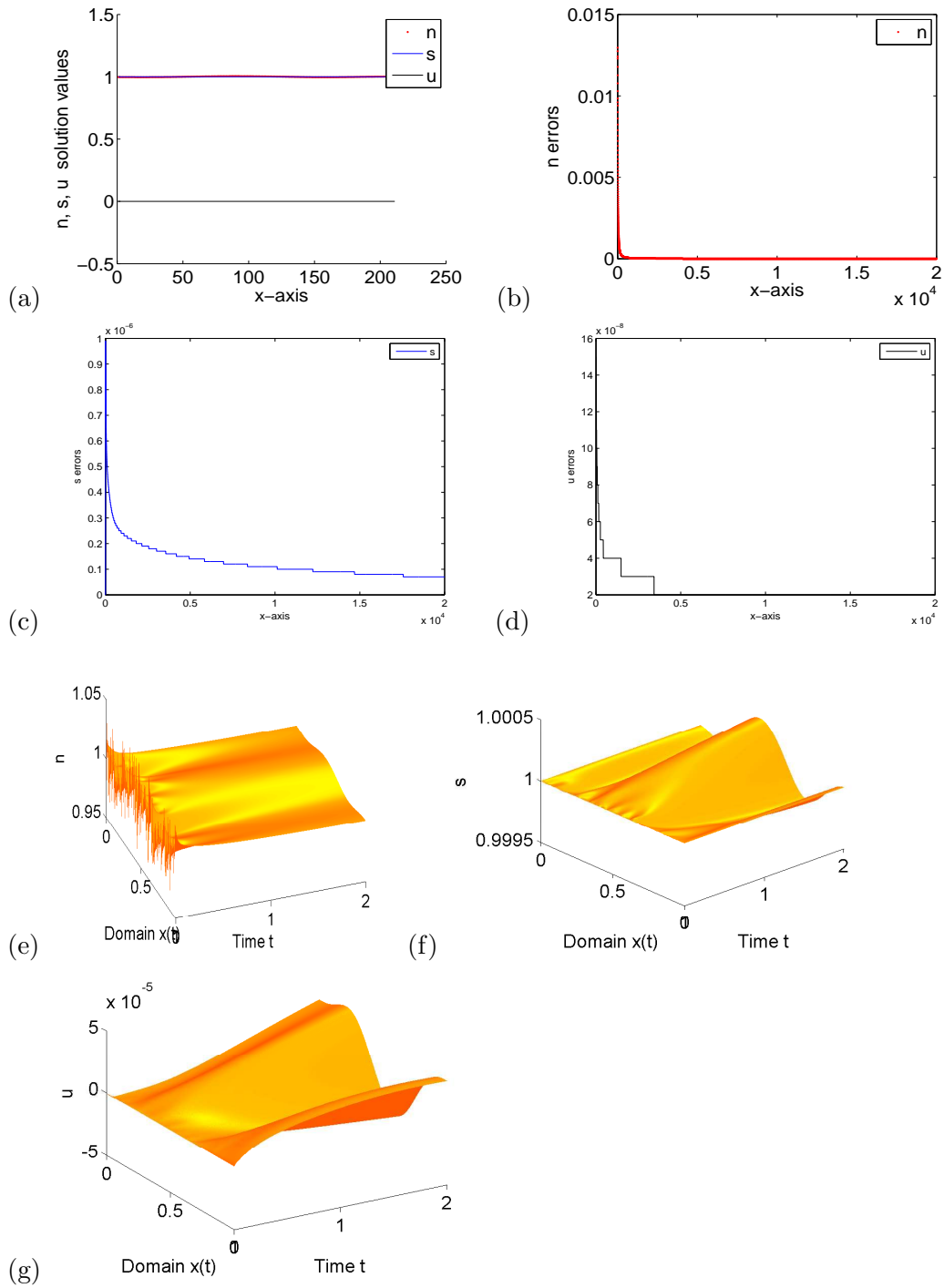


Figure 4.3: Solutions obtained by fixing all the parameters with $\tau = 0.01$ and $b=0.01$ (a) n , s , u solution values, (b)-(d) error graph, (e)-(g) transient solution for n , s and u using the 2-SAFBDF scheme for equation (A).

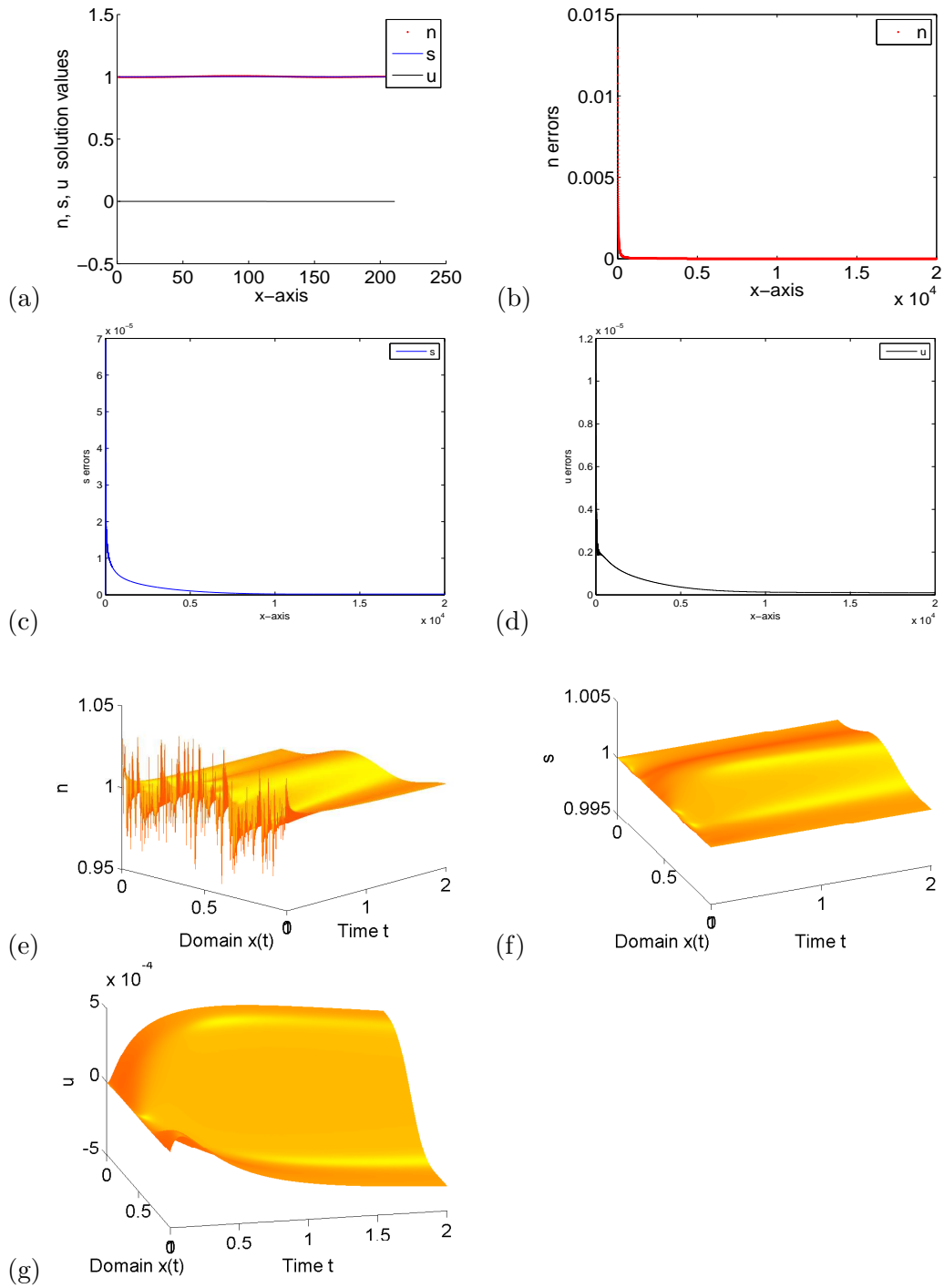


Figure 4.4: Solutions obtained by fixing all the parameters with $\tau = 0.7$ and $b=0.01$ (a) n , s , u solution values, (b)-(d) error graph, (e)-(g) transient solution for n , s and u using the 2-SAFBDF scheme for equation (A).

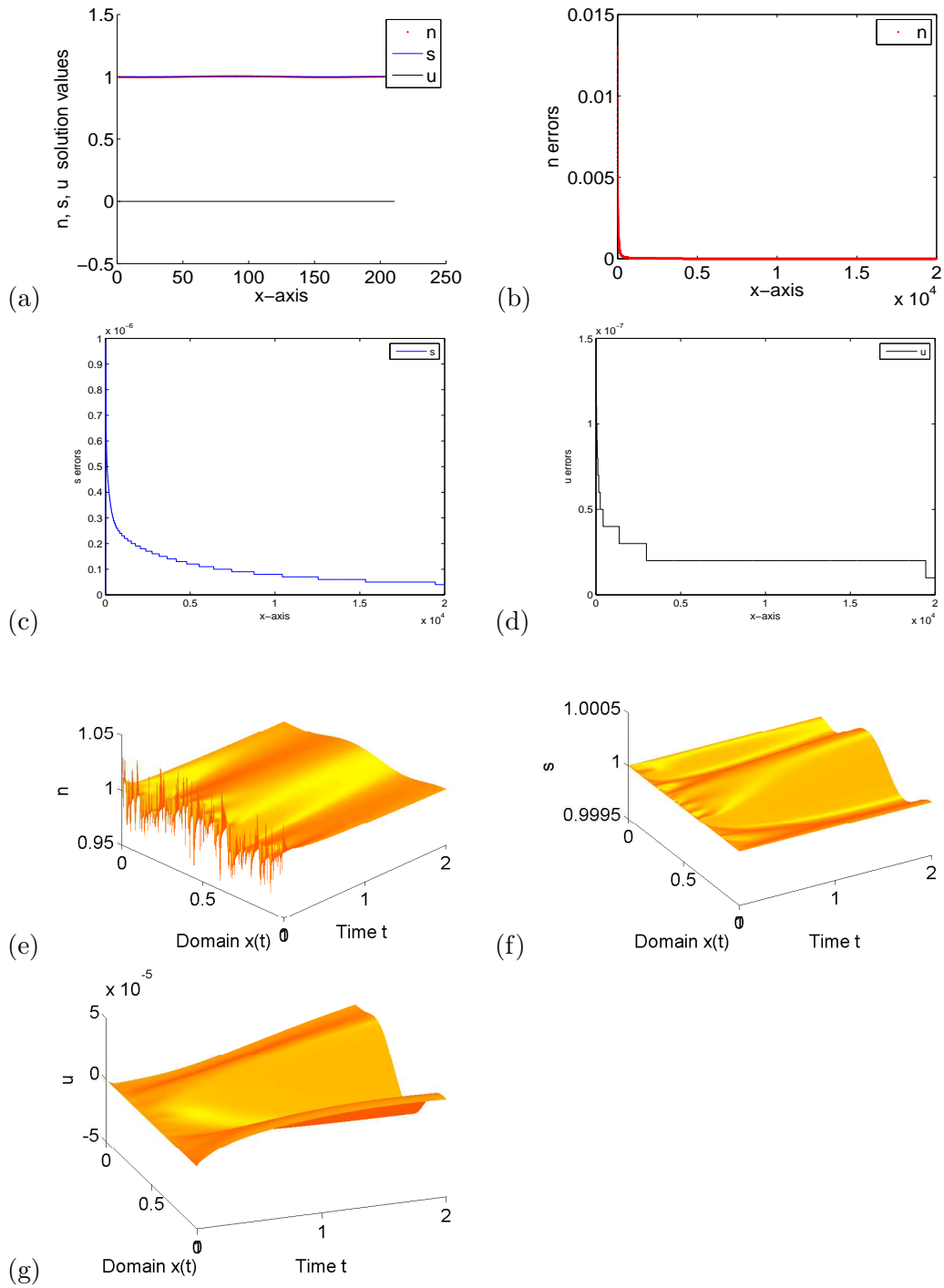


Figure 4.5: Solutions obtained by fixing all the parameters with $\tau = 0.01$ and $b=0.12$ (a) n, s, u solution values, (b)-(d) error graph, (e)-(g) transient solution for n, s and u using the 2-SAFBDF scheme for equation (A).

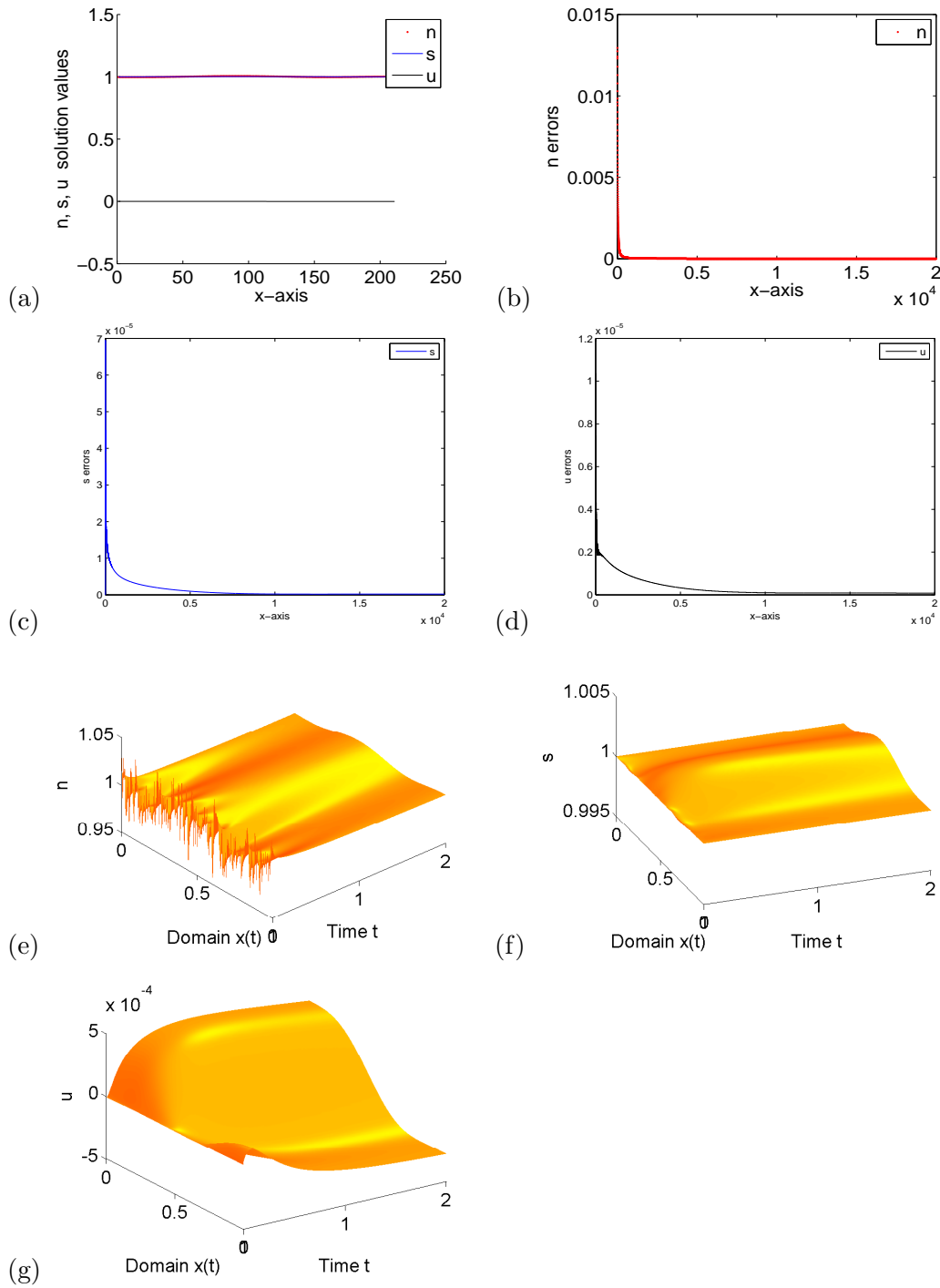


Figure 4.6: Solutions obtained by fixing all the parameters with $\tau = 0.7$ and $b=0.12$ (a) n , s , u solution values, (b)-(d) error graph, (e)-(g) transient solution for n , s and u using the 2-SAFBDF scheme for equation (A).

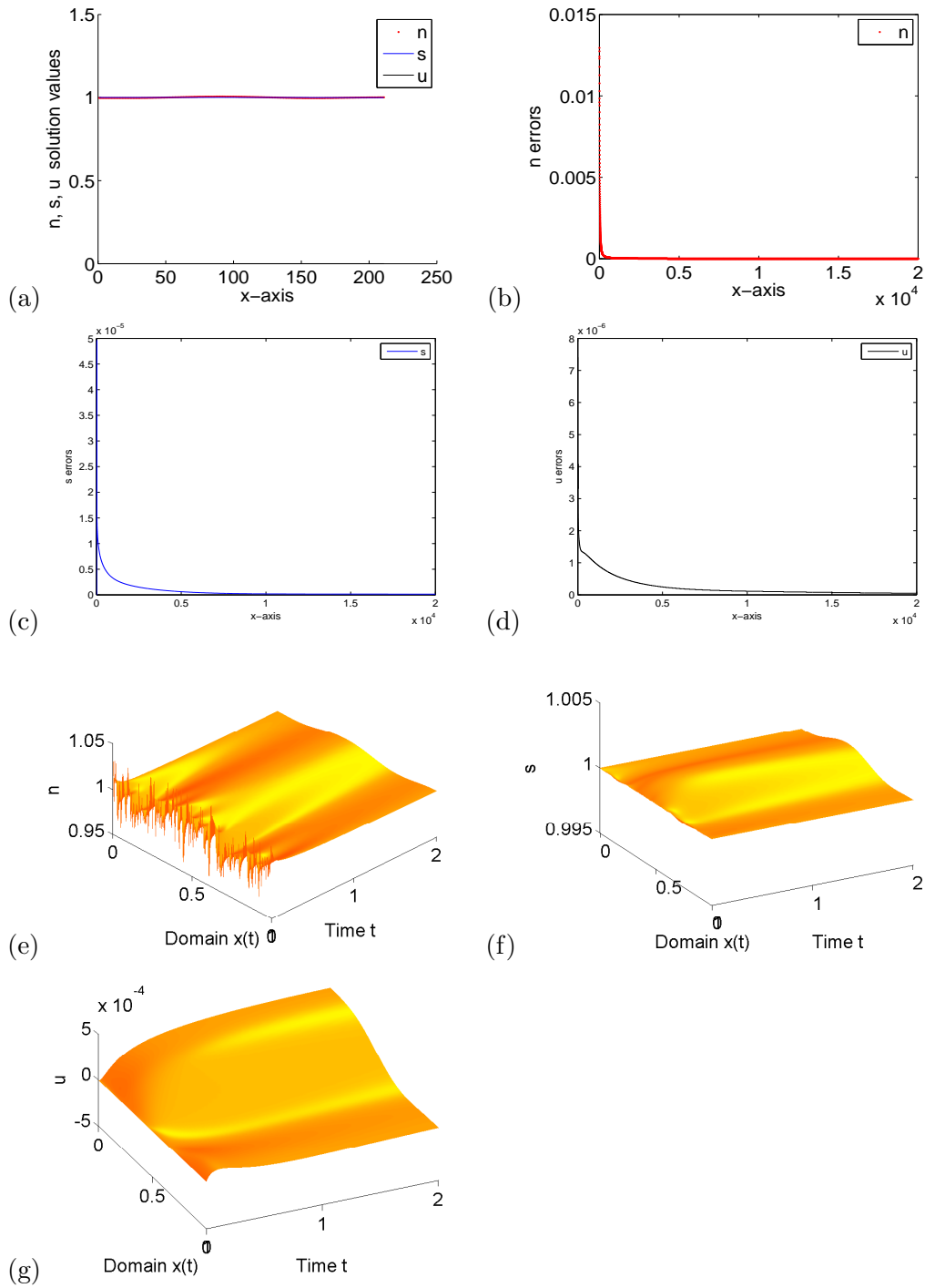


Figure 4.7: Solutions obtained by fixing all the parameters with $\tau = 0.5$ and $b=1$ (a) n, s, u solution values, (b)-(d) error graph, (e)-(g) transient solutions for n, s and u using the 2-SAFBDF scheme for Equation (A).

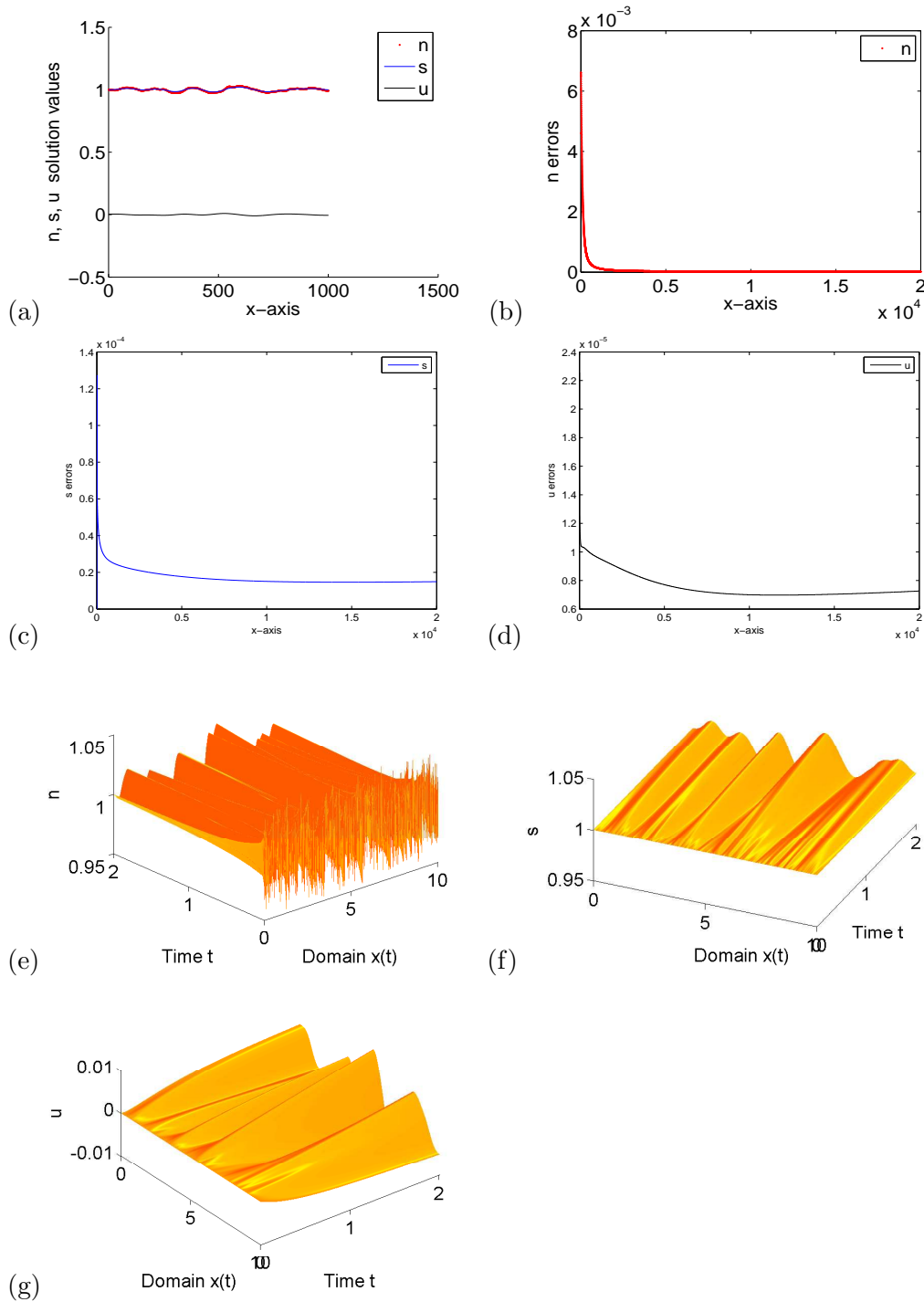


Figure 4.8: Solutions obtained by fixing all the parameters with $\tau = 0.7$, $b = 0.01$, $L = 10$ and $m = 1000$ (a) n , s , u solution values, (b)-(d) error graph, (e)-(g) transient solutions for n , s and u using the 2-SAFBDF scheme for Equation (A).

BIBLIOGRAPHY

- [1] Alaiti, Sameer. Striae Distensae. *eMedicine*, (2006).
- [2] Maini, P. K. A Mechanochemical model for adult dermal wound contraction. *J. of Biol. Systems* **3(4)**, 1021-1031, (1995).
- [3] Hay, E. *Cell Biology of the Extracellular Matrix*, Newyork, Plenum, (1981).
- [4] Harris, A. K., Stopak, D., Wild, D. Fibroblast traction as a mechanism for collagen morphogenesis. *Nature*, **290**, 249-251, (1981).
- [5] Trinkaus, J. P. *Cells Into Organs*. Prentice-Hall, (1984).
- [6] Viennet, C. Mechanical behavior of fibroblasts included in collagen lattices. *J. Soc. Biol.*, **195(4)**, 427-30 (2001).
- [7] Viennet, C. Contractile forces generated by striae distensae fibroblasts embedded in collagen lattices. *Arch. Dermatol. Res.*, **297(1)**, 7-10, (2005).
- [8] Madzvamuse, A. A modified backward Euler scheme for advection-reaction-diffusion systems. *Mathematical Modeling of Biological Systems Volume I*. A. Deutsch, L. Brusch, H. Byrne, G. de Vries and H.-P. Herzel (eds). Birkhuser, Boston, 191-197 (2007).
- [9] Madzvamuse, A. Time-stepping schemes for moving grid finite elements applied to reaction-diffusion systems on fixed and growing domains. *J. of Comp. Phys.*, **24 (1)**, 239-263, (2005).
- [10] Morton, K. W. and Mayers, D. F. *Numerical Solution of Partial Differential Equations*. Cambridge University Press, (1994).
- [11] Murray, J. D. *Mathematical Biology*. Springer Velag, (1989).
- [12] Segel, Lee. *Mathematics Applied to Continuum Mechanics*. Dover Publications, (1987).
- [13] Murray, J. D. A mechanical model for mesenchymal morphogenesis. *J.Math.Biol.*, **17**, 125-129, (1983)
- [14] Murray, J. D., Oster, G.F. Generation of Biological Pattern and Form. *IMA J. Math. Appl. in Medicine and Biol.*, **1**, 51-75, (1984)
- [15] Murray, J. D., Maini, P.K., Tranquillo, R. T. Mechanical models for generating biological patterns and form in development. *Phys. Reports*, **171**, 60-84, (1988)

- [16] Hinchcliffe, J. R., Johnson, D. R. *The development of Vertebrate Limb*. Oxford Clarendon Press, (1980)
- [17] Landau, L., Lifshitz, E. *Theory of Elasticity: 2nd end*. Pergamon, (1970)
- [18] Ruuth, S. Implicit-explicit methods for reaction-diffusion problems in pattern-formation. *J. Math. Biol.*, **34(2)**, 148-176, (1995).
- [19] Sherrat, A. A Mathematical Model for Fibroblast and Collagen Orientation. *Bullet. for Math Biol.*, (1998).
- [20] Smoller, J. *Shock Waves and Reaction Diffusion equations*. Springer, (1983).
- [21] Turing, A. The chemical basis of morphogenesis. *Phil. Trans. R. Soc. Lond. B*, **237**, 37-72, (1952).

APPENDIX A

LINEAR STABILITY ANALYSIS: MATLAB

```
clear all; clear variables; clf; hold on;

axes('FontSize',16);

a=10^(-0.5); b=1; d=10^(-2.5); h=0.1; delta=1; tau=1.86;

gamma=0.01; r=1; j=1; l=1; g=1; f=1; count=1;

for k=0:0.01:40

    alpha=a*k.^2;

    beta=(a*d+tau*h)*k.^4+(tau*(delta-2)+b)*k.^2+1;

    sigma=d*tau*h*k.^6+(d*tau*(delta-1)+b*d)*k.^4+d*k.^2;

    H(g)=(-beta+real(sqrt((beta).^2-4*alpha*sigma)));

        if(H(g)>0)

            c(r)=g-1;

            n(r)=c(r)/100;

            if(r>1)

                if((n(r-1)<pi)&&(n(r)>pi))

                    count=count+1;

                end

            end

            S(r)=H(g-1);

            r=r+1;

            S(r)=H(c(r-1)+1);

        end

end
```

```

        j=j+1;
        g=g+1;
end
if(r==1)
    fprintf('No Wave Numbers Can be Isolated\n')
else
    S(r+1)=H(c(r-1)+2);
    o=[S(2:r)];
fid=fopen('isowave1.txt','wt');
fprintf(fid,'The isolated wave numbers when TAU=%d are\n',tau);
fprintf(fid,'%2.10f\n',o);
fprintf(fid,'The isolated wave numbers along with the lower and
higher boundary limits are\n');
fprintf(fid,'%2.10f\n',S);
fprintf(fid,'The approxiamate interval in which these wave numbers
are situated is\n');
fprintf(fid,'%2.10f\t %2.10f\t',(-S(1)+S(2))/2,(S(r)-S(r+1))/2);
fclose(fid);
n(1)n(r-1)end count g=1
for k=0:0.01:40
    plot(k,H(g),'k');
    g=g+1;
end

```

APPENDIX B

THE THOMAS ALGORITHM

The Thomas algorithm is a simplified form of Gaussian elimination that can be used to solve a tridiagonal systems of equations. A tridiagonal system is written as

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i \quad (\text{B.1})$$

where $a_1 = 0$ and $c_n = 0$. Hence when the each of the coefficients a_i , b_i , c_i and d_i are defined for values of i ranging from 0 to m , which are the number of unknowns, Thomas Algorithm can be used in solving these system of equations. In matrix form, these system of equations are written as

$$\begin{bmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & \cdot & \\ & & \cdot & \cdot & c_{n-1} \\ 0 & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \cdot \\ \cdot \\ d_n \end{bmatrix}$$

The solution of these system of equations can be obtained in $O(n)$ steps instead of $O(n^3)$ steps, as a result of normal Gaussian Elimination. Also this algorithm works only for diagonally dominant matrices and this check is made. The algorithm is implemented in two phases, a forward elimination phase in which the a_i 's are eliminated and a backward substitution phase where the solution is got.

B.1 Description of the algorithm

B.1.1 The forward elimination phase

- Preserve B and D(also called as forcing vectors) by copying them to new arrays for $i=0$ to step m
- For $i = 1$ to m do

$$m = \frac{a_i}{b_{i-1}}$$

$$b_i = b_i - m * c_{i-1}$$

$$d_i = d_k - m d_{i-1}$$

B.1.2 The backward substitution phase

- Compute

$$x_m = \frac{d_m}{b_m}$$

- For $i = m-1$ step down to 1 do

$$x_k = \frac{d_k - c_k x_{k+1}}{b_k}$$

APPENDIX C

IMPLEMENTATION OF THE FINITE DIFFERENCE AND TRI-DIAGONAL SOLVER: FORTRAN

```
implicit none

! -----
! Parameters
! -----

integer i,m,simultime, timestep,snapshot, k,max, j, maxn
parameter (max=50000,maxn=10**6)

double precision paramA, paramB, paramD, delta, paramh, gamma, kappa
double precision phi, gary, mary, lary, pary, dary, cary
double precision tau, theta
double precision lambda1, lambda2
double precision dt, dx, dx2, dx3, length, finaltime,alpha,beta
double precision random,epsilon
double precision An(0:max), Bn(0:max), Cn(0:max)
double precision Dn(0:max), As(0:max), Bs(0:max)
double precision Cs(0:max), Ds(0:max), Av(0:max)
double precision Bv(0:max), Cv(0:max), Dv(0:max)
double precision err1(1:maxn), err2(1:maxn), err3(1:maxn)
double precision errn, errs, erru
double precision Fq(0:max),Fqmin(0:max)
```

```

double precision P1(0:max),P2(0:max)
double precision P3(0:max),P4(0:max)
double precision P5(0:max),P6(0:max)
double precision P7(0:max),P8(0:max)
double precision P9(0:max),P10(0:max)
double precision P11(0:max),P12(0:max)
double precision P13(0:max),P14(0:max)
double precision P21(0:max),P31(0:max)
double precision P41(0:max),P51(0:max)
double precision P61(0:max),P71(0:max)
double precision P81(0:max)
double precision TempBNew1(0:max), TempBNew2(0:max), TempBNew3(0:max)
double precision ResNew1(0:max),ResNew2(0:max),ResNew3(0:max)
double precision mult1,mult2, mult3
double precision n(-1:max), nn(-1:max), nold(-1:max)
double precision s(-2:max), sn(-2:max)
double precision v(-1:max), vn(-1:max), vold(-1:max)
double precision u(-1:max), un(-1:max), uold(-1:max)
! -----
! data
! -----
paramA= 10.0d0**(-0.50d0)
paramB=1.0d0

```

```

paramD=10.0d0**(-2.50d0)

delta=1.0d0

paramh=0.10d0

gamma=0.010d0

tau=0.50d0

dt=10.0d0**(-4d0)

length=1.0d0

m=210

snapshot=10

finaltime=2.0d0

! -----
! The constant parameters
! -----

dx= length/m

dx2= dx*dx

dx3= dx*dx*dx

simultime=INT(finaltime/dt)

print*, simultime

lambda1= 2.0d0*paramD*dt/dx2

lambda2=dt/(2.0d0*dx)

alpha=paramD*gamma/(2.0d0*dx3)

beta=1.0d0/(4.0d0*dx)

theta=(3.0d0*dx2/(2.0d0*dt))

```

```

kappa=paramD*gamma

phi=(dx/4.0d0)

gary=kappa/(2.0d0*dx)

mary=dx2/(2.0d0*dt)

lary=(tau*dx/4.0d0)

pary=(tau*paramh/(2.0d0*dx))

dary=(delta*tau/8.0d0)

cary=(delta*tau*paramh/(4.0d0*dx2))

! -----
! Open Files for storing Results
! -----

open(25, file='wnsol.m', status='unknown')
open(26, file='wssol.m', status='unknown')
open(27, file='wusol.m', status='unknown')
open(50, file='nerr.dat', status='unknown')
open(51, file='serr.dat', status='unknown')
open(52, file='uerr.dat', status='unknown')

write(25,8)

write(26,9)

write(27,10)

8 format(/,'N=[')
9 format(/,'S=[')
10 format(/,'U=[')

```



```

! -----
! define initial condition
! -----

do 1 i=0,m

epsilon=random()

n(i)=1.0d0+(0.5d0-epsilon)/10.0d0

s(i)=1.0d0

u(i)=0.0d0

v(i)=0.0d0

mn(i)=0.0d0

sn(i)=0.0d0

vn(i)=0.0d0

un(i)=0.0d0

nold(i)=1.0d0+(0.5d0-epsilon)/10.0d0

uold(i)=0.0d0

vold(i)=v(i)

1 continue

! -----

! Starting The Iterations

! -----

do 50 timestep= 1,simultime

!do 50 timestep=1,1

do 25 i=0,m

```

An(i)=0.0d0

Bn(i)=0.0d0

Cn(i)=0.0d0

Dn(i)=0.0d0

As(i)=0.0d0

Bs(i)=0.0d0

Cs(i)=0.0d0

Ds(i)=0.0d0

Av(i)=0.0d0

Bv(i)=0.0d0

Cv(i)=0.0d0

Dv(i)=0.0d0

25 continue

```
! -----  
! Zero Flux Boundary Conditions  
! -----
```

n(-1)=n(1)

n(m+1)=n(m-1)

nold(-1)=nold(1)

nold(m+1)=nold(m-1)

$$s(-2)=s(2)$$

$$s(-1)=s(1)$$

$$s(m+1)=s(m-1)$$

$$s(m+2)=s(m-2)$$

$$u(-1)=u(1)$$

$$u(m+1)=u(m-1)$$

$$uold(-1)=uold(1)$$

$$uold(m+1)=uold(m-1)$$

$$v(-1)=v(1)$$

$$v(m+1)=v(m-1)$$

$$vold(-1)=vold(1)$$

$$vold(m+1)=vold(m-1)$$

```
! -----  
! Coefficients A,B,C,D, Solving and Errors  
! -----  
  
do 2 i=0,m  
P1(i)=(n(i+1)-2.0d0*n(i)+n(i-1))  
P2(i)=(u(i+1)-u(i-1))
```

```

P3(i)=(n(i+1)-n(i-1))
P4(i)=(u(i+1)-2.0d0*u(i)+u(i-1))
P5(i)=(n(i+1)+n(i))
P6(i)=(v(i+1)+v(i))
P7(i)=(n(i)+n(i-1))
P8(i)=(v(i)+v(i-1))
P9(i)=(v(i+1)-v(i-1))
P10(i)=(s(i+1)+s(i))
P12(i)=(s(i)+s(i-1))
P13(i)=(s(i+1)-2.0d0*s(i)+s(i-1))
P14(i)=(s(i+2)-2.0d0*s(i+1)+2.0d0*s(i-1)-s(i-2))
P11(i)=(nold(i+1)-2.0d0*nold(i)+nold(i-1))
P21(i)=(uold(i+1)-uold(i-1))
P31(i)=(nold(i+1)-nold(i-1))
P41(i)=(uold(i+1)-2.0d0*uold(i)+uold(i-1))
P51(i)=(nold(i+1)+nold(i))
P61(i)=(vold(i+1)+vold(i))
P71(i)=(nold(i)+nold(i-1))
P81(i)=(vold(i)+vold(i-1))

!SAFBDF

if (i .eq. 0) then
  Bn(i)=theta+2.0d0*paramD+2.0d0*gary*P2(i)+phi*P9(i)

```

```
Cn(i)=-paramD-gary*P2(i)+gary*P4(i)-phi*P8(i)-&
      paramD-gary*P2(i)-gary*P4(i)+phi*P6(i)
```

```
elseif (i .eq. m) then
```

```
An(i)=-paramD-gary*P2(i)+gary*P4(i)-phi*P8(i)-&
      paramD-gary*P2(i)-gary*P4(i)+phi*P6(i)
```

```
Bn(i)=theta+2.0d0*paramD+2.0d0*gary*P2(i)+phi*P9(i)
```

```
else
```

```
An(i)=- (paramD+gary*P2(i)-gary*P4(i)+phi*P8(i))
```

```
Bn(i)=theta+2.0d0*paramD+2.0d0*gary*P2(i)+phi*P9(i)
```

```
Cn(i)=- (paramD+gary*P2(i)+gary*P4(i)-phi*P6(i))
```

```
endif
```

```
Dn(i)=(-nold(i)+4.0d0*n(i))*mary
```

```
!$$$$$$ !SBDF
```

```
!$$$$$$      Fq(i)=alpha*P1(i)*P2(i)+alpha*P3(i)*P4(i)-&
```

```
!$$$$$$          beta*(P5(i)*P6(i)-P7(i)*P8(i))
```

```
!$$$$$$      Fqmin(i)=alpha*P11(i)*P21(i)+alpha*P31(i)*P41(i)-&
```

```
!$$$$$$          beta*(P51(i)*P61(i)-P71(i)*P81(i))
```

```
!$$$$$$      if (i .eq. 0) then
```

```
!$$$$$$      Bn(i)=3.0d0+2.0d0*lambda1
```

```
!$$$$$$      Cn(i)=-2.0d0*lambda1
```

```

!$$$$$$ elseif (i .eq. m) then
!$$$$$$ An(i)=-2.0d0*lambda1
!$$$$$$ Bn(i)=3.0d0+2.0d0*lambda1
!$$$$$$ else
!$$$$$$ An(i)=-lambda1
!$$$$$$ Bn(i)=3.0d0+2.0d0*lambda1
!$$$$$$ Cn(i)=-lambda1
!$$$$$$ endif
!$$$$$$ Dn(i)=4.0d0*n(i)-nold(i)+4.0d0*dt*Fq(i)-2.0d0*dt*Fqmin(i)
2 continue

```

```

!--- Preserve B and forcing vectors by copying them to new arrays

```

```

DO i = 0, m
    TempBNew1(i) = Bn(i)
    ResNew1(i) = Dn(i)
END DO

```

```

!--- Calculating solution to tridiagonal matrix

```

```

DO i = 1, m
    mult1 = An(i)/TempBNew1(i-1)
    TempBNew1(i) = TempBNew1(i) - Cn(i-1) * mult1
    ResNew1(i) = ResNew1(i) - ResNew1(i-1) * mult1
END DO

```

```

nn(m) = ResNew1(m)/TempBNew1(m)

DO i = m-1, 0, -1
    nn(i) = (ResNew1(i) - Cn(i) * nn(i+1)) / &
            TempBNew1(i)

END DO

!error n
errn=0.0d0
do j=0,m
errn=errn+(nn(j)-n(j))**(2.0d0)
end do
err1(timestep)=sqrt(errn)

!update n
do i=0,m
nold(i)=n(i)
n(i)=nn(i)
nn(i)=0.0d0
end do

! Hyperbolic Equation: s
do 30 i=0,m

```

```

if(v(i) .le. 0.0d0)then
  if (i .eq. 0) then
    Bs(i)=1.0d0+lambda2*(v(i+1)-v(i-1))+2.0d0*lambda2*v(i)
    Cs(i)=-2.0d0*lambda2*v(i)
  else
    As(i)=-2.0d0*lambda2*v(i)
    Bs(i)=1.0d0+lambda2*(v(i+1)-v(i-1))+2.0d0*lambda2*v(i)
    Cs(i)=0.0d0
  endif
else
  if (i .eq. m) then
    As(i)=2.0d0*lambda2*v(i)
    Bs(i)=1.0d0+lambda2*(v(i+1)-v(i-1))-2.0d0*lambda2*v(i)
  else
    As(i)=0.0d0
    Bs(i)=1.0d0+lambda2*(v(i+1)-v(i-1))-2.0d0*lambda2*v(i)
    Cs(i)=2.0d0*lambda2*v(i)
  endif
endif
endif
Ds(i)=s(i)
30 continue
!--- Preserve B and forcing vectors by copying them to new arrays
DO i = 0, m

```



```

TempBNew2(i) = Bs(i)

ResNew2(i) = Ds(i)

END DO

!--- Calculating solution to tridiagonal matrix
DO i = 1, m
    mult2          = As(i)/TempBNew2(i-1)
    TempBNew2(i) = TempBNew2(i) - Cs(i-1) * mult2
    ResNew2(i) = ResNew2(i) - ResNew2(i-1) * mult2
END DO

sn(m) = ResNew2(m)/TempBNew2(m)

DO i = m-1, 0, -1
    sn(i) = (ResNew2(i) - Cs(i) * sn(i+1)) / &
            TempBNew2(i)
END DO

! errors for s(x,t)
errs=0.0d0
do j=0,m
    errs=errs+(sn(j)-s(j))**(2.0d0)
end do

```

```

err2(timestep)=sqrt(errs)

!update s
do i=0,m
  s(i)=sn(i)
  sn(i)=0.0d0
end do

!Elliptic Equation
do 4 i=0,m
if (i .eq. 0) then
  Bv(i)=2.0d0*paramA+0.00010d0
  Cv(i)=-2.0d0*paramA

elseif (i .eq. m) then
Av(i)=-2.0d0*paramA
Bv(i)=2.0d0*paramA+0.00010d0
else
  Av(i)=-paramA
  Bv(i)=2.0d0*paramA+0.00010d0
  Cv(i)=-paramA
endif

```

```

Dv(i)=paramB*P4(i)+lary*(P10(i)*P5(i)-P12(i)*P7(i)) &
+pary*P3(i)*P13(i)+pary*n(i)*P14(i)&
+dary*P2(i)*(P10(i)*P5(i)-P12(i)*P7(i)) &
+8.0d0*dary*n(i)*s(i)*P4(i)+cary*P2(i)*P3(i)*P13(i) &
+4.0d0*cary*n(i)*P4(i)*P13(i)+cary*n(i)*P2(i)*P14(i) &
-s(i)*u(i)*dx2

4 continue

! --- Preserve B and forcing vectors by copying them to new arrays

DO i = 0, m
    TempBNew3(i) = Bv(i)
    ResNew3(i) = Dv(i)
END DO

!--- Calculating solution to tridiagonal matrix

DO i = 1, m
    mult3          = Av(i)/TempBNew3(i-1)
    TempBNew3(i) = TempBNew3(i) - Cv(i-1) * mult3
    ResNew3(i) = ResNew3(i) - ResNew3(i-1) * mult3
END DO

vn(m) = ResNew3(m)/TempBNew3(m)

```

```

DO i = m-1, 0, -1
    vn(i) = (ResNew3(i) - Cv(i) * vn(i+1)) / &
            TempBNew3(i)
END DO
!Update v
do i=0,m
    vold(i)=v(i)
    v(i)=vn(i)
    vn(i)=0.0d0
end do

!Calculate u
do 6 i=0,m
    un(i)=v(i)*dt+u(i)
6 continue
! errors for u(x,t)
    erru=0.0d0
    do j=0,m
        erru=erru+(un(j)-u(j))**(2.0d0)
    end do
    err3(timestep)=sqrt(erru)

```

```

!Update u

  do i=0,m

    uold(i)=u(i)

    u(i)=un(i)

    un(i)=0.0d0

  end do

! -----
! Store the results in a file
! -----

k=mod(timestep,snapshot)

if(k.eq.0)then

  write(25,42) (n(i),i=0,m)

  write(26,43) (s(i),i=0,m)

  write(27,44) (u(i),i=0,m)

42  format(1505f15.8,';')
43  format(1505f15.8,';')
44  format(1505f15.8,';')

endif

  write(50,77) err1(timestep)

  write(51,78) err2(timestep)

```

```

write(52,79) err3(timestep)

77 format(f15.8)
78 format(f15.8)
79 format(f15.8)

do i=0,m

if(ABS(u(i)).gt.5.0d0) then
  print*, 'Blow up for u values:u(i)=', ABS(u(i))
  print*, 'Number of iterations for blow up=', timestep
  stop
elseif(ABS(v(i)).gt.5.0d0) then
  print*, 'Blow up for v values:v(i)=', ABS(v(i))
  print*, 'Number of iterations for blow up=', timestep
  stop
elseif(ABS(n(i)).gt.5.0d0) then
  print*, 'Blow up for n values:n(i)=', ABS(n(i))
  print*, 'Number of iterations for blow up=', timestep
  stop
elseif(ABS(s(i)).gt.5.0d0) then
  print*, 'Blow up for s values:s(i)=', ABS(s(i))
  print*, 'Number of iterations for blow up=', timestep

```

```

    stop
end if
end do
50 continue

    close(53)

    close(54)

    close(55)

    close(60)

    close(61)

! Write the output of the solutions at the final time

open(70, file='nfinsol.dat', status='unknown')
open(71, file='sfinsol.dat', status='unknown')
open(72, file='ufinsol.dat', status='unknown')

    write(70,74) (n(i),i=0,m)

    write(71,75) (s(i),i=0,m)

    write(72,76) (u(i),i=0,m)

74 format(f15.8)
75 format(f15.8)
76 format(f15.8)

```

```
close(70)

close(71)

close(72)

    write(25,200)

    write(26,201)

    write(27,202)

    200 format(/,''];')

    201 format(/,''];')

    202 format(/,''];')

close(25)

close(26)

close(27)

stop

end
```


APPENDIX D

GRAPHS OF THE SOLUTIONS OBTAINED: MATLAB

```
clear variables; clf; close all;    format long;
axes('FontSize',18);
    hold on;
load nfinsol.dat;
    load sfinsol.dat;
    load ufinsol.dat;
    plot(nfinsol,'r. ');
    plot(sfinsol,'b: ');
    plot(ufinsol,'k');
    ylabel('n, s, u  solution values');
xlabel('x-axis');
```

APPENDIX E
ERROR CONVERGENCE: MATLAB

```
clear variables;

clf;

close all;

    format long;

axes('FontSize',18);

load nerr.dat;

load serr.dat;

load uerr.dat;

figure(1);

plot(nerr,'r. ');

ylabel('n errors');

xlabel('x-axis');

legend('n');

figure(2);

plot(serr,'b');

ylabel('s errors');

xlabel('x-axis');

legend('s');

figure(3);

plot(uerr,'k');
```

```
ylabel('u errors');  
xlabel('x-axis');  
legend('u')
```

APPENDIX F
TRANSIENT SOLUTIONS: MATLAB

```
clear variables;

close all; clf; whitebg('w')

    axes('FontSize',18);

    fprintf('READING INPUT PLOTTING VALUES \n');

fprintf('1 is to plot n solution \n');
fprintf('2 is to plot s solution \n');

    fprintf('3 is to plot u solution \n');

    nsu = input('Type your choice \n');

    finaltime=2.0;

    snapshot=10;

    dt=10(-4);

    m=100;

    length=1.0;

    dx=length/m;

for i=0:(finaltime/(snapshot*dt))-1

    time(i+1)=i*dt;

end

for j=0:m

    x(j+1)=j*dx;

end
```

```

hold on;

fprintf('Begin plotting \n');

if nsu == 1

wmsol;

[r,c]=size(N);

ONE=ones(c,1);

Time=time'*ONE';

ONE1=ones(r,1);

X=x'*ONE1';

surf1(X',10*Time,N);

shading interp;

colormap autumn;

elseif nsu == 2

wssol;

[r,c]=size(S);

ONE=ones(c,1);

Time=time'*ONE';

ONE1=ones(r,1);

X=x'*ONE1';

surf1(X',10*Time,S)

shading interp;

colormap autumn;

elseif nsu == 3

```

```
wusol;

[r,c]=size(U);

    ONE=ones(c,1);

    Time=time'*ONE';

    ONE1=ones(r,1);

    X=x'*ONE1';

    surf1(X',10*Time,U)

shading interp;

colormap autumn;

    end

ylabel('Time t');

xlabel('Domain x(t)');
```