AGE-STRUCTURED POPULATION MODELS IN CYCLICAL NEUTROPENIA: A NUMERICAL INVESTIGATION.

by

Gordon Hiscott
Bachelor of Science: Mathematics

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the Department
of
Mathematics

SIMON FRASER UNIVERSITY
Spring 2012

APPROVAL

Name: Gordon Hiscott

Degree: Master of Science

Title of thesis: Age-structured population models in cyclical neutropenia: a numerical investigation.

Examining Committee:

Dr. Paul F. Tupper
Associate Professor, Mathematics
Simon Fraser University
Chair

Dr. Nilima Nigam
Associate Professor, Mathematics
Simon Fraser University
Senior Supervisor

Dr. Razvan Fetecau
Associate Professor, Mathematics
Simon Fraser University
Supervisor

Dr. John Stockie
Associate Professor, Mathematics
Simon Fraser University
SFU Examiner

Date Approved: April 25, 2012

Declaration of
Partial Copyright Licence

Abstract

Blood is composed of a variety of cells which play important roles in the health of an organism. Among these cells are white blood cells which are responsible for the body's immune response. An important type of white blood cell is the neutrophil. In this thesis, we investigate a model of cyclical neutropenia, a hematological disease characterized by abnormal oscillations in the neutrophil count of an organism. A standard treatment for this disease is to inject an apoptosis-inhibiting hormone, G-CSF, at periodic intervals.

Mathematical models to simulate the dynamics of neutrophil populations with and without G-CSF treatment were developed by C. Foley, [4]. These models include the populations in the cell line from stem cells to neutrophils, and consist of a nonlinear hyperbolic system of coupled integro-differential equations. The author then reduces the model to a system of delay differential equations which are then discretized to yield approximate solutions.

In this thesis, we first provide a quick overview of age-structured population models. We then discuss the origin of the PDE (partial differential equation) models in [4], and highlight some of their features which render their simulation very challenging. We describe some numerical approximation strategies employed by other authors for age-structured population models which did not converge for our model, and provide some insight into the reasons. We then discuss the modification of a splitting strategy, which does provide a convergent method for the system of PDE. We finally provide some numerical results, and compare our findings to those obtained in [4] on the DDE (delay differential equation) model.

Acknowledgments

I would like to thank to thank Nilima for all of the help, support, and encouragement that she has given me throughout my graduate school career. I would also like to thank my mother, father, and brother for all of their support.

Contents

List of Figures

Chapter 1

Introduction

In this thesis, we investigate the numerical approximation of a system of coupled nonlinear hyperbolic integro-differential equations. Such equations arise in the biological sciences as models of population dynamics where the constituent population is structured in some manner, [7, 2]. Our focus will be on a particular model of age-structured population dynamics, derived by Catherine Foley in [4], describing a blood disease called cyclical neutropenia.

1.1 Background: mathematical models of age-structured populations

It is worth briefly describing the development of simpler models of age-structured population models, before we discuss the more complex system which we will investigate in this thesis. Perhaps the simplest model of population dynamics is that due to Malthus, who assumed the members of a population were identical and did not compete for resources. With these assumptions, the population $u(t)$ at time t satisfies the Malthusian law

$$\dot{u} = \delta u, \tag{1.1}$$

where δ is the constant growth modulus. The assumption of non-competition is rather severe, and unrealistic. A refinement of the model was proposed by Verhulst, who allowed for the growth modulus to change linearly with increases in population.

$$\dot{u} = (\delta_1 - \omega_1 u)u, \qquad \omega_1, \delta_1 \text{ constant.} \tag{1.2}$$

If we set the population $C_o := \frac{\delta_1}{\omega_1}$, this ODE is solved by

$$u(t) = C_o \left(1 + \left(\frac{C_o}{P(0)} - 1\right) e^{-C_o t}\right)^{-1}. \tag{1.3}$$

As described in [7], $P = C_o$ is a stable equilibrium point of the ODE, and populations with initial size less than this will satisfy $\lim_{t\to\infty} P(t) = C_0$.

Both the models above assume all the members of the population are identical, and have the same birth/death rates. This assumption is clearly a poor one, for example, for cells which differentiate only upon reaching a maturation age. Better models which relax this assumption can classified into two categories:

- Discrete models, in which both the age and time variables are discrete, and updates in population only occur at discrete intervals.
- Continuous models, in which both the age and time variables are continuous.

In the next two subsections we give a brief description of these two classes of models. We restrict our attention to single-species models.

1.1.1 Discrete-time age-structured population models

In this section, we derive a discrete-time model for a single species, and follow the discussion by Cushing in Chapter 1 of the text [2].

Suppose there are n distinct classes of individuals in a population, and that transitions, births and deaths occur at discrete times $t = 0, 1, 2, 3...$ If the number of individuals in class i at time t are denoted by $u_i(t)$, we need to specify update rules for the *class distribution vector* $\vec{u}(t) = (u_1(t), ..., u_n(t))^T$.

Now let a_{ij} denote the fraction of individuals in class j who survive and *transition* to class i per unit time. This fraction may depend on time as well as the existing class distribution. Clearly $0 \leq a_{ij} \leq 1$, and $\sum_{i=1}^{n} a_{ij} = 1$ since individuals are only moving between classes, and not being created or destroyed. To account for births and deaths, let b_{ij} be the number of surviving offspring in class i, per j class individual, per unit of time. Again, this quantity may depend on time as well as the existing class distribution, and should be non-negative.

Assuming only births, deaths and inter-class transitions, the update rule for the class distribution vector is

$$\vec{u}(t+1) = (A + B)\vec{u}(t), \qquad t = 0, 1, ... \tag{1.4}$$

where $A = (a_{ij})$ is called the *transition matrix*, and $B = (b_{ij})$ is called the *fertility matrix*. The operator $P := A + B$ is called the *projection matrix*, and discrete models are studied by investigating properties of this matrix. For example, if P is linear and autonomous, and if the population is divided into n categories since birth, then we obtain the *Leslie matrix model*:

$$\vec{u}(t+1) = \begin{bmatrix} 0 & 0 & . & . & . & 0 & 0 \\ a_{21} & 0 & . & . & . & 0 & 0 \\ 0 & a_{32} & . & . & . & 0 & 0 \\ . & . & . & & & . & . \\ . & . & & . & & . & . \\ . & . & & & . & . & . \\ 0 & 0 & . & . & . & a_{n,n-1} & 0 \end{bmatrix} \vec{u} + \begin{bmatrix} b_{11} & b_{12} & . & . & . & b_{1,n-1} & b_{1,n} \\ 0 & 0 & . & . & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 & 0 \\ . & . & . & & & . & . \\ . & . & & . & & . & . \\ . & . & & & . & . & . \\ 0 & 0 & . & . & . & 0 & 0 \end{bmatrix} \vec{u} \quad (1.5)$$

The behaviour of the class distribution vector will then be determined by the spectrum of P.

1.1.2 Continuous-time age-structured population models

In this section, we derive a continuous-time model for a single species, and follow the treatment by Gurtin and MacCamy [7]. It is not difficult to see how this model can be retrieved from the discrete-time model, following the discussion by Cushing in Chapter 2 of the text [2].

We begin by defining a variable $u(a, t)$ which represents the number of individuals at time t whose age lies between a and $a + \Delta a$, where $\Delta a << 1$. If time changes by an increment of h, the group of individuals who are age a at time t will age by h units. Further, we assume $d(a, t)$ individuals of age a die out per unit age and time at time t. Therefore, the rate at which the population changed is

$$\lim_{h \to 0^+} \frac{u(a + h, t + h) - u(a, t)}{h} + d(a, t) = 0. \quad (1.6)$$

We can assume the death term is well-modelled by

$$d(a, t) = \lambda(a, U)u(a, t), \qquad U(t) := \int_0^\infty u(a, t)da. \quad (1.7)$$

Here $\lambda(a, U) \geq 0$ is called the *death modulus*, and is the death rate per unit time at age a, and depends on the total population $U(t)$ at time t. We assume births in the population are described by the so-called renewal equation:

$$u(0,t) = \int_0^\infty \beta(a,U)u(a,t)da, \tag{1.8}$$

where $\beta(a, U) \geq 0$ is the *birth modulus*, describing the number of offspring produced by an individual of age a per unit time.

Formally, $Du := \lim_{h\to 0^+} \frac{u(a+h,t+h)-u(a,t)}{h} = \frac{\partial u}{\partial t} + \frac{\partial u}{\partial a}$ only if u is at least C^1 in its arguments. With this assumption, we arrive at the continuous time model

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial a} + \lambda(a,U)u(a,t) = 0 \tag{1.9}$$

$$U(t) = \int_0^\infty u(a,t)da \tag{1.10}$$

along with (1.8) and an initial condition $u(a, 0) = \phi(a) \geq 0$.

At this juncture, we should ask: what are the important features of this equation? Under what conditions can well-posedness be proved, and can the tools used be generalized to the more complex model of interest in this thesis? Is there an equilibrium solution, and if so, what are its stability properties? We postpone some of these issues to Chapter 3. We point out that the proof of well-posedness of the model of interest is beyond the scope of this thesis, though we outline some of the key ideas behind such a proof.

1.1.3 Numerical methods

Since the work of Gurtin and MacCamy in their original paper, several authors have proposed numerical algorithms to simulate these systems. One of the more important contributions is Deborah Sulsky, who proposes a half-step strategy to update the renewal and advection terms in [14]. Unfortunately, this method did not prove convergent for the more complex system we studied, possibly because the system was not compatible with the method. Other notable works on numerical approximation for age-structured populations include up-winding methods (eg. [9]), spectral methods (eg. [1]), finite element and discontinuous Galerkin methods [11, 10] and splitting methods [8].

These equations typically allow for non-smooth solutions, and there are numerous works on models with some spatial diffusion, for example [6]. We do not discuss these, even though the numerical methods are very well developed for this class of models. Spatial diffusion is not relevant in our system.

1.2 Background: cyclical neutropenia

Blood is composed of a variety of cells. An important role is played by white blood cell, of which a neutrophil is a particular type. The neutrophils are responsible for preserving the immune system. Cyclical neutropenia is a hematological disease in which an individual's neutrophil count oscillates between normal and very low levels. This disease impacts on the health of the individual by undermining the body's defense against infection.

One form of treatment of this disease is the injection of granulocyte colony stimulating factor (G-CSF), beneath the skin of the diseased individual. G-CSF works by hindering the apoptosis, or programmed cell death, of the neutrophils. In order to optimize the neutrophil count without increasing the dosage of G-CSF, two rules must be followed. Firstly, the treatment should begin at the right time after the end of a period of decrease in the neutrophil count. Secondly, the period between each individual injection should be short enough so that oscillations remain stable and with small amplitudes, and long enough to ensure efficiency of the treatment.

A mathematical model of the cell line from stem cells to neutrophils is presented in [4]. The model is an age-structured compartment model with a compartment for each stage of the cell line. Mathematically, this is composed of five partial differential equations, or PDEs. The five stages modelled are resting stem cells, proliferative stem cells, prolifereative precursor cells, non-proliferative precursor cells, and neutrophils. The authors also consider a model derived from this PDE model, composed of delay differential equations (DDEs). Both of these models take into account the influence of G-CSF treatment on the cell line. In [4], results of simulations of the DDE models are presented, but not for any simulations of the PDE model. These simulations were done with a number of simplifying assumptions and estimates of parameters which we describe.

1.3 Goals in this thesis

The goals of this thesis include finding a stable and accurate algorithm to simulate the PDE model for cyclical neutropenia, which was not simulated in [4]. This was done by finding plausible algorithms used for other age-structured models, and then comparing the algorithms to each other in terms of results. After finding a stable algorithm, experiments are run to simulate the situations that were simulated with the DDE model in [4]. The

results of the algorithms could help determine which form of G-CSF treatment is the best of the treatment simulated in [4].

This thesis is organized as follows.

- In Chapter 2, we discuss the development of the PDE model as in [4].
- In Chapter 3, we provide an analytic look at the model and how it can be converted to the DDE model.
- We then discuss, in Chapter 4, numerical methods that were used in attempts to simulate solutions to the model.
- We then provide a look at modifications of existing approximation methods, such as a splitting method, in Chapter 5.
- In Chapter 6, we provide results of the splitting method using parameters based on experiments discussed in [4]
- In the final chapter, we give conclusions of the work on the PDE model as well as possible future work.

Chapter 2

Cyclical neutropenia and G-CSF treatment

In this chapter we provide a brief introduction to cyclical neutropenia and granulocyte-colony simulating factor (G-CSF) treatment. We then discuss two mathematical models developed by C. Foley in [4]. The simpler of these models is the focus of our study in this thesis.

2.1 The cell line and G-CSF treatment

In [4] on G-CSF treatment, Foley provides a mathematical model which models *cell populations and biochemical factors* during G-CSF treatment. The key cells employed in the model are *hematopoietic stem cells (HSC)* and blood cells (which include *neutrophils, platelets and erythrocytes*). The key biochemical factors involved in the dynamics are *G-CSF within the tissue* and *circulating G-CSF.*

The major mechanisms of interest include the proliferation of the hematopoietic stem cells, and the amplifications of each of the blood cell lines between the HSCs and the circulating blood cells. The G-CSF in the tissue, which is added to by subcutaneous injection, enters the circulation. Circulating G-CSF, if it does not return to the tissue, alters one of the following: the rate of apoptosis of the HSCs, the rate of differentiation from HSCs to neutrophils, and the amplification of the neutrophil line between the HSC and the circulating neutrophils. For these reasons, G-CSF is lost at a rate which depends on the neutrophil

count. The amplification factor of the neutrophil line affects the apoptosis and divisions of the HSCs that differentiate into neutrophils. After the amplification, there is an amount of time known as the transit time, which is the time a cell in the neutrophil line spends in the non-proliferative precursor stage. The erythrocyte line and the platelet line work similarly, each with their own rate of differentiation, amplification factor, transit time, and apoptosis rates. The HSCs that do not differentiate into blood cells enter the proliferate phase where they each either die or replicate. After a delay in the proliferate phase, the surviving cells and replicates enter the HSC concentration.

Also provided in [4] is a two-part model which models the production of white blood cells in one part and models the changing concentration of G-CSF in the other. In the white blood cell model, apoptosis rates of proliferate stem cells, proliferate precursor cells, and non-proliferate precursor cells are shown to be dependent on the concentration of G-CSF. Apoptosis of circulating neutrophils occurs independently of the G-CSF concentration. The rate at which resting stem cells reenter proliferation is dependent on the concentration of stem cells. On the other hand, the rate at which the resting stem cells differentiate into white blood cell precursors depends on the concentration of circulating white blood cells. The proliferate precursor cells are amplified during proliferation at a rate dependent on the G-CSF concentraion. After amplification, they become nonproliferate and remain so for an amount of time dependent on the aging velocity, which itself is dependent on the G-CSF concentration, before becoming white blood cells and entering circulation. In the G-CSF concentration model, two concentrations of G-CSF are employed: the concentration in the tissue and the concentration in the blood. The tissue G-CSF concentration is increased by injections of G-CSF as well as the flow of G-CSF from the blood to the tissue, the rate of which is not dependent on time or any concentrations. G-CSF also flows from the tissue to the blood at a rate independent of time and concentrations. The G-CSF concentration in the blood is increased by G-CSF production and decreased at a rate dependent on both the white blood cell concentration and the G-CSF concentration in the blood.

2.1.1 The mathematical model

In the model we consider, the independent variables are time t and cell age a, which is the amount of time a cell has spent in a particular phase. We neglect spatial effects since the system of blood is assumed to be well-mixed on a fast time time scale. The dependent variables of the model are the population densities at time t and age a of the following:

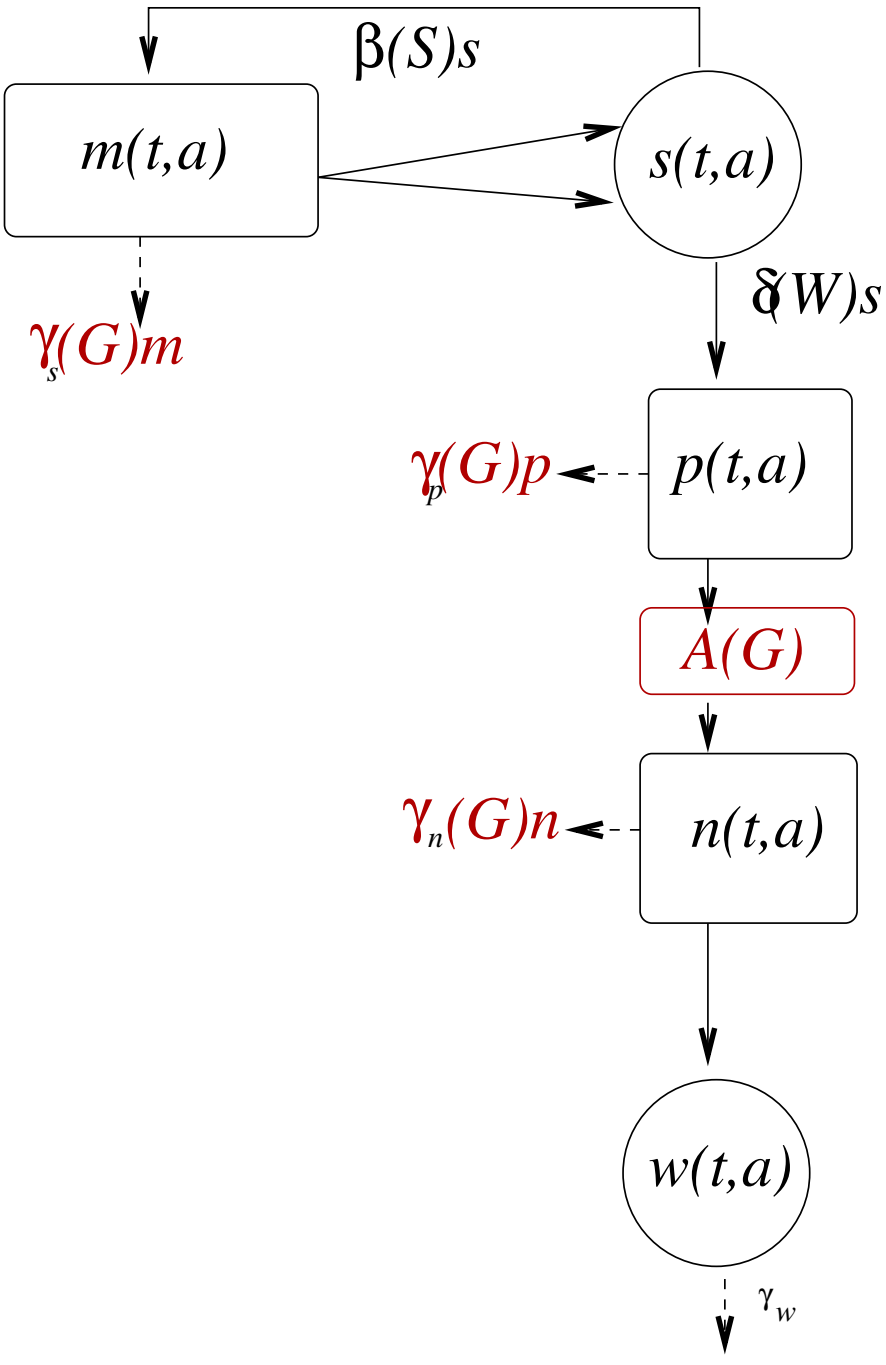

Figure 2.1: The cell line from stem cell to neutrophil. Source: [4]

- stem cells in the proliferative phase (or $m(t,a)$),
- resting stem cells (or $s(t,a)$),
- precursor cells in the proliferative phase (or $p(t,a)$),
- precursor cells in the non-proliferative phase (or $n(t,a)$),
- white blood cells (or $w(t,a)$).

We denote cell population totals at time t as follows:

$$M(t) = \int_0^{\tau_s} m(t,a)\,da;\ S(t) = \int_0^{\infty} s(t,a)\,da;\ P(t) = \int_0^{\tau_p} p(t,a)\,da;$$

$$N(t) = \int_0^{\tau_n} n(t,a)\,da;\ W(t) = \int_0^{\infty} w(t,a)\,da.$$

The model was developed under the following assumptions in [4]:

- Apoptosis occurs in each cell phase, except for the resting stem cell phase, at a rate of γ_i, for $i = s, p, n, w$. Furthermore, all of these apoptosis rates, except for γ_w, depend on G, the concentration of G-CSF. While γ_w is constant, γ_i where $i = s, p, n$ is assumed to be a decreasing linear function of the form $\gamma_i(x) = ({\gamma_i}^{min} - {\gamma_i}^{max})\frac{x}{8} + {\gamma_i}^{max}$.
- In each cell phase, cells age at velocity V_i, where $i = m, s, p, n, w$. Furthermore, V_m, V_s, V_p, V_w are all equal to 1, while V_n is dependent on G and of the form

  $$V_n(G) = (V_{max} - 1)\frac{G}{G + b_v} + 1.$$

  In addition, cells enter the non-proliferative precursor state at age $a = 0$ and exit the phase at age $a = \tau_n$.
- The rate at which resting stem cells differentiate into proliferative precursor cells depends on W and is a monotone decreasing Hill function of the form

  $$\delta(W) = f_0\frac{\theta_1}{\theta_1 + W}.$$

- Resting stem cells reenter proliferation at a rate given by a monotone decreasing function of the form

  $$\beta(S) = k_0\frac{{\theta_2}^2}{{\theta_2}^2 + S^2}.$$

  Furthermore, these cells enter proliferation at age $a = 0$ and exit proliferation at age $a = \tau_s$, after which each cell divides into two daughter cells.

- The cells that leave the proliferative precursor phase are amplified by a factor which is dependent on G and is of the form

$$A(G) = (A_{max} - A_{min})\frac{G}{G + b_A} + A_{min}.$$

Based on the assumptions, the PDE model for the cell dynamics can be written as follows:

$$\frac{\partial m}{\partial t} + \frac{\partial m}{\partial a} = -\gamma_s(G)m \qquad t > 0, a \in [0, \tau_s] \tag{2.1a}$$

$$\frac{\partial s}{\partial t} + \frac{\partial s}{\partial a} = -\delta(W)s - \beta(S)s \qquad t > 0, a > 0 \tag{2.1b}$$

$$\frac{\partial p}{\partial t} + \frac{\partial p}{\partial a} = -\gamma_p(G)p \qquad t > 0, a \in [0, \tau_p] \tag{2.1c}$$

$$\frac{\partial n}{\partial t} + V_n(G)\frac{\partial n}{\partial a} = -\gamma_n(G)n \qquad t > 0, a \in [0, \tau_n] \tag{2.1d}$$

$$\frac{\partial w}{\partial t} + \frac{\partial w}{\partial a} = -\gamma_w w; \qquad t > 0, a > 0 \tag{2.1e}$$

with initial conditions

$$m(0, a) = \phi_m(a) \qquad a \in [0, \tau_s] \tag{2.2a}$$

$$s(0, a) = \phi_s(a) \qquad a > 0 \tag{2.2b}$$

$$p(0, a) = \phi_p(a) \qquad a \in [0, \tau_p] \tag{2.2c}$$

$$n(0, a) = \phi_n(a) \qquad a \in [0, \tau_n] \tag{2.2d}$$

$$w(0, a) = \phi_w(a) \qquad a > 0 \tag{2.2e}$$

and with boundary conditions at $t = 0$ provided by the renewal equations

$$m(t, 0) = \beta(S(t))S(t),\ s(t, 0) = 2m(t, \tau_s),\ p(t, 0) = \delta(W(t))S(t),$$

$$n(t, 0) = A(G(t))p(t, \tau_p),\ w(t, 0) = n(t, \tau_n) \tag{2.3}$$

Foley obtained many of these parameters from the literature of hematological diseases, and inferred the remaining from numerical experiments. We note here that since Foley performed a further reduction of this model to a system of delay-differential equations, specific forms of the initial conditions are not specified in [4]. We shall return to this fact in Chapter 6.

We still have to describe the dynamics of the circulating G-CSF concentration, G, which affects the forcing terms in (2.1)- (2.2). These dynamics are described in the following section, in (2.4).

2.1.2 Equations and parameters of the canine model

In [4], the single-part model for canine cyclical neutropenia uses a set of parameter-based equations which will be discussed in this section. The model uses Q, N, R, and P to denote the respective concentrations of HSC's (in units $10^6 \frac{cells}{kg}$), neutrophils (in units $10^8 \frac{cells}{kg}$), erythrocytes (in units $10^{11} \frac{cells}{kg}$), and platelets (in units $10^{10} \frac{cells}{kg}$).The model uses A_i, γ_i, and κ_i to represent, respectively,the amplification factor of the cell line of i, the apoptosis rate of i, and the rate of differentiation from HSC's into the cell line of i, where $i = N, R, P$. Furthermore, the tissue concentration (in units $\frac{\mu g}{kg}$) of G-CSF is represented by X while the circulating concentration of G-CSF (in units $\frac{\mu g}{mL}$) is represented by G. The injections of G-CSF are modelled by an input function $I(t)$ which satisfies $\int_{t_a}^{t_b} I(t)dt = [\text{dosage of injection}]$, where t_a is a time briefly before the injection and t_b is a time briefly after the injection. The model equations, using the convention $Y_{\tau_v} = Y(t - \tau_v)$ to indicate the existence of a delay, are as follows:

$$\frac{dQ}{dt} = -\beta(Q)Q - (\kappa_N + \kappa_R + \kappa_P)Q + 2e^{-\gamma_S \tau_S}\beta(Q_{\tau_S})Q_{\tau_S}$$

$$\frac{dN}{dt} = -\gamma_N N - A_N \kappa_N(N_{\tau_N})Q_{\tau_N}$$

$$\frac{dR}{dt} = -\gamma_R R - A_R[\kappa_R(R_{\tau_{RM}})Q_{\tau_{RM}} - e^{-\gamma_R \tau_{RS}}\kappa_R(R_{\tau_{RM}+\tau_{RS}})Q_{\tau_{RM}+\tau_{RS}}]$$

$$\frac{dP}{dt} = -\gamma_P P - Ap_[\kappa_P(P_{\tau_{PM}})Q_{\tau_{PM}} - e^{-\gamma_P \tau_{PS}}\kappa_P(P_{\tau_{PM}+\tau_{PS}})Q_{\tau_{PM}+\tau_{PS}}]$$

The dynamics of circulating G-CSF are described by the coupled equations

$$\frac{dX}{dt} = I(t) + k_T G - k_B X \tag{2.4}$$

$$\frac{dG}{dt} = \frac{k_B}{V_B}X - k_T G - (\alpha_N + \gamma_G)G \tag{2.5}$$

Given that κ_i is dependent on the concentration of type i, there is feedback between the concentration of i and κ_i. This feedback is represented by negative feedback functions (negative in that as the concentration of a cell type increases, the corresponding differentiation rate decreases), which are given as follows:

$$\beta(Q) = k_0 \frac{\theta_2^s}{\theta_2^s + Q^s};\ \kappa_N(N) = f_0 \frac{\theta_1^n}{\theta_1^n + N^n};\ \kappa_P(P) = \frac{\bar{\kappa}_p}{1 + K_p P^r};\ \kappa_R(R) = \frac{\bar{\kappa}_r}{1 + K_r R^{m_e}}$$

It is stated in [4] that as G increases, so do A_N, γ_S, and θ_1; therefore, by using the data from the experiments to obtain estimated values of each parameter prior to and after G-CSF

treatment, the parameters are written as functions of G of the following forms :

$$A_N = A_N^{untr}(1 - H(t-d)) + H(t-d)(m_A(G - \bar{G}) + A_N^{tr})$$

$$\gamma_S = \gamma_S^{untr}(1 - H(t-d)) + H(t-d)(m_g(G - \bar{G}) + \gamma_S^{tr})$$

$$\theta_1 = \theta_1^{untr}(1 - H(t-d)) + H(t-d)(m_t(G - \bar{G}) + \theta_1^{tr})$$

The function $H(t)$ is the Heaviside step function, meaning that it satisfies $H(t) = 0$ if $t \leq 0$ and $H(t) = 1$ if $t > 0$. Furthermore, the superscripts "tr" and "untr" are used to indicate data from a treated individual and an untreated individual, respectively. Also, $\bar{G}$ denotes the average value of G for each data set used. The values m_A, m_g, and m_t are slopes of the form, while using parameter m:

$$m_A = m(A_N^{tr} - A_N^{untr})/\bar{G}$$

$$m_g = m(\gamma_S^{tr} - \gamma_S^{untr})/\bar{G}$$

$$m_t = m(\theta_1^{tr} - \theta_1^{untr})/\bar{G}$$

Although the main PDE model in [4] focuses on the cell line from HSC's to neutrophils, the canine model takes into account the effects of the erythrocyte and platelet populations on the HSC population. This makes the canine model more realistic, but it also makes a model that is more difficult to simulate solutions of. Therefore, we simulate the model based on (2.1) since our focus is on the neutrophil population. However, there are functions from the canine model that either are used or can be used in simulations of the main model, such as the input function and the negative feedback functions.

Chapter 3

Analytical observations

We can write the model of interest, (2.1) to (2.3), in abstract form as follows: find $\vec{u}$ with non-negative components such that

$$\mathcal{D}\vec{u} = \mathcal{L}(\vec{u}, \vec{U}, G)\vec{u}, \qquad \vec{u}(0,a) = \vec{\phi}, \qquad \vec{u}(t,0) = \mathcal{I}(\vec{u}, \vec{U}) \tag{3.1}$$

where the directional derivative $\mathcal{D}\vec{u} = \vec{u}_t + \mathcal{V}(G)\vec{u}_a$ if $\vec{u}$ is smooth enough in both its arguments, and $\vec{U}(t)$ is the vector of total cell populations at time t. The operator $\mathcal{L}(\vec{u}, \vec{U}, G)$ contains the information about the transition/apoptosis rates, and $\mathcal{I}$ describes the renewal rates. The vector-valued function $\vec{u} = (m, s, p, n, w)^T$ is not simply a map from $[0,T]\times\mathbb{R}^+ \to \mathbb{R}^{5,+}$. This is because the different components of $\vec{u}$ are well-defined for different age lengths. In particular, we may seek $m \in C([0,T]\times[0,\tau_s])$, $s \in C([0,T]\times[0,\infty)$, $p \in C([0,T]\times[0,\tau_p])$, $n \in C([0,T]\times[0,\tau_n)andw \in C([0,T]\times[0,\infty)$. It is obvious that the analysis of this hyperbolic integro-differential system will not be a straightforward extension of that of general nonlinear first order PDE.

In this chapter, we first discuss approximations of solutions to the PDE model above using the Method of Characteristics as explained in [13]. We will also discuss the conversion of the PDE model into a system of delay differential equations, or DDEs.

We then summarize some prior analytical results concerning a scalar age-structured population model. We point to the challenges in extending this analysis to the model of cyclical neutropenia.

3.1 The Method of Characteristics

We first consider a prototypical age-structured model for a single population $u(a,t)$. This satisfies the PDE

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial a} = f(t,a,u) \qquad t \geq 0, a \in [0,\tau_u] \tag{3.2a}$$

$$u(0,t) = v(t,u) \qquad t \geq 0 \tag{3.2b}$$

$$u(a,0) = \phi_u(a) \qquad a \in [0,\tau_u] \tag{3.2c}$$

where τ_u is equal to the terminal age of the cell if the cell has a finite terminal age. In the case that the cell has no terminal age, τ_u is infinite and we assume that the population density of the cell at any time approaches 0 as a approaches infinity.

In equation (3.2a), if we let $f(t,a,u) = -Bu$, then:

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial a} = -Bu$$

Multiplying by the integrating factor $e^{Bt}u$, we can rewrite the PDE as

$$\frac{\partial}{\partial t}(e^{Bt}u) + \frac{\partial}{\partial a}(e^{Bt}u) = 0$$

which has a solution $U_u = e^{Bt}u$. This has a gradient $\bigtriangledown U_u = \frac{\partial U_u}{\partial t}\hat{j} + \frac{\partial U_u}{\partial a}\hat{i}$, where $\hat{i}$ is the unit vector in the direction of a and $\hat{j}$ is the unit vector in the direction of t. By calculating the rate of change of U_u in the direction of the vector $1\hat{i} + 1\hat{j}$, we obtain $\bigtriangledown U_u \cdot [1\hat{i} + 1\hat{j}] = \frac{\partial}{\partial t}(U_u) + \frac{\partial}{\partial a}(U_u) = 0$, which means that U_u is constant along curves parallel to $1\hat{i}+1\hat{j}$. These curves are the characteristic curves of equation (3.2a) and can be written as $a-t=c$, where c is any real number. For two distinct constants c_1 and c_2, $U_u(a,t) = k_1$, if $a - t = c_1$, and $U_u(a,t) = k_2$, if $a - t = c_2$, for real numbers k_1 and k_2. Then, $U_u(a,t) = g(a-t)$ for some function g of one variable and given that $g(a) = U_u(a,0) = \phi_u(a)$, it must be that $g = \phi_u$ and that $U_u(a,t) = \phi_u(a-t)$ for $t < a$. We then obtain $u(a,t)$ for $t < a$ as follows:

$$U_u(a,t) = \phi_u(a-t) = U_u(a-t,0)$$

$$e^{Bt}u(a,t) = \phi_u(a-t)$$

$$u(a,t) = \phi_u(a-t)e^{-Bt}$$

In order to obtain the characteristic values for $t > a$, we need to use the boundary condition $u(0,t) = v(t,u)$, since $t > 0$. Therefore, $U_u(0,t) = e^{Bt}u(0,t) = e^{Bt}v(t,u)$ and $U_u(a,t) =$

$e^{B(t-a)}v(t-a,u) = U_u(0,t-a)$. The form of $u(a,t)$ for $t > a$ is subsequently obtained as follows:

$$U_u(a,t) = e^{B(t-a)}v(t-a,u) = U_u(0,t-a)$$

$$u(a,t) = \frac{e^{B(t-a)}v(t-a,u)}{e^{Bt}} = e^{-Ba}v(t-a,u)$$

Therefore, $u(a,t) = \phi_u(a-t)e^{-Bt}$ for $t < a$ and $u(a,t) = e^{-Ba}v(t-a,u)$ for $t > a$. Clearly any discontinuities in the initial data will persist along characteristics. Looking at the PDE model in (2.1), we observe the equations for the five cell population densities in the form of equations (3.2a) - (3.2c) and we could imagine using the Method of Characteristics for these. Unfortunately, this approach does not work. First, the equation (2.1d) has a variable coefficient, rendering strategy above invalid. One can address this issue (see below), but there are other challenges.

Let us consider a variable-coefficient scalar equation. Assumption that a cell's population density $x(t,a)$, at time t and age a satisfies the following equations:

$$\frac{\partial x}{\partial t} + V_x(t)\frac{\partial x}{\partial a} = -\gamma_x(t)x \qquad t > 0, a \in [0,\tau_x] \tag{3.3a}$$

$$x(t,0) = H_x(t) \qquad t > 0 \tag{3.3b}$$

$$x(0,a) = \phi_x(a) \qquad a \in [0,\tau_x] \tag{3.3c}$$

We then find an expression for $x(t,\tau_x)$ by using the method of characteristics. We define a new variable s so that $x(s) = x(t(s),a(s))$, which yields

$$\frac{dx}{ds} = \frac{\partial x}{\partial t}\frac{dt}{ds} + \frac{\partial x}{\partial a}\frac{da}{ds} = -\gamma_x(t)x$$

From this equation, we obtain the following

$$\frac{dt}{ds} = 1, \frac{da}{ds} = V_x(t), \frac{dx}{ds} = -\gamma_x(t)x$$

We subsequently obtain, through integration,

$$t(s) = t(0) + s$$

$$a(s) = a(0) + \int_0^s V_x(y)dy$$

$$x(s) = x(0)e^{-\int_0^s \gamma_x(t(y),a(y))dy}$$

We next let C be the characteristic curve which begins at $(t, a) = (0, 0)$ and is defined by $C = \{(t, a)|t(s) = s, a(s) = \int_0^s V_x(y)dy, s \in [0, s_\tau]\}$, where s_τ satisfies the equation $\tau_x = \int_0^{s_\tau} V_x(y)dy$. The solution $x(t, a)$, and therefore the expression of $x(0)$, depends on which of the two disjoint regions of the (t, a)-plane separated by C it lies within. The two regions are R_1, the region which contains the a-axis and R_2, the region which contain the t-axis.

If $(t(0), a(0)) \in R_1$, then $t(0) = 0$, which means that $t(s) = s$ and that $a(s) = a(0) + \int_0^s V_x(y)dy$. By using the initial condition, we see that $x(0) = \phi_x(a - \int_0^s V_x(y)dy)$ and that, therefore,

$$x(t, \tau_x) = \phi_x(\tau_x - \int_0^s V_x(y)dy)e^{-\int_0^t \gamma_x(y)dy}$$

If $(t(0), a(0)) \in R_2$, then $a(0) = 0$, meaning that $a(s) = \int_0^s V_x(y)dy$ and $t(s) = t(0) + s$; furthermore, by using the boundary condition, we obtain $x(0) = H(t - s)$. We then obtain an expression for s based on the expression of $a(s)$ which can be written as follows

$$a(s) = \int_0^s V_x(t(y))dy = \int_0^s V_x(t(0) + y)dy = \int_{t(0)}^{t(0)+s} V_x(z)dz$$

Therefore, for some value T_τ,

$$\tau_x = \int_0^{T_\tau} V_x(y)dy = \int_{t-T_\tau}^{t} V_x(y)dy$$

and subsequently

$$x(t, \tau_x) = H_x(t - T_\tau)e^{-\int_0^{T_\tau} \gamma_x(y)dy}$$

The solution $x(t, \tau_x)$, by the method of characteristics, is of the form

$$x(t, \tau_x) = \phi_x(\tau_x - \int_0^s V_x(y)dy)e^{-\int_0^t \gamma_x(y)dy} \qquad \text{for } (t, a) \in R_1 \tag{3.4}$$

$$= H_x(t - T_\tau)e^{-\int_0^{T_\tau} \gamma_x(y)dy} \qquad \text{for } (t, a) \in R_2 \tag{3.5}$$

Note that x could be discontinuous across the interface between R_1 and R_2, unless the initial data is compatible.

We are still unable to use the method of characteristics on (2.1) with the renewal conditions (2.3) to obtain a closed-form solution. The PDE are nonlinearly coupled through the renewal equations (2.3). Additionally, since some of the populations have a finite (and different) exit times τ_u, this method cannot be used to approximate the entire model.

3.2 PDE to DDE Conversion

In [4], there is a DDE model which models the behaviour of the cells involved in the production of white blood cells and is related to the PDE model. In order to obtain the DDE model from the PDE model, a process of conversion is used starting from the PDE model. Consider again the scalar model for populations considered in the previous section. Equation (3.3a) is integrated with respect to a, resulting in

$$\int_0^{\tau_x} \frac{\partial x(t,a)}{\partial t} da + \int_0^{\tau_x} V_x(t) \frac{\partial x(t,a)}{\partial a} da = -\int_0^{\tau_x} \gamma_x(t) x(t,a) da$$

This equation is rewritten in terms of the variable $X(t)$, the total number of cells at t, which satisfies $X(t) = \int_0^{\tau_x} x(t,a)da$:

$$\frac{dX}{dt} + V_x(t)[x(t,\tau_x) - x(t,0)] = -\gamma_x(t)X(t) \tag{3.6}$$

By rearranging the terms and substituting the boundary condition into equation (3.6), we obtain

$$\frac{dX}{dt} = V_x(t)[H_x(t) - x(t,\tau_x)] - \gamma_x(t)X(t)$$

[4] only considers the case where $(t,a) \in R_2$, due to focus on long term behaviour of cells and the fact that R_2 contains the t-axis. Therefore, $x(t,\tau_x)$ takes the form

$$x(t,\tau_x) = H_x(t - T_\tau) e^{-\int_0^{T_\tau} \gamma_x(y)dy}$$

and by substituting this equation into the equation for $\frac{dX}{dt}$, we obtain the general solution for X(t) which is of the form

$$\frac{dX}{dt} = V_x(t)[H_x(t) - H_x(t - T_\tau) e^{-\int_0^{T_\tau} \gamma_x(y)dy}] - \gamma_x(t)X(t)$$

In [4], this process is applied to obtain DDEs for cell population totals S(t), P(t), N(t), and W(t). (2.1b) is integrated and simplified by using its boundary condition and by assuming that $\lim_{a\to+\infty} s(t,a) = 0$. The resulting system is as follows:

$$\frac{dS}{dt} + \lim_{a\to+\infty} s(t,a) - s(t,0) = -[\delta(W(t)) + \beta(S(t))]S(t),$$

or

$$\frac{dS}{dt} = 2m(t,\tau_s) - [\delta(W(t)) + \beta(S(t))]S(t) \tag{3.7}$$

The expression for $m(t,\tau_s)$ is then obtained by solving (2.1a) using its boundary condtion and takes the form

$$m(t,\tau_s) = m(t-\tau_s,0)e^{-\int_0^{\tau_s}\gamma_s(G(t))dt} = \beta(S(t-\tau_s))S(t-\tau_s)e^{-\int_0^{\tau_s}\gamma_s(G(t))dt}$$

As in [4], $X_{\tau_x} = X(t-\tau_x)$ shall be used to denote a variable with a delay. Therefore, the above equation becomes $m(t,\tau_s) = \beta(S_{\tau_s})S_{\tau_s}e^{-\int_0^{\tau_s}\gamma_s(G(t))dt}$. Next, integrating (2.1c) in the age variable, we obtain

$$\frac{dP}{dt} = -p(t,\tau_p) + \delta(W(t))S(t) - \gamma_p(G(t))P(t)$$

By the method of characteristcs,

$$p(t,\tau_p) = \delta(W_{\tau_p})S_{\tau_p}e^{-\int_0^{\tau_p}\gamma_p(G(t))dt}$$

which leads to the new equation

$$\frac{dP}{dt} = -\delta(W_{\tau_p})S_{\tau_p}e^{-\int_0^{\tau_p}\gamma_p(G(t))dt} + \delta(W(t))S(t) - \gamma_p(G(t))P(t) \tag{3.8}$$

As before we integrate (2.1d) with respect to a to obtain

$$\frac{dN}{dt} = -V_n(G)[n(t,\tau_n) - A(G(t))p(t,\tau_p)] - \gamma_n(G(t))N(t)$$

where, for T_n satisfying $\tau_n = \int_{t-T_n}^{t} V_n(G(y))dy$ and by the method of characteristics,

$$n(t,\tau_n) = A(G_{T_n})\delta(W_{\tau_p})S_{\tau_p}e^{-\int_0^{\tau_p}\gamma_p(G(t))dt - \int_0^{T_n}\gamma_n(G(t))dt}$$

Therefore, the DDE for $N(t)$ is of the form

$$\frac{dN}{dt} = V_n(G)\delta(W_{\tau_p})S_{\tau_p}e^{-\int_0^{\tau_p}\gamma_p(G(t))dt}[A(G) - A(G_{T_n})e^{-\int_0^{T_n}\gamma_n(G(t))dt}] - \gamma_n(G(t))N(t) \tag{3.9}$$

The same process is used again and in addition to the assumption that $\lim_{a\to+\infty} w(t,a) = 0$ to obtain the following DDE for $W(t)$:

$$\frac{dW}{dt} = A(G_{T_n})\delta(W_{\tau_p})S_{\tau_p}e^{-\int_0^{\tau_p}\gamma_p(G(t))dt - \int_0^{T_n}\gamma_n(G(t))dt} - \gamma_p W(t) \tag{3.10}$$

This thesis focuses mainly on the PDE model and on methods to approximate solutions to it. The chapters that follow provide a look into what has not been shown in [4], including methods of simulating solutions to the PDE model whereas the simulations in [4] are related to the DDE model (3.7)-(3.10).

3.3 Well-posedness

We review some classical results [7] for the scalar model

$$Du + \lambda(a, U) = 0, \qquad U(t) = \int_0^\infty u(a,t)\, da, \tag{3.11}$$

with the renewal equation,

$$u(0,t) = \int_0^\infty \beta(a,U)u(a,t)\, da, \tag{3.12}$$

along with the initial condition

$$u(a,0) = \phi(a). \tag{3.13}$$

The model is only biologically meaningful if λ, β, ϕ, u are non-negative. Without more assumptions we cannot, of course, expect solutions of (3.11) to be globally differentiable. When u is smooth, $Du = u_t + u_a$.

Gurtin and MacCamy provide a careful analysis of this model, and begin by precisely defining a notion of solution:

Definition 1. *A function $u : \mathbb{R}^+ \times [0,T] \longrightarrow \mathbb{R}, u \geq 0$ with the properties that $u(\cdot,t) \in L^1(\mathbb{R}^+)$, Du is well-defined on $\mathbb{R}^+ \times [0,T]$, $U(t)$ is continuous and such that it satisfies (3.11) on $\mathbb{R}^+ \times (0,T)$ and (3.12) on $0 < t \leq T$, is called a solution of the population problem up to time T.*

Note in this definition, the renewal equation (3.12) need not hold at $t = 0$. There is no *a priori* reason to expect that the initial data $\phi(a)$ will satisfy (3.12). If it does, we say the initial data satisfies a *compatibility condition.* From the discussion in the previous section, it is clear that discontinuities in the initial data will move along the characteristics. Additionally, unless the initial data is compatible, solutions will be discontinuous across the characteristic $t = a$.

In [7], the authors proceed to reformulate eqs. (3.11) to (3.13) as a pair of integral equations for $B(t) := u(0,t)$ and $U(t)$:

$$U(t) = \int_0^t K(t-a,t;U)B(a)\, da + \int_0^\infty L(a,t;U)\phi(a)\, da \tag{3.14}$$

$$B(t) = \int_0^t \beta(t-a,U)K(t-a,t,U)B(a)\, da + \int_0^\infty \beta(a+t,U)L(a,t,U)\phi(a)\, da \tag{3.15}$$

with kernels

$$K(x,t,U) = \exp(-\int_{t-x}^{t} \lambda(a+\tau-t, U(\tau))\, d\tau) \tag{3.16}$$

$$L(x,t,U) = \exp(\int_{0}^{t} \lambda(t+x, U(\tau))\, d\tau) \tag{3.17}$$

If λ, β do not depend on U, the equation for B reduces to the linear integral equation of Lotka. The authors show that solving eqs. (3.14) and (3.15) is equivalent to solving their scalar model. Solvability of eqs. (3.14) and (3.15) for finite time T is then demonstrated by a contraction mapping argument. The argument is broadly the following: if U is fixed, equation (3.15) is a linear Volterra integral equation, and solvable for finite T. Denote this solution by $B(t) = \mathcal{B}(U)(t)$. The operator $\mathcal{B}(U)(t)$ is a contraction map on the Banach space of non-negative continuous functions on $[0,T]$ equipped with the sup-norm. This operator can then be used in (3.14) to define an integral operator $\mathcal{P}(U)(t)$. Under some further assumptions on the birth rate and transition rates, it can be shown that $\mathcal{P}(U)(t)$ is a contraction map on the same Banach space. This establishes the existence of fixed points for both maps, and hence, for the system of integral equations.

We have not been able to extend this analysis to the far more complex situation of (2.1), primarily given the differing final ages. Additionally, we are interested in simulating the effects of G-CSF treatment, which adds complications to the definition of the integral operators. We defer the analysis of well-posedness to future work.

The authors also study equilibrium age distributions, and their stability, for various cases of transition rates and birth rates.

3.4 Equilibrium age distribution

While the question of interest in the thesis is how to numerically simulate the dynamics of neutrophils with G-CSF treatment, we can analytically study the equilibrium age distribution of the cell line in the absence of such treatments. We ask: is there a time-independent solution of (2.1), when no G-CSF treatment is applied? This question is addressed by examining the equilibrium age equations. Following the description at the beginning of this chapter, we are interested in studying

$$\vec{u}_a = \mathcal{L}(\vec{u}, \vec{U})\vec{u}, \qquad \vec{u}(0) = \mathcal{I}(\vec{u}, \vec{U}). \tag{3.18}$$

The operator $\mathcal{V}(G)$ reduced to the identity matrix in the equilibirum case.The $\vec{U}$ is now the constant vector of total cell populations. The operator $\mathcal{L}(\vec{u}, \vec{U}, G)$ contains the information about the transition/apoptosis rates and reduces to a constant matrix. Again, $\mathcal{I}$ describes the renewal rates. Unfortunately, since the components of the equilibrium age distribution obey ODE over different age-lengths, the study of this system does not follow that of typical ODE system. To highlight the challenge, we write out the system explicitly:

$$\frac{\partial m}{\partial a} = -\gamma_s m \qquad a \in [0, \tau_s] \tag{3.19a}$$

$$\frac{\partial s}{\partial a} = -\delta(W)s - \beta(S)s \qquad a > 0 \tag{3.19b}$$

$$\frac{\partial p}{\partial a} = -\gamma_p p \qquad a \in [0, \tau_p] \tag{3.19c}$$

$$\frac{\partial n}{\partial a} = -\gamma_n n \qquad a \in [0, \tau_n] \tag{3.19d}$$

$$\frac{\partial w}{\partial a} = -\gamma_w w; \qquad a > 0 \tag{3.19e}$$

and with conditions at $a = 0$ provided by the (time-independent) renewal equations

$$m(0) = \beta(S)S, \qquad s(0) = 2m(\tau_s), \qquad p(0) = \delta(W)S, \qquad n(0) = Ap(\tau_p), \qquad w(0) = n(\tau_n) \tag{3.20}$$

The total cell populations M, S, W, N, P are now constants. Further, the apoptosis rates γ_i and the transition rates V_i are constants as well, since we assume $G = 0$.

Clearly the cell populations are all decaying as functions of a. We are interested in non-trivial solutions. Fortunately, the form of the ODE suggests a form for the solutions. To this end, we use the ansatz

$$m(a) = m_0 \exp(-\gamma_s a), s(a) = s_0 \exp(-(\delta(W) + \beta(S))a), p(a) = p_0 \exp(-\gamma_p a),$$

and $n(a) = n_0 \exp(-\gamma_n a), w(a) = w_0 \exp(-\gamma_w a)$. Note this determines the total cell populations, for example,

$$W = \int_0^\infty w_0 \exp(-\gamma_w a)\, da = \frac{w_0}{\gamma_w}.$$

We obtain the following equations for the constants m_0, s_0, p_0, n_0, w_0 and S:

$$S = s_0 \frac{1}{\delta(W) + \beta(S)} = s_0 \left(\frac{f_0 \theta_1}{\theta_1 + w_0/\gamma_w} + \frac{k_0 \theta_2^2}{\theta_2^2 + S^2} \right)^{-1} \tag{3.21a}$$

$$m_0 = \beta(S) S = \frac{k_0 \theta_2^2 S}{\theta_2^2 + S^2} \tag{3.21b}$$

$$s_0 = 2 m_0 e^{-\gamma_s \tau_a} \tag{3.21c}$$

$$p_0 = \delta(W) S = \frac{f_0 \theta_1}{\theta_1 + w_0/\gamma_w} S \tag{3.21d}$$

$$n_0 = A p_0 e^{-\gamma_p \tau_p} \tag{3.21e}$$

$$w_0 = A n_0 e^{-\gamma_n \tau_n} \tag{3.21f}$$

From this system, it can be verified that one possible equilibrium solution is the trivial one. We need to find non-trivial solutions. If we specify that the equilibrium population totals are fixed and non-zero, it is readily seen that the system above cannot be consistently solved unless the other parameters in the problem satisfy some additional constraints. For example, if S and W are given, then the equations for m_0, s_0 yield that

$$\beta(S)[2e^{-\gamma_s \tau_a} - 1] = \delta(W).$$

Since $\beta(S), \delta(W)$ are strictly positive for finite values of their arguments, this implies a condition on $\gamma_s \tau_a$. However, these parameters are determined from physiological conditions, and cannot be chosen independently. Similar reasoning for the other variables leads us to conclude that in general, determining the equilibrium age population given specified total populations will not be possible except in very specific circumstances.

We end this section by noting that while the study of the equilibrium age distributions is important, the actual model of interest operates at conditions far from equilibrium due to the injection of G-CSF. We therefore do not pursue the issue of equilibrium age distributions further in this thesis.

Chapter 4

What numerical methods did not work, and why

In this section, we discuss the numerical methods used when attempting to approximate solutions to the PDE model. Several methods which are usually used for age-structured population models were unstable, unfortunately, and we look at how and why they do not work. Throughout this chapter, the following notation will be used: k for the length of the step in t used for the approximations, h for the length of the step in a, $t_n = nk$ where $0 \leq n$, and $a_j = jh$ where $0 \leq j \leq \frac{A}{h}$ for A the maximum age.

4.1 An upwind scheme

The first algorithm we attempted to use is an upwind scheme where the solution values at one time step are approximated by the values at the previous time step. An upwind scheme for the scalar age-structured population model was presented, for example, in [9]

For each of these experiments, we chose to make the step in tme equal to the step in age because, as stated in Section 3.1, the characteristic curves are parallel to the line $1\hat{i} + 1\hat{j}$. If k and h were not equal, then the direction of the scheme would not be parallel to $1\hat{i} + 1\hat{j}$.

In [9], there are theorems involving the stability and convergence of an upwind scheme

for a PDE model which consists of the following equations:

$$v(t) = \int_0^{a_\dagger} u(a,t)da, \qquad t > 0, \tag{4.1}$$

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial a} + \mu(a, v(t)) = 0, \qquad 0 < a < a_\dagger, \qquad t > 0, \tag{4.2}$$

$$u(0,t) = \int_0^{a_\dagger} \beta(a, v(t))u(a,t)da, \qquad t > 0, \tag{4.3}$$

$$u(a,0) = u_0(a), \qquad 0 \le a < a_\dagger. \tag{4.4}$$

Here, $N = \lfloor \frac{T}{k} \rfloor$ for final time T, $a_\dagger$ is the maximum age, $J = \lfloor \frac{a_\dagger}{h} \rfloor$

The upwind scheme itself is as follows

$$U_j^0 = u_0(a_j), \qquad 0 \le j \le J; \qquad V^0 = \sum_{j=1}^{J} U_j^0 h;$$

$$\frac{U_j^n - U_j^{n-1}}{k} + \frac{U_j^{n-1} - U_{j-1}^{n-1}}{h} + \tilde{\mu}_j^{n-1} U_j^n = 0, \qquad 1 \le l < J + n, \qquad 1 \le n \le N;$$

$$U_{J+n}^n = \frac{\frac{k}{h} U_{J+n-1}^{n-1}}{1 + k\tilde{\mu}_{J+n}^{n-1}}, \qquad 1 \le n \le N$$

$$V^n = \sum_{l=1}^{J+n} U_l^n h; \qquad U_0^n = \sum_{l=1}^{L+j} \tilde{\beta}_l^n U_l^n h, \qquad 1 \le n \le N$$

It is assumed that $U_j^n = 0$ for $n \ge 0$ and $j > J + n$.

The following hypotheses are assumed in [9] for natural death rate μ_n and μ_e, which is the component of the death rate which takes into account external pressures:

1. $u_0 \in C^0[0, a_\dagger)$ and u_0 is nonnegative and compactly supported in $[0, A_1]$ for $A_1 < a_\dagger$.
2. β, $\mu \in C^1(0, a_\dagger) \times (0, \infty)$.
3. $0 \le \beta(a, v) \le \overline{\beta} < \infty$ and $0 \le \mu(a, v)$.
4. $\partial_v \beta \le 0$.
5. $\mu = \mu_n(a, v) + \mu_e(v)$, for $\mu_n, \mu_e \in C^1$, $\mu_n \ge 0$, $\partial_v \mu_n \ge 0$, and $\partial_v \mu_e > 0$.
6. $\beta(a, v) = \mu_n(a, v) = 0$, for $a \ge A_2 > 0$.
7. $\mu_e(0) > 0$ and $\nu > \overline{\beta}$ for $\nu = lim_{v\to\infty} \mu_e(v) \in (0, +\infty]$.

The theorem for the stability of the scheme states that if hypotheses 1 through 3 are satisfied, then there exists $C = C(T) > 0$ such that $0 \leq U_j^n \leq C$ and $0 \leq V^n \leq C$ for all n and j such that $0 \leq n \leq N$ and $0 \leq j \leq J + n$. The same theorem also states that if hypotheses 4 through 7 are satisfied, then there exists a value $C^* > 0$ which is independent of T, h, and k and which satisfies $0 \leq U_j^n \leq C^*$ and $0 \leq V^n \leq C^*$ for all n and j such that $0 \leq n \leq N$ and $0 \leq j \leq J + n$. There is a theorem in [9] for the convergence of the scheme which states that for $u \in C^2$, if T is fixed, then there is a positive value C which is independent of h and which satisfies

$$\max_{0 \leq n \leq N} \max_{0 \leq j \leq J+n} |u_j^n - U_j^n| + \max_{0 \leq n \leq N} |v^n - V^n| \leq C$$

Unfortunately, these theorems cannot be applied to the model we are working on because none of the boundary conditions and mortality rates are of the form of those in the upwind scheme.

4.1.1 Numerical experiments with the upwind scheme

Using the filgrastim parameter estimates from [4], we designed an up-winding algorithm to simulate Foley's PDE model. The algorithm has been tested using different sizes of k and h in order to check for convergence. The values of k and h used in this test are 0.2, 0.1, 0.05, 0.025, and 0.125.

According to figure 4.1, as the time step and age step decrease, the curve between t values 6 and 10 goes higher until the step length reaches some value between .05 and .025, after which the curve is below $M = 0.5$. In figure 4.2, after the step length reaches some value between 0.05 and 0.025, there is no longer a peak around $t = 2.8$. As in figures 4.1 and 4.2, the curve in figure 4.3 changes drastically when the step length reaches some value b between 0.05 and 0.025. For step lengths greater than b, the curve has a peak between $t = 4$ and $t = 6$. For step lengths less than b, there is a peak and the curve decays to zero. As the step length decreases from some value between 0.1 and 0.05, the highest peak value in the graphs in 4.4 increases to unrealistically high levels, with no sign of stability. Similarly in figure 4.5, after somewhere between $k = 0.1$ and $k = 0.05$, the highest peak in the graphs increases as the step length decreases, leading to more abnormally high values.

Based on these graphs, the upwind algorithm using the equations from the PDE model is not convergent, which means that a new algorithm is needed. Similar results are obtained

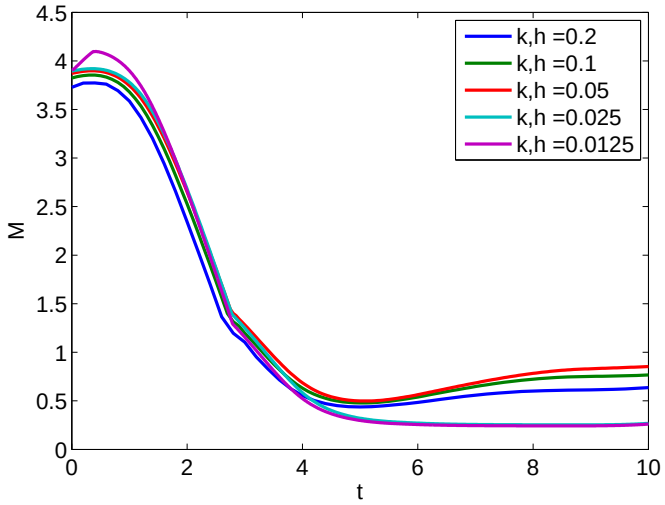

Figure 4.1: Graph for population total M, using an up-winding scheme.

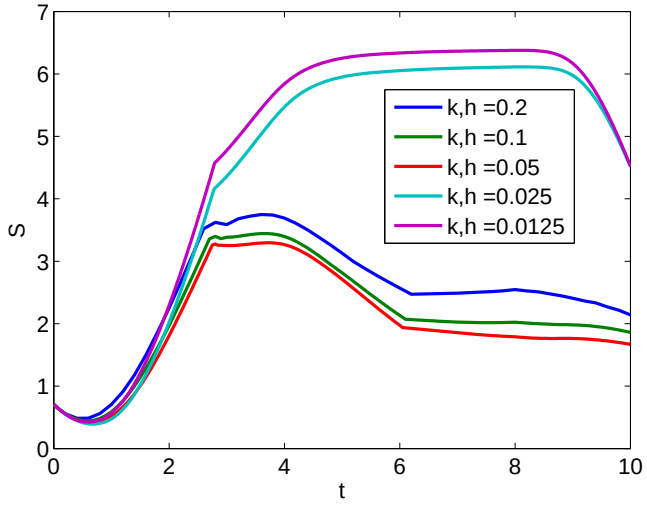

Figure 4.2: Graph for S, using an up-winding scheme.

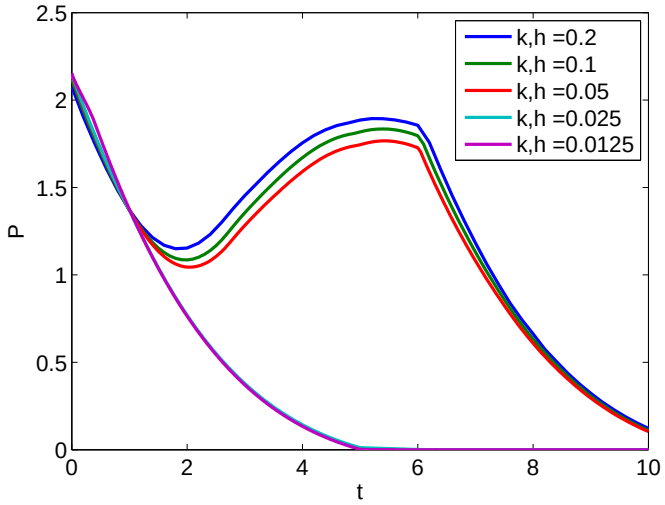

Figure 4.3: Graph for P, using an up-winding scheme.

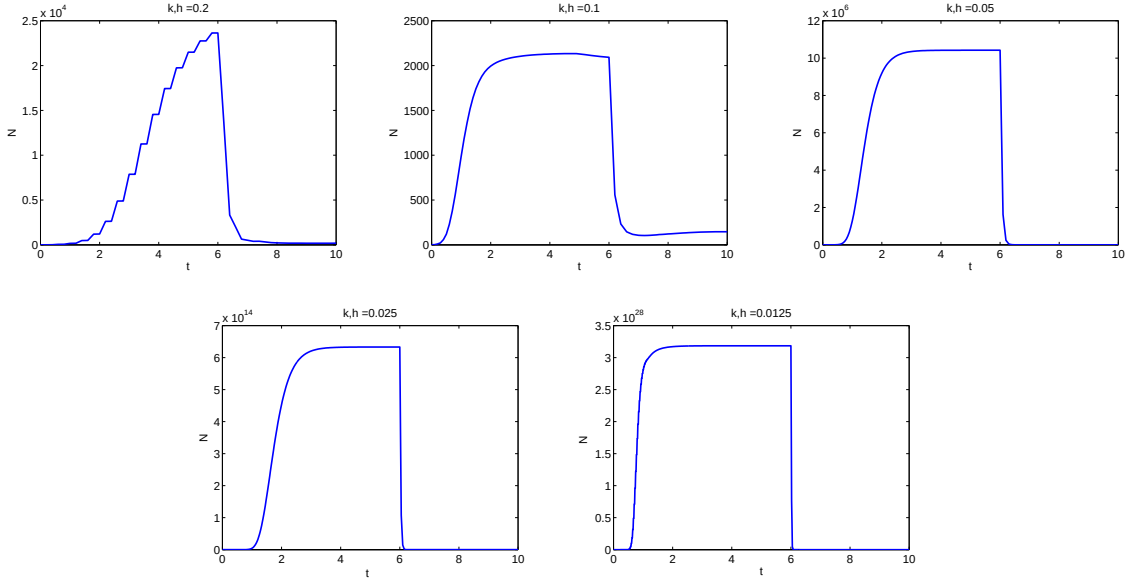

Figure 4.4: Graphs for N for k and $h = 0.2$, 0.1, 0.05, 0.025, and 0.0125

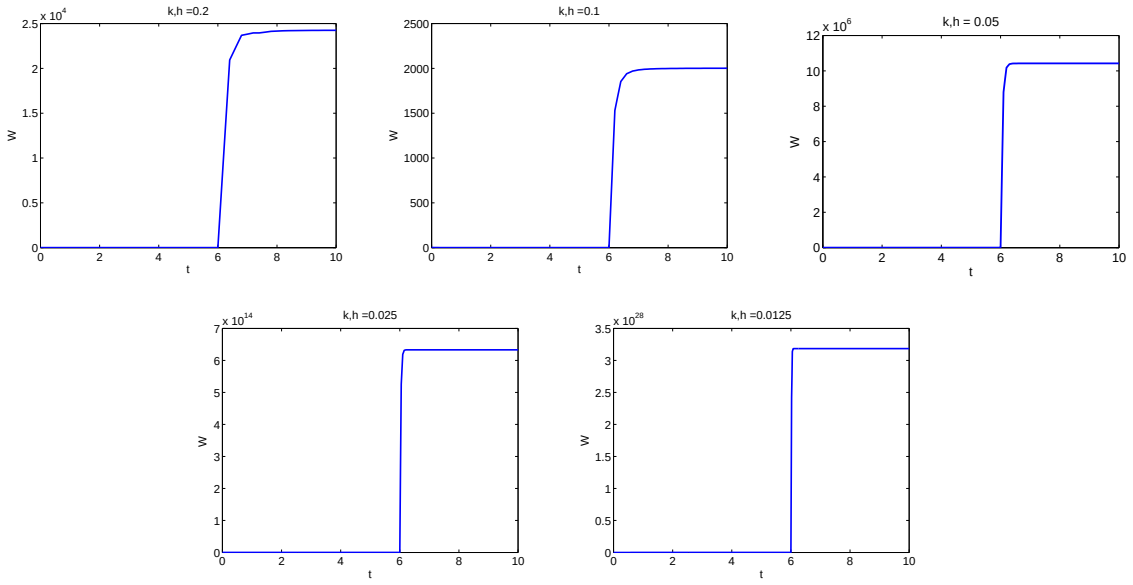

Figure 4.5: Graphs for W for k and $h = 0.2$, 0.1, 0.05, 0.025, and 0.0125

using built-in ODE solvers in Matlab.

4.2 A semi-implicit method

Another technique that may be used to solve the model is to use the Transport Equation as described in [3]. By using that in (2.1a), $\frac{\partial m}{\partial t} + \frac{\partial m}{\partial a} = -\gamma_s(G)m$ in $(0,\infty) \times \mathbb{R}$ and that $m(t,a) = \phi_m(a)$ in $\{0\} \times \mathbb{R}$, we get, using the technique in [3], that

$$m(t,a) = \phi_m(a-t) - \int_0^t \gamma_s(G(s))m(s, a+(s-t))ds$$

By using the approximation $\int_0^t f(s)ds \simeq \frac{\triangle s}{2}(f(0,a-t) - 2(\Sigma_{i=1}^{\lfloor \frac{t}{\triangle s} - 1 \rfloor} f(i\triangle s)) + f(t,a))$ where $\lfloor \frac{t}{\triangle s} - 1 \rfloor$ is the floor function, we get

$$m(t,a) = \phi_m(a-t) - \frac{\triangle s}{2}(\gamma_s(G(0))m(0,a-t) - 2(\Sigma_{i=0}^{\lfloor \frac{t}{\triangle s} - 1 \rfloor} \gamma_s(G(i\triangle s))m(i\triangle s, a+(i\triangle s - t))))$$

$$- \frac{\triangle s}{2}\gamma_s(G(t))m(t,a)$$

$$m(t,a) + \frac{\triangle s}{2}\gamma_s(G(t))m(t,a) = \phi_m(a-t) - \frac{\triangle s}{2}(\gamma_s(G(0))m(0,a-t)$$

$$-2(\Sigma_{i=0}^{\lfloor \frac{t}{\triangle s}-1 \rfloor}\gamma_s(G(i\triangle s))m(i\triangle s, a+(i\triangle s-t))))$$

$$m(t,a)=\frac{\phi_m(a-t)-\frac{\triangle s}{2}(\gamma_s(G(0))m(0,a-t)-2(\Sigma_{i=0}^{\lfloor \frac{t}{\triangle s}-1 \rfloor}\gamma_s(G(i\triangle s))m(i\triangle s,a+(i\triangle s-t))))}{1+\frac{\triangle s}{2}\gamma_s(G(t))}$$

By applying this same technique to (2.1b), (2.1c), and (2.1e), we get, for $DS(t)=\delta(W(t))+\beta(S(t))$,

$$s(t,a)=\frac{\phi_s(a-t)-\frac{\triangle s}{2}(DS(0)s(0,a-t)-2(\Sigma_{i=0}^{\lfloor \frac{t}{\triangle s}-1 \rfloor}(DS(i\triangle s)s(i\triangle s,a+(i\triangle s-t))))}{1+\frac{\triangle s}{2}DS(t)}$$

$$p(t,a)=\frac{\phi_p(a-t)-\frac{\triangle s}{2}(\gamma_p(G(0))p(0,a-t)-2(\Sigma_{i=0}^{\lfloor \frac{t}{\triangle s}-1 \rfloor}\gamma_p(G(i\triangle s))m(i\triangle s,a+(i\triangle s-t))))}{1+\frac{\triangle s}{2}\gamma_p(G(t))}$$

$$w(t,a)=\frac{\phi_w(a-t)-\frac{\triangle s}{2}(\gamma_w(G(0))w(0,a-t)-2(\Sigma_{i=0}^{\lfloor \frac{t}{\triangle s}-1 \rfloor}\gamma_w(G(i\triangle s))w(i\triangle s,a+(i\triangle s-t))))}{1+\frac{\triangle s}{2}\gamma_w(G(t))}$$

Due to the fact that $V(G)$ is non-constant, the transport technique cannot be used on (2.1d) for the entire interval of t employed and, therefore, cannot be used to solve the entire model.

4.3 A Half-step Algorithm

In the calculation of the solution of a PDE model at specific grid points, information from between grid points can be used in addition to information *at* grid points. In [14], there is a formula which uses such information and is of the form

$$\rho(t+k,a)=\rho(t,a)-\frac{k}{h}\{\rho(t+\frac{1}{2}k,a+\frac{1}{2}h)-\rho(t+\frac{1}{2}k,a-\frac{1}{2}h)\}-k*d(t+\frac{1}{2}k,a)\rho(t+\frac{1}{2}k,a)$$

where $\rho(t,a)$ is population density and satisfies $\frac{\partial\rho}{\partial t}+\frac{\partial\rho}{\partial a}=-d(t,a)\rho$. Two approximations for the midpoint value are presented in [14] for $\rho(t+\frac{1}{2}k,a+\frac{1}{2}h)$. We decided to use the formula of the form

$$\rho(t+\frac{1}{2}k,a+\frac{1}{2}h)=\rho(t,a)+\frac{1}{2}(1-\frac{k}{h})s(t,a)-\frac{k}{2}d(t,a)\rho(t,a).$$

Here, $s(t,a)$ is of the form $m(\rho(t,a+h)-\rho(t,a),\rho(t,a)-\rho(t,a-h))$, where $m(x,y)=sgn(x)*min(|x|,|y|)$ if $sgn(x)=sgn(y)$ and $=0$, otherwise.

The convergence and stability of the scheme cannot be determined when the scheme is applied to the model we are working on due to the coupled relationship between the five populations as well as the boundary conditions.

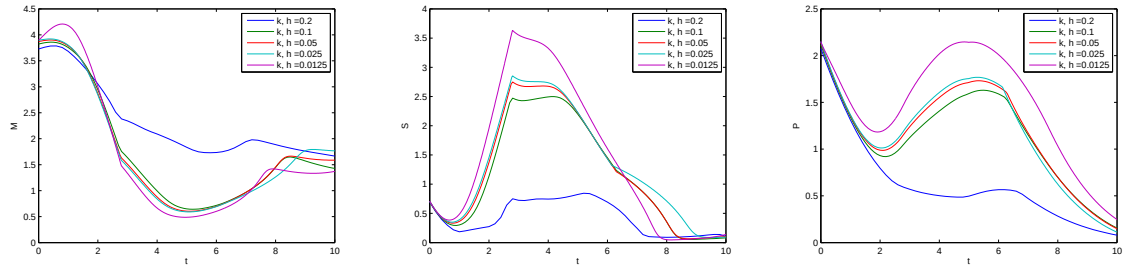

Figure 4.6: Output graphs of the half-step algorithm for M, S, and P. Each graph display behaviour similar to the behaviour of its corresponding graph from the upwind scheme when h amd k are between 0.1 and 0.025

4.3.1 Numerical experiments using a half-step algorithm

We implemented the half-step algorithm of [14], modified to our system of PDE. As before, we used filgrastim parameter estimates from [4]. The graphs obtained from this algorithm indicate instability and lack of convergence.

According to the graphs in Figure 4.6, M, S, and P appear convergent until k and h reach some value between 0.025 and 0.0125. Meanwhile, in Figure 4.7, N appears normal only for certain values of k and h that are less than 0.2 and greater than .05. According to Figure 4.8, W also appears normal only for certain values of k and h that are less than 0.2 and greater than .05.

The attempted algorithms in this section have shown to be unstable in n and w. In order to properly simulate all of the population densities, an algorithm unlike standard approximations will have to be used. In addition, certain components of the model may have to be simplified for the algorithm to work or to be easier to implement. In the next section, we discuss an algorithm derived from an existing splitting algorithm.

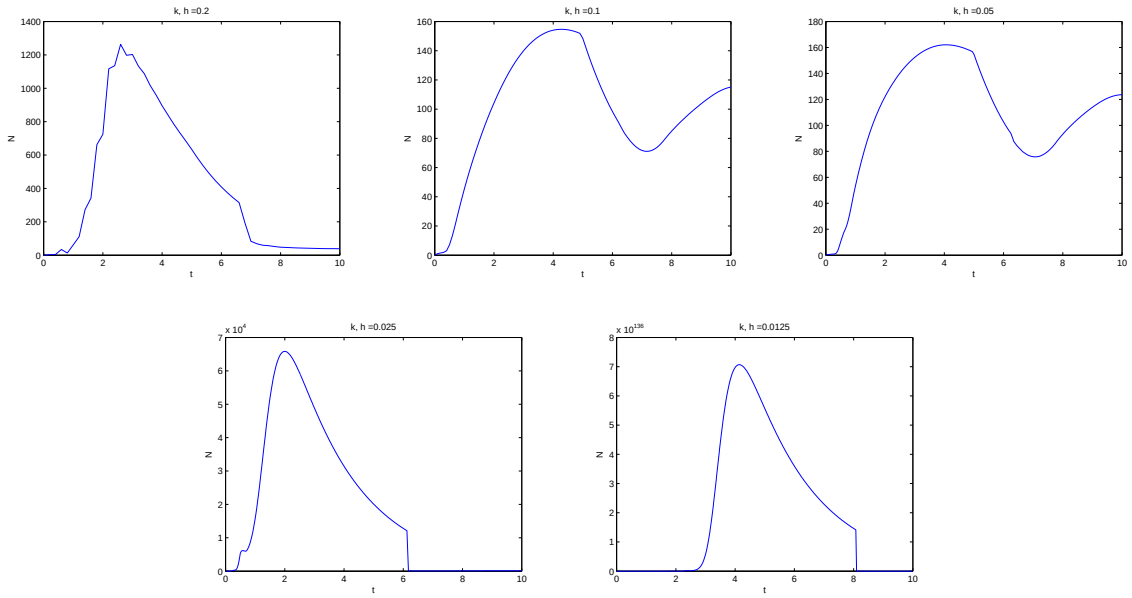

Figure 4.7: Output graphs of the half-step algorithm for N for k and $h = 0.2$, 0.1, 0.05, 0.025, and 0.0125

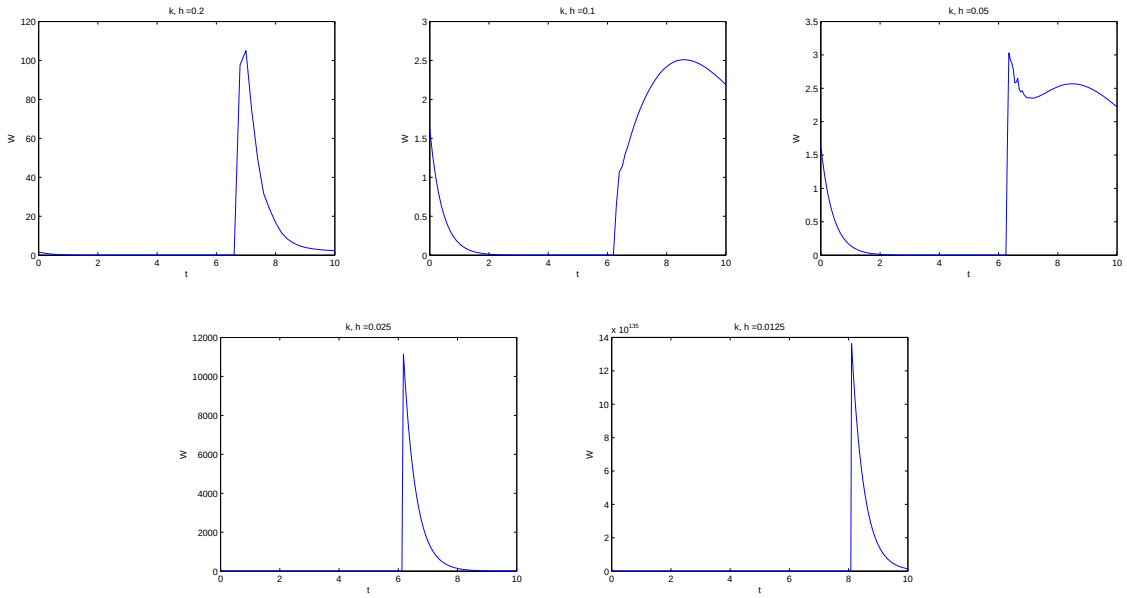

Figure 4.8: Output graphs of the half-step algorithm for W for k and $h = 0.2$, 0.1, 0.05, 0.025, and 0.0125

Chapter 5

Modified approximation schemes

In this chapter, we discuss the use of algorithms that are different from algorithms that have been used before. These algorithms are modifications of existing algorithms. The first is a version of the algorithm from [14] modified so that at each time step, a limited number of populations is updated instead of all five. The second algorithm is a modification of a splitting method in [8]. This section shows and compares the results of the algorithms.

5.1 Modified Half-step scheme

After lack of success with algorithms where the populations are updated simultaneously, we decided to try an algorithm where only some of the populations are updated at each time step. We decided to make two loops based on the relationships among the cells. In the first loop, m cells affect the growth of s cells, the total of which affects the growth of m cells. In the second loop, the total number of w cells affects the growth of p cells, the p cells become n cells and are amplified after reaching age τ_p, and the n cells become w cells after reaching age τ_n. In the new algorithm, the loops are updated simultaneously at each time step; since the first loop is two steps long and the second loop is three steps long, a bigger loop which contains both of these loops is six steps long. The new algorithm is as follows:

loop

$i \leftarrow \frac{t}{k}$

$x \leftarrow min(i+1, 2)$

$y \leftarrow min(i+1, 3)$

if $mod(i,2) = 0$ **then**
$s \leftarrow s + xkf_s(\tilde{W}, S, s)$
$s(t,0) \leftarrow 2\tilde{m}(t, \tau_s)$
else
$m \leftarrow m + xkf_m(G, m)$
$m(t,0) \leftarrow \beta(\tilde{S})\tilde{S}$
end if
if $mod(i,3) = 0$ **then**
$w \leftarrow w + ykf_w(w)$
$w(t,0) \leftarrow \tilde{n}(t, \tau_n)$
else
if $mod(i,3) = 1$ **then**
$p \leftarrow p + ykf_p(G, p)$
$p(t,0) \leftarrow \delta(\tilde{W})\tilde{S}$
else
$n \leftarrow n + ykf_n(G, n)$
$n(t,0) \leftarrow A(G)\tilde{p}(t, \tau_p)$
end if
end if
$t \leftarrow t + k$
end loop

The populations are updated by using the half-step algorithm based on [14].

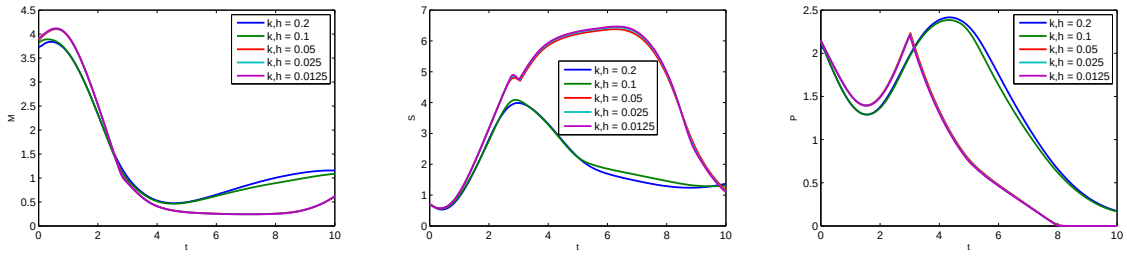

Figure 5.1: Graphs from the modified half-step algorithm for M, S, and P

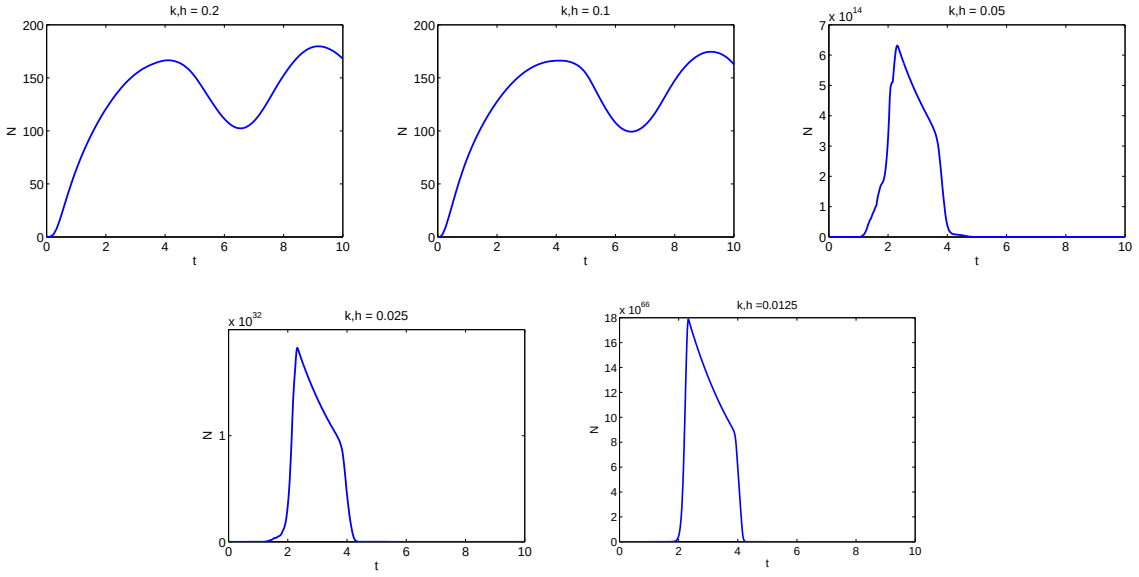

Figure 5.2: Graphs from the modified half-step algorithm for N for k and $h = 0.2$, 0.1, 0.05, 0.025, and 0.0125

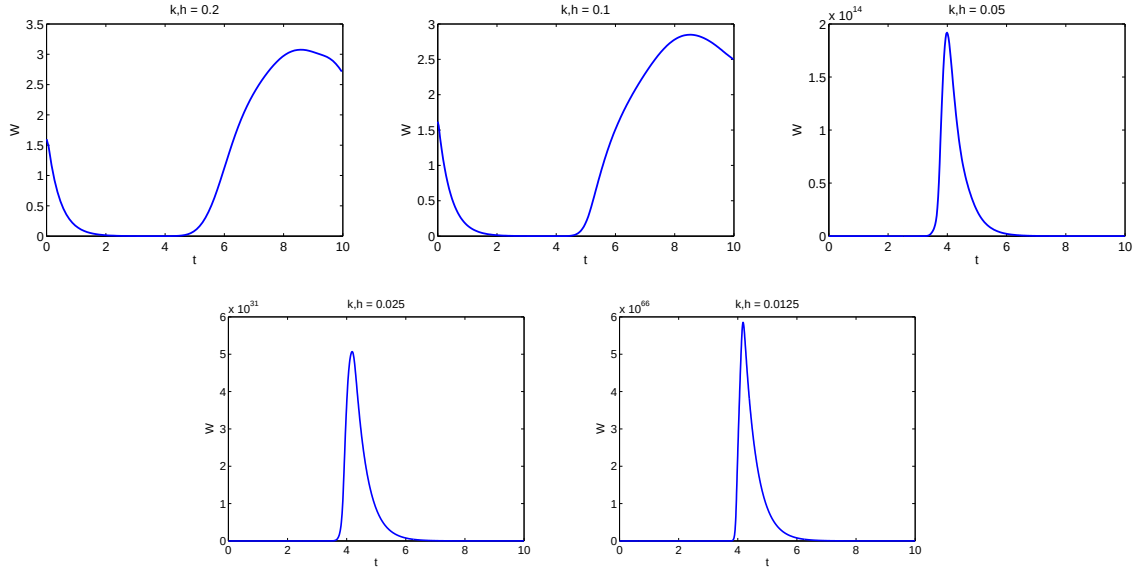

Figure 5.3: Graphs from the modified half-step algorithm for W for k and $h = 0.2$, 0.1, 0.05, 0.025, and 0.0125

The graphs obtained from simulations demonstrate instability, and therefore even this modified half-step strategy is not suitable for our problem.

5.1.1 A splitting method

In [8] the model discussed is of the following form for population density $u(a,t)$ and maximum age $a_\dagger$:

$$u_t + u_a + \mu(a,p(t))u = 0, \quad 0 \le a \le a_\dagger,\ t > 0$$

$$u(a,0) = u_0(a), \quad 0 \le a \le a_\dagger$$

$$u(0,t) = \int_0^{a_\dagger} \beta(a,p(t))u(a,t)da, \quad t > 0$$

$$p(t) = \int_0^{a_\dagger} u(a,t)da, \quad t \ge 0$$

Among the methods applied to the model is a splitting method which assumes that u_0 is nonnegative, β and μ are differentiable, β is nonnegative and bounded. Furthermore, the authors assume $0 \le \mu(a,p) = m(a) + M(a,p)$ for differentiable M and continuous m such that $\int_0^{a_\dagger} m(a)da = +\infty$. The method uses time-step $h = T/N$, where T is the final time and N is the number of time-steps, and $A_\dagger$ such that $A_\dagger h < a_\dagger \le (A_\dagger + 1)h$. For $a_j = jh$ with $j = 0, 1, ..., J$ and for $t^n = nh$ with $n = 0, 1, ...N$, U_j^n is the approximation of $u(a_j, t^n)$ and P^n is the approximation of $p(t^n)$. The method has initial values $U_j^0 = u_0(a_j)$ for $0 \le j \le A_\dagger$, and $P^0 = \sum_{j=0}^{A_\dagger} U_j^0 h$. For $n > 0$, the following splitting algorithm is used:

$$\frac{U_j^{n-1/2} - U_{j-1}^{n-1}}{h} + m_j U_j^{n-1/2} = 0, \quad 1 \le j \le A_\dagger$$

$$\frac{U_j^n - U_j^{n-1/2}}{h} + \tilde{M}_j^{n-1} U_j^n = 0, \quad 1 \le j \le A_\dagger$$

$$U_0^n = \sum_{j=0}^{A_\dagger} w_j \tilde{\beta}_j^{n-1} U_j^n h, \qquad P^n = \sum_{j=0}^{A_\dagger} \bar{w}_j U_j^n h$$

where $\tilde{M}_j^n = M(a_j, P^n)$, $\tilde{\beta}_j^n = \beta(a_j, P^n)$ and w and $\bar{w}$ are numerical quadrature weights. The first two equations are simplified to one equation:

$$U_j^n = \frac{U_{j-1}^{n-1}}{\left(1 + \tilde{M}_j^{n-1} h\right)(1 + m_j h)}$$

This method is clearly only first-order accurate.

Modified splitting method

We derived an algorithm based on the splitting method from [8] with a few differences. Firstly, the equation that relates the cell densities $c(jh, nk)$ and $c((j-1)h, (n-1)k)$ are based on the following equations for parameter $\omega \in (0, 1)$:

$$\frac{c_j^{n-1/2} - c_{j-1}^{n-1}}{h} + \omega\gamma_c(G^{n-1})c_j^{n-1/2} = 0$$

$$\frac{c_j^{n} - c_{j}^{n-1/2}}{h} + (1-\omega)\gamma_c(G^{n-1})c_j^{n} = 0$$

These equations lead to the relation of the form:

$$c_j^n = \frac{c_{j-1}^{n-1}}{(1+\omega\gamma_c(G^{n-1})h)\,(1+(1-\omega)\gamma_c(G^{n-1})h)}$$

Using this, we obtain the following algorithm:

$$m_j^n = \frac{m_{j-1}^{n-1}}{(1+\omega\gamma_s(G^{n-1})h)\,(1+(1-\omega)\gamma_s(G^{n-1})h)}$$

$$m_0^n = \beta(S^n)S^n$$

$$s_j^n = \frac{s_{j-1}^{n-1}}{(1+\omega(\delta(W^{n-1})+\beta(S^{n-1}))h)\,(1+(1-\omega)(\delta(W^{n-1})+\beta(S^{n-1}))h)}$$

$$s_0^n = 2m_{a_s}^n,\ a_s = \lceil\frac{\tau_s}{h}\rceil$$

$$p_j^n = \frac{p_{j-1}^{n-1}}{(1+\omega\gamma_n(G^{n-1})h)\left(1+(1-\omega)\gamma_p(G^{n-1})\frac{h}{Vn(G^{n-1})}\right)}$$

$$p_0^n = \delta(W^n)S^n$$

$$n_j^n = \frac{n_{j-1}^{n-1}}{(1+\omega\gamma_p(G^{n-1})h)\,(1+(1-\omega)\gamma_n(G^{n-1})h)}$$

$$n_0^n = A(G^n)p_{a_p}^n,\ a_p = \lceil\frac{\tau_p}{h}\rceil$$

$$w_j^n = \frac{w_{j-1}^{n-1}}{(1+\omega\gamma_w h)\,(1+(1-\omega)\gamma_w h)}$$

$$w_0^n = n_{a_n}^n,\ a_n = \lceil\frac{\tau_n}{h}\rceil$$

After scaling the populations, we found that the algorithm outputs graphs such as in Figures 5.4 and 5.5 demonstrate convergence. Depending on the time step used, the graphs

could show convergence limited to a finite interval. However, if the step is sufficiently small, the method at the very least is convergent within the intervals of time used in our experiments. The populations were scaled as follows:

$$m = \tilde{m} \times 10^6,\ s = \tilde{s} \times 10^6,\ p = \tilde{p} \times 10^6,\ n = \tilde{n} \times 10^9,\ w = \tilde{w} \times 10^8 \tag{5.1}$$

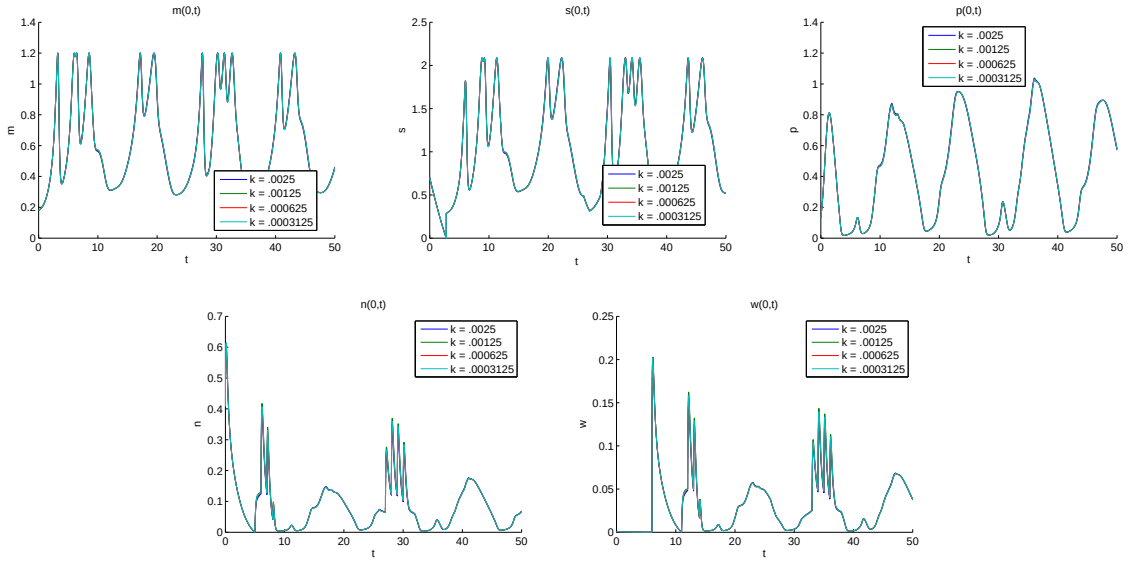

Figure 5.4: Graphs at $a = 0$ obtained from the modified splitting algorithm

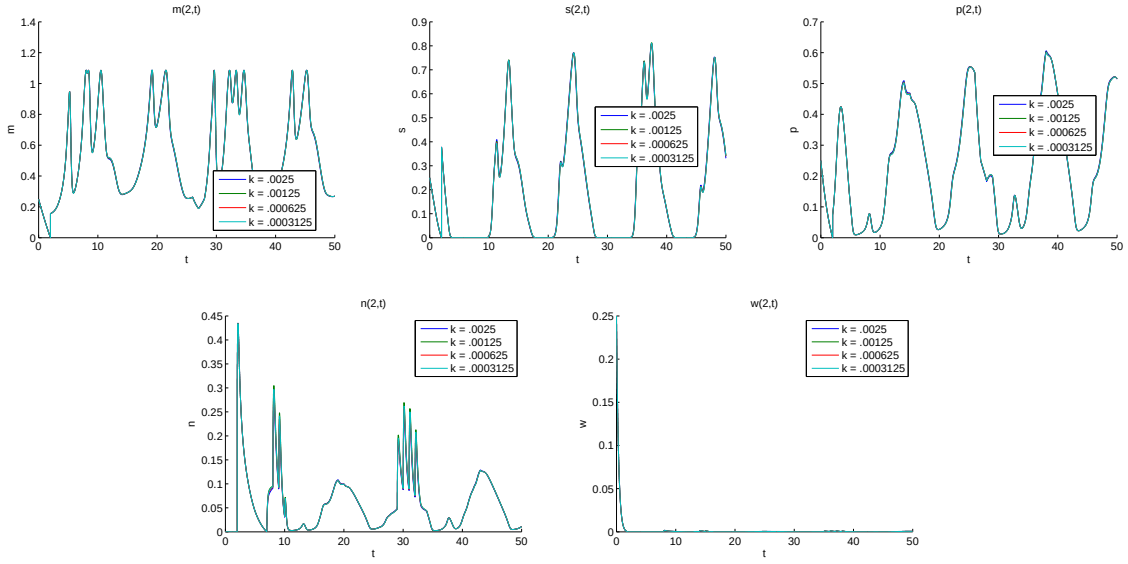

Figure 5.5: Graphs at $a = 2$ obtained from the modified splitting algorithm

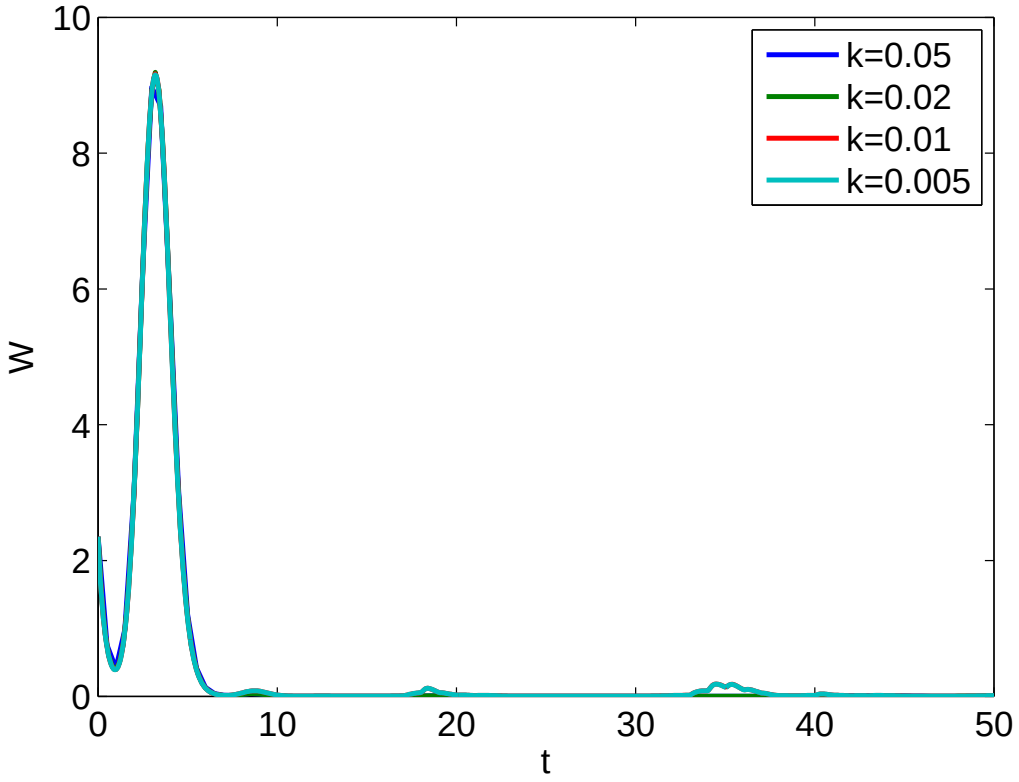

Figure 5.6: Neutrophil count obtained from the modified splitting algorithm.

Chapter 6

Results

In this section, we look at the results of various experiments ran with the modified splitting algorithm discussed in the previous section. Each experiment is ran in two ways: with simplifying assumptions and without simplifying assumptions. We then compare the results of each model to find out how much impact changing the parameters of the run of the model can have. In addition, we compare the two models in terms of their results to see the difference that the simplifications make.

The experiments from which the results discussed in this chapter were obtained, were run with the population densities all scaled as in (5.1). These scales are based on the steady state values of the population totals used for the solutions to the DDE model simulated in [4]; $S = 3.1 \times 10^6$ cells/kg, $P = 0.46 \times 10^6$ cells/kg, $N = 8.45 \times 10^9$ cells/kg, $W = 2.35 \times 10^8$ cells/kg. These steady state values are not the same values shown in Appendix A.

6.1 Model A: Simplified PDE model

Athough the PDE model (2.1a) - (2.1e) is not solved in [4], the DDE model (3.7) - (3.10) is simulated using a number of approximations which can be applied to PDE model. The apoptosis rates γ_s, γ_p, and γ_n are assumed to be constant, as is V_n. If these assumptions, as well as the assumption that $\delta(W)$ and $\beta(S)$ are constant, are applied to the PDE model, then the method of characteristics can be used to find a solution as can the transport equation, which previously could not be used due to the variability of V_n.

We can simulate the original PDE model by making the same assumptions, and call this

Model A. The PDE model with the same simplifying assumptions is as follows:

$$\frac{\partial \tilde{m}}{\partial t} + \frac{\partial \tilde{m}}{\partial a} = -0.05\tilde{m}; \qquad t > 0, a \in [0, 2.8] \tag{6.1a}$$

$$\frac{\partial \tilde{s}}{\partial t} + \frac{\partial \tilde{s}}{\partial a} = -\delta(100\tilde{W})\tilde{s} - \beta(\tilde{S})\tilde{s}; \qquad t > 0, a > 0 \tag{6.1b}$$

$$\frac{\partial \tilde{p}}{\partial t} + \frac{\partial \tilde{p}}{\partial a} = -0.27\tilde{p}; \qquad t > 0, a \in [0, 5] \tag{6.1c}$$

$$\frac{\partial \tilde{n}}{\partial t} + 6\frac{\partial \tilde{n}}{\partial a} = -0.27\tilde{n}; \qquad t > 0, a \in [0, 6] \tag{6.1d}$$

$$\frac{\partial \tilde{w}}{\partial t} + \frac{\partial \tilde{w}}{\partial a} = -2.4\tilde{w}; \qquad t > 0, a > 0 \tag{6.1e}$$

with initial conditions

$$\tilde{m}(0, a) = \phi_m(a)/10^6, a \in [0, 2.8]$$

$$\tilde{s}(0, a) = \phi_s(a)/10^6, a > 0$$

$$\tilde{p}(0, a) = \phi_p(a)/10^6, a \in [0, 5]$$

$$\tilde{n}(0, a) = \phi_n(a)/10^9, a \in [0, 6]$$

$$\tilde{w}(0, a) = \phi_w(a)/10^8, a > 0$$

and with renewal conditions

$$\tilde{m}(t, 0) = \beta(\tilde{S}(t))\tilde{S}(t)$$

$$\tilde{s}(t, 0) = 2\tilde{m}(t, 2.8)$$

$$\tilde{p}(t, 0) = \delta(100\tilde{W}(t))\tilde{S}(t)$$

$$\tilde{n}(t, 0) = A(G(t))\tilde{p}(t, 5)/1000$$

$$\tilde{w}(t, 0) = 10\tilde{n}(t, 6)$$

The experiments ran with the algorithm are as follows:

Expt. 1 no simulation of G-CSF treatment or chemotherapy

Expt. 2 simulation of filgrastim treatment with a period of 14 days, with daily dosage of 5 μg/kg or 10 μg/kg, and without chemotherapy

Expt. 3 simulation of filgrastim treatment with different period lengths, different delays after chemotherapy, and daily dosage of 5 μg/kg.

Expt. 4 simulation of pegfilgrastim treatment with dosage of 100 μg/kg and a post-chemotherapy delay of 1 day.

The initial profiles used for the population densities are normally distributed curves each of which has a mean value occurring in the center of the age interval used. For example, the mean of $\tilde{m}(0,a)$ occurs at $a = 1.4$. In addition, in the experiments where chemotherapy is simulated, chemotherapy begins on day 9 in the first cycle and on day 30 in the second cycle.

6.2 Numerical experiments for the simplified model

In all of the graphs in Figures 6.1 through 6.7, we used the modified splitting scheme of Section 5.1.1 and used steps in age and time such that the graph were converged.

6.3 Model B: the full age-structured model for cyclical neutropenia

The full PDE model is as follows:

$$\frac{\partial \tilde{m}}{\partial t} + \frac{\partial \tilde{m}}{\partial a} = -\gamma_s(G)\tilde{m}; \qquad t > 0, a \in [0, \tau_s] \tag{6.2a}$$

$$\frac{\partial \tilde{s}}{\partial t} + \frac{\partial \tilde{s}}{\partial a} = -\delta(100\tilde{W})\tilde{s} - \beta(\tilde{S})\tilde{s}; \qquad t > 0, a > 0 \tag{6.2b}$$

$$\frac{\partial \tilde{p}}{\partial t} + \frac{\partial \tilde{p}}{\partial a} = -\gamma_p(G)\tilde{p}; \qquad t > 0, a \in [0, \tau_p] \tag{6.2c}$$

$$\frac{\partial \tilde{n}}{\partial t} + V_n(G)\frac{\partial \tilde{n}}{\partial a} = -\gamma_n(G)\tilde{n}; \qquad t > 0, a \in [0, \tau_n] \tag{6.2d}$$

$$\frac{\partial \tilde{w}}{\partial t} + \frac{\partial \tilde{w}}{\partial a} = -\gamma_w\tilde{w}; \qquad t > 0, a > 0 \tag{6.2e}$$

with initial conditions

$$\tilde{m}(0,a) = \phi_m(a)/10^6, a \in [0, \tau_s]$$

$$\tilde{s}(0,a) = \phi_s(a)/10^6, a > 0$$

$$\tilde{p}(0,a) = \phi_p(a)/10^6, a \in [0, \tau_p]$$

$$\tilde{n}(0,a) = \phi_n(a)/10^9, a \in [0, \tau_n]$$

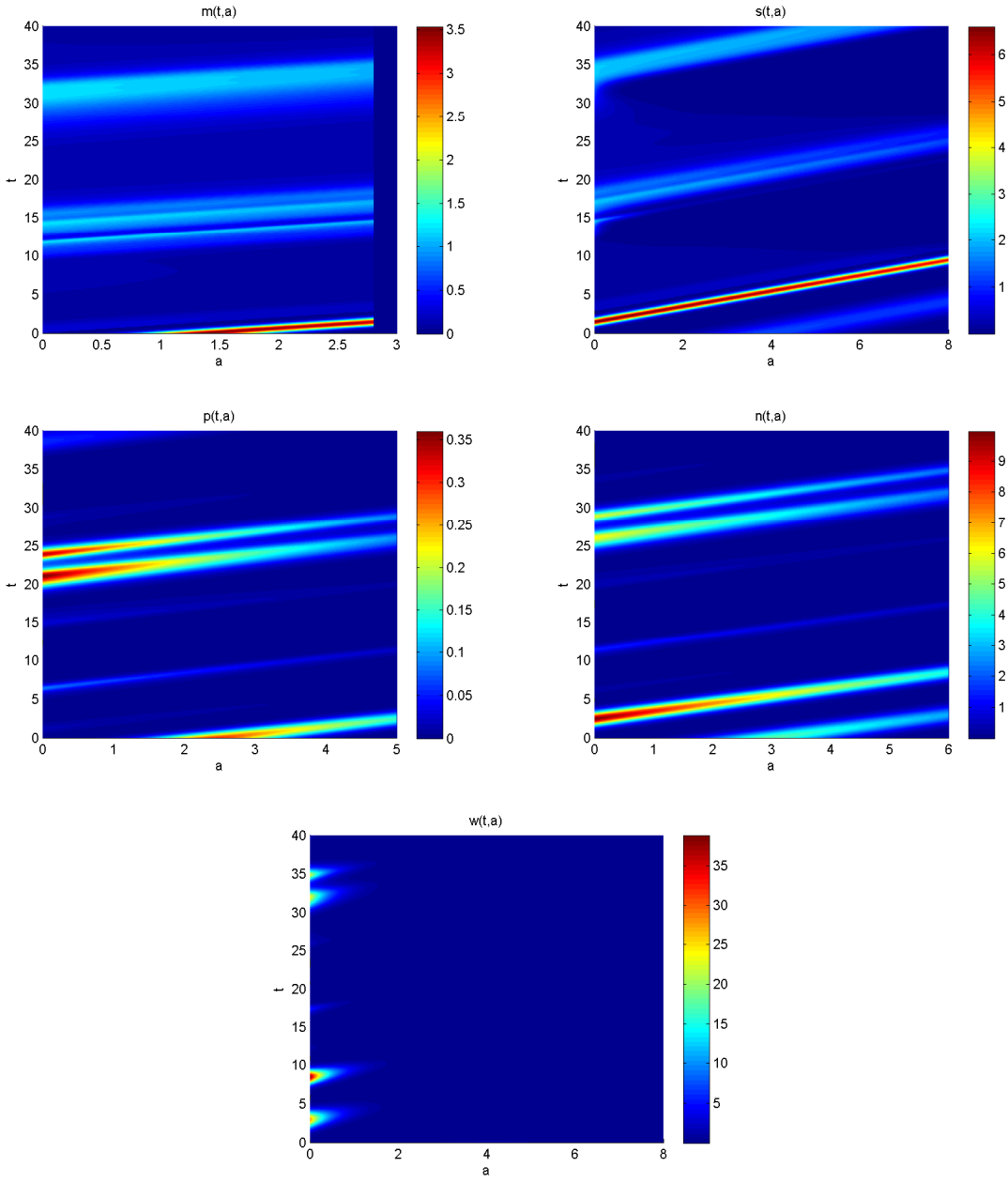

Figure 6.1: Expt.1, Model A: Cell population densities without G-CSF treatment.

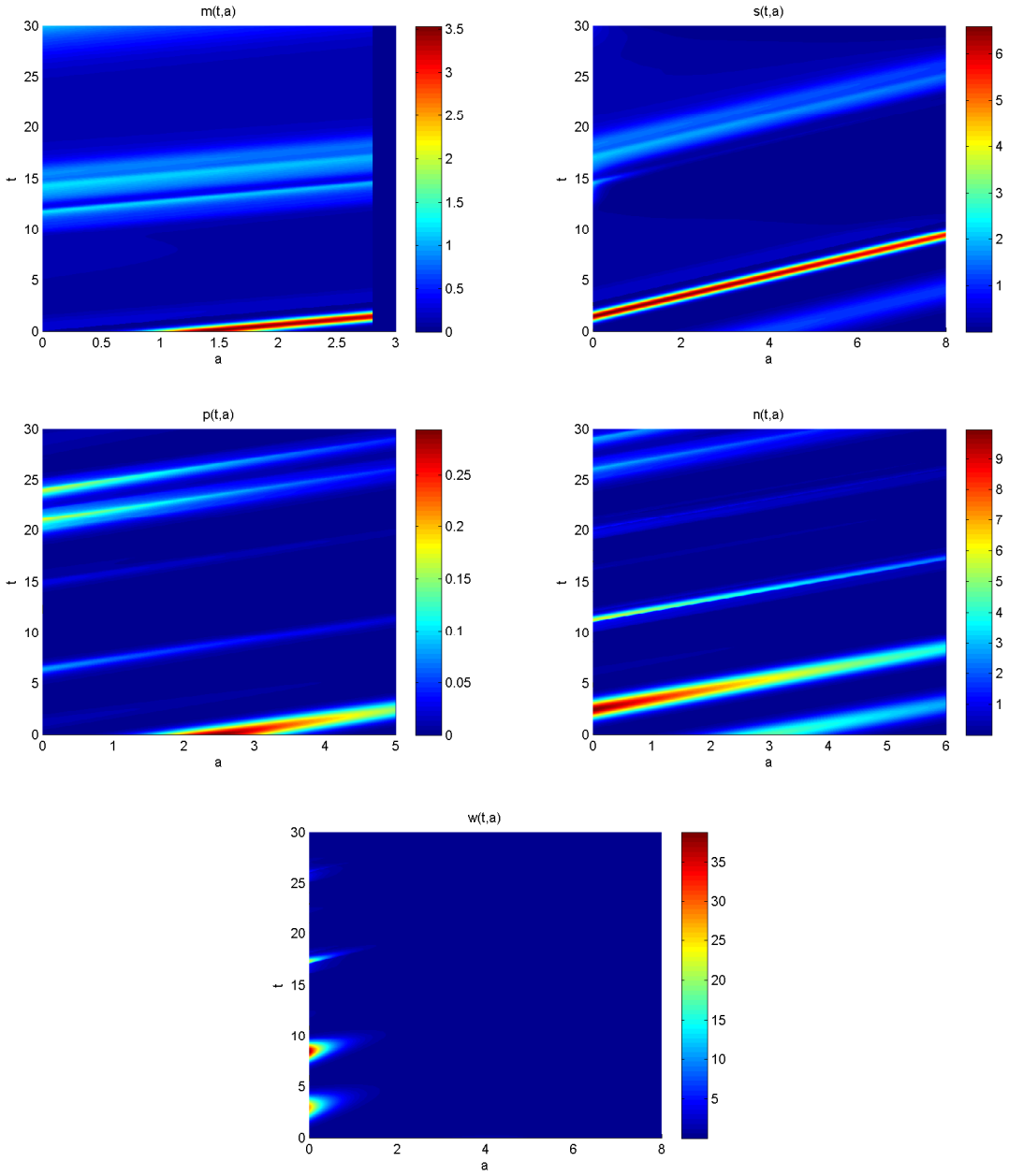

Figure 6.2: Expt. 2, Model A: Simulating filgrastim treatment starting at day 5 with period of 14 days, with daily dosage of 5 μg/kg, and without chemotherapy. The graphs of m, s, n, and w are virtually unchanged from the $t \leq 30$ region of their corresponding graphs in the previous figure. The only major change in p is the slightly different oscillations in the region $20 \leq t \leq 25$.

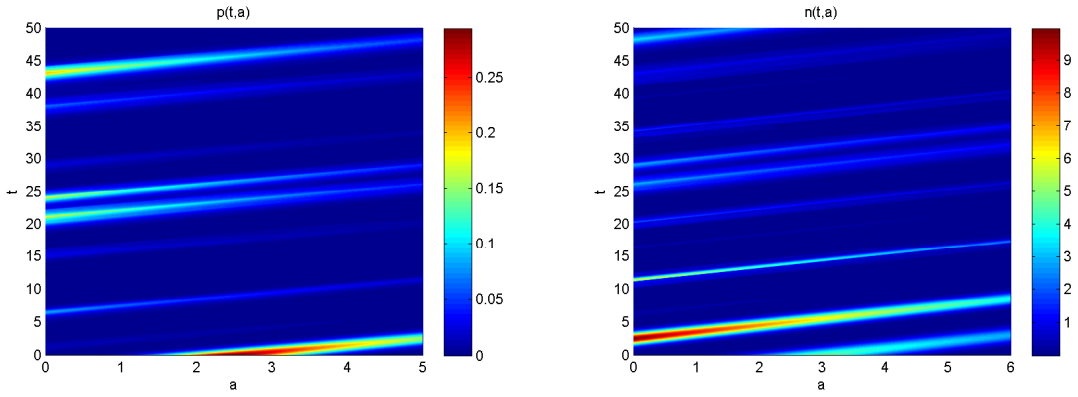

Figure 6.3: Expt. 3, Model A: The graphs of p and n obtained by simulating filgrastim treatment with period of 11 days, with daily dosage of 5 μg/kg, and with each treatment starting 1 day after chemotherapy. Unlike in the previous figure, p does not go higher than 0.5 in this figure. In addition, the oscillations in n are smaller here.

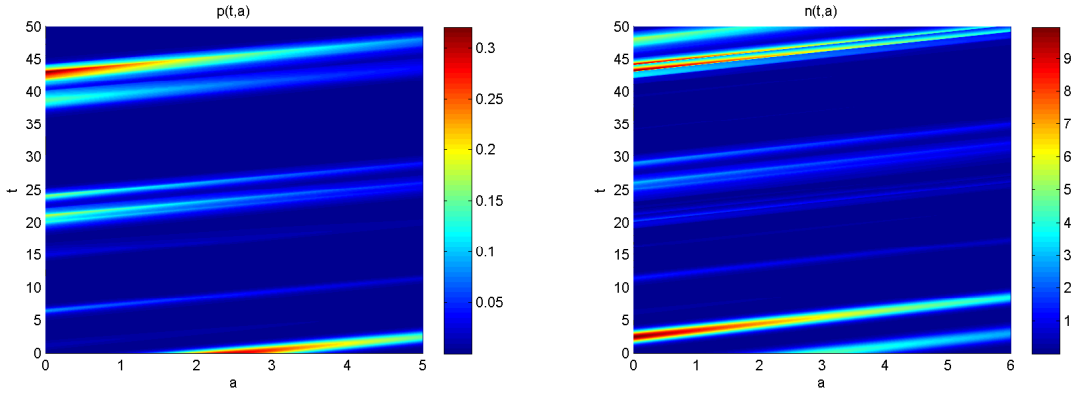

Figure 6.4: Expt. 3, Model A: The graphs of p and n obtained by simulating filgrastim treatment with period of 11 days, with daily dosage of 5 μg/kg, and with each treatment starting 4 days after chemotherapy. In this figure, p has a lower maximum value than in the previous figure and n shows slightly different oscillations.

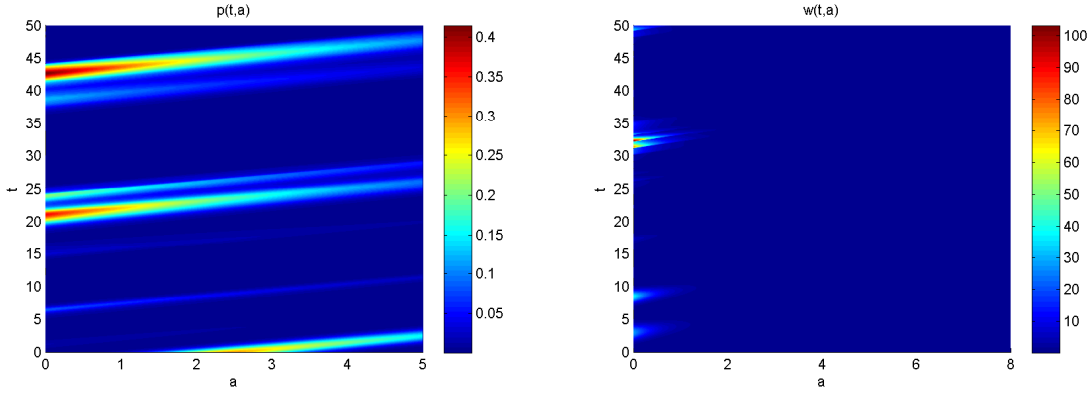

Figure 6.5: Expt. 3, Model A: The graphs of p and w obtained by simulating filgrastim treatment with period of 11 days, with daily dosage of 5 μg/kg, and with each treatment starting 8 days after chemotherapy. p is noticeably different from in the previous figure, with slightly different oscillations and a higher maximum value. w is slightly different in the $40 \leq t \leq 50$ region.

$$\tilde{w}(0,a) = \phi_w(a)/10^8, a > 0$$

and with boundary conditions

$$\tilde{m}(t,0) = \beta(\tilde{S}(t))\tilde{S}(t)$$

$$\tilde{s}(t,0) = 2\tilde{m}(t,\tau_s)$$

$$\tilde{p}(t,0) = \delta(100\tilde{W}(t))\tilde{S}(t)$$

$$\tilde{n}(t,0) = A(G(t))\tilde{p}(t,\tau_p)/1000$$

$$w(t,0) = 10\tilde{n}(t,\tau_n)$$

6.3.1 Numerical approximations

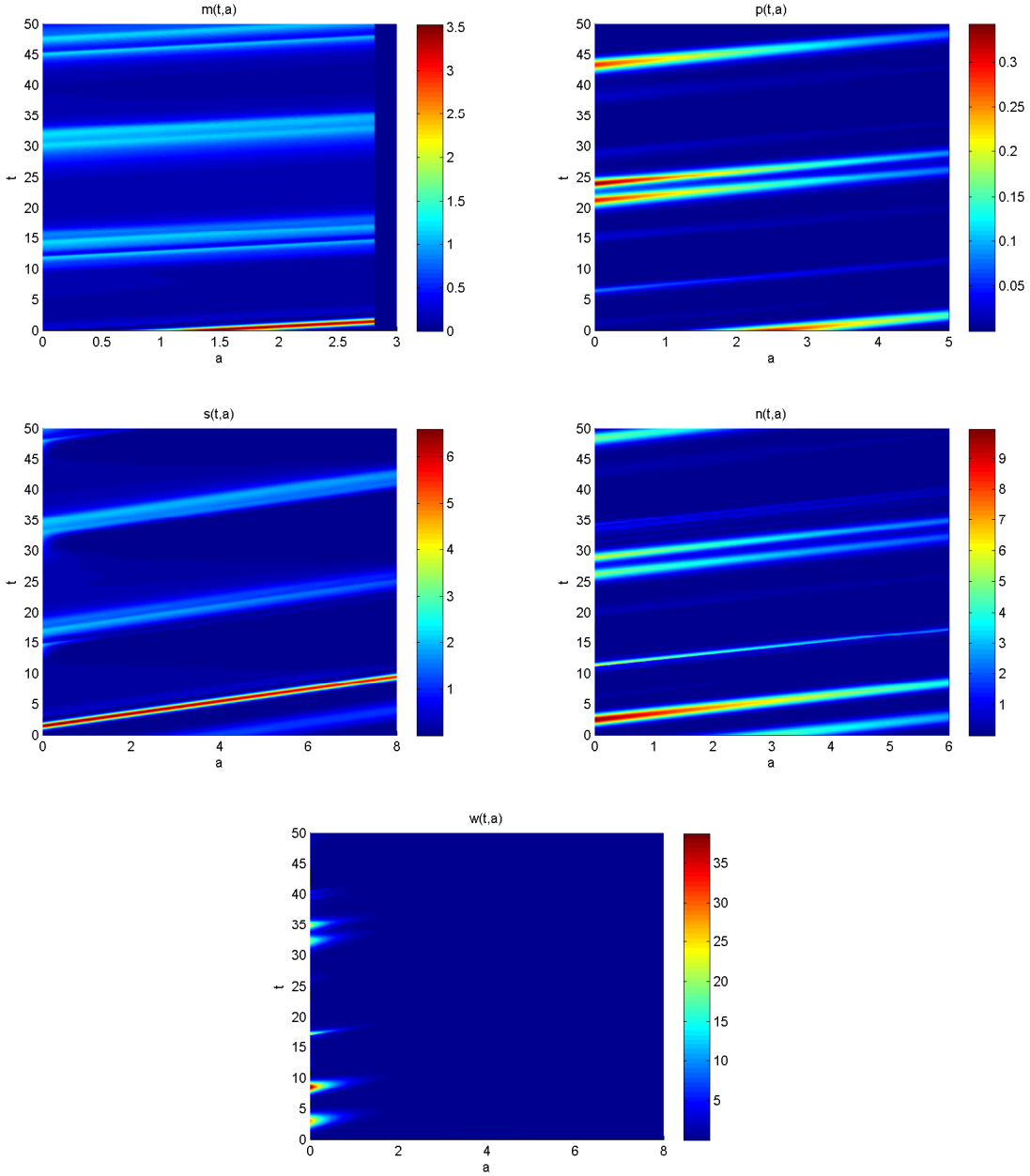

Figure 6.6: Expt. 3, Model A: The graphs obtained via the simplified model by simulating filgrastim treatment with period of 4 days, with daily dosage of 5 μg/kg, and with each treatment starting 1 day after chemotherapy. The graph here, in comparison to the graph of p in the previous figure, shows different oscillations in the regions $20 \leq t \leq 30$ and $40 \leq t \leq 50$.

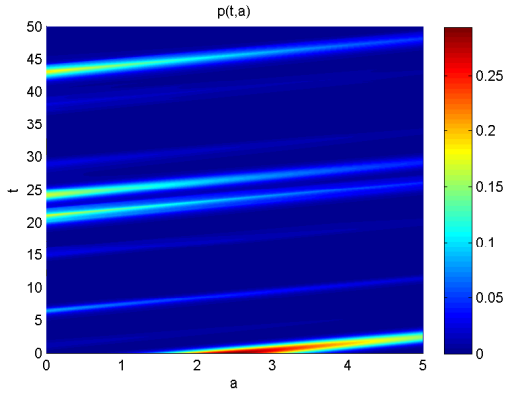

Figure 6.7: Expt. 3, Model A: The graph of p obtained via the simplified model by simulating filgrastim treatment with period of 8 days, with daily dosage of 5 μg/kg, and with each treatment starting 1 day after chemotherapy. The major differences between this graph and graph of p in the previous figure are in the regions $20 \leq t \leq 30$ and $40 \leq t \leq 50$.

6.4 Conclusion

Treatment Protocol	Model A Observations	Model B Observations
No G-SCF treatment or chemotherapy	All populations show oscillations without any pattern. p shows peaks when $20 \leq t \leq 25$, n shows peaks when $t \leq 5$ and when $25 \leq t \leq 30$, and w shows peaks when $0 \leq t \leq 10$ and when $30 \leq t \leq 37$.	m shows several oscillations after $t = 10$; s shows several oscillations and, when $10 \leq t \leq 32$, shows cells of age no greater than 1; p shows several oscillations without any pattern; n shows several oscillations, but with values greater than 2 only occuring when $t \leq 5$; w shows few oscillations.

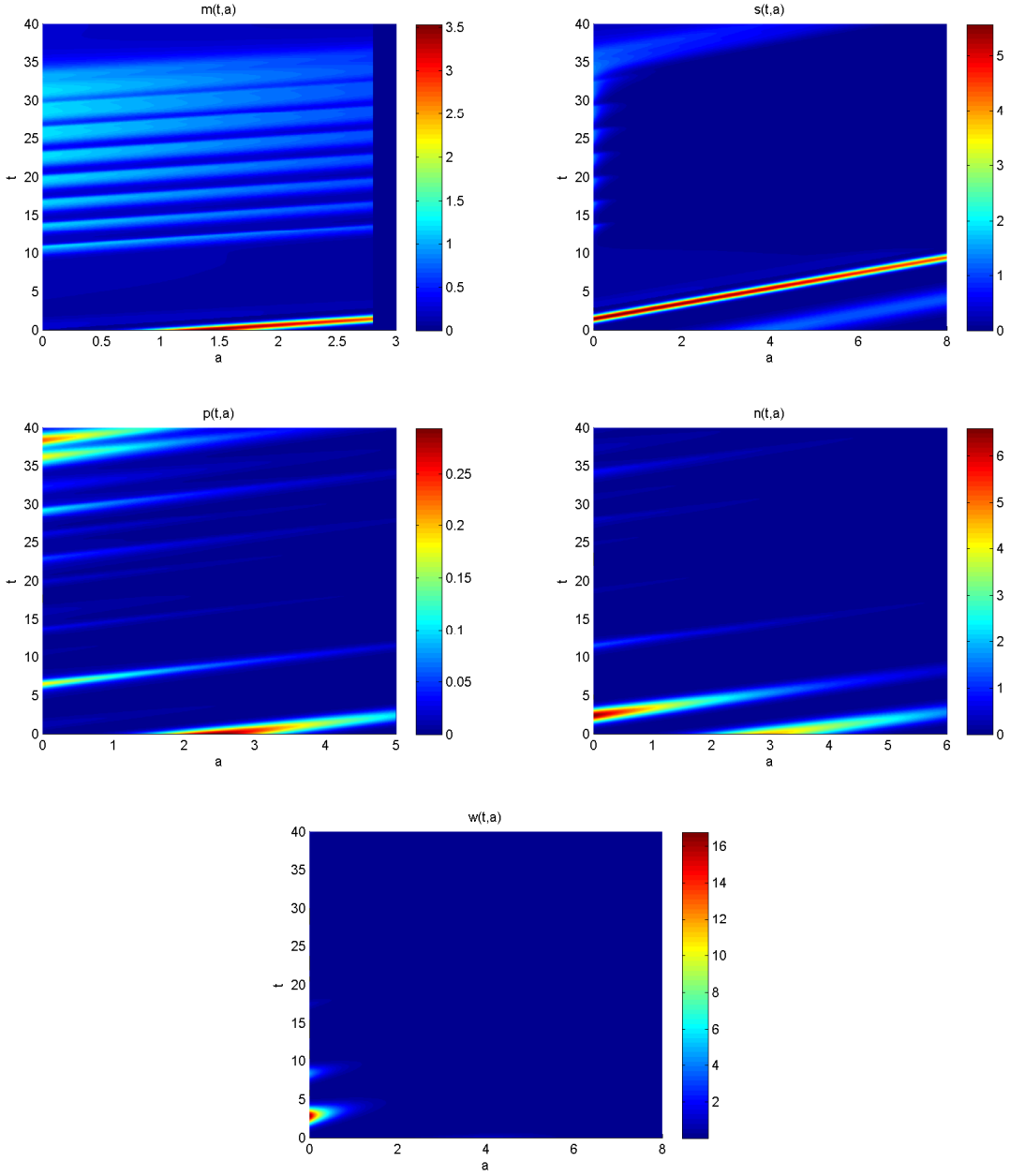

Figure 6.8: Expt. 1, Model B: The graphs obtained via the full model by running the algorithm without simulating G-CSF treatment. The graphs of m, s, and p here are radically different from what is obtained from the simplified model with the same treatment protocol. The graph of m shows several more oscillations than in Figure 6.1, the cell age in s is no larger than 1 when $t \geq 10$, and the graph of p shows an early peak in the region $5 \leq t \leq 10$ followed by several small oscillations.

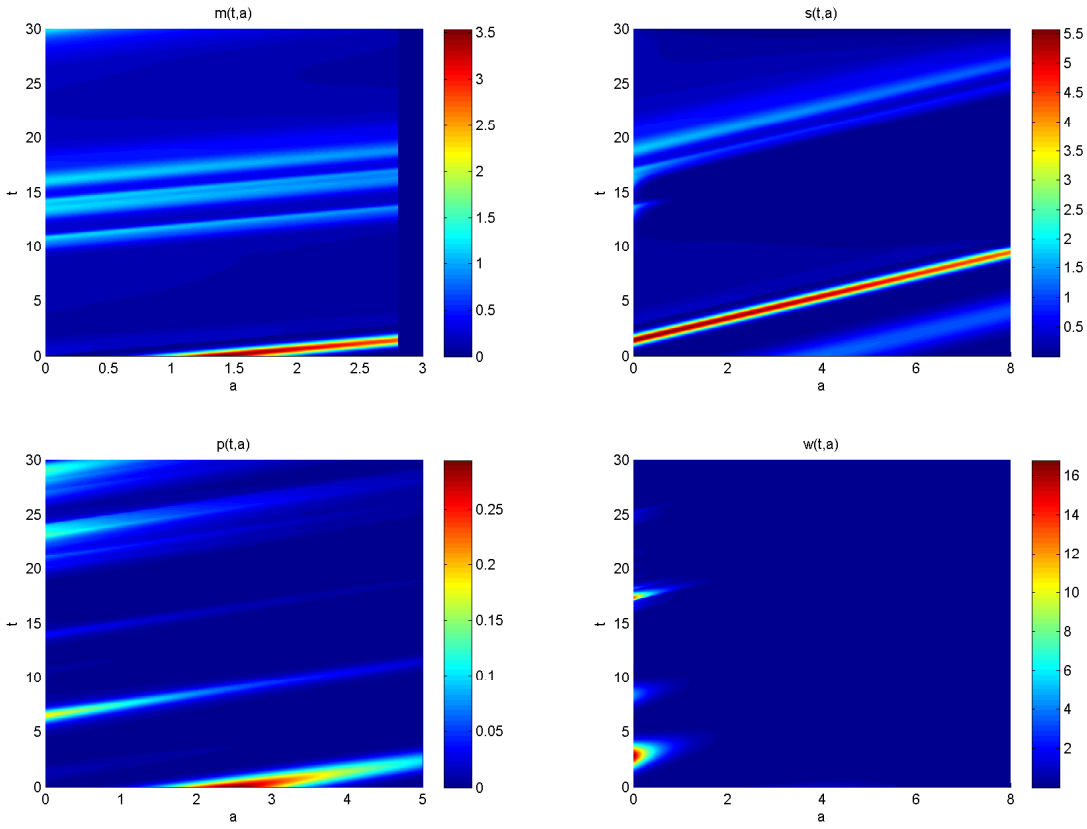

Figure 6.9: Expt. 2, Model B: The graphs of m, s, p, and w obtained by simulating filgrastim treatment starting at day 5 treatment with period of 14 days, with daily dosage of 5 μg/kg, and without chemotherapy. There are less oscillations in m here than in the $t \leq 30$ region of the graph of m in the previous figure, the $t \geq 10$ region of s shows more cells of age greater than 1, and p shows a peak in the region $20 \leq t \leq 25$ in addition to the peak in the region $5 \leq t \leq 10$ from the previous figure.

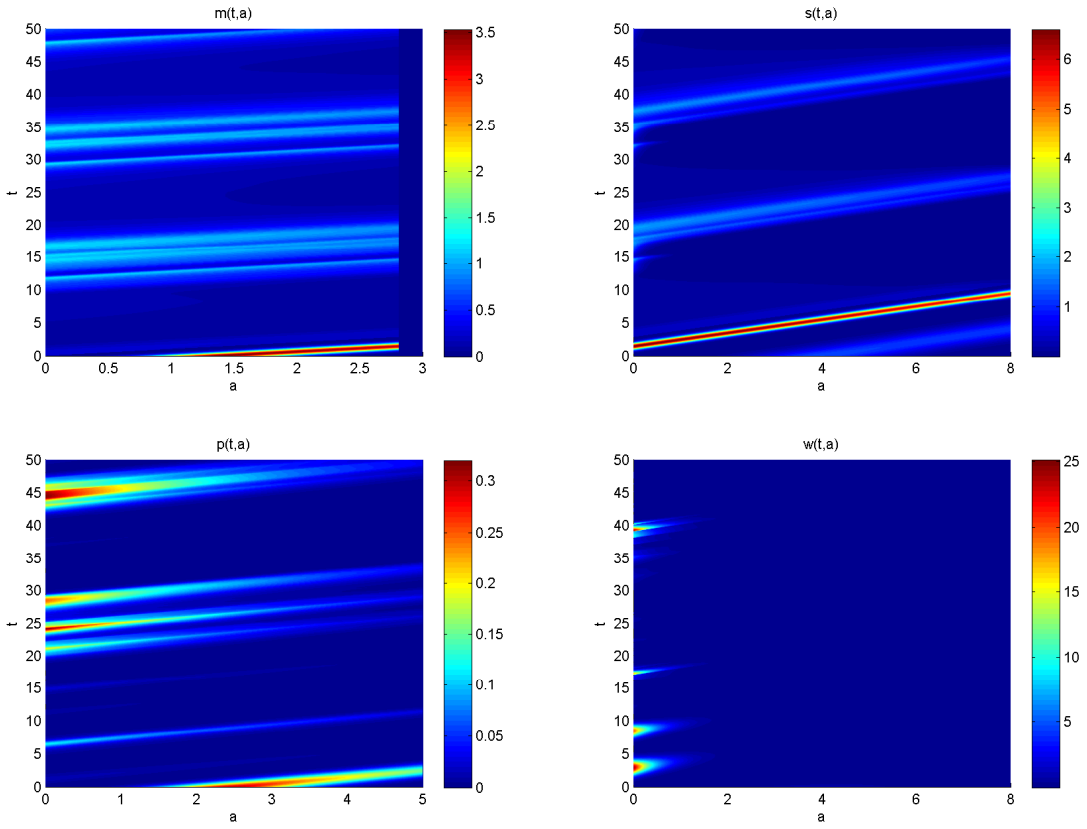

Figure 6.10: Expt. 3, Model B: The graphs of m, s, p, and w obtained by simulating filgrastim treatment with period of 11 days, with daily dosage of 5 μg/kg, and with each treatment starting 1 day after chemotherapy. The $t \leq 30$ region of m here shows more oscillations than in the previous graph of m. The oscillations in s that occur in the $15 \leq t \leq 25$ region are slightly different from the oscillations that occur in the same region of the previous graph of s.

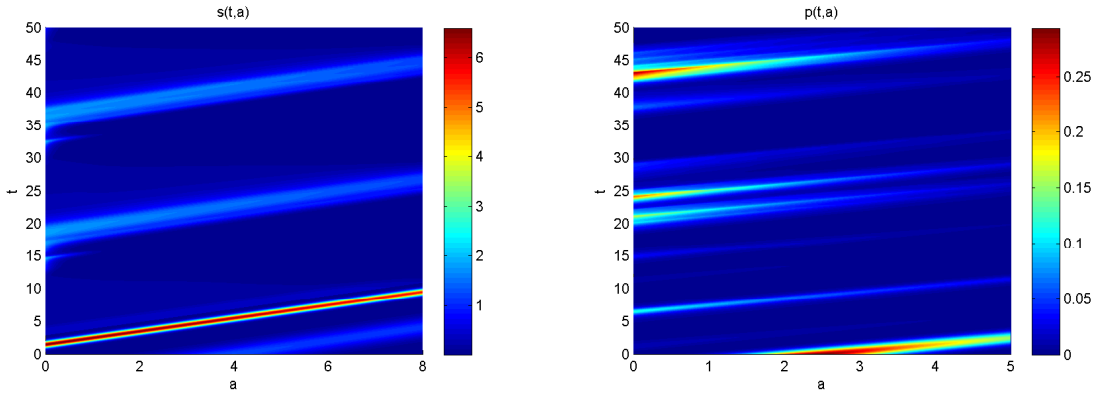

Figure 6.11: Expt. 3, Model B: The graphs of s and p obtained by simulating filgrastim treatment with period of 11 days, with daily dosage of 5 μg/kg, and with each treatment starting 4 days after chemotherapy. The $15 \leq t \leq 25$ region of the graph of s here shows more cells of age greater than 4 than in the same region in the previous figure's graph of s. There are slightly different oscillations in the $20 \leq t \leq 30$ region of p here than in the previous figure.

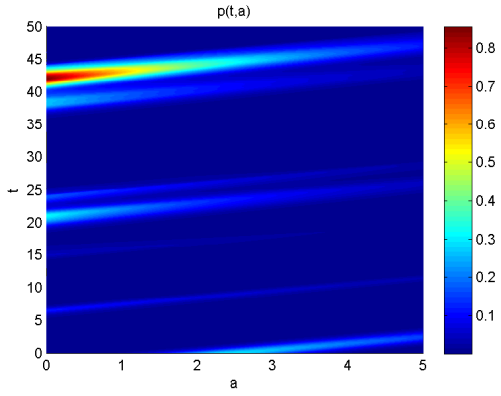

Figure 6.12: Expt. 3, Model B: The graph of p obtained by simulating filgrastim treatment with period of 11 days, with daily dosage of 5 μg/kg, and with each treatment starting 8 days after chemotherapy. The graph, in comparison to the graph of p in the previous figure, shows slightly different oscillations in the $20 \leq t \leq 30$ region and $35 \leq t \leq 45$ region.

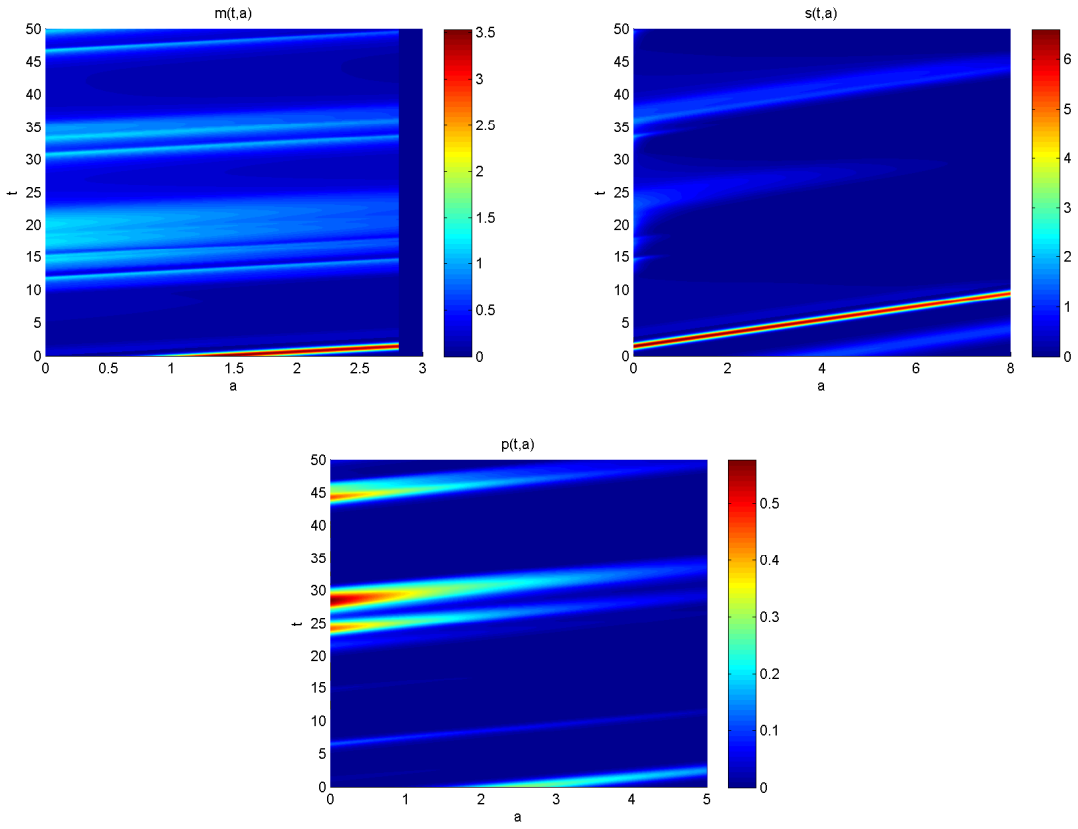

Figure 6.13: Expt. 3, Model B: The graphs of m, s, and p obtained by simulating filgrastim treatment with period of 4 days, with daily dosage of 5 μg/kg, and with each treatment starting 1 day after chemotherapy. In comparison to the previous figures obtained by approximating the full model, the graph of m shows more oscillations, the graph of s shows cells of age greater than 1 in two intervals of t instead of three, and the graph of p shows fewer intervals where the peaks are higher than 0.3.

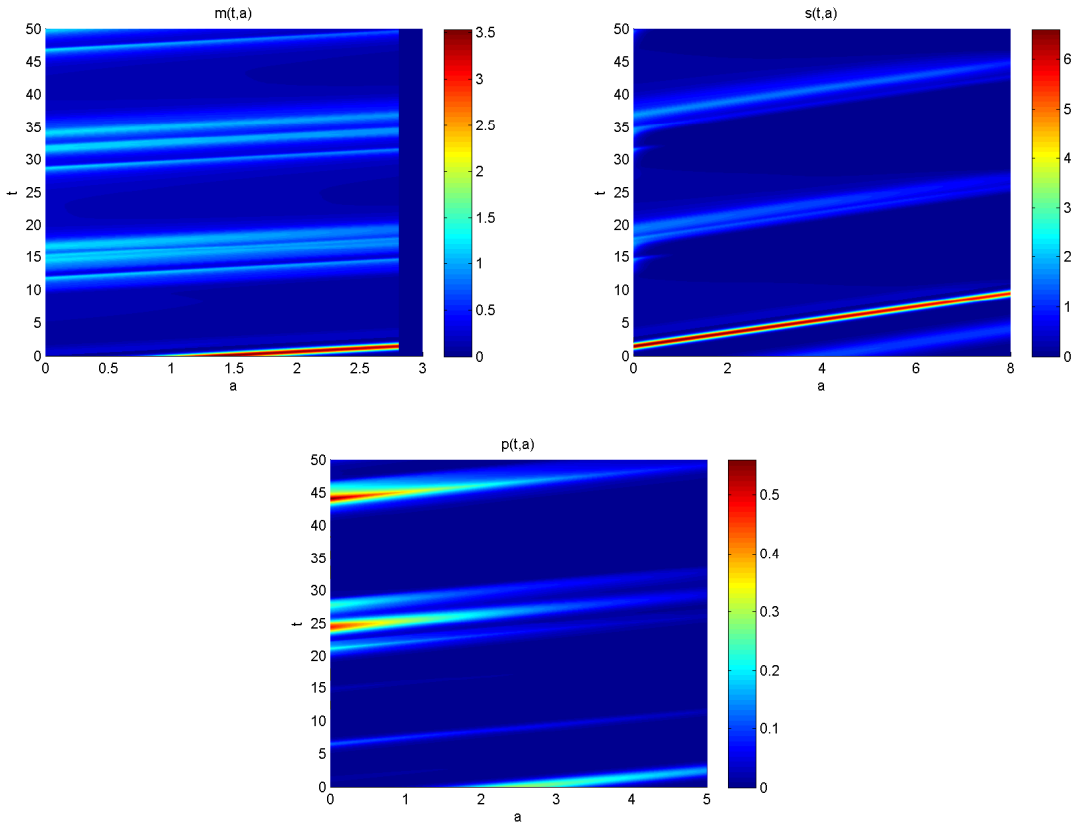

Figure 6.14: Expt. 3, Model B: The graphs of m, s, and p obtained by simulating filgrastim treatment with period of 8 days, with daily dosage of 5 μg/kg, and with each treatment starting 1 day after chemotherapy. The oscillations shown in the graph of m are slightly different from in the previous figure. The graph of s also shows behaviour slightly different from in its corresponding graph in the previous figure. In the graph of p, the most noticeable differences the oscillations and the higher peak in the interval $20 \leq t \leq 40$.

Treatment Protocol	Model A Observations	Model B Observations
Filgrastim treatment with period of 14 days, with daily dosage of 5 μg/kg, and without chemotherapy	Oscillations in m occur where $10 \leq t \leq 20$, s shows oscillations where $15 \leq t \leq 25$, p oscillates and the largest values occur in the intervals $t \leq 10$ and $20 \leq t \leq 30$, and n shows several oscillations with the highest peak occuring when $t \leq 5$, and w shows few oscillations.	m shows oscillations in the interval $10 \leq t \leq 20$; s shows few oscillations in the interval $15 \leq t \leq 30$; p shows two peaks of height at least 0.15 with small oscillations in between; n shows peaks in the intervals $t \leq 5$ and $10 \leq t \leq 15$; w shows few oscillations.

Treatment Protocol	Model A Observations	Model B Observations
Filgrastim treatment with period of 11 days, daily dosage of 5 μg/kg, and a post-chemotherapy delay of 1 day	Oscillations occur in m in the intervals $10 \leq t \leq 20$, $30 \leq t \leq 35$, and $45 \leq t \leq 50$; oscillations occur in s in the intervals $15 \leq t \leq 25$ and $30 \leq t \leq 45$; p, n, and w all show several oscillations without any pattern.	When $t > 10$, m and s both show oscillations with that almost show patterns; p shows inconsistent oscillations and shows peaks occuring in the intervals $20 \leq t \leq 30$ and $43 \leq t \leq 50$; n shows peaks in the intervals $t \leq 15$ and $30 \leq t \leq 40$; w shows several oscillations, but also shows very few cells of age greater than 2.
Filgrastim treatment with period of 11 days, daily dosage of 5 μg/kg, and a post-chemotherapy delay of 4 days	m, s, p, and w all show behaviour similar to that in the above experiment while n shows different and higher oscillations in the interval $40 \leq t \leq 50$.	m and s show behaviour similar to in the above experiment; p shows slightly different oscillations and a smaller peak in the interval $35 \leq t \leq 50$; n shows several oscillations with height no greater than 3 when $t \leq 10$; w shows oscillations which have height no greater than 10 when $t \leq 10$ and very few cells of age greater than 2.
Filgrastim treatment with period of 11 days, daily dosage of 5 μg/kg, and a post-chemotherapy delay of 8 days	m, s, and p all show behaviour similar to that in the above experiment; n and w all show several oscillations with relatively large heights.	Behaviour in m and s is very similar to that in the previous experiment; n shows few oscillations in the intervals $20 \leq t \leq 25$ and $37 \leq t \leq 45$; n shows several inconsistent oscillations with the highest values occuring in the interval $45 \leq t \leq 50$; w shows several oscillations and still shows very few cells of age greater than 2.

Treatment Protocol	Model A Observations	Model B Observations
Filgrastim treatment with period of 4 days, daily dosage of 5 μg/kg, and a post-chemotherapy delay of 1 day	All populations show several oscillations and peaks without any pattern.	All populations show several oscillations and peaks without any pattern.
Filgrastim treatment with period of 8 days, daily dosage of 5 μg/kg, and a post-chemotherapy delay of 1 day	Behaviour is very similar to that in the previous experiment.	Behaviour is similar to that in the previous experiment, except that m shows slightly different oscillations in $17 \leq t \leq 23$, s and p show slightly different behaviour in $15 \leq t \leq 30$, and n and w show higher maximum values.

6.4.1 Comparison to results of past experiments

Returning to the goals of G-CSF treatment, which were to find the correct time to begin treatment and to find the right length of time to wait between injections (see Chapter 1), we see that models A and B predict different results for the same treatment protocol.

A key difference between our work and [4] is that we are able to provide details of the cell population distributions within each phase of the cell line. Because we used the full model whereas Foley used a simplified model to obtain the results in [4], the results of our splitting method are different from Foley's results.

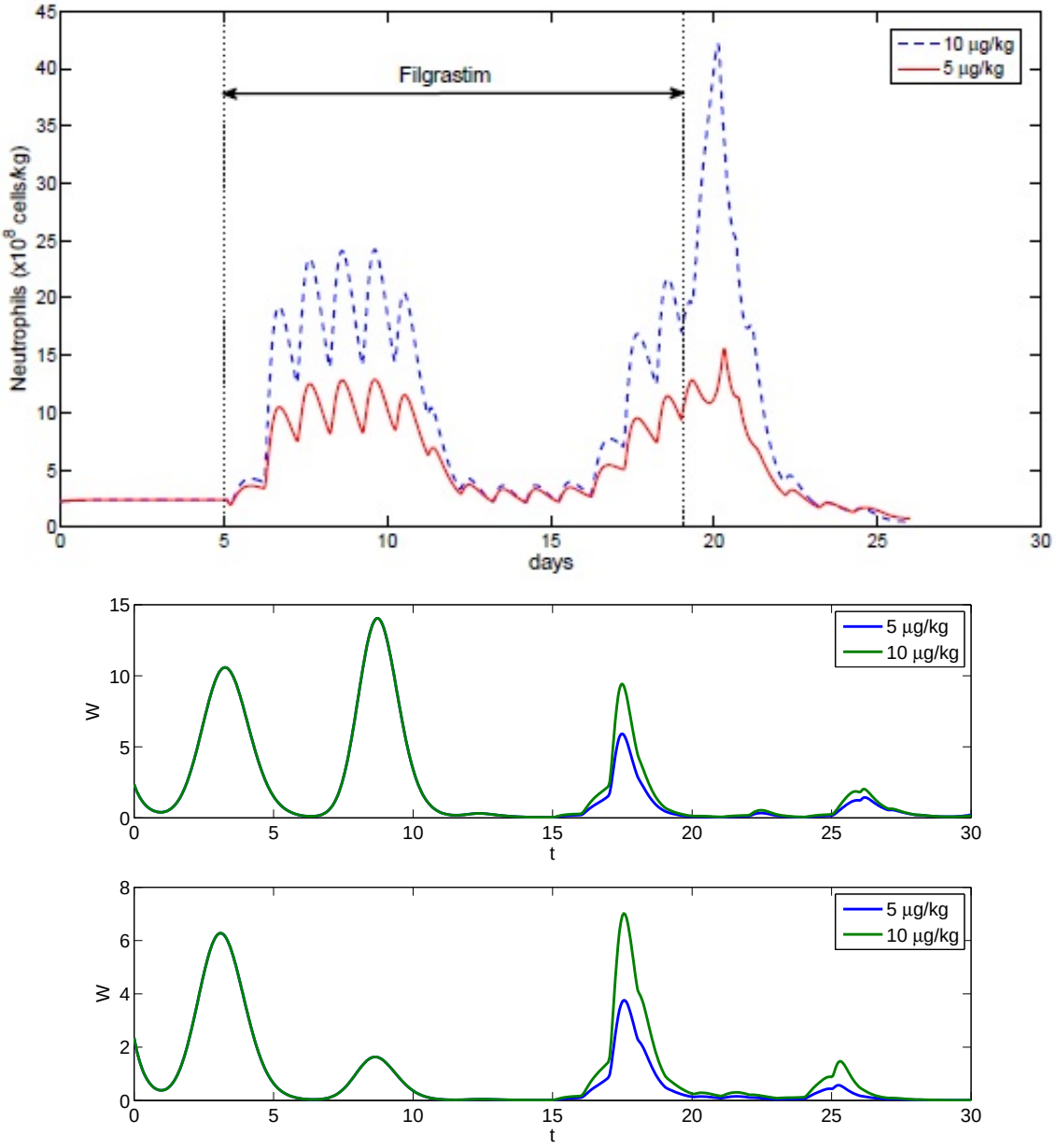

Figure 6.15: Neutrophil counts from varying the dosage of G-CSF and simulating without chemotherapy. Top: Neutrophil counts from [4]. Middle: Neutrophil counts based on Model A. Bottom: Neutrophil counts based on Model B.

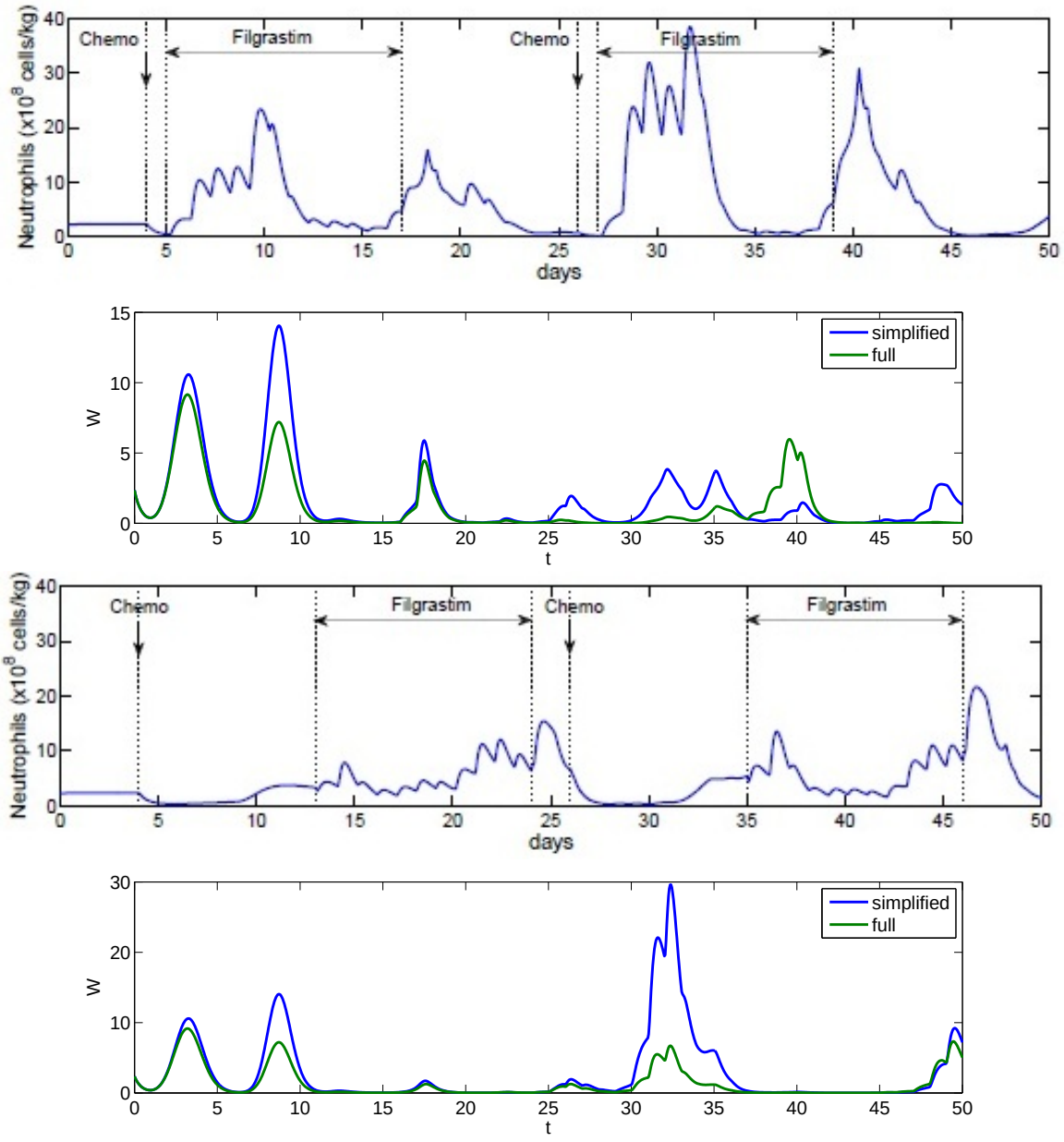

Figure 6.16: Top: Neutrophil counts from [4] from simulating filgrastim with post-chemotherapy delay of 1 day and treatment cycle of 12 days. Second from top: Neutrophil total graph obtained from the spliiting scheme when simulating the same treatment as above. Second from bottom: Graph from [4] from simulating filgrastim with post-chemotherapy delay of 8 days and treatment cycle of 11 days. Bottom: Graph obtained from splitting scheme when simulating the same treatment as for the graph immediately above.

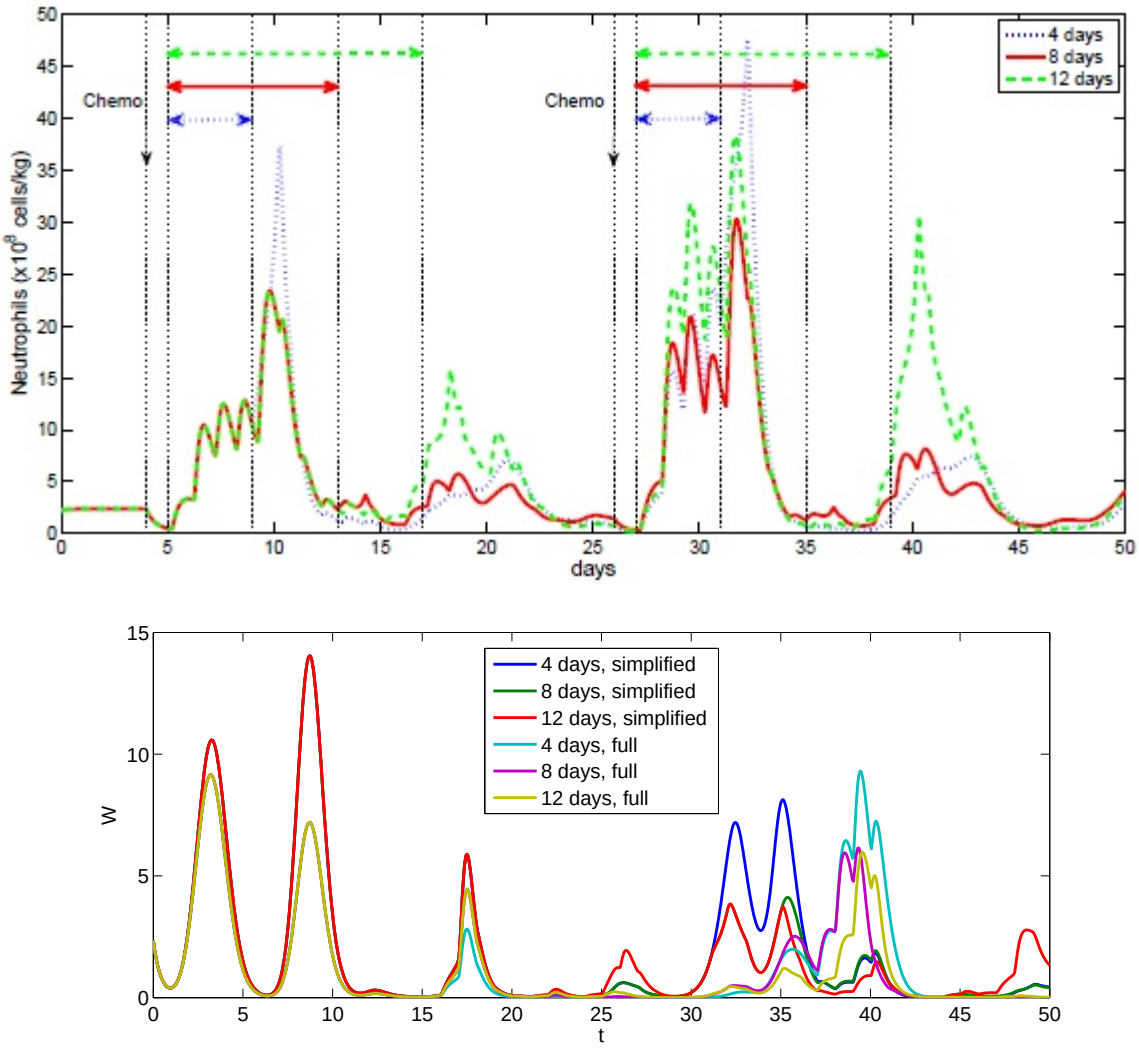

Figure 6.17: Neutrophil counts from simulating filgrastim with post-chemotherapy delay of 1 day and different treatment period lengths. Top: graph from [4]. Bottom: Graph obtained from splitting scheme.

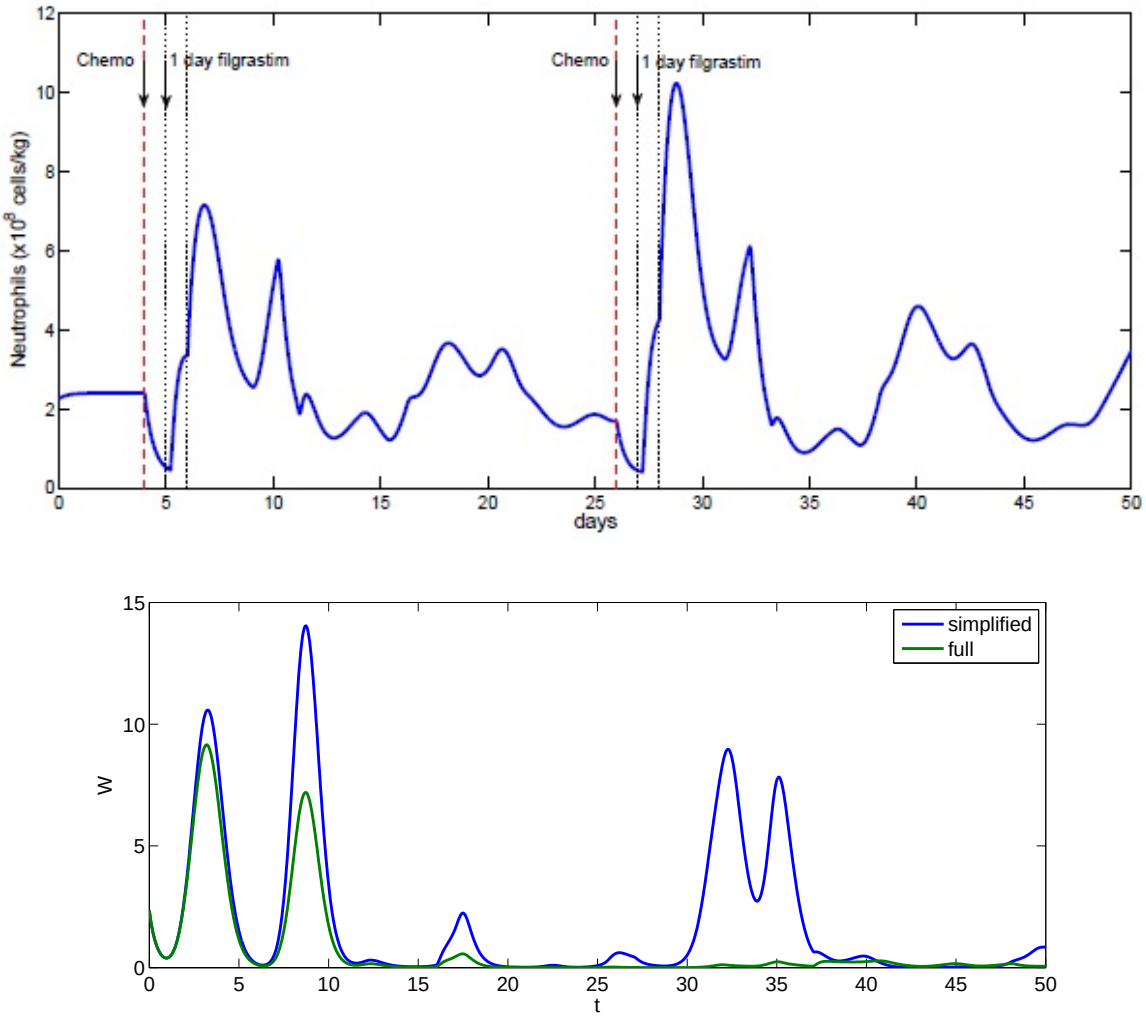

Figure 6.18: Neutrophil counts from simulating filgrastim with post-chemotherapy delay of 1 day and treatment cycle of 1 day. Top: graph from [4]. Bottom: Graph obtained from splitting scheme.

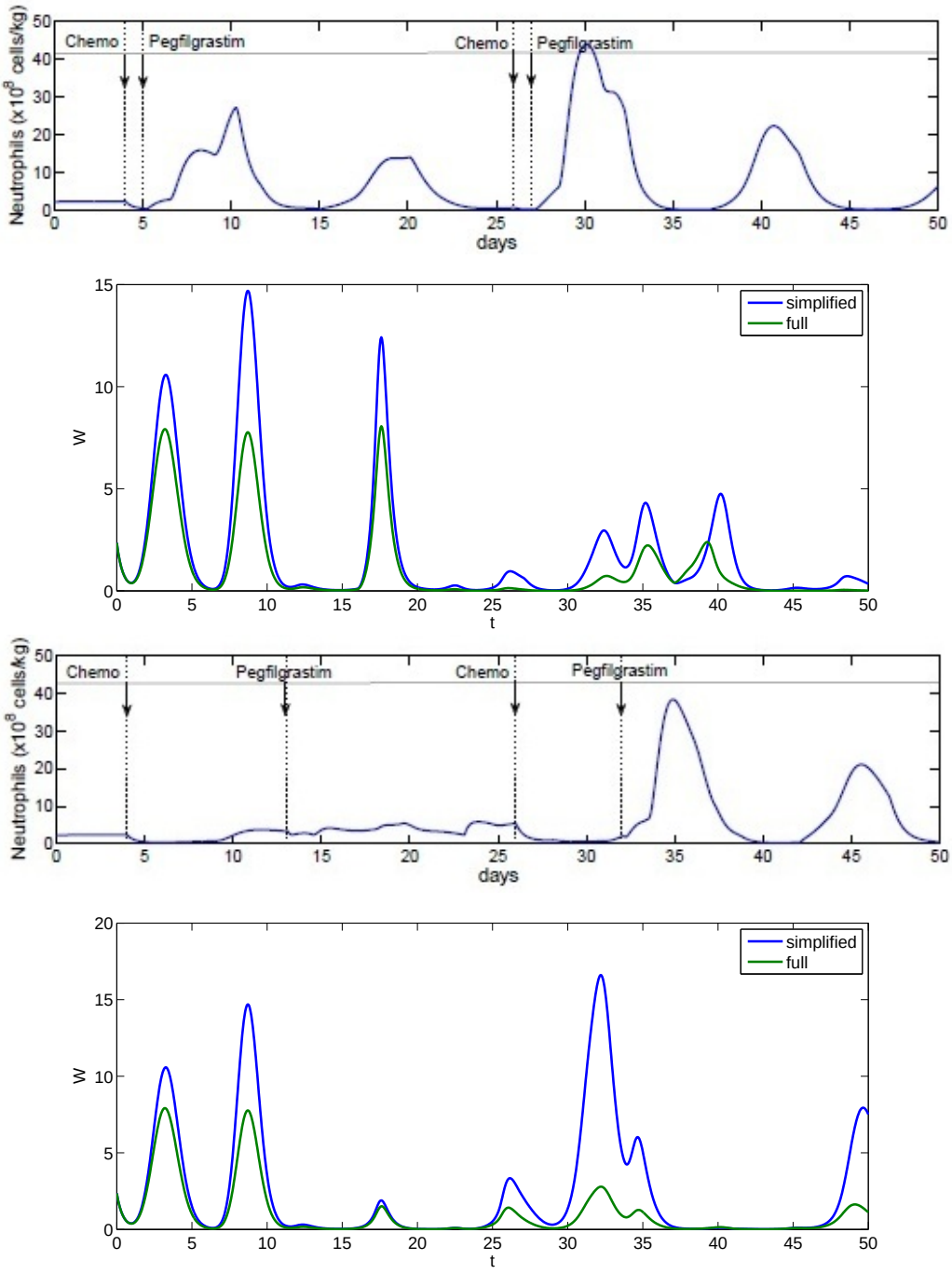

Figure 6.19: Top: Neutrophil counts from [4] from simulating pegfilgrastim with post-chemotherapy delay of 1 day. Second from top: Graph obtained from splitting scheme when simulating the same treatment. Second from bottom: Graph from [4] from simulating pegfilgrastim with post-chemotherapy delay of 8 days in the first cycle and 5 days in the second cycle. Bottom: Graph obtained from splitting scheme when simulating the same treatment.

Chapter 7

Conclusions and future work

In this thesis, we discussed a compartment model for cyclical neutropenia. We described several solution algorithms, both analytic and numerical, and the challenge of simulating the PDE model in [4]. We first tried an upwind scheme, but it was unsuccessful because of the different terminating ages for the various cell populations. The method of characteristics could not be readily used due to the nonlinear integral coupling on the boundary. The half-step algorithm in [14] as well as our modification of the algorithm did not appear to converge either. Our modification of the splitting method from [8], on the other hand, appeared convergent and gave conclusive results from different experiments.
We may consider the following work in the future:

- **Convergence and stability of splitting algorithm:** Although the modified splitting method discussed in Chapter 5 appears to be a convergent and stable scheme for simulating the PDE model in [4], this must be proven.

- *Analysis of the PDE model:* Though scalar age-structured population models are well-studied, the corresponding theory for systems is not well developed. Demonstrating well-posedness of the model (without G-CSF treatment) under physiologically reasonable assumptions on the transition rates and birth/death rates will require further analysis, which is another direction of future research.

Appendix A

Parameter values

The parameter values used in the thesis are taken from Chapter 4 in [4]. The table below listing these values is also reproduced from [4]:

Parameter Name	Value Used	Unit	Sources
Stem cell compartment			
S_*	1.1 (0.0001-1.1)	$\times 10^6$ cells/kg	Mackey (2001)
γ_s	0.05 (0.01-0.20)	days^{-1}	Bernard et al. (2003)
γ_s^{min}	0.05	days^{-1}	calculated
γ_s^{max}	0.20	days^{-1}	calculated
b_s	0.01	-	calculated
τ_s	2.8 (1.4 - 4.2)	days	Bernard et al. (2003)
k_0	8.0 (2.0-10.0)	days^{-1}	Colijn and Mackey (2005a)
θ_2	0.3	$\times 10^6$ cells/kg	Colijn and Mackey (2005a)
f_0	0.40	days^{-1}	Colijn and Mackey (2005a)
θ_1	0.36 (0.1-2.0)	$\times 10^8$ cells/kg	Colijn and Mackey (2005a)
Prolif. precursors compartment			
P_*	2.11	$\times 10^9$ cells/kg	Dancey et al. (1976)
γ_p	0.27	days^{-1}	Mackey et al. (2003)
γ_p^{min}	0.27	days^{-1}	Mackey et al. (2003)
γ_p^{max}	0.45	days^{-1}	calculated
b_p *(filgrastim)*	0.05	-	fit
b_p *(pegfilgrastim)*	1	-	fit
τ_p	5	days	Israels and Israels (2002)
A_{max}	20972	100	Bernard et al. (2003)
A_{min}	655	100	Bernard et al. (2003)
b_A *(filgrastim)*	0.35	-	fit
b_A *(pegfilgrastim)*	1.05	-	fit
Non-prolif. precursors compartment			
N_*	5.59	$\times 10^9$ cells/kg	Dancey et al. (1976)
γ_n	0.27	days^{-1}	Mackey et al. (2003)
γ_n^{min}	0.27	days^{-1}	Mackey et al. (2003)
γ_n^{max}	0.45	days^{-1}	calculated
b_n *(filgrastim)*	0.05	-	fit
b_n *(pegfilgrastim)*	1	-	fit
τ_N	6 (3.27-8.4)	days	Price et al. (1996)
V_{max}	6	-	calculated
b_v *(filgrastim)*	0.001	-	fit
b_v *(pegfilgrastim)*	0.08	-	fit
Neutrophils compartment			
W_*	6.9 (4.0 - 10.0)	$\times 10^8$ cells/kg	Abkowitz et al. (1988); Beutler et al. (1995)
γ_w	2.4 (2.2-2.5)	days^{-1}	Bernard et al. (2003)
G-CSF compartment			
X_*	0.1	μg/kg	Colijn et al. (2007)
G_*	0	μg/ml	Colijn et al. (2007)
V_B	76	mL/kg	Hayashi et al. (2001), Colijn et al. (2007)
G_{prod}	7.2 $\times 10^{-29}$	μg/(ml*day)	Vainstein et al. (2005)
Filgrastim			
k_T	1.68	day^{-1}	Hayashi et al. (2001),Colijn et al. (2007)
k_B	9.84	day^{-1}	Colijn et al. (2007)
σ	0.72	kg/day	Stute et al. (1992); Kearns et al. (1993a); Colijn et al. (2007)
γ_G	3.36	day^{-1}	fit
a	1200	μg/(kg*day)	(calculated)
s	0.0083	day	(calculated)
t_{on}	0.0083	day	(calculated)
k	10	-	fit
Pegfilgrastim			
k_T	0	day^{-1}	Roskos et al. (2006)
k_B	0.32	day^{-1}	fit
σ	0.01	kg/day	fit
γ_G	1.4	day^{-1}	fit
a	12048	μg/(kg*day)	(calculated)
s	0.0083	day	(calculated)
t_{on}	0.0083	day	(calculated)
k	0.01	-	fit

Table 4.1 Parameters of the model (steady state values).

Figure A.1: Parameter values for unscaled models. Source: [4]

Appendix B

MatLab Codes

```
\\ means "continued from previous line"

%upwind scheme
function [T,Mt,St,Pt,Nt,Wt] = PDEsolver(intT,intA,t,a)
tauM = 2.8;tauP = 5;tauN = 6;G=0;X=0.1;
A = 0:intA:a;T = 0:intT:t;
m = zeros( t/intT + 1, a/intA +1);s=m;p=m;n=m;w=m;
%proliferative stem cells
m(1,2:(round(tauM/intA))) = ((tauM-intA:-intA:intA));
%resting stem cells
s(1,2:end-1) = (A(end-1:-1:2)+.03)*(1.1/49.797);
%proliferative precursors
p(1,2:(round(tauP/intA))) = ((tauP-intA:-intA:intA)+.09)/6;
%non-proliferative precursers
n(1,2:(round(tauN/intA))) = ((tauN-intA:-intA:intA)+.25)/20;
%neutrophils
w(1,2:end-1) = (A(end-1:-1:2)+.1)./20;
s(1,1) = 2*m(1,(round(tauM/intA)+1));
w(1,1) = n(1,(round(tauN/intA)+1));
S = intA*sum(s(1,:) + [0,s(1,2:end-1),0])/2;
W = intA*sum(w(1,:) + [0,w(1,2:end-1),0])/2;
```

```
St = S; Wt = W;
m(1,1) = beta(S)*S;
p(1,1) = delta(W)*S;
n(1,1) = Amp(G)*p(1,(round(tauP/intA)+1));
M = intA*sum(m(1,1:(round(tauM/intA))+1)
\\+ [0,m(1,2:(round(tauM/intA))),0])/2;
P = intA*sum(p(1,1:(round(tauP/intA))+1)
\\+ [0,p(1,2:(round(tauP/intA))),0])/2;
N = intA*sum(n(1,1:(round(tauN/intA))+1)
\\+ [0,n(1,2:(round(tauN/intA))),0])/2;
Mt = M; Pt = P; Nt = N;
[dX,dG] = GCSF(X,G,W,0);
X = X + intT*dX; X = max(X,0);
G = G + intT*dG; G = max(G,0);
for i = 2:(round(t/intT)+1)
m(i,2:round(tauM/intA)+1) = m(i-1,2:(round(tauM/intA)+1))
\\ + intT*(((m(i-1,1:round(tauM/intA))- m(i-1,2:round(tauM/intA)+1))/intA)
\\ - gamma1('m',G)*m(i-1,2:(round(tauM/intA)+1)));
s(i,2:end) = s(i-1,2:end)+ intT*(((s(i-1,1:end-1)-s(i-1,2:end))/intA)
\\ - (delta(W) + beta(S))*s(i-1,2:end));
p(i,2:round(tauP/intA)+1) = p(i-1,2:round(tauP/intA)+1)
\\ + intT*(((p(i-1,1:round(tauP/intA))-p(i-1,2:round(tauP/intA)+1))/intA)
\\ - gamma1('p',G)*p(i-1,2:round(tauP/intA)+1));
n(i,2:round(tauN/intA)+1) = n(i-1,2:round(tauN/intA)+1)
\\ + intT*(Vn(G)*((n(i-1,1:round(tauN/intA))
\\- n(i-1,2:round(tauN/intA)+1))/intA)
\\ - gamma1('n',G)*n(i-1,2:round(tauN/intA)+1));
w(i,2:end) = w(i-1,2:end)+ intT*(((w(i-1,1:end-1)-w(i-1,2:end))/intA)
\\ - 2.4*w(i-1,2:end));
m = max(m,0);s = max(s,0);n = max(n,0);p = max(p,0);w = max(w,0);
s(i,1) = 2*m(i,round(tauM/intA)+1);
w(i,1) = n(i,round(tauN/intA)+1);
S = intA*sum(s(i,:) + [0,s(i,2:end-1),0])/2;
```

```
W = intA*sum(w(i,:) + [0,w(i,2:end-1),0])/2;
St = [St,S];Wt = [Wt,W];
m(i,1) = beta(S)*S;
p(i,1) = delta(W)*S;
n(i,1) = Amp(G)*p(i,round(tauP/intA)+1);
M = intA*sum(m(i,1:(round(tauM/intA))+1)
\\+ [0,m(i,2:(round(tauM/intA))),0])/2;
P = intA*sum(p(i,1:(round(tauP/intA))+1)
\\ + [0,p(i,2:(round(tauP/intA))),0])/2;
N = intA*sum(n(i,1:(round(tauN/intA))+1)
\\ + [0,n(i,2:(round(tauN/intA))),0])/2;
Mt = [Mt,M]; Pt = [Pt,P]; Nt = [Nt,N];
[dX,dG] = GCSF(X,G,W,intT*(i-1));
X = X + intT*dX; X = max(X,0);
G = G + intT*dG; G = max(G,0);
end;

 %half-step scheme
function [T,Mt,St,Pt,Nt,Wt] = PDEsolver2(intT,intA,t,a)
tauM = 2.8;tauP = 5;tauN = 6;G=0;X=0.1;
A = 0:intA:a;T = 0:intT:t;
m = zeros( t/intT + 1, a/intA +1);s=m;p=m;n=m;w=m;
m(1,2:(round(tauM/intA))) = ((tauM-intA:-intA:intA));
s(1,2:end-1) = (A(end-1:-1:2)+.03)*(1.1/49.797);
p(1,2:(round(tauP/intA))) = ((tauP-intA:-intA:intA)+.09)/6;
n(1,2:(round(tauN/intA))) = zeros(1,(round(tauN/intA))-1);
w(1,2:end-1) = (A(end-1:-1:2)+.1)./20;
s(1,1) = 2*m(1,(round(tauM/intA)+1));
w(1,1) = n(1,(round(tauN/intA)+1));
S = intA*sum(s(1,:) + [0,s(1,2:end-1),0])/2;
W = intA*sum(w(1,:) + [0,w(1,2:end-1),0])/2;
St = S; Wt = W;
m(1,1) = beta(S)*S;
```

```
p(1,1) = delta(W)*S;
n(1,1) = Amp(G)*p(1,(round(tauP/intA)+1));
M = intA*sum(m(1,1:(round(tauM/intA))+1)
\\ + [0,m(1,2:(round(tauM/intA))),0])/2;
P = intA*sum(p(1,1:(round(tauP/intA))+1)
\\ + [0,p(1,2:(round(tauP/intA))),0])/2;
N = intA*sum(n(1,1:(round(tauN/intA))+1)
\\ + [0,n(1,2:(round(tauN/intA))),0])/2;
Mt = M; Pt = P; Nt = N; Gt = G;
[dX,dG] = GCSF(X,G,W,0);
X = X + intT*dX; X = max(X,0);
G = G + intT*dG; G = max(G,0); Gt = [Gt,G];
for i = 2:(round(t/intT)+1)
G2 = .5*(Gt(end)+Gt(end-1));
m2 = m(i-1,2:(round(tauM/intA)+1))+ .5*intT*(((m(i-1,1:round(tauM/intA))
\\-m(i-1,2:round(tauM/intA)+1))/intA)
\\- gamma1('m',Gt(end-1))*m(i-1,2:(round(tauM/intA)+1)));
m3 = mini(m(i-1,1:(round(tauM/intA)+1))-[0,m(i-1,1:(round(tauM/intA)))],
\\m(i-1,2:(round(tauM/intA)+2))-m(i-1,1:(round(tauM/intA)+1)));
m4 = m(i-1,1:(round(tauM/intA)+1)) + .5*(1-(intT/intA))*m3
\\- (intT/2)*gamma1('m',Gt(end-1))*m(i-1,1:(round(tauM/intA)+1));
m(i,2:round(tauM/intA)+1) = m(i-1,2:(round(tauM/intA)+1)
\\- intT*(((m4(2:end)-m4(1:end-1))/intA) + gamma1('m',G2)*m2);
p2 = p(i-1,2:(round(tauP/intA)+1))+ .5*intT*(((p(i-1,1:round(tauP/intA))
\\-p(i-1,2:round(tauP/intA)+1))/intA)
\\- gamma1('p',Gt(end-1))*p(i-1,2:(round(tauP/intA)+1)));
p3 = mini(p(i-1,1:(round(tauP/intA)+1))-[0,p(i-1,1:(round(tauP/intA)))],
\\p(i-1,2:(round(tauP/intA)+2))-p(i-1,1:(round(tauP/intA)+1)));
p4 = p(i-1,1:(round(tauP/intA)+1)) + .5*(1-(intT/intA))*p3
\\- (intT/2)*gamma1('p',Gt(end-1))*p(i-1,1:(round(tauP/intA)+1));
p(i,2:round(tauP/intA)+1) = p(i-1,2:round(tauP/intA)+1)
\\- intT*(((p4(2:end)-p4(1:end-1))/intA) + gamma1('p',G2)*p2);
n2 = n(i-1,2:(round(tauN/intA)+1))
```

```
\\+ .5*intT*(Vn(Gt(end-1))*((n(i-1,1:round(tauN/intA))
\\-n(i-1,2:round(tauN/intA)+1))/intA)
\\- gamma1('n',Gt(end-1))*n(i-1,2:(round(tauN/intA)+1)));
n3 = mini(n(i-1,1:(round(tauN/intA)+1))-[0,n(i-1,1:(round(tauN/intA)))],
\\n(i-1,2:(round(tauN/intA)+2))-n(i-1,1:(round(tauN/intA)+1)));
n4 = n(i-1,1:(round(tauN/intA)+1)) + .5*(1-(Vn(Gt(end-1))*intT/intA))*n3
\\- (intT/2)*gamma1('n',Gt(end-1))*n(i-1,1:(round(tauN/intA)+1));
n(i,2:round(tauN/intA)+1) = n(i-1,2:round(tauN/intA)+1)
\\- intT*(Vn(G2)*((n4(2:end)-n4(1:end-1))/intA) + gamma1('n',G2)*n2);
w2 = w(i-1,2:end)+ .5*intT*(((w(i-1,1:end-1)-w(i-1,2:end))/intA)
\\- 2.4*w(i-1,2:end));
w3 = mini(w(i-1,1:end)-[0,w(i-1,1:end-1)],[w(i-1,2:end),0]-w(i-1,1:end));
w4 = w(i-1,1:end) + .5*(1-(intT/intA))*w3 - (intT/2)*2.4*w(i-1,1:end);
w(i,2:end) = w(i-1,2:end)+ intT*(((w4(1:end-1)-w4(2:end))/intA) - 2.4*w2);
W2 = .5*intT*(sum(w4)+sum(w2));
s2 = s(i-1,2:end)+ .5*intT*(((s(i-1,1:end-1)-s(i-1,2:end))/intA)
\\- (delta(W) + beta(S))*s(i-1,2:end));
s3 = s(i-1,1:end)+.5*(1-(intT/intA))*mini(s(i-1,1:end)-[0,s(i-1,1:end-1)],
\\[s(i-1,2:end),0]-s(i-1,1:end));
s4 = s(i-1,1:end) + .5*(1-(intT/intA))*s3
\\- (intT/2)*(delta(W) + beta(S))*s(i-1,1:end);
S2 = .5*intT*(sum(s4)+sum(s2));
s(i,2:end) = s(i-1,2:end)
\\+ intT*(((s4(1:end-1)-s4(2:end))/intA) - (delta(W2) + beta(S2))*s2);
m = max(m,0);s = max(s,0);n = max(n,0);p = max(p,0);w = max(w,0);
s(i,1) = 2*m(i,round(tauM/intA)+1);
w(i,1) = n(i,round(tauN/intA)+1);
S = intA*sum(s(i,:) + [0,s(i,2:end-1),0])/2;
W = intA*sum(w(i,:) + [0,w(i,2:end-1),0])/2;
St = [St,S];Wt = [Wt,W];
m(i,1) = beta(S)*S;
p(i,1) = delta(W)*S;
n(i,1) = Amp(G)*p(i,round(tauP/intA)+1);
```

```
M = intA*sum(m(i,1:(round(tauM/intA))+1)
\\ + [0,m(i,2:(round(tauM/intA))),0])/2;
P = intA*sum(p(i,1:(round(tauP/intA))+1)
\\ + [0,p(i,2:(round(tauP/intA))),0])/2;
N = intA*sum(n(i,1:(round(tauN/intA))+1)
\\ + [0,n(i,2:(round(tauN/intA))),0])/2;
Mt = [Mt,M]; Pt = [Pt,P]; Nt = [Nt,N];
[dX,dG] = GCSF(X,G,W,intT*(i-1));
X = X + intT*dX; X = max(X,0);
G = G + intT*dG; G = max(G,0); Gt = [Gt,G];
end;

%modified half-step scheme
function [T1,T2,T3,T4,T5,Mt,St,Pt,Nt,Wt] = PDEsolver3(intT,intA,t,a)
tauM = 2.8;tauP = 5;tauN = 6;G=0;X=0.1;
A = 0:intA:a;
T1=0;T2=T1;T3=T1;T4=T1;T5=T1;
m = zeros(1, a/intA +1);s=m;p=m;n=m;w=m;
m(2:(round(tauM/intA))) = ((tauM-intA:-intA:intA));
s(2:end-1) = (A(end-1:-1:2)+.03)*(1.1/49.797);
p(2:(round(tauP/intA))) = ((tauP-intA:-intA:intA)+.09)/6;
n(2:(round(tauN/intA))) = zeros(1,(round(tauN/intA))-1);
w(2:end-1) = (A(end-1:-1:2)+.1)./20;
s(1,1) = 2*m((round(tauM/intA)+1));
w(1,1) = n((round(tauN/intA)+1));
S = intA*sum(s(1,:) + [0,s(1,2:end-1),0])/2;
W = intA*sum(w(1,:) + [0,w(1,2:end-1),0])/2;
St = S; Wt = W;
m(1,1) = beta(S)*S;
p(1,1) = delta(W)*S;
n(1,1) = Amp(G)*p(1,(round(tauP/intA)+1));
M = intA*sum(m(1,1:(round(tauM/intA))+1)
\\ + [0,m(1,2:(round(tauM/intA))),0])/2;
```

```
P = intA*sum(p(1,1:(round(tauP/intA))+1)
\\ + [0,p(1,2:(round(tauP/intA))),0])/2;
N = intA*sum(n(1,1:(round(tauN/intA))+1)
\\ + [0,n(1,2:(round(tauN/intA))),0])/2;
Mt = M; Pt = P; Nt = N; Gt = G;
[dX,dG] = GCSF(X,G,W,0);
X = X + intT*dX/6; X = max(X,0);
G = G + intT*dG/6; G = max(G,0); Gt = [Gt,G];
sm = 0;wpn = 0;
for i = 2:(round(t*6/intT)+1)
        k = min([2,3],[i-1,max(i-2,1)]);
G2 = .5*(Gt(end)+Gt(end-1));
if sm == 1
m3 = mini(m(end,1:(round(tauM/intA)+1))-[0,m(end,1:(round(tauM/intA)))],
\\m(end,2:(round(tauM/intA)+2))-m(end,1:(round(tauM/intA)+1)));
m4 = m(end,1:(round(tauM/intA)+1))+ .5*(1-(k(1)*intT/(intA*6)))*m3
\\- (k(1)*intT/12)*gamma1('m',Gt(end-1))*m(end,1:(round(tauM/intA)+1));
m2 = (m4(2:end)+m4(1:end-1))/2;
m5 = zeros(1, a/intA +1);
m5(2:round(tauM/intA)+1) = m(end,2:(round(tauM/intA)+1))
\\- k(1)*intT/6*(((m4(2:end)-m4(1:end-1))/intA) + gamma1('m',G2)*m2);
m5(1) = beta(S)*S;
m = [m;m5];
M = intA*sum(m(end,1:(round(tauM/intA))+1)
\\+ [0,m(end,2:(round(tauM/intA))),0])/2;
Mt = [Mt,M];
T1 = [T1,i];
end
if wpn == 0
w3 = mini(w(end,1:end)-[0,w(end,1:end-1)],[w(end,2:end),0]-w(end,1:end));
w4 = w(end,1:end) + .5*(1-((k(2)*intT/6)/intA))*w3
\\- ((k(2)*intT/6)/2)*2.4*w(end,1:end);
w2 = (w4(2:end)+w4(1:end-1))/2;
```

```
w5 = w(end,2:end)+ k(2)*(intT/6)*(((w4(1:end-1)-w4(2:end))/intA) - 2.4*w2);
W2 = .5*intA*(sum(w4)+sum(w2));
w5 = [n(end,round(tauN/intA)+1),w5];
w = [w;w5];
W = intA*sum(w(end,:) + [0,w(end,2:end-1),0])/2;
Wt = [Wt,W];
T5 = [T5,i];
end
if wpn == 2
n3 = mini(n(end,1:(round(tauN/intA)+1))-[0,n(end,1:(round(tauN/intA)))],
\\n(end,2:(round(tauN/intA)+2))-n(end,1:(round(tauN/intA)+1)));
n4 = n(end,1:(round(tauN/intA)+1))
\\+ .5*(1-(Vn(Gt(end-1))*(k(2)*intT/6)/intA))*n3
\\- ((k(2)*intT/6)/2)*gamma1('n',Gt(end-1))*n(end,1:(round(tauN/intA)+1));
n2 = (n4(2:end)+n4(1:end-1))/2;
n5 = zeros(1, a/intA +1);
n5(2:round(tauN/intA)+1) = n(end,2:round(tauN/intA)+1)
\\- k(2)*(intT/6)*(Vn(G2)*((n4(2:end)-n4(1:end-1))/intA)
\\ + gamma1('n',G2)*n2);
n5(1) = Amp(G)*p(end,round(tauP/intA)+1);
n = [n;n5];
N = intA*sum(n(end,1:(round(tauN/intA))+1)
\\+ [0,n(end,2:(round(tauN/intA))),0])/2;
Nt = [Nt,N];
T4 = [T4,i];
end
if wpn == 1
p3 = mini(p(end,1:(round(tauP/intA)+1))-[0,p(end,1:(round(tauP/intA)))],
\\p(end,2:(round(tauP/intA)+2))-p(end,1:(round(tauP/intA)+1)));
p4 = p(end,1:(round(tauP/intA)+1)) + .5*(1-((k(2)*intT/6)/intA))*p3
\\- ((k(2)*intT/6)/2)*gamma1('p',Gt(end-1))*p(end,1:(round(tauP/intA)+1));
p2 = (p4(2:end)+p4(1:end-1))/2;
p5 = zeros(1, a/intA +1);
```

```
p5(2:round(tauP/intA)+1) = p(end,2:round(tauP/intA)+1)
\\- k(2)*(intT/6)*(((p4(2:end)-p4(1:end-1))/intA) + gamma1('p',G2)*p2);
p5(1) = delta(W)*S;
p = [p;p5];
P = intA*sum(p(end,1:(round(tauP/intA))+1)
\\+ [0,p(end,2:(round(tauP/intA))),0])/2;
Pt = [Pt,P];
T3 = [T3,i];
end
if sm == 0
s3 = s(end,1:end)
\\+.5*(1-((k(1)*intT/6)/intA))*mini(s(end,1:end)-[0,s(end,1:end-1)],
\\[s(end,2:end),0]-s(end,1:end));
s4 = s(end,1:end) + .5*(1-((k(1)*intT/6)/intA))*s3
\\- ((k(1)*intT/6)/2)*(delta(W) + beta(S))*s(end,1:end);
s2 = (s4(2:end)+s4(1:end-1))/2;
S2 = .5*intA*(sum(s4)+sum(s2));
s5 = s(end,2:end)
\\+ k(1)*(intT/6)*(((s4(1:end-1)-s4(2:end))/intA)
\\ - (delta(W2) + beta(S2))*s2);
s5 = [2*m(end,round(tauM/intA)+1),s5];
s = [s;s5];
S = intA*sum(s(end,:) + [0,s(end,2:end-1),0])/2;
St = [St,S];
T2 = [T2,i];
end
m = max(m,0);s = max(s,0);n = max(n,0);p = max(p,0);w = max(w,0);
[dX,dG] = GCSF(X,G,W,intT*(i-1)/6);
X = X + intT*dX/6; X = max(X,0);
G = G + intT*dG/6; G = max(G,0); Gt = [Gt,G];
sm = abs(sm - 1);
wpn = mod(wpn+1,3);
end;
```

```
T1=T1*(intT/6);
T2=T2*(intT/6);
T3=T3*(intT/6);
T4=T4*(intT/6);
T5=T5*(intT/6);

%splitting scheme with full model
function [T,A,M1,S1,P1,N1,W1,W2]= PDEsolver4(intA,a,t,fil,chemo,trtmnt,run)
    [tM,tP,tN,gW,omega,Gi,Xi,VB,Gp] = param2();
    tplot=10*intA; plotgap=round(tplot/intA); nplots = round(t/tplot);
    [k,kT,kB,gammaG,sgm,a1,dur,ton,bp,bA,bn,bv] = filg(fil);
    A = 0:intA:a;T = 0;
    [m,s,p,n,w] = init2(tM,tP,tN,a,A,intA);
    st = (sum(s(1,:))+sum(s(1,2:(end-1))))*(intA/2);
    wt = (sum(w(1,:))+sum(w(1,2:(end-1))))*(intA/2);
    m(1,1) = beta(st)*st;
    p(1,1) = delta(wt*100)*st;
    p0 = p(1,round(tP/intA)+1);
    n(1,1) = 100*Amp2(Gi,bA)*p0/1000;
    G = Gi;X = Xi;
    M1 = [m(1,:);zeros(nplots,length(A))];
    S1 = [s(1,:);zeros(nplots,length(A))];
    P1 = [p(1,:);zeros(nplots,length(A))];
    N1 = [n(1,:);zeros(nplots,length(A))];
    W1 = [w(1,:);zeros(nplots,length(A))];
    W2 = wt;
    j=1;tt=0;
    for i = 1:nplots
        for I = 1:plotgap
            tt = tt + intA;
            if tt > run(j+1)
                j = j+1;
            end
```

```
            ii = max(tt-run(j),0);
            if ii < chemo(j)
                gM = gamma3('m',ii*intA);
                gP = gamma3('p',ii*intA);
                gN = gamma3('n',ii*intA);
            else
                gM = gamma2('m',G(1),bp,bn);
                gP = gamma2('p',G(1),bp,bn);
                gN = gamma2('n',G(1),bp,bn);
            end
            dS = delta(wt*100)+beta(st);
            m1 = m(1,1:round(tM/intA))/((1+omega*gM*intA)*(1+(1-omega)*
\\gM*intA));
            m(2,2:round(tM/intA)+1) = m1;
            s(2,1) = 2*m1(end);
            s(2,2:end) = s(1,1:(end-1))/((1+omega*dS*intA)*(1+(1-omega)*
\\dS*intA));
            p1 = p(1,1:round(tP/intA))/((1+omega*gP*intA)*(1+(1-omega)*
\\gP*intA));
            p(2,2:round(tP/intA)+1) = p1;
            n1 = n(1,1:round(tN/intA))/((1+omega*gN*intA)*
\\(1+(1-omega)*gN*(intA/Vn2(G(1),bv))));
            p0 = p(2,round(tP/intA)+1);
            n(2,1) = 100*Amp2(G(1),bA)*p0/1000;
            w(2,1) = n1(end)*10;
            n(2,2:round(tN/intA)+1) = n1;
            w(2,2:end) = w(1,1:(end-1))/((1+omega*gW*intA)*(1+(1-omega)*
\\gW*intA));
            st = (sum(s(2,:))+sum(s(2,2:(end-1))))*(intA/2);
            wt = (sum(w(2,:))+sum(w(2,2:(end-1))))*(intA/2);
            m(2,1) = beta(st)*st;
            p(2,1) = delta(wt*100)*st;
            m(1,:) = m(2,:);
```

```
            s(1,:) = s(2,:);
            p(1,:) = p(2,:);
            n(1,:) = n(2,:);
            w(1,:) = w(2,:);
            F = (G(1)^2)/(G(1)^2 + k);
            ik = 0;
            if ii >= chemo(j)
                ik = max(ii-chemo(j),0);
            end;
            il = mod(mod(min(ik,trtmnt),trtmnt),1);
            dX = I2(il,a1,ton,dur) + kT*VB*G(1) - kB*X(1);
            dG = Gp + kB*X(1)/VB - kT*G(1) - G(1)*(gammaG + sgm*wt*100*F);
            G = G+(intA*dG);
            X = X+(intA*dX);
        end;
        M1(i+1,:) = m(2,:);
        S1(i+1,:) = s(2,:);
        P1(i+1,:) = p(2,:);
        N1(i+1,:) = n(2,:);
        W1(i+1,:) = w(2,:);
        T = [T,tt];
        W2 = [W2,wt];
    end

%splitting scheme with simplified model
function [T,A,M1,S1,P1,N1,W1,W2]= PDEsolver5(intA,a,t,fil,chemo,trtmnt,run)
    [tM,tP,tN,gW,omega,Gi,Xi,VB,Gp] = param2();
    tplot=10*intA; plotgap=round(tplot/intA); nplots = round(t/tplot);
    [k,kT,kB,gammaG,sgm,a1,dur,ton,~,bA,~,~] = filg(fil);
    A = 0:intA:a;T = 0;
    [m,s,p,n,w] = init2(tM,tP,tN,a,A,intA);
    st = (sum(s(1,:))+sum(s(1,2:(end-1))))*(intA/2);
    wt = (sum(w(1,:))+sum(w(1,2:(end-1))))*(intA/2);
```

```
    m(1,1) = beta(st)*st;
    p(1,1) = delta(wt*100)*st;
    p0 = p(1,round(tP/intA)+1);
    n(1,1) = 100*Amp2(Gi,bA)*p0/1000;
    G = Gi;X = Xi;
    M1 = [m(1,:);zeros(nplots,length(A))];
    S1 = [s(1,:);zeros(nplots,length(A))];
    P1 = [p(1,:);zeros(nplots,length(A))];
    N1 = [n(1,:);zeros(nplots,length(A))];
    W1 = [w(1,:);zeros(nplots,length(A))];
    W2 = wt;
    j=1;tt=0;
    for i = 1:nplots
        for I = 1:plotgap
            tt = tt + intA;
            if tt > run(j+1)
                j = j+1;
            end
            ii = max(tt-run(j),0);
            if ii < chemo(j)
                gM = gamma3('m',ii*intA);
                gP = gamma3('p',ii*intA);
                gN = gamma3('n',ii*intA);
            else
                gM = .05;
                gP = .27;
                gN = .27;
            end
            dS = delta(wt*100)+beta(st);
            m1 = m(1,1:round(tM/intA))/((1+omega*gM*intA)*
\\(1+(1-omega)*gM*intA));
            m(2,2:round(tM/intA)+1) = m1;
            s(2,1) = 2*m1(end);
```

```
            s(2,2:end) = s(1,1:(end-1))/((1+omega*dS*intA)*
\\(1+(1-omega)*dS*intA));
            p1 = p(1,1:round(tP/intA))/((1+omega*gP*intA)*(1+(1-omega)*
\\gP*intA));
            p(2,2:round(tP/intA)+1) = p1;
            n1 = n(1,1:round(tN/intA))/((1+omega*gN*intA)*(1+(1-omega)*
\\gN*(intA/6)));
            p0 = p(2,round(tP/intA)+1);
            n(2,1) = 100*Amp2(G(1),bA)*p0/1000;
            w(2,1) = n1(end)*10;
            n(2,2:round(tN/intA)+1) = n1;
            w(2,2:end) = w(1,1:(end-1))/((1+omega*gW*intA)*(1+(1-omega)*
\\gW*intA));
            st = (sum(s(2,:))+sum(s(2,2:(end-1))))*(intA/2);
            wt = (sum(w(2,:))+sum(w(2,2:(end-1))))*(intA/2);
            m(2,1) = beta(st)*st;
            p(2,1) = delta(wt*100)*st;
            m(1,:) = m(2,:);
            s(1,:) = s(2,:);
            p(1,:) = p(2,:);
            n(1,:) = n(2,:);
            w(1,:) = w(2,:);
            F = (G(1)^2)/(G(1)^2 + k);
            ik = 0;
            if ii >= chemo(j)
                ik = max(ii-chemo(j),0);
            end;
            il = mod(mod(min(ik,trtmnt),trtmnt),1);
            dX = I2(il,a1,ton,dur) + kT*VB*G(1) - kB*X(1);
            dG = Gp + kB*X(1)/VB - kT*G(1) - G(1)*(gammaG + sgm*wt*100*F);
            G = G+(intA*dG);
            X = X+(intA*dX);
        end;
```

```
        M1(i+1,:) = m(2,:);
        S1(i+1,:) = s(2,:);
        P1(i+1,:) = p(2,:);
        N1(i+1,:) = n(2,:);
        W1(i+1,:) = w(2,:);
        T = [T,tt];
        W2 = [W2,wt];
    end

function [tM,tP,tN,gW,omega,Gi,Xi,VB,Gp] = param2()
    tM = 2.8;tP = 5;tN = 6;
    gW = 2.4;
    omega = .5;
    Gi = 0;
    Xi = 1;
    VB =76;Gp=7.2*(10^(-29));

function [k,kT,kB,gammaG,sgm,a,s,ton,bp,bA,bn,bv] = filg(i)
    ki = [10,.01]; k = ki(1);
    kTi = [1.68,0]; kT = kTi(i);
    kBi = [9.84,.32]; kB = kBi(i);
    gammaGi = [3.36,1.4]; gammaG = gammaGi(i);
    sgmi = [.72,.01]; sgm = sgmi(i);
    ai = [600,12048]; a = ai(i);
    si = [.0083,.0083]; s = si(i);
    toni = [.0083,.0083]; ton = toni(i);
    bpi = [.05,1]; bp = bpi(i);
    bAi = [.35,1.05]; bA = bAi(i);
    bni = [.05,1]; bn = bni(i);
    bvi = [.001,.08]; bv = bvi(i);

function [m,s,p,n,w] = init2(tM,tP,tN,a,A,intA)
    m = zeros(2, round(a/intA) +1);p=m;n=m;
```

```
    m(1,1:round(tM/intA)+1) = 3.1*normpdf(A(1:round(tM/intA)+1),tM/2,tM/8);
    s(1,:) = 3.1*normpdf(A,a/2,a/8);
    s(1,1) = 2*m(1,round(tM/intA)+1);
    p(1,1:round(tP/intA)+1) = 0.46*normpdf(A(1:round(tP/intA)+1),tP/2,tP/8);
    n(1,1:round(tN/intA)+1) = 8.45*normpdf(A(1:round(tN/intA)+1),tN/2,tN/8);
    w(1,:) = 2.35*normpdf(A,a/2,a/8);
    w(1,1) = n(1,round(tN/intA)+1)*10;

function A = Amp(G)
Amax = 20972;
Amin = 655;
bA = .35;
A = (Amax-Amin)*G/(G + bA) + Amin;

function A = Amp2(G,bA)
Amax = 20972;
Amin = 655;
A = (Amax-Amin)*G/(G + bA) + Amin;

function b = beta(S)
k0 = 8;
theta2 = .3;
b = k0*(theta2^2)/(theta2^2 + S^2);

function d = delta(W)
f0 = .4;
theta1 = .36;
d = f0*theta1/(theta1+W);

function g = gamma1(script,G)
%determines the value of gamma_script
b = 1;
gmax = 1;
```

```
gmin = 1;
if (script == 'm')
b = .01;
gmax = .2;
gmin = .05;
elseif (script == 'p')
b = .05;
gmax = .45;
gmin = .27;
else
b = .05;
gmax = .45;
gmin = .27;
end;
g = (gmax-gmin)*b/(G + b) + gmin;

function g = gamma2(script,G,bp,bn)
%determines the value of gamma_script
b = 1;gmax=0;gmin=0;
if (script == 'm')
b = .01;
gmax = .2;
gmin = .05;
elseif (script == 'p')
b = bp;
gmax = .45;
gmin = .27;
else
b = bn;
gmax = .45;
gmin = .27;
end;
g = (gmax-gmin)*b/(G + b) + gmin;
```

```
function g = gamma3(script,t)
%determines the value of gamma_script
gmax=0;gmin=0;
if (script == 'm')
gmax = .2;
gmin = .05;
elseif (script == 'p')
gmax = .45;
gmin = .27;
else
gmax = .45;
gmin = .27;
end;
g = (gmax-gmin)*t/8 + gmin;

function v = Vn(G)
Vmax = 6;
bv = .001;
v = (Vmax-1)*G/(G + bv) + 1;

function v = Vn2(G,bv)
Vmax = 6;
v = (Vmax-1)*G/(G + bv) + 1;

function m = mini(s1,s2)
    w1 = 0; w2 = w1;
if s1<=0 & 0<s2
       w1 = 1;
    end
if s2<=0 & 0<s1
        w2 = 1;
    end;
```

```
w = abs(w1+w2-1);
m = w.*sign(s1).*min(abs(s1),abs(s2));

function [dX,dG] = GCSF(X,G,W,t)
k=10;kT=1.68;VB =76;kB=9.84;
Gprod=7.2*(10^(-29));gammaG=3.36;sigma=.72;
F = (G^2)/(G^2 + k);
dX = I(t) + kT*VB*G - kB*X;
dG = Gprod + kB*X/VB - kT*G - G*(gammaG + sigma*W*F);
```

Bibliography

[1] Dimitri Breda, Caterina Cusulin, Mimmo Iannelli, Stefano Maset, and Rossana Vermiglio. Stability analysis of age-structured population equations by pseudospectral differencing methods. *J. Math. Biol.*, 54(5):701–720, 2007.

[2] J. M. Cushing. *An introduction to structured population dynamics*, volume 71 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998.

[3] Lawrence C. Evans. *Partial differential equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, second edition, 2010.

[4] Catherine Foley. *Mathematical modeling for designing new treatment strategies with Granulocyte-Colony Stimulating Factor*. PhD thesis, McGill University, 2008.

[5] Catherine Foley, Samuel Bernard, and Michael C. Mackey. Cost-effectve G-CSF therapy strategies for cyclical neutropenia: mathematical modelling based hypotheses. *J. Theoret. Biol.*, 238(4):754–763, 2006.

[6] M. E. Gurtin and R. C. MacCamy. Diffusion models for age-structured populations. *Math. Biosci.*, 54(1-2):49–59, 1981.

[7] Morton E. Gurtin and Richard C. MacCamy. Non-linear age-dependent population dynamics. *Arch. Rational Mech. Anal.*, 54:281–300, 1974.

[8] Mimmo Iannelli, Mi-Young Kim, and Eun-Jae Park. Splitting methods for the numerical approximation of some models of age-structured population dynamics and epidemiology. *Appl. Math. Comp.*, 87(1):69–93, 1997.

[9] M.-Y. Kim and E.-J. Park. An upwind scheme for a nonlinear model in age-structured population dynamics. *Comput. Math. Appl.*, 30(8):5–17, 1995.

[10] Mi-Young Kim and Ts. Selenge. Age-time discontinuous Galerkin method for the Lotka-McKendrick equation. *Commun. Korean Math. Soc.*, 18(3):569–580, 2003.

[11] Mi-Young Kim and Tsendauysh Selenge. Age-time continuous Galerkin method for a model of population dynamics. *J. Comput. Appl. Math.*, 223(2):659–671, 2009.

[12] Randall J. LeVeque. *Finite difference methods for ordinary and partial differential equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2007. Steady-state and time-dependent problems.

[13] Walter A. Strauss. *Partial differential equations*. John Wiley & Sons Ltd., Chichester, second edition, 2008. An introduction.

[14] Deborah Sulsky. Numerical solution of structured population models. I. Age structure. *J. Math. Biol.*, 31(8):817–839, 1993.