Research Blood Coagulation

The Blood Coagulation Process: A Mathematical Model

Final Report Internship

Author:
Nicky van den Berg
0923424
n.j.v.d.berg@student.tue.nl

Western University Supervisor:
Mikko Karttunen
mkarttu@uwo.ca

TU/e Supervisor:
Barry Koren
b.koren@tue.nl

September 7, 2020

Abstract

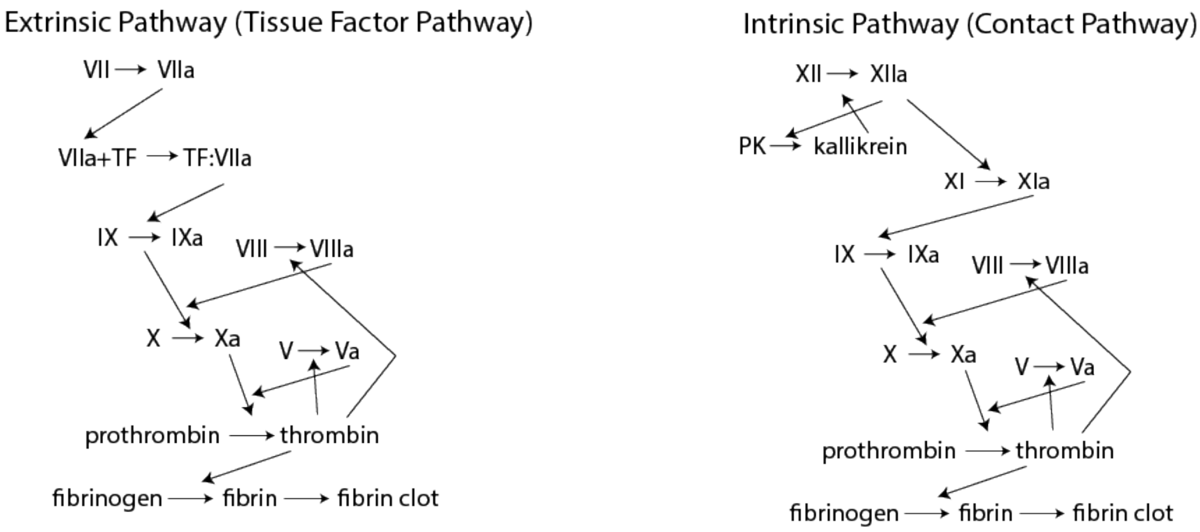
In this paper, the model introduced by Tatiana Galochkina, Anass Bouchnita, Polina Kurbatova and Vitaly Volpert is reconstructed and discussed [1]. In Section 1, we get insight in how the blood coagulation system works and which models have been made before. In Section 2, we introduce the model of Galochkina et al. [1]. In Section 3, we show that there exists a solution that is stable. We simplify the model in Section 5, after which we determine an approximation of the wave propagation speed using the narrow reaction zone method and a piecewise linear approximation. In Section 6, we use a simulation to determine the propgation of the wave speed. In Section 7, we summarize the results, and in Section 8 we recommend some topics that need further research.

Contents

1 Introduction

It is essential that, when someone falls and a blood vessel ruptures, the bleeding is stopped as fast as possible. Our body makes sure this happens by activating the blood coagulation system. Different chemical substances are present in our blood. When our blood starts clotting, different factors are activated consecutively. We denote the different factors by Roman numerals and an ‘a’ is added when the factor is in its activated form.

The blood coagulation system can be activated in different ways. The first one that we will discuss is when the blood comes into contact with the cells just outside of the vessel wall. We call this the extrinsic pathway or the tissue factor pathway. The second activation method is when the blood is in contact with an artificial surface. We call this the intrinsic pathway or the contact pathway. Both pathways are activated when a blood vessel rupture takes place. The extrinsic pathway is explosive (clotting can occur after only 15 seconds), while the intrinsic pathway is slower (takes 1 to 6 minutes to cause clotting) [2].



(a) Extrinsic pathway. (b) Intrinsic pathway.

Figure 1: Visual representation of the different pathways.

One should be aware that calcium ions promote and accelerate the blood clotting process [2].

1.1 Extrinsic Pathway/Tissue Factor Pathway

Factor VII circulates in plasma, mostly in inactive form (about 1% in active form). This substance is synthesized in the liver. The non-activated form has a half-life of 5 hours while activated Factor VII has a half-life of 2 hours. It is not completely clear how *in vivo* activation works, but it can be activated *in vitro* by a number of factors including Factor IXa, Factor Xa, Factor XIIa, thrombin, plasmin, Factor VII-activating protease and TF:VIIa complex [3].

Tissue factor (TF, Factor III) is mainly present in the cells surrounding the blood vessels that are larger than capillaries, in keratinocytes in the skin, in several epithelial layers (e.g. organs) and especially in large quantities around organs where bleeding has disastrous consequences, e.g. the brain or the kidneys. When the (activated) Factor VII comes in contact with the tissue factor, a chemical bond between both factors arises. If TF binds with the non-activated form of Factor VII, it is rapidly activated. The created substance is denoted by TF:VIIa. Factor VIIa can activate substrates (Factor VII, Factor IX, Factor X) extremely slow, but TF:VIIa enhances this process with at least 5 orders of magnitude [3].

The TF:VIIa complex catalyses the activation of Factor IX. Factor IX is synthesized in the liver. On its turn, Factor IXa activates Factor X. This process is catalysed by Factor VIIIa. Factor Xa activates the process that changes prothrombin into thrombin. This process is catalysed by Factor Va. Thrombin also catalyses the activation of Factor V and of Factor VIII, so a positive feedback-loop is created. The thrombin makes sure that the fibrinogen is converted into fibrin strings, which will form a fibrin clot later on. This will form a hemostatic plug or thrombosis [3].

A visual representation of this process is given in Figure 1a.

1.2 Intrinsic Pathway/Contact Pathway

The intrinsic pathway is initiated by the activation of Factor XII. This occurs when Factor XII comes into contact with artificial surfaces (e.g. glass test tubes [3]) and it leads to a change in the production of Factor XII, such that small amounts of Factor XIIa are produced. This substance is produced in the liver, and circulates in plasma. It can be activated via limited proteolysis (breakdown of proteins into smaller polypeptides or amino acids) by kallikrein, plasmin or Factor XIIa (autoactivation) [3].

Prekallikrein (PK) is synthesized in the liver and circulates in plasma. It is activated via limited proteolysis by Factor XIIa and results in the formation of kallikrein. HK is a protein in the plasma and most likely synthesized in the liver. It is required for efficient formation of kallikrein. Kallikrein accelerates the activation of Factor XII [3].

Factor XIIa activates Factor XI and Factor XIa activates Factor IX after which the same process takes place as for the extrinsic pathway. Shortly summarized, this means that consecutively Factor X, prothrombin, Factor V and Factor VIII are activated, and this leads to the convertion of fibrinogen into fibrin which results in a fibrin clot [3].

A visual representation of this process is given in Figure 1b.

1.3 Previous Mathematical Models

People have created different mathematical models of the blood coagulation system. These models aim to help us understand the process in a better, deeper way. They can help us to understand the effects of different medical defects and we can calculate the time that it takes to form a fibrin clot to stop the bleeding. Besides that, they can also help us gain more insight into the role of different factors as thrombin and flow in the coagulation process. We will discuss some of these models here.

In the model presented by Zarnitsina, Pokhilko and Atauliakhanov [4], we consider the factors IXa, Xa, IIa (thrombin), II (prothrombin), VIIIa, Va, APC (activated protein C) and Ia (fibrin). This model is given by a set of 8 differential equations that describe the blood coagulation process. Since they take the prothrombin levels into account, the generation of thrombin will not be constant, but decrease after it has reached a peak, as can be read in [5].

Another mathematical model is introduced by Donghui Zhu. In this model [6], the intrinsic pathway and the extrinsic pathway are modelled separately. The model describes both pathways with respectively 32 and 27 equations, which describe the model as given in Figure 2.

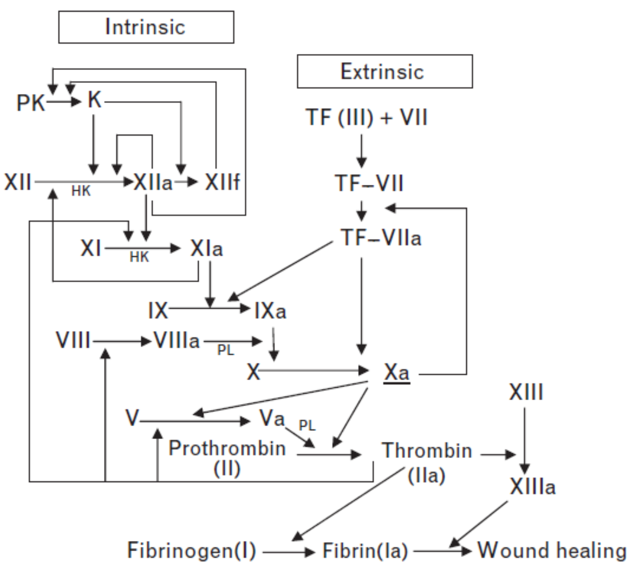


Figure 2: Visual representation of the intrinsic and extrinsic pathway described in the model presented by Zhu [6].

In this report, we will recreate and discuss a model that was proposed by Galochkina, Bouchnita, Kurbatova and Volpert [1].

2 Mathematical Model

We consider the mathematical model introduced by Galochkina et al. [1] to describe the blood coagulation process. That means that we use the following system of equations:

$$\frac{\partial T}{\partial t} = D\Delta T + \left(k_2 U_{10} + \overline{k_2}\frac{k_{510}}{h_{510}} U_{10} U_5\right)\left(1 - \frac{T}{T_0}\right) - h_2 T, \tag{1a}$$

$$\frac{\partial U_5}{\partial t} = D\Delta U_5 + k_5 T - h_5 U_5, \tag{1b}$$

$$\frac{\partial U_8}{\partial t} = D\Delta U_8 + k_8 T - h_8 U_8, \tag{1c}$$

$$\frac{\partial U_9}{\partial t} = D\Delta U_9 + k_9 U_{11} - h_9 U_9, \tag{1d}$$

$$\frac{\partial U_{10}}{\partial t} = D\Delta U_{10} + k_{10} U_9 + \overline{k_{10}}\frac{k_{89}}{h_{89}} U_9 U_8 - h_{10} U_{10}, \tag{1e}$$

$$\frac{\partial U_{11}}{\partial t} = D\Delta U_{11} + k_{11} T - h_{11} U_{11}. \tag{1f}$$

Here, T and U_i denote the concentrations of thrombin and the ith factors in activated form respectively. Furthermore, T_0 denotes the initial prothrombin concentration. The constants k_i and h_i respectively represent the activation rate and the deactivation rate of factor i. The first term represents the diffusion of the different factors in the plasma, the rest of the factors describe the chemical reactions that take place [1]. We assume that all factors diffuse in the same way in the plasma. Therefore, we use the same diffusion constant D for all equations. This system has boundary conditions given by

$$\left.\frac{\partial U_{11}}{\partial x}\right|_{x=0} = -A, \left.\frac{\partial U_{11}}{\partial x}\right|_{x=L} = 0, \tag{2a}$$

$$\left.\frac{\partial F}{\partial x}\right|_{x=0} = 0, \left.\frac{\partial F}{\partial x}\right|_{x=L} = 0, \qquad F = T, U_5, U_8, U_9, U_{10}, \tag{2b}$$

where $x = 0$ is at the blood vessel where the rupture took place and $x = L$ is at some distance L from the vessel wall. Here, A denotes the activation level at the point $x = 0$ [4]. We will assume that the value of A is the same as T_0. This model is derived from a previously published model by Zarnitsina, Pokhilko and Ataullakhanov [4]. The model of Galochkina et al. is more simplified and considers fewer different variables [1].

We note that with the boundary conditions given in Eq. (2) are all Neumann Boundary conditions. By consequence, the solution could differ up to a constant. The system of equations given in Eq. (1) makes sure that the system converges to one solution as we will see in Section 6.

3 Existence, Uniqueness and Stability

We will consider the existence and uniqueness of the model given in Eq. (1). In order to do that, we will rewrite the model to the more general form

$$\frac{\partial u}{\partial t} = D\Delta u + F(u), \tag{3}$$

where $u = (T, U_5, U_8, U_9, U_{10}, U_{11})$ and $F(u)$ is a vector containing all reaction rates of Eq. (1). In addition to that, it satisfies

$$\frac{\partial F_i}{\partial u_j} \geq 0, \qquad \forall i \neq j.$$

By definition this system is monotone.

Definition 3.1 (Monotone Systems [7])**.** Consider the equation

$$\frac{\partial \underline{u}}{\partial t} = a\frac{\partial^2 \underline{u}}{\partial t^2} + \underline{F}(\underline{u}),$$

where $\underline{u} = (u_1, \ldots, u_n)$ and $\underline{F} = (F_1, \ldots, F_n)$.

If $a \in \mathbb{R}^{n\times n}$ is a diagonal matrix and $\underline{F}\left(\underline{u}\right)$ satisfies

$$\frac{\partial F_i}{\partial u_j} \geq 0, \quad i, j = 1, \ldots, n, \;\; i \neq j,$$

then the system is called monotone.

Monotone systems form an important class of differential equations. They occur amongst others in biological systems [8]. They have several properties that are very similar to those for one scalar equations including the maximum principle [7]. This makes it possible to prove the existence and stability of the wave solutions for the monotone systems and to make the estimations of the wave propagation speed. First, we need to analyse the existence and stability of the stationary points of the system.

3.1 Stationary Points

Consider the ODE (ordinary differential equation)

$$\frac{\mathrm{d}u}{\mathrm{d}t} = F(u). \tag{4}$$

We start determining the values of the stationary points of the system (4). We found some deviations from the expressions given in the paper by Galochkina et al. [1]. The values that we added are written in orange. That means that we can neglect the temporal derivatives in the system. One should find that

$$0 = \left(k_2 U_{10} + \overline{k_2}\frac{k_{510}}{h_{510}}U_{10}U_5\right)\left(1 - \frac{T}{T_0}\right) - h_2 T, \tag{5a}$$

$$0 = k_5 T - h_5 U_5, \tag{5b}$$

$$0 = k_8 T - h_8 U_8, \tag{5c}$$

$$0 = k_9 U_{11} - h_9 U_9, \tag{5d}$$

$$0 = k_{10} U_9 + \overline{k_{10}}\frac{k_{89}}{h_{89}}U_9 U_8 - h_{10} U_{10}, \tag{5e}$$

$$0 = k_{11} T - h_{11} U_{11}. \tag{5f}$$

This can be simplified to the stationary points

$$U_5 = \frac{k_5}{h_5}T, \quad U_8 = \frac{k_8}{h_8}T, \quad U_9 = \frac{k_9}{h_9}U_{11}, \quad U_{10} = \frac{k_9 k_{11}}{h_9 h_{10} h_{11}}T\left(k_{10} + \overline{k_{10}}\frac{k_{89}}{h_{89}}\frac{k_8}{h_8}T\right), \quad U_{11} = \frac{k_{11}}{h_{11}}T, \tag{6}$$

where T is the solution of (5a). Note that all variables U_i with $i = 5, 8, 9, 10, 11$ depend on T. Hence T is the solution of $P(T) = 0$ where $P(T) = aT^4 + bT^3 + cT^2 + dT$,

$$a = \frac{\overline{k_2}k_5k_{510}k_8k_{89}k_9\overline{k_{10}}k_{11}}{h_5h_{510}h_8h_{89}h_9h_{10}h_{11}}, \quad b = \frac{k_2k_8k_{89}k_9\overline{k_{10}}k_{11}}{h_8h_{89}h_9h_{10}h_{11}} + \frac{\overline{k_2}k_5k_{510}k_9k_{10}k_{11}}{h_5h_{510}h_9h_{10}h_{11}} - \frac{\overline{k_2}k_5k_{510}k_8k_{89}k_9\overline{k_{10}}k_{11}}{h_5h_{510}h_8h_{89}h_9h_{10}h_{11}}T_0$$

$$c = \frac{k_2k_9k_{10}k_{11}}{h_9h_{10}h_{11}} - \frac{k_2k_8k_{89}k_9\overline{k_{10}}k_{11}}{h_8h_{89}h_9h_{10}h_{11}}T_0 - \frac{\overline{k_2}k_5k_{510}k_9k_{10}k_{11}}{h_5h_{510}h_9h_{10}h_{11}}T_0, \quad d = h_2T_0 - \frac{k_2k_9k_{10}k_{11}}{h_9h_{10}h_{11}}T_0.$$

That means that the stationary points of Eq. (1) are the same as the stationary points given by

$$\frac{dT}{dt} = -P(T). \tag{7}$$

and substituting the found value into (6).

Since we know that the concentration of thrombin has to be positive, we determine the number of positive roots of $P(T)$. First, we note that $T = 0$ is a solution of $P(T) = 0$. Therefore, we rewrite $P(T) = TQ(T)$, where $Q(T) = aT^3 + bT^2 + cT + d$. In order to determine the number of positive roots, we take a closer look at the first derivative of Q, which yields $Q'(T) = 3aT^2 + 2bT + c$. We note that if $Q'(T)$ has no zeros, the function $Q(T)$ is increasing since $a > 0$. In addition to that, we have that $\lim_{T\to\infty} = \infty$. The system has:

- one positive root if $Q'(T)$ has no zeros and $Q(0) < 0$ (Figure 3a);
- one positive root if $Q'(T)$ has two zeros, denoted by $T_{1,2} = \frac{-2b\pm\sqrt{4b^2-12ac}}{6a} = \frac{-b\pm\sqrt{b^2-2ac}}{3a}$ and
  - $T_1 \leq 0$, $Q(0) < 0$ (Figure 3b,3c);
  - $0 \leq T_1 < T_2$, $Q(0) < 0$ and $Q(T_1) > 0, Q(T_2) > 0$ or $Q(T_1) < 0$ (Figure 3d,3e);
  - $Q(0) > 0, Q(T_1) > 0, Q(T_2) = 0$ (Figure 3f,3g);
- Two positive roots if $0 < T_2, Q(0) > 0, Q(T_2) < 0$ or $Q(0) < 0, Q(T_2) = 0$ (Figure 3h,3i,3j).

Theorem 3.1 (Relation between stability of Eq. (1) and Eq. (7) [1])**.** *There is one to one correspondence between stationary solutions $u^\star = (T^\star, U_5^\star, U_8^\star, U_9^\star, U_{10}^\star, U_{11}^\star)$ of system* (1) *and the stationary points $T^\star$ of Eq.* (7) *given by* (6)*. The principal eigenvalue of the matrix $F'(u^\star)$ is positive (negative) if and only if $P'(T^\star) < 0$ ($P'(T^\star) > 0$).*

Proof. We consider the following two systems:

$$\frac{\mathrm{d}u}{\mathrm{d}t} = F(u) \tag{8}$$

$$\frac{\mathrm{d}u}{\mathrm{d}t} = F_\tau(u) \tag{9}$$

where $\tau \in [0,1]$. The only difference between $F(u)$ and $F_\tau(u)$ is the equation for T, which is given by

$$\frac{\mathrm{d}T}{\mathrm{d}t} = (\tau U_{10} + (1-\tau)\,\varphi_{10}(T))\left(k_2 + \overline{k_2}\frac{k_{510}}{h_{510}}(\tau U_5 + (1-\tau)\,\varphi_5(T))\right)\left(1 - \frac{T}{T_0}\right) - h_2 T,$$

in $F_\tau(u)$ where functions $\varphi_i(T)$ are determined by the equalities

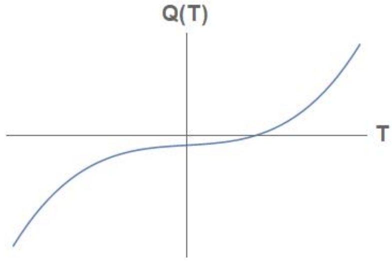
$$\varphi_{11}(T) = \frac{k_{11}}{h_{11}}T, \quad \varphi_9(T)\frac{k_9 k_{11}}{h_9 h_{11}}T, \quad \varphi_5(T) = \frac{k_5}{h_5}T, \quad \varphi_8(T) = \frac{k_8}{h_8}T, \quad \varphi_{10}(T) = \frac{k_9 k_{11}}{h_9 h_{11}}T\left(k_{10} + \overline{k_{10}}\frac{k_{89}}{h_{89}}\frac{k_8}{h_8}T\right).$$

We can express U_i, $i = 5, 8, 9, 10, 11$, as functions of T for both Eq. (8) and (9). More specifically, we find $U_i = \varphi_i(T)$. Therefore, we also find that the solutions of equations $F_\tau(T) = 0$ if and only if $F(T) = 0$. One can check this very easily by checking that

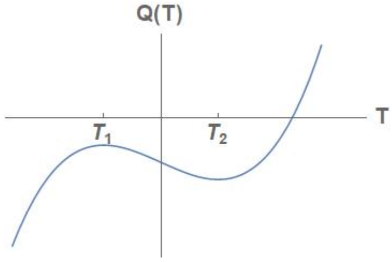
$$\begin{aligned}\frac{\mathrm{d}T}{\mathrm{d}t} &= (\tau U_{10} + (1-\tau)\,\varphi_{10}(T))\left(k_2 + \overline{k_2}\frac{k_{510}}{h_{510}}(\tau U_5 + (1-\tau)\,\varphi_5(T))\right)\left(1 - \frac{T}{T_0}\right) - h_2 T \\ &= (\tau U_{10} + (1-\tau)\,U_{10})\left(k_2 + \overline{k_2}\frac{k_{510}}{h_{510}}(\tau U_5 + (1-\tau)\,U_5)\right)\left(1 - \frac{T}{T_0}\right) - h_2 T = -P(T).\end{aligned}$$

Therefore, we can conclude that system (8) and system (9) have the same stationary solutions for all $\tau \in [0,1]$. Note that for $\tau = 1$, the two systems are exactly the same. For $\tau = 0$, the equation for T in system (9) does not depend on variables different from T. This allows us to determine the eigenvalues from a linearized Jacobian matrix.

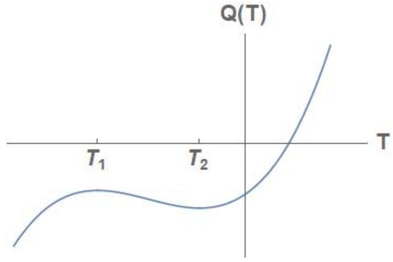
We verify that $\det DF_\tau(u^\star) = 0$ if and only if $\det DF(u^\star) = 0$ for all $\tau \in [0,1]$. To do this, we first determine



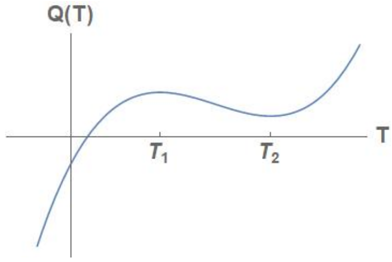
(a) $Q'(T)$ has no roots.



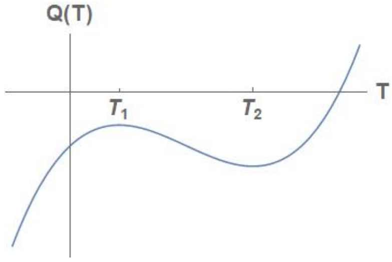
(b) $T_1 \leq 0, Q(0) < 0$



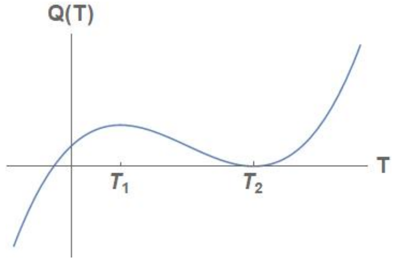
(c) $T_1 \leq 0, Q(0) < 0$



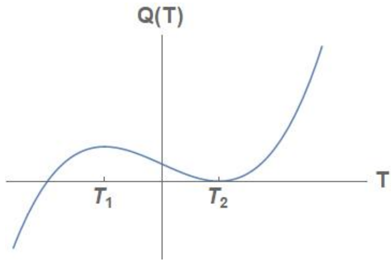
(d) $0 \leq T_1 < T_2, Q(0) < 0, Q(T_1) > 0, Q(T_2) > 0$



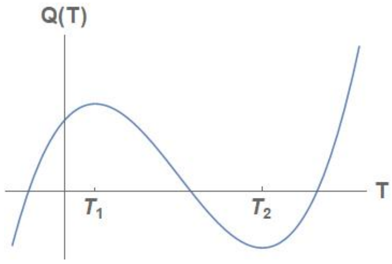
(e) $0 \leq T_1 < T_2, Q(0) < 0, Q(T_1) > 0, Q(T_1) < 0$



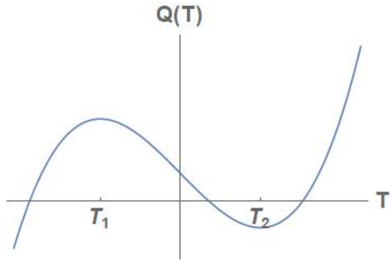
(f) $Q(0) > 0, Q(T_1) > 0, Q(T_2) = 0$



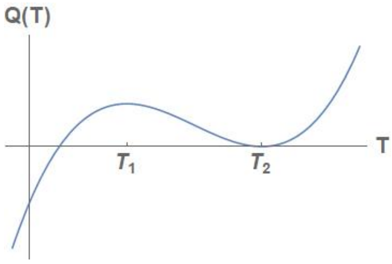
(g) $Q(0) > 0, Q(T_1) > 0, Q(T_2) = 0$



(h) $0 < T_2, Q(0) > 0, Q(T_2) < 0$



(i) $0 < T_2, Q(0) > 0, Q(T_2) < 0$



(j) $Q(0) < 0, Q(T_2) = 0$

Figure 3: Visualization of the number of positive roots depending on the criteria.

the Jacobians of both systems:

$$D\tilde{F}(u) = \begin{array}{c} \\ T \\ U_5 \\ U_8 \\ U_9 \\ U_{10} \\ U_{11} \end{array} \!\!\begin{array}{c} \begin{array}{cccccc} T & U_5 & U_8 & U_9 & U_{10} & U_{11} \end{array} \\ \begin{pmatrix} \tilde{a} & \tilde{b} & 0 & 0 & \tilde{c} & 0 \\ k_5 & -h_5 & 0 & 0 & 0 & 0 \\ k_8 & 0 & -h_8 & 0 & 0 & 0 \\ 0 & 0 & 0 & -h_9 & 0 & k_9 \\ 0 & 0 & d & e & -h_{10} & 0 \\ k_{11} & 0 & 0 & 0 & 0 & -h_{11} \end{pmatrix} \end{array}, \tag{10}$$

where $d = \overline{k_{10}} \frac{k_{89}}{h_{89}} U_9$, $e = k_{10} + \overline{k_{10}} \frac{k_{89}}{h_{89}} \frac{k_8}{h_8} U_8$, and $\tilde{a}$, $\tilde{b}$, $\tilde{c}$ depend on the system that we are considering. For $\tilde{F}(u) = F(u)$, they are denoted by a, b and c respectively, and for $\tilde{F}(u) = F_\tau(u)$ by a_τ, b_τ and c_τ. There values are given by

$$\begin{aligned}
a &= \left(k_2 U_{10} + \overline{k_2} \frac{k_{510}}{h_{510}} U_{10} U_5\right) \left(-\frac{1}{T_0}\right) - h_2; \\
b &= \overline{k_2} \frac{k_{510}}{h_{510}} U_{10} \left(1 - \frac{T}{T_0}\right); \\
c &= \left(k_2 + \overline{k_2} \frac{k_{510}}{h_{510}} U_5\right) \left(1 - \frac{T}{T_0}\right); \\
a_\tau &= (1-\tau)\, \varphi_{10}'(T) \left(k_2 + \overline{k_2} \frac{k_{510}}{h_{510}} \left(\tau U_5 + (1-\tau)\, \varphi_5(T)\right)\right) \left(1 - \frac{T}{T_0}\right) \\
&\quad + \left(\tau U_{10} + (1-\tau)\, \varphi_{10}(T)\right) \left(\overline{k_2} \frac{k_{510}}{h_{510}} (1-\tau)\, \varphi_5'(T)\right) \left(1 - \frac{T}{T_0}\right) \\
&\quad - \frac{1}{T_0} \left(\tau U_{10} + (1-\tau)\, \varphi_{10}(T)\right) \left(k_2 + \overline{k_2} \frac{k_{510}}{h_{510}} \left(\tau U_5 + (1-\tau)\, \varphi_5(T)\right)\right) - h_2; \\
b_\tau &= \tau \overline{k_2} \frac{k_{510}}{h_{510}} U_{10} \left(1 - \frac{T}{T_0}\right); \\
c_\tau &= \tau \left(k_2 + \overline{k_2} \frac{k_{510}}{h_{510}} U_5\right) \left(1 - \frac{T}{T_0}\right).
\end{aligned}$$

The determinant is then given by

$$\det DF(u) = -\left(h_8 h_9 h_{10} h_{11} \left(h_5 a - k_5 b\right) - h_5 c \left(d k_8 h_9 h_{11} + e h_8 k_9 k_{11}\right)\right).$$

We note that for both systems, the only variables that change will be a, b and c. Therefore, we calculate the relation between them for both systems in the stationary points $u^\star$. We find

$$\begin{aligned}
b_\tau^\star &= \tau b^\star, \\
c_\tau^\star &= \tau c^\star, \\
a_\tau^\star &= \tau a^\star + (1-\tau) \left(U_{10}^\star \frac{U_5^\star}{T_0} \overline{k_2} \frac{k_{510}}{h_{510}} \left(1 - \frac{T^\star}{T_0}\right) - h_2\right) - \tau U_{10}^\star \frac{U_5^\star}{T_0} \overline{k_2} \frac{k_{510}}{h_{510}} \frac{T^\star}{T_0} \\
&= (1-\tau) \left(\varphi_{10}'(T^\star)\, c^\star + \varphi_5'(T^\star)\, b^\star + a^\star\right) + \tau a^\star \\
&= (1-\tau) \left(\frac{k_9 k_{11}}{h_9 h_{10} h_{11}} \left(k_{10} + 2\overline{k_{10}} \frac{k_{89}}{h_{89}} \frac{k_8}{h_8} T^\star\right) c^\star + \frac{k_5}{h_5} b^\star + a^\star\right) + \tau a^\star.
\end{aligned}$$

We will show that the determinant of $DF(u^\star)$ only is zero if and only if $DF_\tau(u^\star)$ is zero. First, we note that the determinant of $DF(u^\star)$ is given by

$$\det DF(u^\star) = -\mu \left(h_5 a^\star - k_5 b^\star\right) - \sigma^\star c^\star,$$

where $\mu := h_8 h_9 h_{10} h_{11}$ and $\sigma^\star := h_5 \left(d^\star h_9 h_{11}\right) = h_5 k_9 k_{11} \left(h_8 k_{10} + 2 k_8 \overline{k_{10}} \frac{k_{89}}{h_{89}} T^\star\right)$. Then, we consider the determinant of $DF_\tau(u^\star)$ which is given by

$$\begin{aligned}
\det DF_\tau(u^\star) &= -\mu\left(h_5 a_\tau^\star - k_5 b_\tau^\star\right) - c_\tau^\star \sigma^\star \\
&= \tau \det DF(u^\star) - (1-\tau)\,\mu\left(h_5 a^\star + k_5 b^\star + h_5 c^\star\left(\frac{k_9 k_{11}}{h_9 h_{10} h_{11}}\left(k_{10} + 2\overline{k_{10}}\frac{k_{89}}{h_{89}}\frac{k_8}{h_8}\right)\right)\right) \\
&= \tau \det DF(u^\star) - (1-\tau)\left(\mu\left(h_5 a^\star - k_5 b^\star\right) + c^\star h_5 k_9 k_{11}\left(h_8 k_{10} + 2 k_8 \overline{k_{10}}\frac{k_{89}}{h_{89}} T^\star\right)\right) \\
&= \tau \det DF(u^\star) - (1-\tau)\left(\mu\left(h_5 a^\star - k_5 b^\star\right) + \sigma^\star c^\star\right) \\
&= \det DF(u^\star).
\end{aligned}$$

This also implies that $\det DF_\tau(u^\star) = 0$ if and only if $\det DF(u^\star) = 0$.

Suppose that the determinant of $DF(u^\star) \neq 0$. Then, the principal eigenvalue of the matrix $DF_\tau(u^\star)$ cannot change sign when τ changes from 0 to 1 since the determinant is continuous. Hence, the signs of the principal eigenvalues of $DF(u^\star)$ and $DF_\tau(u^\star)$ are the same.

When we calculate the Jacobian of the system, we find

$$DF_{\tau=0}(u) = \begin{array}{c} \\ T \\ U_5 \\ U_8 \\ U_9 \\ U_{10} \\ U_{11} \end{array}
\begin{array}{c}
\begin{array}{cccccc} T & U_5 & U_8 & U_9 & U_{10} & U_{11} \end{array} \\
\begin{pmatrix}
-P'(T) & 0 & 0 & 0 & 0 & 0 \\
k_5 & -h_5 & 0 & 0 & 0 & 0 \\
k_8 & 0 & -h_8 & 0 & 0 & 0 \\
0 & 0 & 0 & -h_9 & 0 & k_9 \\
0 & 0 & \overline{k_{10}}\frac{k_{89}}{h_{89}} U_9 & k_{10} + \overline{k_{10}}\frac{k_{89}}{h_{89}} U_8 & -h_{10} & 0 \\
k_{11} & 0 & 0 & 0 & 0 & -h_{11}
\end{pmatrix}
\end{array}.$$

Calculation of the eigenvalues (in the stationary points) yields

$$\lambda_1 = -P'(T), \quad \lambda_2 = -h_5, \quad \lambda_3 = -h_8, \quad \lambda_4 = -h_9, \quad \lambda_5 = -h_{10}, \quad \lambda_6 = -h_{11}.$$

Hence, the principal eigenvalue of the system is positive if $P'(T) < 0$ and negative if $P'(T) > 0$. □

This means that we can make the following conclusions:

- The trivial solution $u^\star = \underline{0}$ is always a stationary point of the system.
- The system has one (two) positive solution(s) if and only if the polynomial $P(T)$ has one (two) positive root(s).
- A positive solution $u^\star$ is stable if and only if $P'(T^\star) > 0$.

3.2 Wave Existence and Stability

Theorem 3.2 (Wave Existence and Stability [1])**.** *Suppose that $P(T^\star) = 0$ for some $T^\star > 0$ and $P'(0) \neq 0$, $P'(T^\star) \neq 0$. Let $u^\star = (T^\star, U_5^\star, U_8^\star, U_9^\star, U_{10}^\star, U_{11}^\star)$ be the corresponding stationary solutions of system* (3) *determined by* (6)*.*

- *Monostable case. If there are no other positive roots of the polynomial $P(T)$, then system* (1) *has monotonically decreasing traveling wave solutions $u(x,t) = w(x-ct)$ with the limits $u(+\infty) = 0, u(-\infty) = u^\star$ for all values of the speed c greater than or equal to the minimal speed c_0.*
- *Bistable case. If there is one more positive root of the polynomial $P(T)$ in the interval $0 < T < T^\star$, then system* (1) *has a monotonically decreasing traveling wave solution $u(x,t) = w(x-ct)$ with the limits $u(+\infty) = 0, u(-\infty) = u^\star$ for a unique value of c.*

Proof. First, we will consider the monostable case. In that case $P(T)$ has no other positive roots besides $T^\star$. That means that $u^\star$ is the only positive stationary solution of (1). Since $P(T)$ has no other positive roots, we know that $P(T)$ does not change sign on the interval $0 < T < T^\star$. By consequence, that also means that the function $F(u)$ does not change sign on $0 < u < u^\star$. We are interested in the sign of $F(u)$ on the interval $0 < u < u^\star$. Therefore, it is relevant to know what sign $P(T)$ has on this interval. As we already mentioned, we know that $P(T)$ does not change sign on this interval. In addition to that, we know that $P(T)$ does change sign when $T > T^\star$. The easiest way to determine the sign is by looking at the sign of the function $P(T)$ in the limit to ∞ which is given by

$$P(T) \overset{T\to\infty}{\to} \infty, \text{ so } -P(T) \overset{T\to\infty}{\to} -\infty.$$

Since Eq. (1a) is given by Eq. (7), we find that $F(u) > 0$ for $0 < u < u^\star$. By the application of Theorem B.1, we find the desired result.

Next, we prove the statement about the bistable case. In this case $P(T)$ does have another positive root, denoted by $\tilde{T}$, besides $T^\star$. We assume that $\tilde{T} < T^\star$. This means that the system (1) also has two stationary solutions, $u^\star$ and $\tilde{u}$ where the first value of $u^\star$ is the same as $T^\star$ and the first value of $\tilde{u}$ is the same as $\tilde{T}$. Since $u^\star$ and $\tilde{u}$ are stationary solutions, we find $F(u^\star) = 0 = F(\tilde{u})$, and since 0 is also a stationary solution, we also know that $F(0) = 0$ with $0 < \tilde{u} < u^\star$. When we consider the eigenvalues of the system, we find that the eigenvalues of the system are given by

$$\lambda_1 = -P'(T), \quad \lambda_2 = -h_5, \quad \lambda_3 = -h_8, \quad \lambda_4 = -h_9, \quad \lambda_5 = -h_{10}, \quad \lambda_6 = -h_{11}.$$

That means that either all eigenvalues are negative, or all but one. When we take a look at the value of λ_1 for the different stationary points that we are considering, we find $P'(0) > 0$ and $P'(u^\star) > 0$, but $P'(\tilde{u}) < 0$. When we apply Theorem B.2, we find the desired result. $\square$

The traveling wave solution of Eq. (3) describes the thrombin concentration in the blood plasma. This concentration is under influence of the reactions that take place in the coagulation process. In the system, the convergence only takes place if the initial concentrations of the blood factors exceed some critical value. When that is not the case, the blood clot formation does not start (because of the deactivation factors). This dependency on the initial conditions and the stability of the zero-solution correspond with the bistable case. In the monostable case, a small perturbation would lead to convergence of the solution towards the nonzero solution. This would not be a good thing, because that means that a small change in the levels of the different factors would immediately lead to convergence towards the nonzero solution, i.e. blood clotting.

In Theorem 3.2 we only consider one or two positive roots. In case that $P(T)$ has three positive roots, one will find that the system will be monostable, with a stable intermediate stationary point. This would be interesting to study in more detail, but it is less relevant to our application, and therefore, we will not discuss it here [1].

4 Speed of Wave Propagation

Our goal is to find an analytical approximation of the wave speed for the blood coagulation model. In order to do so, we will reduce the system to a one equation model in Section 5. Therefore, we first take a look at what this means for the propagation speed of the wave.

Let us consider the following model:

$$u'' + c_\varepsilon u' + f(u,v) = 0, \tag{11a}$$

$$v'' + c_\varepsilon v' + \frac{1}{\varepsilon}(au - bv) = 0, \tag{11b}$$

where $\varepsilon > 0$ is a very small parameter, $\frac{\partial f}{\partial v} > 0$ and the system is bistable, $f \in C^2(\mathbb{R})$. When we multiply Eq. (11b) with ε, and then take the formal limit to 0, we find:

$$\lim_{\varepsilon \downarrow 0} \varepsilon v'' + \varepsilon c_\varepsilon v' + (au - bv) = 0,$$

$$\Leftrightarrow \quad au - bv = 0 \quad \Leftrightarrow \quad v = \frac{a}{b}u.$$

Now, we can rewrite (11a) to

$$u'' + c_0 u' + f\left(u, \frac{a}{b}u\right). \tag{12}$$

Note that c_ε and c_0 are both unknown, and in general, they do not have the same value. However, we will show that $c_\varepsilon \to c_0$ when $\varepsilon \downarrow 0$.

Theorem 4.1 (Converging Speeds of Wave Propagation [1])**.** *The speed of wave propagation for system* (11a)-(11b) *converges to the speed of the wave propagation for Eq.* (12) *as* $\varepsilon \to 0$.

Proof. First, we recall the minimax representation of the wave speed in the bistable case.

Theorem 4.2 (Minimax Representation of the Wave Speed [7])**.** *Consider the system*

$$w'' + cw' + F(w) = 0.$$

Let K *be a class of continuous functions with their second derivatives monotonically decreasing and having limits*

$$\lim_{x \to \pm\infty} w(x) = w_\pm, \qquad -\infty < w_+ < w_- < \infty.$$

Then

$$\inf_{x \in \mathbb{R}} \frac{\rho'' + F(\rho)}{-\rho'} \leq c \leq \sup_{x \in \mathbb{R}} \frac{\rho'' + F(\rho)}{-\rho'} \qquad \forall \rho \in K. \tag{13}$$

The minimax representation is then given by

$$c = \inf_{\rho \in K} \sup_{x \in \mathbb{R}} \frac{\rho'' + F(\rho)}{-\rho'} = \sup_{\rho \in K} \inf_{x \in \mathbb{R}} \frac{\rho'' + F(\rho)}{-\rho'}.$$

When we apply this theorem to our system of equations, we find

$$\max\left\{\inf_x S_1(\rho), \inf_x S_2(\rho)\right\} \leq c \leq \min\left\{\sup_x S_1(\rho), \sup_x S_2(\rho)\right\}, \tag{14}$$

where

$$S_1(\rho) = \frac{\rho_1'' + f(\rho_1, \rho_2)}{-\rho_1'}, S_2 = \frac{\rho_2'' + (a\rho_1 - b\rho_2)/\varepsilon}{-\rho_2'},$$

with $\rho = (\rho_1, \rho_2)$ arbitrary test functions in K (as defined in Theorem 4.2), and

$$\rho(+\infty) = 0, \rho(-\infty) = u^\star.$$

We choose the following test functions:

$$\rho_1 = u_0, \quad \rho_2 = \frac{a}{b}u_0 - \varepsilon f\left(u_0, \frac{a}{b}u_0\right)\frac{a}{b^2},$$

where u_0 is the solution of Eq. (12). Then, S_1 can be reduced to:

$$\begin{aligned}
S_1(\rho) &= \frac{\rho_1'' + f\left(\rho_1, \rho_2\right)}{-\rho_1'} \\
&= \frac{u_0'' + f\left(u_0, \frac{a}{b}u_0 - \varepsilon f\left(u_0, \frac{a}{b}u_0\right)\frac{a}{b^2}\right)}{-u_0'} \\
&\doteq \frac{u_0'' + f\left(u_0, \frac{a}{b}u_0\right) - \varepsilon \frac{a}{b^2} f\left(u_0, \frac{a}{b}u_0\right) f_v\left(u_0, \frac{a}{b}\right)}{-u_0'} \\
&= c_0 - \varepsilon \frac{\frac{a}{b^2} f\left(u_0, \frac{a}{b}u_0\right) f_v\left(u_0, \frac{a}{b}\right)}{-u_0'} =: c_0 + \varepsilon\varphi(x),
\end{aligned}$$

where c_0 is the propagation speed as in Eq. (12) and $\varphi(x) = \varepsilon \frac{\frac{a}{b^2} f\left(u_0, \frac{a}{b}u_0\right) f_v\left(u_0, \frac{a}{b}\right)}{-u_0'}$. A similar reduction can be done for S_2:

$$\begin{aligned}
S_2(\rho) &= \frac{\rho_2'' + \left(a\rho_1 - b\rho_2\right)/\varepsilon}{-\rho_2'} \\
&= \frac{\left(\frac{a}{b}u_0 - \varepsilon f\left(u_0, \frac{a}{b}u_0\right)\frac{a}{b^2}\right)'' + \left(au_0 - b\left(\frac{a}{b}u_0 - \varepsilon f\left(u_0, \frac{a}{b}u_0\right)\frac{a}{b^2}\right)\right)/\varepsilon}{-\left(\frac{a}{b}u_0 - \varepsilon f\left(u_0, \frac{a}{b}u_0\right)\frac{a}{b^2}\right)'} \\
&= \frac{u_0'' - \frac{\varepsilon}{b} f''\left(u_0, \frac{a}{b}u_0\right) - f\left(u_0, \frac{a}{b}u_0\right)}{-u_0' + \varepsilon f'\left(u_0, \frac{a}{b}u_0\right)/b} \\
&= \frac{u_0'' - \frac{\varepsilon}{b} f''\left(u_0, \frac{a}{b}u_0\right) - f\left(u_0, \frac{a}{b}u_0\right)}{-u_0'} \left(1 + \frac{\varepsilon}{b}\frac{\left(f\left(u_0, \frac{a}{b}u_0\right)\right)'}{u_0'} + \mathcal{O}\left(\varepsilon^2\right)\right) \\
&\doteq \frac{u_0'' - f\left(u_0, \frac{a}{b}u_0\right)}{-u_0'} + \varepsilon\left(\frac{f''\left(u_0, \frac{a}{b}u_0\right)}{bu_0'} + \frac{\left(u_0'' - f\left(u_0, \frac{a}{b}u_0\right)\right)\left(f\left(u_0, \frac{a}{b}u_0\right)\right)'}{-b\left(u_0'\right)^2}\right) =: c_0 + \varepsilon\psi(x)
\end{aligned}$$

where $\psi(x) := \frac{f''\left(u_0, \frac{a}{b}u_0\right)}{bu_0'} + \frac{c_0}{bu_0'}\left(f\left(u_0, \frac{a}{b}u_0\right)\right)'$. This means that we can rewrite Eq. (14) to

$$c_0 + \varepsilon \max\left\{\inf_x \varphi(x), \inf_x \psi(x)\right\} \leq c \leq c_0 + \varepsilon \min\left\{\sup_x \varphi(x), \sup_x \psi(x)\right\},$$

in which c_0 denotes the propagation speed of Eq. (12). Taking the limit of ε to 0 yields $c = c_0$ since $\varphi(x)$ and $\psi(x)$ are bounded. $\square$

5 One Equation Model

5.1 Reduction to the Equation on Thrombin Concentration

If the reaction rate constants in the equations describing U_5, U_8, U_9 and U_{10} are sufficiently large, we are able to replace these variables by their stationary points. This can be considered to be a similar approach as in Section 4. Hence, it yields

$$U_5 = \frac{k_5}{h_5}T, \quad U_8 = \frac{k_8}{h_8}T, \quad U_9 = \frac{k_9}{h_9}U_{11}, \quad U_{10} = U_{11}\frac{k_9}{h_9 h_{10}}\left(k_{10} + \frac{\overline{k_{10}}k_{89}}{h_{89}}\frac{k_8}{h_8}T\right).$$

Then system (1) simplifies to

$$\frac{\partial T}{\partial t} = D\Delta T + U_{11}\frac{k_9}{h_9 h_{10}}\left(k_{10} + \frac{\overline{k_{10}}k_{89}}{h_{89}}\frac{k_8}{h_8}T\right)\left(k_2 + \frac{\overline{k_2}k_{510}}{h_{510}}\frac{k_5}{h_5}T\right)\left(1 - \frac{T}{T_0}\right) - h_2 T, \tag{15a}$$

$$\frac{\partial U_{11}}{\partial t} = D\Delta U_{11} + k_{11}T - h_{11}U_{11}, \tag{15b}$$

as suggested by Galochkina et al. [1]. With a similar reasoning, we are able to reduce the two equation system (15) to a one equation model by expressing U_{11} in function of T. This means that we will replace U_{11} by $\frac{k_{11}}{h_{11}}T$ in the first equation, which yields [1]

$$\frac{\partial T}{\partial t} = D\Delta T + \frac{k_9 k_{11}}{h_9 h_{10} h_{11}}T\left(k_{10} + \frac{\overline{k_{10}}k_{89}}{h_{89}}\frac{k_8}{h_8}T\right)\left(k_2 + \frac{\overline{k_2}k_{510}}{h_{510}}\frac{k_5}{h_5}T\right)\left(1 - \frac{T}{T_0}\right) - h_2 T. \tag{16}$$

We did this reduction in two steps such that we are able to consider both models in the simulations. The two equation model gives a better approximation of Eq. (1) than Eq. (16) since fewer assumptions have been made and fewer small deviations were ignored. One should be able to deduce the theoretical wave speed of Eq. (16).

5.2 Dimensionless Model

Galochkina et al. [1] also suggested to rewrite the system to a dimensionless form. In order to do so, we introduce the following dimensionless variables u, $\tilde{t}$ and $\tilde{D}$ defined as

$$T = T_0 u, \quad t = \frac{\tilde{t}}{h_2}, \quad D = \breve{D}h_2. \tag{17}$$

This means that we can rewrite Eq. (16) to

$$\begin{aligned}
& T_0\frac{\partial u}{\partial \tilde{t}}\frac{\mathrm{d}\tilde{t}}{\mathrm{d}t} = \tilde{D}h_2 T_0\Delta u + \frac{k_9 k_{11}}{h_9 h_{10} h_{11}}T_0 u\left(k_{10} + \frac{\overline{k_{10}}k_{89}}{h_{89}}\frac{k_8}{h_8}T_0 u\right)\left(k_2 + \frac{\overline{k_2}k_{510}}{h_{510}}\frac{k_5}{h_5}T_0 u\right)(1-u) - h_2 T_0 u;\\
\Leftrightarrow \quad & h_2\frac{\partial u}{\partial \tilde{t}} = \tilde{D}h_2\Delta u + \frac{k_9 k_{11}}{h_9 h_{10} h_{11}}u\left(k_{10} + \frac{\overline{k_{10}}k_{89}}{h_{89}}\frac{k_8}{h_8}T_0 u\right)\left(k_2 + \frac{\overline{k_2}k_{510}}{h_{510}}\frac{k_5}{h_5}T_0 u\right)(1-u) - h_2 u;\\
\Leftrightarrow \quad & \frac{\partial u}{\partial \tilde{t}} = \tilde{D}\Delta u + \frac{k_9 k_{11}}{h_2 h_9 h_{10} h_{11}}u\left(k_{10} + \frac{\overline{k_{10}}k_{89}}{h_{89}}\frac{k_8}{h_8}T_0 u\right)\left(k_2 + \frac{\overline{k_2}k_{510}}{h_{510}}\frac{k_5}{h_5}T_0 u\right)(1-u) - u;\\
\Leftrightarrow \quad & \frac{\partial u}{\partial \tilde{t}} = \tilde{D}\Delta u + \tilde{b}u^3(1-u) - u,
\end{aligned} \tag{18}$$

where

$$\tilde{b} = M_1 M_2 M_3, \quad M_1 = \frac{k_9 k_{11}}{h_2 h_9 h_{10} h_{11}}, \quad M_2 = \frac{k_8 k_{89}\overline{k_{10}}}{h_8 h_{89}}T_0, \quad M_3 = \frac{\overline{k_2}k_5 k_{510}}{h_5 h_{510}}T_0.$$

The simplification in the last calculation step to Eq. (18) can be done since $\mathcal{O}(k_{10}) = 10^{-4}$ and $\mathcal{O}(M_2) = 10^1$, and $\mathcal{O}(k_2) = 10^0$ and $\mathcal{O}(M_3) = 10^6$ with constants as introduced in App. C. Addition of these small factors does not really influence the equilibrium when $\mathcal{O}(u) > 10^{-5}$.

5.3 Wave Speed Estimate

In Section 4, we showed that the speed of wave propagation of Eq. (11) converges to the speed of wave propagation of Eq. (12) as $\varepsilon \to 0$. We are able to apply Theorem 4.1 on Eq. (15) which propagation speed converges to the propagation speed of Eq. (16). Therefore it is sufficient to find an estimate for the wave speed propagation of Eq. (16). Hence, we rewrite it to the more general form

$$\frac{\partial u}{\partial t} = \breve{D}\Delta u + \breve{b}u^n (1-u) - \sigma u. \tag{19}$$

Equation (19) has traveling wave solution which satisfies

$$\breve{D}w'' + cw' + \breve{b}w^n (1-w) - \sigma w = 0. \tag{20}$$

We will use two different analytical methods to approximate the propagation speed of the wave; the narrow reaction zone method and the piecewise linear approximation.

5.3.1 Narrow Reaction Zone Method

The first method that we will discuss is the narrow reaction zone method. This method was first used by Zeldovich and Frank-Kamenetskii [7]. It assumes that the reaction takes place at one point $x = 0$ in the coordinates of the moving front. When we consider the blood coagulation system, we will see in Section 6 that the equilibrium is reached by a moving front (starting at $x = 0$). One could say that it is a system in which the reaction takes place at one specific point in the moving front. We will consider the equation

$$\breve{D}w'' + c_1 w' + F(w) - \sigma w = 0, \quad F(w) = \breve{b}w^n (1-w). \tag{21}$$

Outside the reaction zone, we consider the following set of linear equations:

$$\begin{cases} \breve{D}w'' + c_1 w' - \sigma w = 0, & x > 0. \\ \breve{D}w'' + c_1 w' = 0, & x < 0. \end{cases} \tag{22}$$

There is a jump condition in the reaction zone $x = 0$. In order to derive this condition, we consider

$$\breve{D}w'' + F(w) = 0. \tag{23}$$

Despite the jump condition, the solution is continuous in the reaction zone, that means that we have

$$\lim_{\varepsilon \to 0} w(\varepsilon) = \lim_{\varepsilon \downarrow 0} w(-\varepsilon). \tag{24}$$

From Eq. (23) we can deduce the jump condition

$$\begin{aligned}
& \breve{D}w''w' + F(w)w' = 0 \\
\Leftrightarrow \quad & \frac{\mathrm{d}\left(\frac{\breve{D}}{2}(w')^2\right)}{\mathrm{d}x} + F(w)w' = 0 \\
\Leftrightarrow \quad & \int_{-\varepsilon}^{\varepsilon} \frac{\breve{D}}{2}\frac{\mathrm{d}\left((w')^2\right)}{\mathrm{d}x}\mathrm{d}x = -\int_{-\varepsilon}^{\varepsilon} F(w)w'\mathrm{d}x \\
\Leftrightarrow \quad & \frac{\breve{D}}{2}\left((w'(\varepsilon))^2 - (w'(-\varepsilon))^2\right) = -\int_{w(-\varepsilon)}^{w(\varepsilon)} F(w)\mathrm{d}w \\
\Leftrightarrow \quad & (w'(\varepsilon))^2 - (w'(-\varepsilon))^2 = -\frac{2}{\breve{D}}\int_{w(-\varepsilon)}^{w(\varepsilon)} F(w)\mathrm{d}w.
\end{aligned} \tag{25}$$

According to Vitaly Volpert [7], the jump conditions are given by

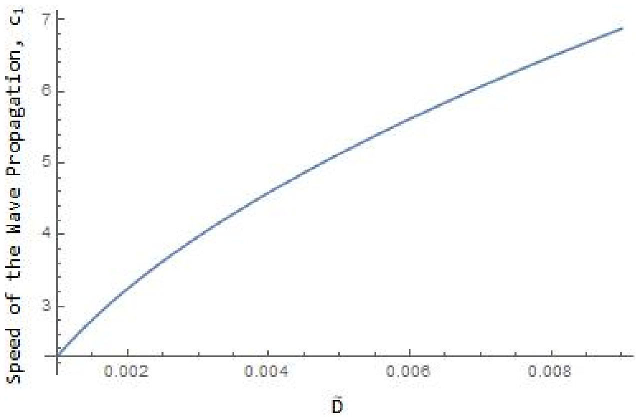
$$(w'(\varepsilon))^2 - (w'(-\varepsilon))^2 = -\frac{2}{\breve{D}}\int_{w(-\infty)}^{w(\infty)} F(w)\mathrm{d}w.$$

Unfortunately, we were not able to find out why the value of ε on the right-hand side are going to infinity, so further research is needed. Since we know that the solution has limits

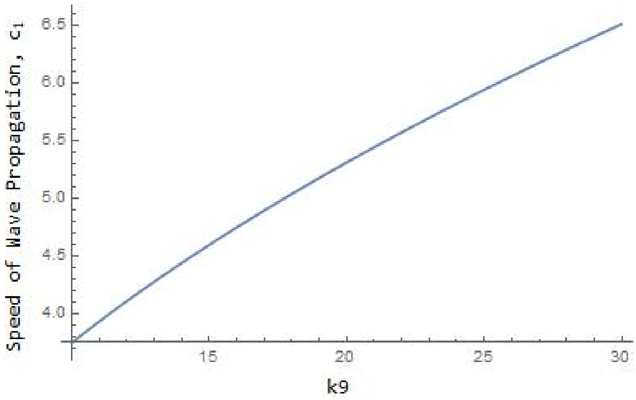
$$\lim_{R\to\infty} w(R) = 0, \qquad \lim_{R\to-\infty} w(R) = w_\star, \tag{26}$$

our jump condition in Eq. (25) changes to

$$(w'(\varepsilon))^2 - (w'(-\varepsilon))^2 = \frac{2}{\breve{D}}\int_0^{w_\star} F(w)\mathrm{d}w. \tag{27}$$



(a) Speed of wave propagation in function of D



(b) Speed of wave propagation in function of k_9

Figure 4: Speed of the wave propagation for the narrow reaction zone method in function of D (left) and k_9 (right) using formula 30 with $w^\star = 1$, $\sigma = 1$.

Solving Eq. (22) with the boundary conditions given in Eq. (24) and Eq. (26) yields

$$w(x) = \begin{cases} w_\star, & x < 0. \\ w_\star \exp\left(\frac{-c_1 - \sqrt{c_1^2 + 4\breve{D}\sigma}}{2\breve{D}}\right), & x > 0. \end{cases} \tag{28}$$

From Eq. (27) and Eq. (28), one can obtain

$$\begin{aligned} & \left(w'(\varepsilon)\right)^2 - \left(w'(-\varepsilon)\right)^2 = \frac{2}{\breve{D}}\int_0^{w_\star} F(w)\mathrm{d}w \\ \Leftrightarrow \quad & \left(w_\star \frac{-c_1^2 - \sqrt{c_1^2 + 4\breve{D}\sigma}}{2\breve{D}}\right)^2 = \frac{2}{\breve{D}}\int_0^{w_\star} F(w)\mathrm{d}w \\ \Leftrightarrow \quad & w_\star^2 \frac{c_1^2 + c_1\sqrt{c_1^2 + 4\breve{D}\sigma} + 2\breve{D}\sigma}{2\breve{D}^2} = \frac{2}{\breve{D}}\int_0^{w_\star} F(w)\mathrm{d}w \\ \Leftrightarrow \quad & c_1^2 + c_1\sqrt{c_1^2 + 4\breve{D}\sigma} + 2\breve{D}\sigma = \frac{4\breve{D}}{w_\star^2}\int_0^{w_\star} F(w)\mathrm{d}w =: A. \end{aligned} \tag{29}$$

Solving Eq. (29) to find the exact value of c_1 yields

$$\begin{aligned} & c_1\sqrt{c_1^2 + 4\breve{D}\sigma} = A - c_1^2 - 2\breve{D}\sigma \\ \Leftrightarrow \quad & c_1^2\left(\cancel{c_1^2} + \cancel{4\breve{D}\sigma}\right) = A^2 + \cancel{c_1^4} + 4\breve{D}^2\sigma^2 - 2Ac_1 - 4A\breve{D}\sigma + \cancel{4\breve{D}\sigma c_1^2} \\ \Leftrightarrow \quad & 2Ac_1^2 = A^2 - 4A\breve{D}\sigma + 4\breve{D}^2\sigma^2 = \left(A - 2\breve{D}\sigma\right)^2 \\ \Leftrightarrow \quad & c_1 = \pm\frac{A - 2\breve{D}\sigma}{\sqrt{2A}}. \end{aligned}$$

Since we need to find $c_1 > 0$, we use

$$c_1 = \frac{\left|A - 2\breve{D}\sigma\right|}{\sqrt{2A}}, \qquad A = 4\breve{D}\breve{b}\left(\frac{w_\star^{n-1}}{n+1} - \frac{w_\star^n}{n+2}\right). \tag{30}$$

When we use the variables as given in Appendix C, we find that the propagation speed evolves as in Figure 4.

5.3.2 Piecewise Linear Approximation

The second approximation that we consider is the piecewise linear approximation. We consider Eq. (20) in the form

$$\breve{D}w'' + cw' + f(w) = 0, \tag{31}$$

where $f(w) = \breve{b}w^n(1-w) - \sigma w$. In addition, we define $w_\star$ such that $f(0) = f(w_\star) = 0$. We approximate Eq. (31) by

$$\breve{D}w'' + cw' + f_0(w) = 0, \tag{32}$$

where

$$f_0(w) = \begin{cases} \alpha w, & 0 < w < w_0 \\ \beta(w - w_\star) & w_0 < w < w_\star, \end{cases}$$

with $\alpha = f'(0)$ and $\beta = f'(w_\star)$. In the specific situation of Eq. (19), this means that we have

$$\alpha = -\sigma, \quad \beta = \breve{b}nw_\star^{n-1} - \breve{b}(n+1)w_\star^n - \sigma. \tag{33}$$

The value of w_0 is determined such that

$$\int_0^{w_\star} f(w)\mathrm{d}w = \int_0^{w_\star} f_0(w)\mathrm{d}w.$$

That means that we have

$$\begin{aligned} \int_0^{w_\star} f(w)\mathrm{d}w &= \breve{b}\frac{w_\star^{n+1}}{n+1} - \breve{b}\frac{w_\star^{n+2}}{n+2} - \frac{\sigma}{2}w_\star^2; \\ \int_0^{w_\star} f_0(w)\mathrm{d}w &= -\frac{\sigma}{2}w_0^2 - w_\star(\beta w_\star - \beta w_0) + \frac{\beta}{2}\left(w_\star^2 - w_0^2\right) = \frac{\alpha-\beta}{2}w_0^2 + \beta w_0 w_\star - \frac{\beta}{2}w_\star^2. \end{aligned}$$

Putting this together means that we have

$$\begin{aligned} & \frac{\alpha-\beta}{2}w_0^2 + \beta w_0 w_\star - \frac{\beta}{2}w_\star^2 - \breve{b}\frac{w_\star^{n+1}}{n+1} + \breve{b}\frac{w_\star^{n+2}}{n+2} + \frac{\sigma}{2}w_\star^2 = 0 \\ \Leftrightarrow \quad & \frac{\alpha-\beta}{2}w_0^2 + \beta w_0 w_\star + r = 0, \end{aligned} \tag{34}$$

where

$$r = -\frac{\beta}{2}w_\star^2 - \frac{\breve{b}}{n+1}w_\star^{n+1} + \frac{\breve{b}}{n+2}w_\star^{n+2} + \frac{\sigma}{2}w_\star^2 = \breve{b}\left(\frac{1}{n+2} + \frac{n+1}{2}\right)w_\star^{n+2} - \breve{b}\left(\frac{1}{n+1} + \frac{n}{2}\right)w_\star^{n+1} + \sigma w_\star^2.$$

From Eq. (34), we deduce that

$$w_0 = \frac{-\beta w_\star \pm \sqrt{\beta^2 w_\star^2 - 2(\alpha-\beta)r}}{\alpha - \beta}.$$

With the variables given in Appendix C, we find that this value is very close to zero. Therefore, we say that Eq. (32) can be approximated by

$$\begin{cases} \breve{D}w'' + c_2 w' + \beta(w - w_\star) = 0, & x < 0, \\ \breve{D}w'' + c_2 w' + \alpha w = 0, & x > 0. \end{cases}$$

In order to solve this system, we need additional conditions on the continuity of this system. The additional conditions are given by

$$w(0) = w_0, \quad w'(-0) = w'(+0).$$

Solving this set of differential equations with the given conditions yields

$$\begin{cases} w = (w_0 - w_\star)\exp\left(\frac{\sqrt{c_2^2 - 4\beta\breve{D}} - c_2}{2\breve{D}}x\right) + w_\star, & x < 0, \\ w = w_0 \exp\left(\frac{-\sqrt{c_2^2 - 4\alpha\breve{D}} - c_2}{2\breve{D}}x\right), & x > 0. \end{cases}$$

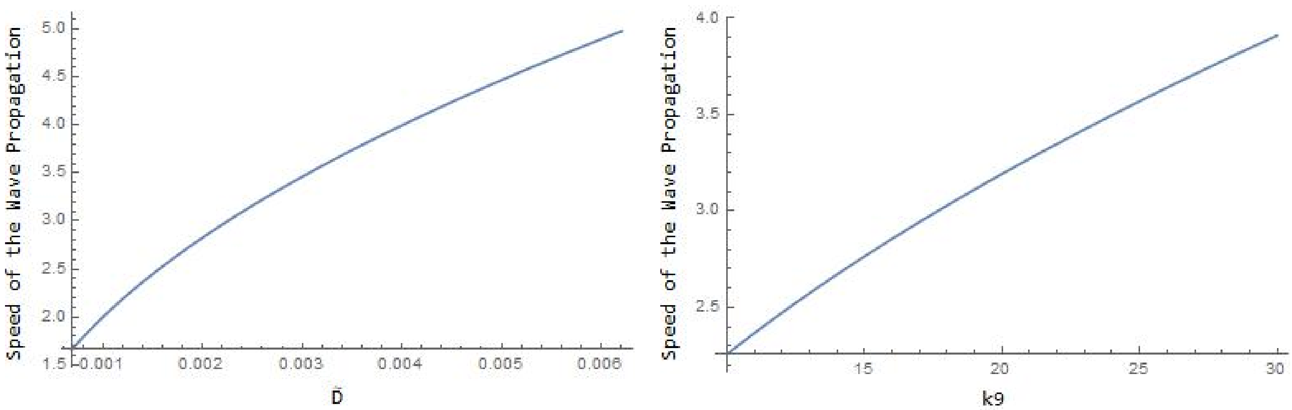


Figure 5: Speed of the wave propagation for the piecewise linear approximation in function of D (left) and k_9 (right) formula (35) with $w_0 = 0.999981$, $\sigma = 1$.

We still need to make sure that the first derivative is continuous as well. We introduce $\overline{w} = \frac{w_0}{w_0 - w_\star}$. That means that we should have

$$
\begin{aligned}
& (w_0 - w_\star)\frac{\sqrt{c_2^2 - 4\beta\breve{D}} - c_2}{2\breve{D}} = w_0\frac{-\sqrt{c_2^2 - 4\alpha\breve{D}} - c_2}{2\breve{D}} \\
\Leftrightarrow\quad & \sqrt{c_2^2 - 4\beta\breve{D}} - c_2 = -\overline{w}\sqrt{c_2^2 - 4\alpha\breve{D}} - \overline{w}c_2 \\
\Leftrightarrow\quad & \sqrt{c_2^2 - 4\beta\breve{D}} + \overline{w}\sqrt{c_2^2 - 4\alpha\breve{D}} = (1 - \overline{w})\, c_2 \\
\Leftrightarrow\quad & \cancel{c_2^2} - 4\beta\breve{D} + 2\overline{w}\sqrt{\left(c_2^2 - 4\beta\breve{D}\right)\left(c_2^2 - 4\alpha\breve{D}\right)} + \overline{w}^2\left(\cancel{c_2^2} - 4\alpha\breve{D}\right) = \left(\cancel{1} - 2\overline{w} + \cancel{\overline{w}^2}\right) c_2^2 \\
\Leftrightarrow\quad & \overline{w}\sqrt{c_2^4 - 4(\alpha+\beta)\,\breve{D}c_2^2 + 16\alpha\beta\breve{D}} = -\overline{w}c_2^2 + 2\left(\alpha\overline{w}^2 + \beta\right)\breve{D} \\
\Leftrightarrow\quad & \overline{w}^2\left(\cancel{c_2^4} - 4(\alpha+\beta)\,\cancel{\breve{D}}c_2^2 + 16\alpha\beta\cancel{\breve{D}}\right) = \cancel{\overline{w}^2 c_2^4} - 4\overline{w}\left(\alpha\overline{w}^2 + \beta\right)\cancel{\breve{D}}c_2^2 + 4\left(\alpha\overline{w}^2 + \beta\right)^2 \breve{D}^{\cancel{2}} \\
\Leftrightarrow\quad & \left(\alpha\overline{w}^3 + \beta\overline{w} - \alpha\overline{w}^2 - \beta\overline{w}^2\right)c_2^2 = \left(\alpha\overline{w}^2 + \beta\right)^2\breve{D} - 4\alpha\beta\overline{w}^2 \\
\Leftrightarrow\quad & \overline{w}\,(\alpha\overline{w} - \beta)\,(\overline{w} - 1)\, c_2^2 = \left(\alpha\overline{w}^2 - \beta\right)^2\breve{D} \\
\Leftrightarrow\quad & c_2 = \pm\frac{\sqrt{\breve{D}}\left(\alpha\overline{w}^2 - \beta\right)}{\sqrt{\overline{w}\,(\alpha\overline{w} - \beta)\,(\overline{w} - 1)}}.
\end{aligned}
$$

We have to find $\overline{w}\,(\alpha\overline{w} - \beta)\,(\overline{w} - 1) > 0$. Since $\overline{w} > 1$, we have to find $\alpha\overline{w} - \beta > 0$ as well. And hence, we find

$$
c_2 = \frac{\sqrt{\breve{D}}\left|\alpha\overline{w}^2 - \beta\right|}{\sqrt{(\overline{w} - 1)\left(\alpha\overline{w}^2 - \beta\overline{w}\right)}}, \quad \overline{w} = \frac{w_0}{w_0 - w_\star}. \tag{35}
$$

When we calculate the values of this estimated speed for different values of the diffusion coefficient $\tilde{D}$ or for different values for the variable k_9, we find figures as given in Figure 5.

5.3.3 Wave Speed Estimates for the Full Model

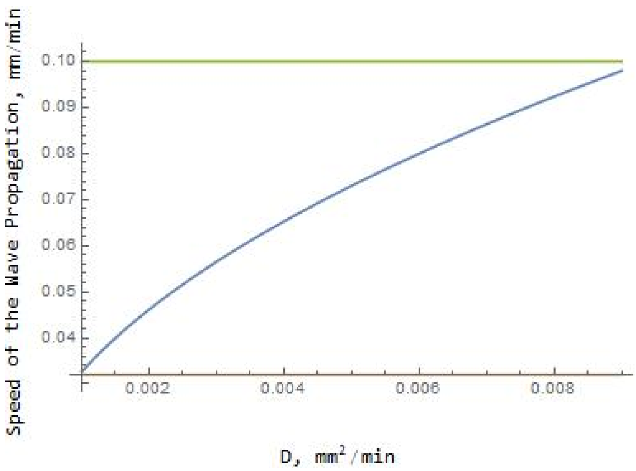
Since we want to find an estimate propagation speed for Eq. (16), we need to change the approximations of the propagation speed back to their dimensional form. In order to do that, we need to write Eq. (16) to the form of Eq. (19). That means that we find

$$
\frac{\partial\hat{T}}{\partial t} = D\Delta\hat{T} + \hat{b}T_0^3\left(1 - \hat{T}\right) - h_2\hat{T},
$$

where $\hat{T} = \frac{T}{T_0}$ and $\hat{b} = \frac{\overline{k_2}k_5k_{510}k_8k_{89}k_9\overline{k_{10}}k_{11}}{h_5h_{510}h_8h_{89}h_9h_{10}h_{11}}$.

We start with the expression for the narrow reaction zone method which was derived in Section 5.3.1 and given in Eq. (30). Rewriting the estimated propagation speed to the estimate of Eq. (1) means that we have to rewrite Eq. (30) towards an expression for Eq. (16). We note that for the full model, we have

$$
T_\star = \lim_{R\to-\infty} T(R) \approx T_0 = 1400\text{ nM}.
$$



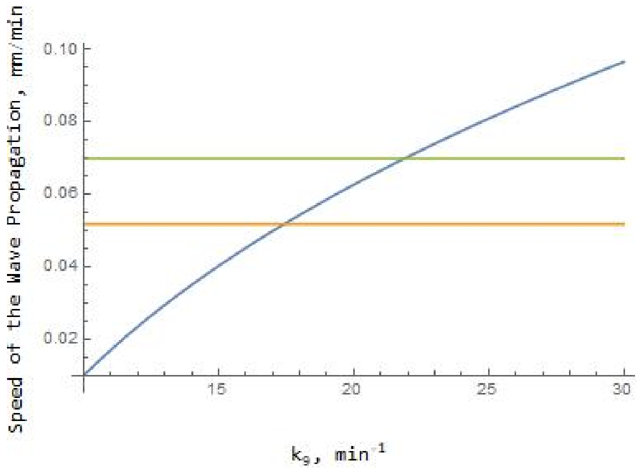


Figure 6: Propagation speed for the narrow reaction zone method applied to Eq. (16).

In addition to that, we have $n = 3$, $\breve{D} = D$, $\sigma = h_2$ and $\breve{b} = \hat{b}T_0^2$. First, we rewrite the expression for A which gives

$$\begin{aligned} A &= 4\breve{b}\breve{D}\left(\frac{w_\star^2}{4} - \frac{w_\star^3}{5}\right) \\ &= 4\hat{b}T_0^2 D\left(\frac{w_\star^2}{4} - \frac{w_\star^3}{5}\right) \\ &= 4D\hat{b}T_0^2\left(\frac{w_\star^2}{4} - \frac{w_\star^3}{5}\right) \\ &= D\hat{b}T_0^2\left(w_\star^2 - \frac{4}{5}w_\star^3\right). \end{aligned} \tag{36}$$

We note that $bh_2 = \hat{b}T_0$. This together with Eq. (36) results in the estimated speed expression

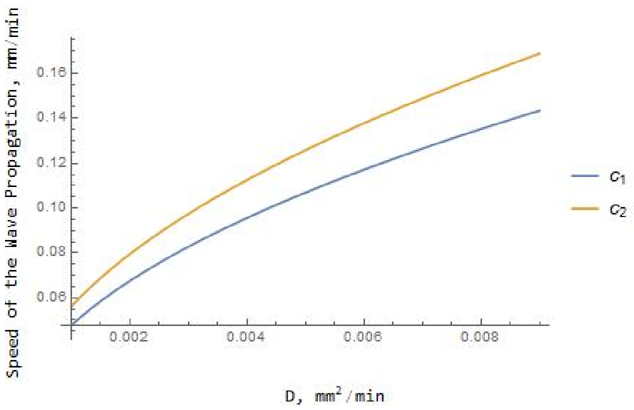
$$\begin{aligned} c_1 &= \frac{A - 2\breve{D}\sigma}{\sqrt{2A}} \\ &= \frac{A - 2Dh_2}{\sqrt{2A}} \\ &= Dh_2\frac{bw_\star^2 - \frac{4}{5}bw_\star^3 - 2}{\sqrt{2bh_2D\left(w_\star^2 - \frac{4}{5}w_\star^3\right)}} \\ &= \sqrt{Dh_2}\frac{bw_\star^2 - \frac{4}{5}bw_\star^3 - 2}{\sqrt{2b\left(w_\star^2 - \frac{4}{5}w_\star^3\right)}}. \end{aligned} \tag{37}$$

In order to get real, valuable results (so a positive square root), we should have $w_\star < \frac{5}{4}$. However, this is not the case for $T_0 = 1400$. We have tried to find the right value for $w_\star$ to get the same figures as Galochkina et al. [1]. For the value $T_0 = 1.2499$, we find the results as given in Figure 6. However, these are not the same as the ones Galochkina et al. found [1], since the curve should be between the orange and green line. We see that this is the case for D, but not for k_9. For k_9, we did not manage to find a right value for T_0 that we can use. However, one should note that we have to use $T_0 = 1400$, although this does not give values in $\mathbb{R}$.

We also want to find an estimate for the propagation speed of Eq. (16) using piecewise linear approximation. Therefore, we rewrite the Eq. (35) to a dimensionless form. That means that we substitute the same variables as above. When we replace n, $w_\star$, $\breve{D}$, $\breve{b}$, σ and $\overline{w}$ with 3, T_0, D, $\hat{b}T_0^2$, h_2 and $\overline{T}$ respectively, we find the following expression for the propagation speed

$$c_2 = \frac{\sqrt{\breve{D}}\left|\alpha\overline{w}^2 - \beta\right|}{\sqrt{(\overline{w} - 1)\left(\alpha\overline{w}^2 - \beta\overline{w}\right)}}, \quad \overline{w} = \frac{w_0}{w_0 - w_\star} \tag{38}$$

$$= \sqrt{D}h_2\frac{-\overline{T}^2 - 3bT_0^2 + 4bT_0^3 + 1}{\sqrt{\left(\overline{T} - 1\right)\overline{T}h_2\left(1 - \overline{T} - 3bT_0^2 + 4bT_0^3\right)}} \tag{39}$$



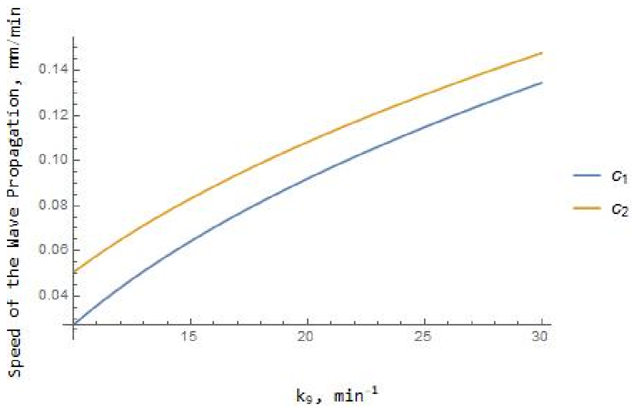


Figure 7: Speed of the wave propagation for the narrow reaction zone and the piecewise linear approximation in function of D (left) and k_9 (right). Both expressions use $T_0 = 1.24988$.

where

$$\overline{T} = \frac{T_\star}{T_\star - T_0};$$
$$T_\star = \frac{-3bT_0^2 + 4bT_0^3 + 1 + \sqrt{\left(3bT_0^2 - 4bT_0^3 - 1\right)^2 - 2b\left(4T_0 - 3\right)T_0^2\left(\frac{11}{5}bT_0^3 - b\left(\frac{1}{4} + \frac{3}{2}\right)T_0^2 + 1\right)}}{4bT_0^2 - 3bT_0}.$$

The graphs that we get when we plot Eq. (37) and Eq. (39) is given in Figure 7. We used $T_0 = 1.24988$ in this plot. As discussed earlier, we should use $T_0 = 1400$, but this does not give real results. When one does a dimensional analysis of Eq. (37) and Eq. (39), we find that T_0 should be dimensionless in order to find the right dimensional results:

$$[c_1] = \left(\text{mm}^2/\text{min} \cdot \text{min}^{-1}\right)^{1/2} \frac{1}{(1)^{1/2}} = \text{mm/min},$$
$$[c_2] = \left(\text{mm}^2/\text{min}\right)^{1/2} \frac{\text{min}^{-1}}{\left(\text{min}^{-1}\right)^{1/2}} = \text{mm/min},$$

since b is dimensionless as well.

6 Simulation

In the Section 5.3 we derived a theoretical approximation of the propagation speed of the wave. With the help of a simulation, we are able to identify the wave speed of the blood coagulation model given by Eq. (1). We use Merson's method for the time-integration and the five-point stencil for the second derivative in space.

6.1 Five-Point Stencil for Second Derivative

For the second order derivative in space, we use the five-point stencil since according to Zarnitsina et al. [4], we need a high order of accuracy. This means that we use the both neighbors, and the neighbors of the neighbors to estimate the value in a specific point. We discretize the space by dividing it into m points as can be seen in Figure 8. The distance h between all points is the same, and is given by $h = \frac{1}{m-1}$.

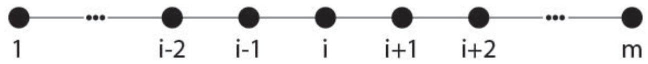


Figure 8: Discretization of space.

We will use the values of points $x_{i-2}, x_{i-1}, x_i, x_{i+1}$ and x_{i+2} to estimate the value of y_i''. First, we do a Taylor expansion of all points around the point x_i that has value y_i. That gives

$$\begin{aligned}
y_{i-2} &= y_i - 2hy_i' + 2h^2 y_i'' - \frac{4h^3}{3} y_i''' + \frac{2h^4}{3} y_i^{(4)} + \mathcal{O}\left(h^5\right), \\
y_{i-1} &= y_i - hy_i' + \frac{h^2}{2} y_i'' - \frac{h^3}{6} y_i''' + \frac{h^4}{24} y_i^{(4)} + \mathcal{O}\left(h^5\right), \\
y_i &= y_i, \\
y_{i+1} &= y_i + hy_i' + \frac{h^2}{2} y_i'' + \frac{h^3}{6} y_i''' + \frac{h^4}{24} y_i^{(4)} + \mathcal{O}\left(h^5\right), \\
y_{i+2} &= y_i + 2hy_i' + 2h^2 y_i'' + \frac{4h^3}{3} y_i''' + \frac{2h^4}{3} y_i^{(4)} + \mathcal{O}\left(h^5\right).
\end{aligned}$$

Since we want to estimate the value of y_i'', we need to find an expression in the form

$$y_i'' = \frac{ay_{i-2} + by_{i-1} + cy_i + dy_{i+1} + ey_{i+2}}{h^2}.$$

That means that we have to find a, b, c, d, e such that

$$\begin{aligned}
a + b + c + d + e &= 0, \\
-2a - b + d + 2e &= 0, \\
2a + \frac{b}{2} + \frac{d}{2} + 2e &= -1, \\
-\frac{4}{3}a - \frac{1}{6}b + \frac{1}{6}d + \frac{4}{3}e &= 0, \\
\frac{2}{3}a + \frac{1}{24}b + \frac{1}{24}d + \frac{2}{3}e &= 0.
\end{aligned}$$

Solving this system of equations yields

$$a = -\frac{1}{12}, \quad b = \frac{4}{3} = \frac{16}{12}, \quad c = -\frac{5}{2} = -\frac{30}{12}, \quad d = \frac{4}{3} = \frac{16}{12}, \quad e = -\frac{1}{12}.$$

That means that we find

$$y_i'' = \frac{-y_{i-2} + 16y_{i-1} - 30y_i + 16y_{i+1} - y_{i+2}}{12h^2} + \mathcal{O}\left(h^4\right) \tag{40}$$

for the discretization of the second order spatial derivative.

We consider the points on the boundary and their direct neighbors separately. We recall that the boundary conditions as given in Eq. (2) are Neumann Boundary Conditions in the form:

$$\left.\frac{\partial G}{\partial x}\right|_{x=0} = a_G, \tag{41a}$$

$$\left.\frac{\partial G}{\partial x}\right|_{x=L} = 0, \tag{41b}$$

where a_G is a variable that takes on a different value depending on the function that we are considering ($a_G = T_0$ if $G = U_{11}$ and $a_G = 0$ otherwise).

First, we will look at what happens at the left boundary ($x = 0$) where the boundary condition is as given in Eq. (41a). In order to determine what influence this will have on the discretization of the second order derivative term, we discretize this first-order derivative which leads us to find

$$\frac{y_2 - y_0}{2h} = a_G \Leftrightarrow y_0 = y_2 - 2ha_G,$$

and

$$\frac{-y_3 + 8y_2 - 8y_0 + y_{-1}}{12h} = a \Leftrightarrow y_{-1} = y_3 - 4ha_G.$$

That means that we find for the discretized expressions on the left hand side of the domain:

$$\begin{aligned} y_1'' &= \frac{-30y_1 + 32y_2 - 2y_3 - 28a_G h}{12h^2}, \\ y_2'' &= \frac{16y_1 - 31y_2 + 16y_3 - y_4 + 2a_G h}{12h^2}. \end{aligned}$$

Next, we take a look at what is happening at the right boundary ($x = L$). Here, the boundary conditions are given by Eq. (41b). These boundary conditions can be rewritten into specific conditions for the values at locations x_{n+1} and x_{n+2}. That means that we find

$$\begin{aligned} \frac{y_{n+1} - y_{n-1}}{2h} &= a_G \Leftrightarrow y_{n+1} = y_{n-1} + 2ha_G, \\ \frac{-y_{n+2} + 8y_{n+1} - 8y_{n-1} + y_{n-2}}{12h} &= a_G \Leftrightarrow y_{n+2} = y_{n-2} + 4ha_G. \end{aligned}$$

This results in the discretized expressions

$$\begin{aligned} y_{n-1}'' &= \frac{-y_{n-3} + 16y_{n-2} - 31y_{n-1} + 16y_n - 2ha_G}{12h^2}, \\ y_n'' &= \frac{-2y_{n-2} + 32y_{n-1} - 30y_n + 28ha_G}{12h^2}, \end{aligned}$$

for the second order space derivative.

6.2 Merson's Method

For the time-integration, we use Merson's method because Zarnitsa used this method for the more extensive model presented by Zarnitsina et al. as well [4]. This is a 5-stage Runge-Kutta method. We recall the definition of Runge-Kutta methods:

Definition 6.1 (s-stage Explicit Runge-Kutta Method [9])**.** Let s be an integer (the "number of stages") and $a_{21}, a_{31}, a_{32}, \ldots, a_{s1}, a_{s2}, \ldots, a_{s,s-1}, b_1, \ldots, b_s, c_2, \ldots, c_s$ be real coefficients. Then the method

$$k_1 = f(x_0, y_0) \tag{42a}$$
$$k_2 = f(x_0 + c_2 h, y_0 + h a_{21} k_1) \tag{42b}$$
$$k_3 = f(x_0 + c_3 h, y_0 + h(a_{31}k_1 + a_{32}k_2)) \tag{42c}$$
$$\vdots \tag{42d}$$
$$k_s = f(x_0 + c_s h, y_0 + h(a_{s1}k_1 + \ldots + a_{s,s-1}k_{s-1})) \tag{42e}$$
$$y_1 = y_0 + h(b_1 k_1 + \ldots b_s k_s) \tag{42f}$$

is called an s-stage explicit Runge-Kutta method (ERK) for the initial value problem

$$y' = f(x, y), \quad y(x_0) = y_0,$$

with maximal step size h.

We know that Merson's method is a fourth order method [9]. We recall the definition of the order of an integration method.

Definition 6.2 (Order of a Runge-Kutta Method [9])**.** A Runge-Kutta Method (42) has order p if for sufficiently smooth problems

$$\|y(x_0 + h) - y_1\| \leq Kh^{p+1},$$

i.e., if the Taylor series for the exact solution $y(x_0 + h)$ and for y_1 coincide up to (and including) the term h^p.

That means that the error that we make in every calculation step is smaller than $K\left(\Delta t\right)^6$ for some constant K. The constants that are used for Merson's method are given in the Butcher tableau in Table 1b. All of this

$$\begin{array}{c|cccccc} 0 & & & & & \\ c_2 & a_{21} & & & & \\ c_3 & a_{31} & a_{32} & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & a_{s2} & \dots & a_{s,s-1} & \\ \hline y_1 & b_1 & b_2 & \dots & b_{s-1} & b_s \end{array}$$

(a) General format of a Butcher Tableau.

$$\begin{array}{c|ccccc} 0 & & & & & \\ \frac{1}{3} & \frac{1}{3} & & & & \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} & & & \\ \frac{1}{2} & \frac{1}{8} & 0 & \frac{3}{8} & & \\ 1 & \frac{1}{2} & 0 & -\frac{3}{2} & 2 & \\ \hline y_1 & \frac{1}{6} & 0 & 0 & \frac{2}{3} & \frac{1}{6} \end{array}$$

(b) Butcher Tableau for Merson's Method.

Table 1: Butcher Tableau's [9].

together means that in order to apply Merson as a method for the time integration of

$$\frac{\partial y\left(t,x\right)}{\partial t} = f\left(t,x\right), \tag{43}$$

we need to calculate

$$\begin{aligned} k_1 &= f(t_n, y_n), \\ k_2 &= f\left(t_n + \frac{\Delta t}{3}, y_n + \frac{\Delta t}{3} k_1\right), \\ k_3 &= f\left(t_n + \frac{\Delta t}{3}, y_n + \frac{\Delta t}{6}\left(k_1 + k_2\right)\right), \\ k_4 &= f\left(t_n + \frac{\Delta t}{2}, y_n + \frac{\Delta t}{8}\left(k_1 + 3k_3\right)\right), \\ k_5 &= f\left(t_n + \Delta t, y_n + \frac{\Delta t}{2}\left(k_1 - 3k_3 + 4k_4\right)\right), \\ y_{n+1} &= y_n + \frac{\Delta t}{6}\left(k_1 + 4k_4 + k_5\right). \end{aligned}$$

For Eq. (16), this means that f is given by

$$f\left(t,T\right) = D\Delta x + \frac{k_9 k_{11}}{h_9 h_{10} h_{11}} T\left(k_{10} + \frac{\overline{k_{10}} k_{89}}{h_{89}} \frac{k_8}{h_8} T\right)\left(k_2 + \frac{\overline{k_2} k_{510}}{h_{510}} \frac{k_5}{h_5} T\right)\left(1 - \frac{T}{T_0}\right) - h_2 T. \tag{44}$$

For Eq. (1), Eq. (15) and Eq. (18), similar expressions for the function f can be found. In the numerical simulation, the second derivative in space will be discretized using the five point stencil as explained in Section 6.1.

6.3 Stability Analysis

We do a stability analysis for the Eq. (1), Eq. (15), Eq. (16) and Eq. (18). We do a Von Neumann stability analysis under the assumption that the time and space integration respectively use Forward Euler and Central Differences.

6.3.1 Dimensionless One Equation Model

We start our analysis with the dimensionless one equation model given by Eq. (18). First, we consider the linearized model. This is given by

$$\frac{\partial u}{\partial t} = \tilde{D}\Delta u - \sigma u. \tag{45}$$

Discretizing Eq. (45) using Forward Euler for the time integration and Central Differences for the space integration yields

$$u_j^{n+1} = \left(1 - 2\alpha - \sigma\Delta t\right) u_j^n + \alpha\left(u_{j+1}^n + u_{j-1}^n\right) \tag{46}$$

at time step n and spatial location j where $\alpha = \frac{\tilde{D}\Delta t}{(\Delta x)^2}$. We will perform the Von Neumann Stability analysis. Therefore, we consider $u_j^n = \xi^n e^{ikj\Delta x}$ where k is a real spatial wave number and $\xi = \xi(k)$ is a complex number that depends on k. In order to have stability, we should find $|\xi(k)| < 1 \quad \forall k$. Substituting this into Eq. (46) yields

$$\xi(k) = (1 - 2\alpha - \sigma\Delta t) + \alpha\left(e^{ik\Delta x} + e^{-ik\Delta x}\right) = (1 - 2\alpha - \sigma\Delta t) + 2\alpha\cos(k\Delta x). \tag{47}$$

Checking the stability condition of Eq. (47) means:

$$\begin{aligned}
& |(1 - 2\alpha - \sigma\Delta t) + 2\alpha\cos(k\Delta x)| \leq 1 \\
\Leftrightarrow\quad & -2 \leq (-2\alpha - \sigma\Delta t) + 2\alpha\cos(k\Delta x) \leq 0 \\
\Leftrightarrow\quad & -2 + \sigma\Delta t \leq -2\alpha + 2\alpha\cos(k\Delta x) \leq \sigma\Delta t \\
\Leftrightarrow\quad & -2 + \sigma\Delta t \leq -4\alpha\sin^2\left(\frac{k\Delta x}{2}\right) \leq \sigma\Delta t \\
\Leftrightarrow\quad & -\frac{\sigma\Delta t}{4} \leq \alpha\sin^2\left(\frac{k\Delta x}{2}\right) \leq \frac{1}{2} - \frac{\sigma\Delta t}{4} \\
\Leftrightarrow\quad & \alpha = \frac{\tilde{D}\Delta t}{(\Delta x)^2} \leq \frac{1}{2} - \frac{\sigma\Delta t}{4} \\
\Leftrightarrow\quad & \frac{4\tilde{D}\Delta t + \sigma\Delta t\,(\Delta x)^2}{4\,(\Delta x)^2} \leq \frac{1}{2} \\
\Leftrightarrow\quad & \Delta t \leq \frac{2\,(\Delta x)^2}{4\tilde{D} + \sigma\,(\Delta x)^2}.
\end{aligned} \tag{48}$$

Note that this stability is in line with our expectation, since choosing $\sigma = 0$ gives the standard diffusion equation. The corresponding stability condition is the same as the stability condition one would expect to find $\frac{D\Delta t}{(\Delta x)^2} \leq \frac{1}{2}$.

Next, we consider the nonlinearized model of Eq. (18). We linearize this system by substituting $u = u_0 + \delta u$ which we expand to linear order in δu. We choose u_0 such that it satisfies the difference equation exactly. That means that we find:

$$\begin{aligned}
& \frac{\partial u}{\partial t} = \tilde{D}\Delta u + bu^n(1-u) - \sigma u \\
\Leftrightarrow\quad & \frac{\partial u_0}{\partial t} + \frac{\partial \delta u}{\delta t} = \tilde{D}\Delta u_0 + \tilde{D}\Delta\delta u + b(u_0 + \delta u)^n(1 - u_0 - \delta u) - \sigma u_0 - \sigma\delta u \\
\Leftrightarrow\quad & \frac{\partial \delta u}{\partial t} = \tilde{D}\Delta\delta u - bu_0^n\delta u + nbu_0^{n-1}\delta u(1 - u_0) - \sigma\delta u.
\end{aligned} \tag{49}$$

We can choose u_0 such that it is the value of the steady state of the system. That means that its derivatives are zero. By consequence, we find that u_0 has value 0:

$$bu^n(1-u) - \sigma u = 0 \Leftrightarrow u = 0 \vee bu^{n-1}(1-u) - \sigma = 0.$$

That leads us to find that Eq. (49) has the same stability condition as calculated earlier in this section since Eq. (49) has the same discretization equation. With a similar way of reasoning, we find the similar conclusions for the 'extended' one equation model given in Eq. (16), but now with $\sigma = \frac{k_2k_9k_{10}k_{11}}{h_9h_{10}h_{11}} - h2$ and $\tilde{D} = D$ which leads us to find the stability condition

$$\Delta t \leq \frac{2\,(\Delta x)^2}{4D + \left(\frac{k_2k_9k_{10}k_{11}}{h_9h_{10}h_{11}} - h2\right)(\Delta x)^2}. \tag{50}$$

6.3.2 Two Equation Model

First, we consider the linearized model of Eq. (15). The linearized model is given by

$$\frac{\partial T}{\partial t} = D\Delta T + \frac{k_9k_{10}k_2}{h_9h_{10}}U_{11} - h_2T, \tag{51a}$$

$$\frac{\partial U_{11}}{\partial t} = D\Delta U_{11} + k_{11}T - h_{11}U_{11}. \tag{51b}$$

Discretization in time with Forward Euler and in space with central differences of system (51) yields

$$T_j^{n+1} = T_j^n + \Delta t\left(D\frac{T_{j+1}^n - 2T_j^n + T_{j-1}^n}{(\Delta x)^2} + \frac{k_9k_{10}k_2}{h_9h_{10}}u_j^n - h_2T_j^n\right), \tag{52a}$$

$$u_j^{n+1} = u_j^n + \Delta t\left(D\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2} + k_{11}T_j^n - h_{11}u_j^n\right), \tag{52b}$$

where T_j^n denotes the thrombin level at time $n\Delta t$ and at location $j\Delta t$ and where u_j^n the level of U_{11} at time $n\Delta t$ and at location $j\Delta t$.

We substitute

$$\begin{bmatrix} T_j^n \\ u_j^n \end{bmatrix} = \xi^n e^{ikj\Delta x} \begin{bmatrix} T^0 \\ u_0 \end{bmatrix}.$$

That means that Eq. (52) can be rewritten to

$$\begin{aligned} \xi \begin{bmatrix} T^0 \\ u_0 \end{bmatrix} &= \begin{bmatrix} T^0 \\ u_0 \end{bmatrix} + \frac{D\Delta t}{(\Delta x)^2} \begin{bmatrix} -2 + 2\cos(k\Delta x)\, T^0 \\ -2 + 2\cos(k\Delta x)\, u_0 \end{bmatrix} + \Delta t \begin{bmatrix} -h_2 & \frac{k_9 k_{10} k_2}{h_9 h_{10}} \\ k_{11} & -h_{11} \end{bmatrix} \begin{bmatrix} T^0 \\ u_0 \end{bmatrix} \\ \Leftrightarrow \begin{bmatrix} 1 - 4\frac{D\Delta t}{(\Delta x)^2}\sin^2\frac{k\Delta x}{2} - h_2\Delta t - \xi & \frac{k_9 k_{10} k_2}{h_9 h_{10}}\Delta t \\ k_{11}\Delta t & 1 - 4\frac{D\Delta t}{(\Delta x)^2}\sin^2\frac{k\Delta x}{2} - h_{11}\Delta t - \xi \end{bmatrix} &\begin{bmatrix} T^0 \\ u_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \end{aligned} \tag{53}$$

We want to make sure that Eq. (53) is always satisfied. That means that the determinant of the matrix should be zero and thus

$$\begin{aligned} &\left(1 - 4\frac{D\Delta t}{(\Delta x)^2}\sin^2\frac{k\Delta x}{2} - h_2\Delta t - \xi\right)\left(1 - 4\frac{D\Delta t}{(\Delta x)^2}\sin^2\frac{k\Delta x}{2} - h_{11}\Delta t - \xi\right) - \frac{k_9 k_{10} k_2 k_{11}}{h_9 h_{10}}(\Delta t)^2 = 0 \\ \Leftrightarrow\quad &\xi^2 - \xi\left(2 - 8\frac{D\Delta t}{(\Delta x)^2}\sin^2\frac{k\Delta x}{2} - (h_{11} + h_2)\Delta t\right) \\ &+ \left(1 - 4\frac{D\Delta t}{(\Delta x)^2}\sin^2\frac{k\Delta x}{2} - h_2\Delta t\right)\left(1 - 4\frac{D\Delta t}{(\Delta x)^2}\sin^2\frac{k\Delta x}{2} - h_{11}\Delta t\right) - \frac{k_9 k_{10} k_2 k_{11}}{h_9 h_{10}}(\Delta t)^2 = 0 \\ \Leftrightarrow\quad &\xi = \frac{\left(2 - 8\frac{D\Delta t}{(\Delta x)^2}\sin^2\frac{k\Delta x}{2} - (h_{11} + h_2)\Delta t\right) \pm \sqrt{B}\Delta t}{2}, \end{aligned} \tag{54}$$

where

$$B = \left((h_{11} - h_2)^2 + 4\frac{k_9 k_{10} k_2 k_{11}}{h_9 h_{10}}\right).$$

The stability condition is found when we have $|\xi| \leq 1$. That means that Eq. (54) becomes

$$\begin{aligned} &\left|1 - 4\frac{D\Delta t}{(\Delta x)^2}\sin^2\frac{k\Delta x}{2} - \frac{1}{2}(h_{11} + h_2)\Delta t \pm \frac{\Delta t}{2}\sqrt{B}\right| \leq 1 \\ \Leftrightarrow\quad &0 \leq 4\frac{D\Delta t}{(\Delta x)^2}\sin^2\frac{k\Delta x}{2} + \frac{\Delta t}{2}\left(h_{11} + h_2 \mp \sqrt{B}\right) \leq 2 \\ \Leftrightarrow\quad &-\frac{\Delta t}{8}\left(h_{11} + h_2 \mp \sqrt{B}\right) \leq \frac{D\Delta t}{(\Delta x)^2}\sin^2\frac{k\Delta x}{2} \leq \frac{1}{2} - \frac{\Delta t}{8}\left(h_{11} + h_2 \mp \sqrt{B}\right). \end{aligned}$$

In order to have stability, we should have $h_{11} + h_2 \mp \sqrt{B} > 0$. Note that this is the case for the given variables in Appendix C. That means that we find the stability condition

$$\Delta t \leq \frac{4(\Delta x)^2}{8D + (\Delta x)^2\left(h_{11} + h_2 \mp \sqrt{B}\right)}. \tag{55}$$

Next, we consider the nonlinearized model of Eq. (15). We linearize this system substituting

$$\begin{cases} T = \tilde{T} + \delta T; \\ U_{11} = \tilde{U} + \delta U, \end{cases}$$

where $\tilde{U}$ and $\tilde{T}$ are chosen such that they satisfy Eq. (15) exactly (steady state solutions). First, we determine values for $\tilde{U}$ and $\tilde{T}$. We use the steady states of Eq. (15). These are given by

$$\begin{cases} \tilde{U} = \frac{k_{11}}{h_{11}}\tilde{T}, \\ \tilde{T} = 0 \quad \vee \quad \frac{k_9 k_{11}}{h_9 h_{10} h_{11}}\left(k_{10} + \frac{\overline{k_{10}} k_{89} k_8}{h_{89} h_8}\tilde{T}\right)\left(k_2 + \frac{\overline{k_2} k_{510} k_5}{h_{510} h_5}\tilde{T}\right)\left(1 - \frac{\tilde{T}}{T_0}\right) - h_2 = 0. \end{cases}$$

We consider the case

$$\begin{cases} \tilde{T} = 0, \\ \tilde{U} = 0. \end{cases}$$

That means that we find that the linearized system of Eq. (15) reduces back to the linearized model as given in Eq. (51) treated earlier in this section. Therefore, the stability condition is the same.

6.3.3 Full Model

First, we consider the linearized model of Eq. (1). This model is given by the following system of equations:

$$\frac{\partial T}{\partial t} = D\Delta T + k_2 U_{10} - h_2 T, \tag{56a}$$
$$\frac{\partial U_5}{\partial t} = D\Delta U_5 + k_5 T - h_5 U_5, \tag{56b}$$
$$\frac{\partial U_8}{\partial t} = D\Delta U_8 + k_8 T - h_8 U_8, \tag{56c}$$
$$\frac{\partial U_9}{\partial t} = D\Delta U_9 + k_9 U_{11} - h_9 U_9, \tag{56d}$$
$$\frac{\partial U_{10}}{\partial t} = D\Delta U_{10} + k_{10} U_9 - h_{10} U_{10}, \tag{56e}$$
$$\frac{\partial U_{11}}{\partial t} = D\Delta U_{11} + k_{11} T - h_{11} U_{11}. \tag{56f}$$

Discretizing this system gives

$$\begin{aligned}
T_j^{n+1} &= T_j^n + \Delta t \left(D\frac{T_{j+1}^n - 2T_j^n + T_j^n}{(\Delta x)^2} + k_2 v_j^n - h_2 T_j^n \right), \\
r_j^{n+1} &= r_j^n + \Delta t \left(D\frac{r_{j+1}^n - 2r_j^n + r_{j-1}^n}{(\Delta x)^2} + k_5 T_j^n - h_5 r_j^n \right), \\
s_j^{n+1} &= s_j^n + \Delta t \left(D\frac{s_{j+1}^n - 2s_j^n + s_{j-1}^n}{(\Delta x)^2} + k_8 T_j^n - h_8 s_j^n \right), \\
u_j^{n+1} &= u_j^n + \Delta t \left(D\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2} + k_9 w_j^n - h_9 u_j^n \right), \\
v_j^{n+1} &= v_j^n + \Delta t \left(D\frac{v_{j+1}^n - 2v_j^n + v_{j-1}^n}{(\Delta x)^2} + k_{10} u_j^n - h_{10} v_j^n \right), \\
w_j^{n+1} &= w_j^n + \Delta t \left(D\frac{w_{j+1}^n - 2w_j^n + w_{j-1}^n}{(\Delta x)^2} + k_{11} T_j^n - h_{11} w_j^n \right),
\end{aligned}$$

where the different variables are related to the variables in the PDE as in Table 2. We substitute for variable z the function $z_j^n = \xi^n e^{ikj\Delta x} z_0$ with $z = T, r, s, u, v, w$. That leaves us with the system

$$\xi \begin{bmatrix} T^0 \\ r^0 \\ s^0 \\ u_0 \\ v^0 \\ w^0 \end{bmatrix} = \begin{bmatrix} T^0 \\ r^0 \\ s^0 \\ u_0 \\ v^0 \\ w^0 \end{bmatrix} + \frac{D\Delta t}{(\Delta x)^2}\left(-2 + 2\cos(k\Delta x)\right) \begin{bmatrix} T^0 \\ r^0 \\ s^0 \\ u_0 \\ v^0 \\ w^0 \end{bmatrix} + \Delta t \begin{bmatrix} k_2 v^0 \\ k_5 T^0 \\ k_8 T^0 \\ k_9 w^0 \\ k_{10} u_0 \\ k_{11} T^0 \end{bmatrix} - \Delta t \begin{bmatrix} h_2 T^0 \\ h_5 r^0 \\ h_8 s^0 \\ h_9 u_0 \\ h_{10} v^0 \\ h_{11} w^0 \end{bmatrix}.$$

Rewriting this system yields

$$\left(\left(1 - 4\frac{D\Delta t}{(\Delta x)^2}\sin^2\frac{k\Delta x}{2} - \xi\right)\mathbf{I} + \begin{bmatrix} -h_2\Delta t & 0 & 0 & 0 & k_2\Delta t & 0 \\ k_5\Delta t & -h_5\Delta t & 0 & 0 & 0 & 0 \\ k_8\Delta t & 0 & -h_8\Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & -h_9\Delta t & 0 & k_9\Delta t \\ 0 & 0 & 0 & k_{10}\Delta t & -h_{10}\Delta t & 0 \\ k_{11}\Delta t & 0 & 0 & 0 & 0 & -h_{11}\Delta t \end{bmatrix} \right) \begin{bmatrix} T^0 \\ r^0 \\ s^0 \\ u_0 \\ v^0 \\ w^0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{57}$$

The determinant of this system is given by

$$\det(A) = (a - h_5\Delta t)(a - h_8\Delta t)\left[(a - h_2\Delta t)(a - h_9\Delta t)(a - h_{10}\Delta t)(a - h_{11}\Delta t) - (\Delta t)^4 k_{11}k_2k_9k_{10}\right]$$

Variable in PDE	Variable in Discretization
T	T
U_5	r
U_8	s
U_9	u
U_{10}	v
U_{11}	w

Table 2: Relation between variables in PDE and discretization.

where A is the matrix in Eq. (57) and $a := 1 - 4\frac{D\Delta t}{(\Delta x)^2}\sin^2\frac{k\Delta x}{2} - \xi$. By setting the determinant to 0, we can find the following expressions of ξ:

$$\begin{aligned}
\xi &= 1 - 4\frac{D\Delta t}{(\Delta x)^2}\sin^2\frac{k\Delta x}{2} - h_5\Delta t,\\
\xi &= 1 - 4\frac{D\Delta t}{(\Delta x)^2}\sin^2\frac{k\Delta x}{2} - h_8\Delta t,\\
&(a - h_2\Delta t)(a - h_9\Delta t)(a - h_{10}\Delta t)(a - h_{11}\Delta t) - (\Delta t)^4 k_{11}k_2k_9k_{10} = 0.
\end{aligned}$$

For now, we assume that $k_{11}k_2k_9k_{10} \approx 0$. This is a reasonable assumption since, using the constants in Appendix C, we find $\mathcal{O}(k_{11}k_2k_9k_{10}) = 10^{-7}$. In that case, the other expressions for ξ would be given by

$$\begin{aligned}
\xi &= 1 - 4\frac{D\Delta t}{(\Delta x)^2}\sin^2\frac{k\Delta x}{2} - h_2\Delta t,\\
\xi &= 1 - 4\frac{D\Delta t}{(\Delta x)^2}\sin^2\frac{k\Delta x}{2} - h_9\Delta t,\\
\xi &= 1 - 4\frac{D\Delta t}{(\Delta x)^2}\sin^2\frac{k\Delta x}{2} - h_{10}\Delta t,\\
\xi &= 1 - 4\frac{D\Delta t}{(\Delta x)^2}\sin^2\frac{k\Delta x}{2} - h_{11}\Delta t.
\end{aligned}$$

To obtain stability, we need to make sure $|\xi| \leq 1$, and therefore we find the stability conditions:

$$\begin{aligned}
\Delta t &\leq \frac{2(\Delta x)^2}{4D + h_2(\Delta x)^2},\\
\Delta t &\leq \frac{2(\Delta x)^2}{4D + h_5(\Delta x)^2},\\
\Delta t &\leq \frac{2(\Delta x)^2}{4D + h_8(\Delta x)^2},\\
\Delta t &\leq \frac{2(\Delta x)^2}{4D + h_9(\Delta x)^2},\\
\Delta t &\leq \frac{2(\Delta x)^2}{4D + h_{10}(\Delta x)^2},\\
\Delta t &\leq \frac{2(\Delta x)^2}{4D + h_{11}(\Delta x)^2}.
\end{aligned}$$

These conditions follow since all constants are positive. If $h_i > 0$ ($i = 5, 8, 9, 10, 11$) is not satisfied, the system is unstable. For the constants given in Appendix C, all these conditions are satisfied.

Next, we consider the nonlinearized model of Eq. (1). We linearize the system by substituting

$$\begin{cases}
T = \tilde{T} + \delta T,\\
U_5 = \tilde{U}_5 + \delta U_5,\\
U_8 = \tilde{U}_8 + \delta U_8,\\
U_9 = \tilde{U}_9 + \delta U_9,\\
U_{10} = \tilde{U}_{10} + \delta U_{10},\\
U_{11} = \tilde{U}_{11} + \delta U_{11},
\end{cases}$$

where $\tilde{T}$, $\tilde{U}_6$, $\tilde{U}_8$, $\tilde{U}_9$, $\tilde{U}_{10}$ and $\tilde{U}_{11}$ are the steady state solutions. First, we determine these values. That means that we find

$$\begin{aligned}
\tilde{U}_{11} &= \frac{k_{11}}{h_{11}}\tilde{T},\\
\tilde{U}_5 &= \frac{k_5}{h_5}\tilde{T},\\
\tilde{U}_8 &= \frac{k_8}{h_8}\tilde{T},\\
\tilde{U}_9 &= \frac{k_9}{h_9}\tilde{U}_{11} = \frac{k_9 k_{11}}{h_9 h_{11}}\tilde{T},\\
\tilde{U}_{10} &= \frac{k_{10}h_{89} + \overline{k_{10}}k_{89}\tilde{U}_8}{h_{10}h_{89}}\tilde{U}_9 = \frac{k_{10}h_{89}h_8 + \overline{k_{10}}k_{89}k_8\tilde{T}}{h_{10}h_{89}h_8}\frac{k_9}{h_9}\tilde{T},\\
\tilde{T} &= 0 \quad \vee \quad \left(\frac{k_{10}h_{89}h_8 + \overline{k_{10}}k_{89}k_8\tilde{T}}{h_{10}h_{89}h_8}\frac{k_9}{h_9}k_2 + \overline{k_2}\frac{k_{510}k_5}{h_{510}h_5}\tilde{U}_{10}\right)\left(1 - \frac{\tilde{T}}{T^0}\right) - h_2 = 0.
\end{aligned}$$

We choose to consider $\tilde{T} = 0$, which results in all variables being equal to 0. That means that our nonlinearized model reduces to Eq. (56) considered earlier in this section and therefore it has the same stability conditions, so

$$\Delta t \leq \min_{i=2,5,8,9,10,11}\left\{\frac{2\left(\Delta x\right)^2}{4D + h_i\left(\Delta x\right)^2}\right\}.$$

6.4 Results

In the previous sections, we described the discretization of Eq. (1), Eq. (15), Eq. (16) and Eq. (18). We created a Python-script that automatically calculates the propagation of these equations. This script can be found in Appendix D.1. In this section, we will discuss the results obtained by the simulation.

First, we will take a look at the results of the one equation model given in Eq. (16). The propagation speeds for different values of D are given in Figure 9. We see that these results are significantly different from the results presented in [1]. Although the propagation speed does increase when the diffusion constant D increases, the (scaled) values are not the same. This could be explained by a discrepancy in the boundary conditions or the initial conditions. If the left-boundary condition, given in Eq. (2a), is bigger, the propagation speed will be higher. Hence, this would suggest that the value of A should be lower than $T_0 = 1400$.

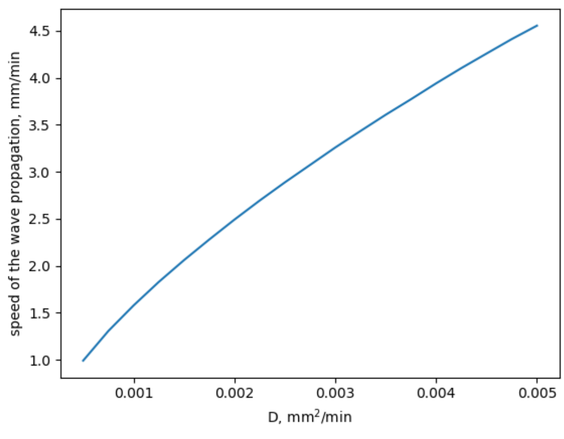


Figure 9: Propagation Speed (mm/min) of the one equation model (16) as a function of D using the parameters as given in Appendix C.

We move on to the simulation of the two equation model represented by Eq. (15). This simulation becomes numerically unstable before it is able to reach an equilibrium for every value of D. The thrombin concentration starts to fluctuate at the left boundary. Then, the simulation becomes unstable. This is most likely related to the left boundary condition of the system given in Eq. (2a). Looking back at the results of the simulation of the one equation model, this could possibly be solved when we use a smaller value for A.

Last, we take a look at the simulation of the full model given by Eq. (1). This simulation becomes unstable when the diffusion coefficient D is bigger than 0.00225. When we compare the propagation speeds up till that specific value of D (given in Figure 10) to the propagation speeds given in [1], we see that the speeds are a lot higher than one expects (about a factor 10). This could be related to using a different value for A in Eq. (2a). However, we also note that the propagation speeds of the full model in Eq. (1) and the one equation model in Eq. (16) are not (approximately) the same. This is interesting, since we simplified Eq. (1) to Eq. (16). Therefore, we conclude that we should have adapted the boundary condition in a suitable way.

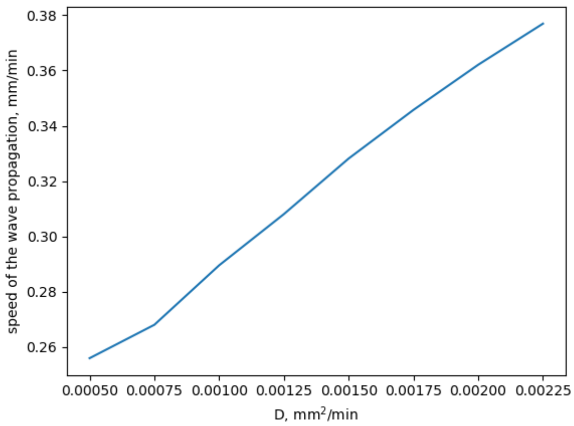


Figure 10: Propagation Speed (mm/min) of the one equation model (1) as a function of D using the parameters as given in Appendix C.

6.4.1 Error Analysis

In order to make sure that the discrepancies are not a consequence of mistake(s) in the script, we compare the results of the space and time integration separately to the theoretical (exact) solution they should have. We start with the space integration. We consider three different boundary conditions (periodic, homogeneous Dirichlet and Neumann). Afterwards, we take a look at the time integration.

Periodic Boundary Conditions We start with a system with periodic boundary conditions. We consider the sine function on the interval $[0, 2\pi)$. Since we have periodic boundary conditions, the results should be the negative sine on the interval $[0, 2\pi)$, since

$$\frac{\partial^2}{\partial x^2}\sin(x) = -\sin(x).$$

When we look at the results created with the script that can be found in Appendix D.2, we see that this is the case as visible in Figure 11. We see that the error in the second derivative is very small ($\mathcal{O}(10^{-7})$). Therefore, we conclude that the second derivative for the space integration is correctly implemented in this simulation.

Homogeneous Dirichlet Boundary Conditions Next, we consider a system with homogeneous Dirichlet boundary conditions. Again we consider the sine function on the interval $[0, 2\pi)$. The results of the second order derivative (created with the script in Appendix D.3) are given in Figure 12. We see that the results are very similar to the results when we used the periodic boundary conditions. However, we note that the errors at the boundary are bigger. This is caused because we give the second order derivative some restrictions at the boundary. Therefore, it will behave differently at the boundary, which is depicted in the middle plot of Figure 12. We conclude that this is implemented correctly as well.

Neumann Boundary Conditions We consider a system with Neumann Boundary Conditions. In our test, we have a homogeneous Neuman boundary condition at the right boundary, and a nonhomogeneous one at the left boundary, given by

$$\left.\frac{\partial f(x)}{\partial x}\right|_{x=0} = -1, \qquad \left.\frac{\partial f(x)}{\partial x}\right|_{x=2\pi} = 0. \tag{58}$$

We calculate the results for the sine $f(x) = \sin x$ using the script in Appendix D.4. The results are as given in Figure 13. We see that the values at the boundary result in discontinuity of the solution. This can be explained by looking at the first order space derivative. For both sides of the interval, these values are different from the values that we force them to be. Under normal circumstances, we would have $f'(0) = 1$ and $f'(2\pi) = -1$, but now, the boundary conditions are determined by Eq. (58). We see that Forward Euler time integration of the

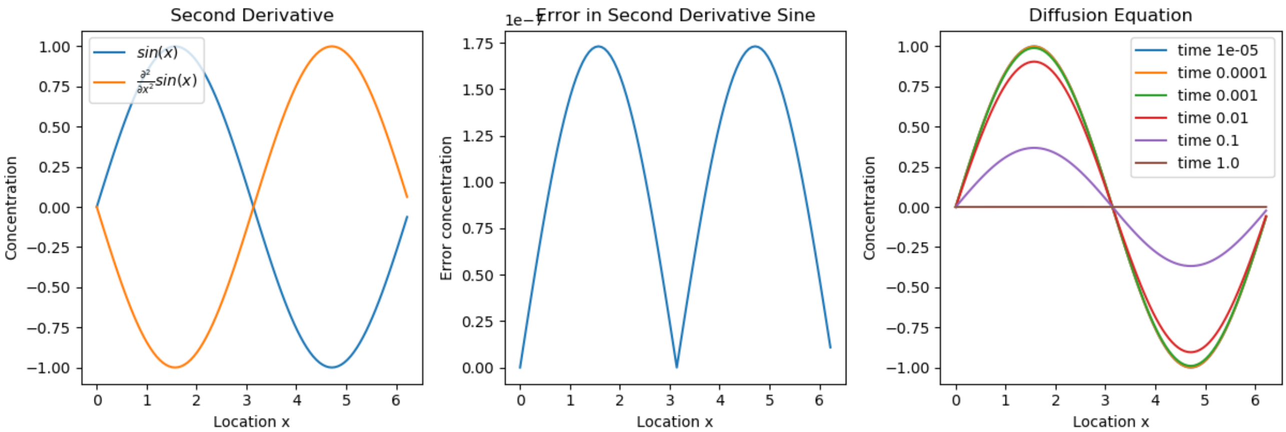


Figure 11: In this figure, we see the results for periodic boundary conditions. In the left figure, the sine and its plotted second derivative are visible. In the middle, one can see the error between the calculated result and the exact result. At the right, we see that the diffusion equation goes to zero if we use Forward Euler for time integration with a diffusion coefficient $D = 10$.

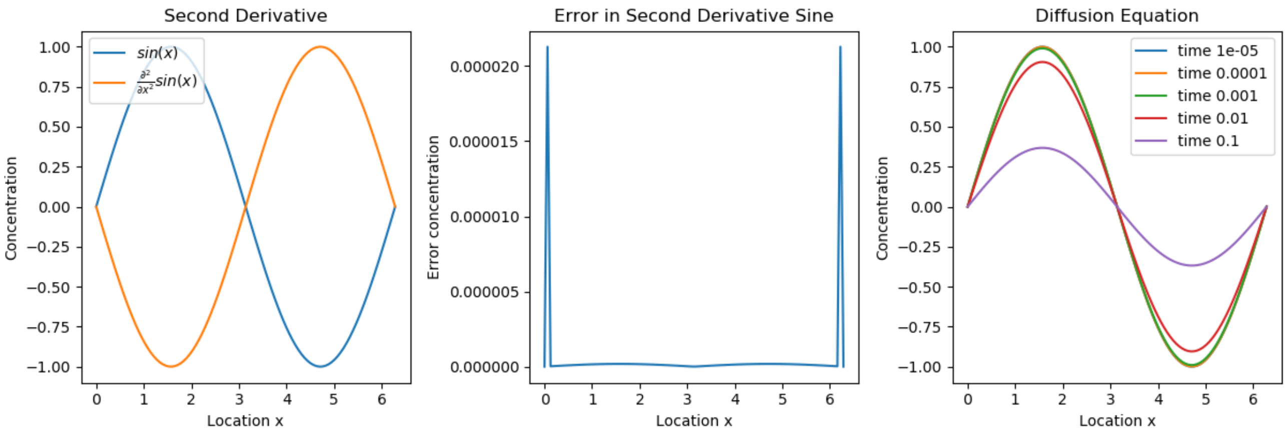


Figure 12: In this figure, we see the results for homogeneous Dirichlet Boundary Conditions. In the left figure, the sine and its plotted second derivative are visible. In the middle, one can see the error between the calculated result and the exact result. At the right, we see that the diffusion equation goes to zero if we use Forward Euler for time integration with a diffusion coefficient $D = 10$.

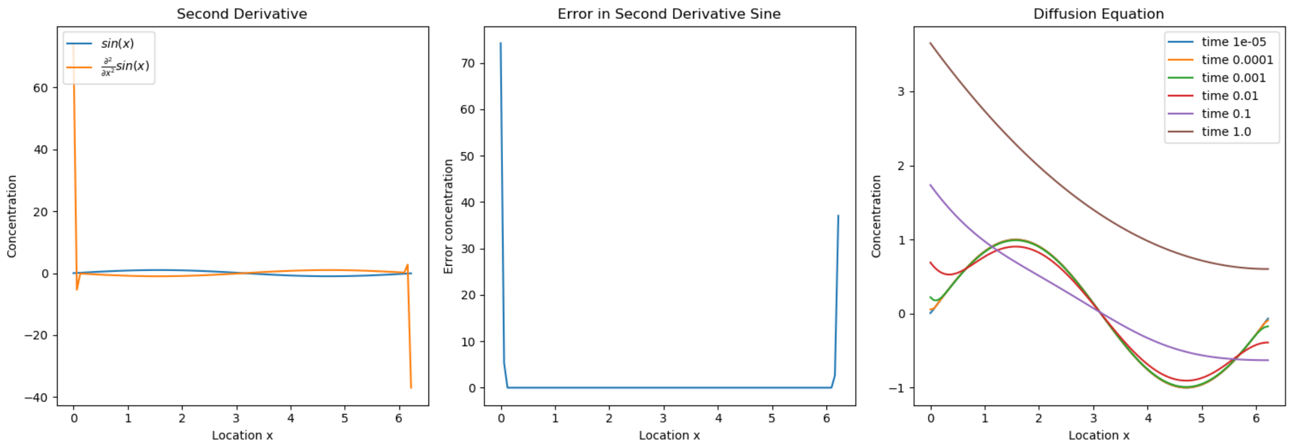


Figure 13: In this figure, we see the results for Neumann boundary conditions as given in Eq. 58. In the left figure, the sine and its plotted second derivative are visible. In the middle, one can see the error between the calculated result and the exact result. At the right, we see that the diffusion equation goes to zero if we use Forward Euler for time integration with a diffusion coefficient $D = 10$.

diffusion equation results in a line that satisfies the boundary conditions. The solution did not converge because of the time integration method.

Merson's Time Integration Last, we consider Merson's time integration separately. We consider two different functions. First, we take a look at the results of the function $y(t) = 1$. When we calculate the result of this first order integration, we should find

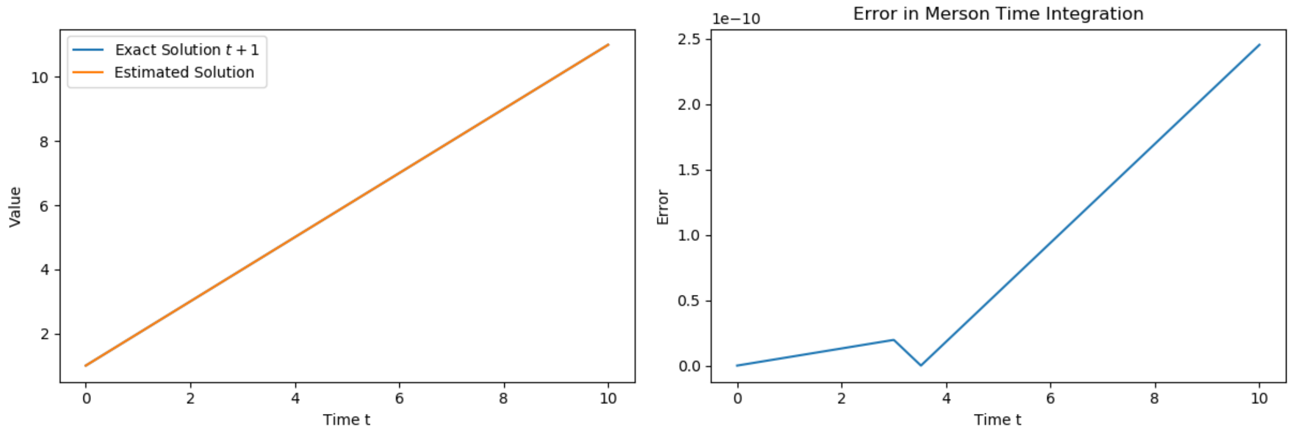
$$\int y(t)\mathrm{d}t = \int 1\mathrm{d}t = t + C.$$

We calculate the results with initial condition $y(0) = 1$. Therefore, we have $C = 1$. Then, the simulation with the script as in Appendix D.5 finds the results as given in Figure 14a. We note that the error is very small ($\mathcal{O}(10^{-10})$). The error grows over time. This is in line with our expectations.

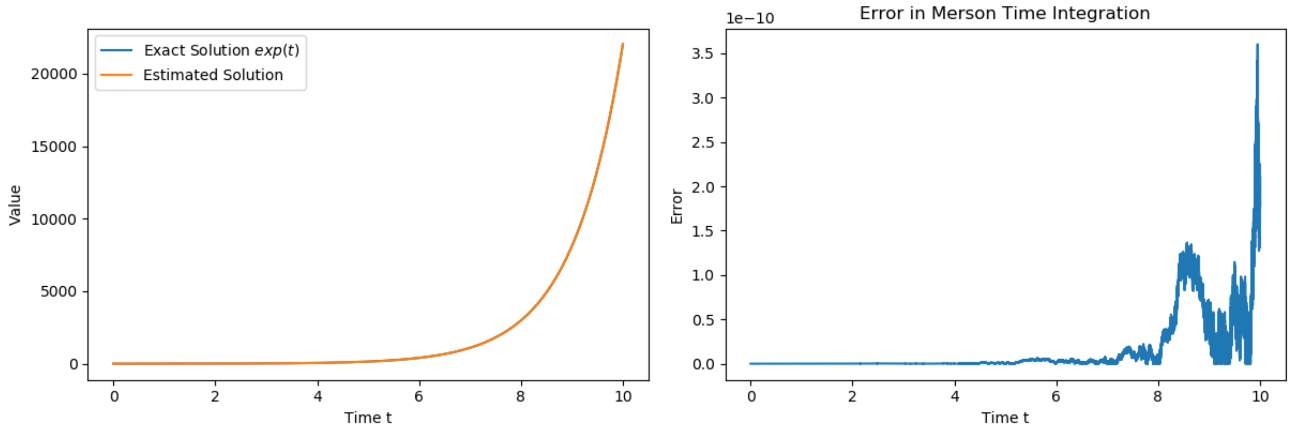
In addition to the constant function $y(t) = 1$, we also consider the exponential function $y(t) = \exp(t)$. The exact solution of one step of time integration, is given by

$$\int y(t)\mathrm{d}t = \int \exp(t)\mathrm{d}t = \exp(t) + C.$$

In the simulation we chose $y(0) = 1$, and therefore, we have $C = 0$. The results of the simulation are given in Figure 14b. The error grows over time. This is what we expect since we only consider the previous value (and some intermediate values) when we calculate the next value. The error is of order $\mathcal{O}(10^{-10})$.



(a) $y(t) = 1$



(b) $y(t) = \exp(t)$

Figure 14: Merson time integration for two different functions. On the left side, both the exact and estimated solution are plotted. At the right side, the error in the time integration in depicted.

7 Conclusion

In this report, we discussed the model of blood coagulation as given in Eq. (1). In Section 3, we proved that there exists a unique solution to the system given by Eq. (1). In addition to that, we also showed that this solution is stable. Since the solution reaches it equilibrium in a wave-like manner, we were interested in the speed of wave propagation. In order to find an estimate for this propagation speed, we simplified the model to a one equation model given by Eq. (16). In Section 4, we proved that the propagation speed of Eq. (16) is approximately the same as the propagation speed of Eq. (15). Therefore, the estimates calculated in Section 5.3 are approximations of the propagation speed of the full model given by Eq. (1) as well. However, the approximated propagation speeds do not give real values as we saw in Section 5.3.3 (in contrast to what is suggested by [1]). Therefore, further research is needed on this topic. In Section 6, we created a Python simulation for the blood coagulation model. We noted that the program is not always able to calculate the propagation speed of the system, since it becomes unstable for some of the models. Therefore, further research is needed on why the simulation becomes unstable and what the propagation speeds of the systems should be.

8 Future Research

In this report, we mainly focused on recreating the same results that were achieved by Galochkina et al. [1]. However, there were some results that we were not able to reproduce exactly the same. Therefore, it is important to try to identify why the estimated propagation speeds calculated in Section 5.3.3 are imaginary. While looking into the differences in the estimated propagation speeds, it is also advised to look into more detail to the narrow reaction zone method in Section 5.3.1 and specifically to why the boundaries of the integral in Eq. (25) can be substituted by the values of the limits given in Eq. (26) as suggested by Vitaly Volpert [7]. In addition to that, one should identify why the simulation discussed in Section 6 becomes unstable for some of the systems.

One of the things that would be interesting to look into beyond reproducing the results of Galochkina et al. [1], is looking into the possibilities of making a generalization of the model. The model that is considered in the paper written by Galochkina et al. [1] is reduced to a one-equation model. In order to achieve that, a lot of simplifications were made. It would be interesting to look into the possibilities of finding expressions for the theoretical wave-speed of the two equation model or even the full model.

It would be interesting to see how big the influence of different blood coagulation factors in diseases is. In Appendix A one can find an overview of different diseases and the influence they should have on the blood coagulation process. It would be interesting to try to find the critical values for which the blood coagulation is not (properly) activated, and when the time it takes to activate is too long. Here, one can look into how much different (de)activation factors can differ from the used values given in Appendix C. In addition to that, one can attempt to find the critical values for which a disease becomes fatal after making sure that there are no calculation errors in the methods and simulation used to find the results. This can then be translated back to the diseases, and one can derive when a disease is fatal for the person suffering from it. It would also be possible to see which influence different treatments would have, and how one can solve the problems arising around different diseases.

References

[1] Tatiana Galochkina, Anass Bouchnita, Polina Kurbatova, and Vitaly Volpert. Reaction-diffusion waves of blood coagulation. *Mathematical Biosciences*, 288:130 – 139, 2017.

[2] Arthur C. Guyton and John E. Hall. *Textbook of Medical Physiology*, chapter Hemostasis and Blood Coagulation, pages 457–468. Elsevier Saunders, 1600 John F. Kennedy Blvd., Suite 1800; Philadelphia, Pennsylvania, 11 edition, 2006.

[3] Stephanie A. Smith, Richard J. Travers, and James H. Morrissey. How it all starts: Initiation of the clotting cascade. *Critical Reviews in Biochemistry and Molecular Biology*, 50(4):326–336, 2015. PMID: 26018600.

[4] V.I. Zarnitsina, A.V. Pokhilko, and F.I. Ataullakhanov. A mathematical model for the spatio-temporal dynamics of intrinsic pathway of blood coagulation. i. the model description. *Thrombosis Research*, 84(4):225 – 236, 1996.

[5] V.I. Zarnitsina, A.V. Pokhilko, and F.I. Ataullakhanov. A mathematical model for the spatio-temporal dynamics of intrinsic pathway of blood coagulation. ii. results. *Thrombosis Research*, 84(5):333 – 344, 1996.

[6] Donghui Zhu. Mathematical modeling of blood coagulation cascade: kinetics of intrinsic and extrinsic pathways in normal and deficient conditions. *Blood Coagulation & Fibrinolysis*, 18(7):637 – 646, 2007.

[7] Vitaly Volpert. *Methods of Analysis*, pages 70,202–220,294–295. Springer Basel, Basel, 2014.

[8] David Angeli. *Monotone Systems in Biology*, pages 769–776. Springer London, London, 2015.

[9] Ernst Hairer, Gerhard Wanner, and Syvert P Nørsett. *Solving Ordinary Differential Equations I*, pages 134–135, 167. Springer, Berlin, Heidelberg, Berlin, 1993.

[10] Lucy A Norris. Blood coagulation. *Best Practice & Research Clinical Obstetrics & Gynaecology*, 17(3):369 – 383, 2003. Inherited Thrombophilias.

[11] Jerrold Levy, Roman Sniecinski, and Ian Welsby. *Blood and Coagulation*, pages 837–848. Elsevier, 01 2019.

A Diseases

Since there are a lot of different factors that play a role in the blood coagulation process, there are numerous things that can go wrong. Either the blood clotting process does not start properly, and therefore does not manage to stop the bleeding (in time). An alternative would be when the blood coagulation process is triggered without a valid reason, resulting in thrombosis. We divided the diseases in two different categories.

A.1 Thrombosis

This is when the blood clot formation process is started without a clear reason. It can have different causes such as loss of the vascular wall integrity, an increased tissue factor expression, increased levels (or activity) of factor VII. [3] One is at risk if they have increased levels of factor XI, or when there are factor V mutations (which are an inherited risk factor for thrombosis). [10] When the antithrombin concentration is concentration is less than 60 to 70% it is associated with thrombosis. [11, 10]

Elevated levels of factor VII may be a risk factor for thrombotic diseases, but not all studies agree on this. There are mixed results of the correlation between factor VIIa and the risk of thrombotic diseases. [3]

A deficiency of factor XII is protective against thrombus formation in arteries and veins (in animal models) and protects them from ischemic brain injury (cause by lack of blood and oxygen reaching the brain). [3]

Atherosclerosis Blood is separated from the tissue factor by a thin monolayer of endotheliail cells resulting in easy rupturing of the vessel wall. Increased levels of factor XII, factor XI or kallikrein activity are associated with atherosclerosis. A severe factor XI deficiency leads to a reduced risk at a stroke. [3]

Myocardinal Infarction (heart attack) This is triggered by a rupture of an atherosclerotic plaque in the coronary artery. By consequence is TF exposed to factor VII/VII(a) within the blood. When the coagulation activation is extensive enough, this leads to occlusive thrombosis within the coronary vessel, and a heart attack follows. Increased levels of factor XII, factor XI or kallikrein activity are associated with myocardinal infarction. [3]

Trousseau Syndrome This is also called ‘cancer-associated thrombosis’. Sometimes, a cancer cell itself can express tissue factor which leads to activation of the blood coagulation system. [3]

Sepsis The contact pathway is (unnecessarily) activated when an individual suffers from sepsis. This leads to a continuous generation of factor XIIa and kallikrein which can deplete the zymogen levels. [3]

Lack of Factor XII Individuals suffering from this disease have no bleeding tendencies. [3]

Systemic Amyloidosis This leads to increased *in vivo* activation of factor XII and prekallikrein, and therefore increased risks for thrombosis. [3]

Disseminated Intravascular Coagulation This is often the result of the presence of large amounts of traumatized or dying tissue (which means that there is a lot of Tissue Factor). Therefore it leads to often small, but numerous clots. [2]

Homozygous Protein C Deficiency This leads to life-threatening thrombotic complications, often occurring immediately after birth. [10]

Deficiency in Protein S About 2 to 5% of the patients with a history in thrombosis suffers from this condition. There are three different types. Type I means that there is a reduction in free and total protein S. Type II means that the protein S antigen levels are normal, but their functional activity is reduced. Type III means that the protein S levels are normal, but there is a reduction i the free unbound protein S only. [10]

A.2 Excessive Bleeding

Vitamin K Deficiency or Liver-Disease Without vitamin K, the factors synthesized in the liver are not functional (II, VII, IX, X, fibrinogen (I)). All these factors undergo vitamin K-dependent carboxylation. This results in a bleeding tendency. [2]

Hemophilia Individuals suffering from hemophilia have a lack of factor VIII or factor IX. Therefore, they do not tend to bleed too much when they have a superficial cut where a lot of tissue factor is present, but a lot at places where less TF is present (because of the extra activation step needed for factor X to be activated. [3]

Hemophilia A This disease only occurs in males because it is X-linked and is also referred to as classic hemophilia. [2] It is caused by a factor IX deficiency and is the most common form of hemophilia. [11]

Hemophilia B This diseases causes a deficiency in factor IX. It results in about ten times lower levels of factor VIIa in plasma. Therefore, it suggests that factor IX contributes to the activation of factor VII *in vivo*. The concentration of factor VIIa increases in a factor IX-dependent manner, especially after eating fatty meals. [3]

von Willebrand Disease (vWD) This is a rare bleeding disorder that can lead to unforeseen bleeding in surgical patients. There are three different types that all have different causes of the defects in the von Willebrand Factor, resulting in different treatments. [11]

Factor VII deficiency This is very rare, but it leads to severe bleeding. [10]

Alzheimer's disease Individuals suffering from Alzheimer's disease have an increased generation of factor XIIa, particularly in the central nervous system. This makes them more vulnerable for excessive bleeding (in these specific areas). [3]

Hereditary Angioedema This disease leads to intermittent episodes of edema and pain due to a dysregulation of the contact pathway. It is usually cause by a deficiency of C1-inhibitor. [3]

Absence of Calcium Ions This leads to no blood clotting. In the living body, this seldom occurs.

(Idiopathic) Thrombocytopenia This is caused when the number of platelets in the blood is very low. It leads to many hemorrhages throughout the body. Failure of clot retraction is an indication that the number of placelets is too low. [2]

B Existence of Waves for Monotone Systems

In the proofs provided in Section 3.2 the following two theorems are used.

Theorem B.1. *[7] Consider the function $F(w)$ such that*

$$F(w) > 0, w_+ < w < w_-.$$

Then there exists a minimal speed c_0 such that for all $c \geq c_0$ there exist monotonically decreasing solutions $w(x)$ of equation

$$w'' + cw' + F(w) = 0, \quad \lim_{x \to \pm\infty} w(x) = w_\pm, \ w_+ < w_-.$$

Such solutions do not exist for $c < c_0$.

Theorem B.2. *[7] Suppose that $F(w_+) = F(w_-) = 0$, where $w_+ < w_-$ (the inequality is component-wise) and the matrices $F'(w_\pm)$ have all eigenvalues in the left half-plane. If there exists a finite number of points $w^j \neq w_\pm$, $j = 1, \ldots, k$ such that $w_+ \leq w^j \leq w_-$ and each matrix $F'(w^j)$ has at least one eigenvalue in the right half-plane, then there exists a unique monotonically decreasing traveling wave solution $u(x,t) = w(x - ct)$ of system (1) with the limits $w(\pm\infty) = w_\pm$. Its velocity admits the following minimax representation*

$$c = \inf_{\rho \in K} \sup_{x,i} \frac{\rho_i'' + F_i(\rho)}{-\rho'} = \sup_{\rho \in K} \inf_{x,i} \frac{\rho_i'' + F_i(\rho)}{-\rho'},$$

where K is the class of monotonically decreasing vector-functions ρ continuous with their second derivatives and having limits $\rho(\pm\infty) = w_\pm$ at infinity.

C Parameter Values Used for the Simulations

The parameter values are given in Table 3.

Parameter	Value	Units
k_{11}	$1.1 \cdot 10^{-5}$	min^{-1}
h_{11}	0.5	min^{-1}
k_{10}	$3.3 \cdot 10^{-4}$	min^{-1}
$\overline{k_{10}}$	500	min^{-1}
h_{10}	1	min^{-1}
k_9	20	min^{-1}
h_9	0.2	min^{-1}
k_{89}	100	$\text{nM}^{-1}\text{min}^{-1}$
h_{89}	100	min^{-1}
k_8	$1 \cdot 10^{-5}$	min^{-1}
h_8	$3.1 \cdot 10^{-1}$	min^{-1}
k_5	$1.7 \cdot 10^{-1}$	min^{-1}
h_5	$3.1 \cdot 10^{-1}$	min^{-1}
k_{510}	100	$\text{nM}^{-1}\text{min}^{-1}$
h_{510}	100	min^{-1}
k_2	2.45	min^{-1}
$\overline{k_2}$	$2 \cdot 10^3$	min^{-1}
h_2	1.45	min^{-1}
D	$3.7 \cdot 10^{-3}$	$\text{mm}^2\text{min}^{-1}$
T_0	1400	nM

Table 3: Parameter Values Used for the Simulation [1].

D Script Simulation

D.1 Script Propagation of Eq. (1), Eq. (15), Eq. (16) and Eq. (18)

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.sparse import diags
import scipy.sparse as sp
import matplotlib.animation as animation
import time
from sys import exit

nreq = 1                          # determine which model we are considering (either 1,2 or 6)
dimensions = 1                    # 0 dimensionless, 1 dimensions

# introducing Merson's coefficients
cM = np.array([0,1/3,1/3,1/2,1])
bM = np.array([1/6,0,0,2/3,1/6])
bM = bM[:,np.newaxis]
aM = np.array([[0,0,0,0,0],[1/3,0,0,0,0],[1/6,1/6,0,0,0],[1/8,0,3/8,0,0],[1/2,0,-3/2,2,0]])
s = 5 #order of Merson's Method

# introducing 5-point scheme for space-derivative
def der2_5pointScheme(deltax,D,A):
    # boundary conditions
    BC = np.zeros((nreq,m))
    if nreq == 6:
        a=-A
        startBC0 = np.array([0,0,0,0,0,-28*deltax*a])
        startBC1 = np.array([0,0,0,0,0,2*deltax*a])
        endBC0 = np.array([0,0,0,0,0,0])
        endBC1 = np.array([0,0,0,0,0,0])
    elif nreq == 2:
        a=-A
        startBC0 = np.array([0,-28*deltax*a])
        startBC1 = np.array([0,2*deltax*a])
        endBC0 = np.array([0,0])
        endBC1 = np.array([0,0])
    elif nreq == 1:
        a=-A
        startBC0 = -28*a*deltax
        startBC1 = 2*a*deltax
        endBC1 = 0
        endBC0 = 0
    BC[:,0]=startBC0
```

```
    BC[:,1]=startBC1
    BC[:,-2]=endBC1
    BC[:,-1]=endBC0
    BC = D*BC/(12*deltax**2)

    # create matrix
    der2mat = diags([-1,16,-30,16,-1],[-2,-1,0,1,2],shape=(m,m))
    der2mat = sp.csr_matrix(der2mat)
    der2mat[0,1]=32
    der2mat[0,2]=-2
    der2mat[1,1]=-31
    der2mat[m-2,m-2]=-31
    der2mat[m-1,m-3]=-2
    der2mat[m-1,m-2]=32

    der2mat2 = D*sp.kron(np.eye(nreq),der2mat)/(12*deltax**2)

    BC = np.reshape(BC,(-1,1))
    print("A =",A)
    return der2mat2,BC.T

# introducing function to calculate the nonlinear part of the system
def fnonlin(time,tn,yn,matrix,boundaryconditions,m):
    y=matrix*yn+boundaryconditions+nonlin(tn,yn,matrix,boundaryconditions,m)
    return y

# introducing function to calculate Merson's time integration
def Merson(f,tn,deltat,w,matrix,i,boundaryconditions,m):
    kM = np.zeros((s,nreq*m))
    kM[0] = f(tn,0,w,matrix,boundaryconditions,m)
    kM[1] = f(tn,deltat*cM[1],w+deltat*aM[1,0]*kM[0],matrix,boundaryconditions,m)
    kM[2] = f(tn,deltat*cM[2],w+deltat*(aM[2,0]*kM[0]+aM[2,1]*kM[1]),matrix,boundaryconditions,m)
    kM[3] = f(tn,deltat*cM[3],w+deltat*(aM[3,0]*kM[0]+aM[3,1]*kM[1]+aM[3,2]*kM[2]),matrix,boundaryconditio
    kM[4] = f(tn,deltat*cM[4],w+deltat*(aM[4,0]*kM[0]+aM[4,1]*kM[1]+aM[4,2]*kM[2]+aM[4,3]*kM[3]),matrix,bo
    w = w+deltat*np.sum(bM*kM, axis=0)
    return w

# introducing function to calculate at which location the value h is passed
def hval(u,h):
    hsmall = np.where(u>=h)
    if len(hsmall[0])!=0:
        hlocmin = hsmall[0][-1]
        hloc = np.array([hlocmin, hlocmin+1])
    else:
        hloc = np.zeros((1,2))
    return hloc

# introducing function to calculate the "exact" location at which value h is passed
def exloc(u,height,h,xax,m): #exact location when assuming points are connected linearly
    h1 = int(height[0])
    h2 = int(height[1])
    if h1 == h2 or height[1]>=m:
        x = 0
    else:
        b = (h-u[h1])/(u[h2]-u[h1])
        a = 1-b
        x = a*xax[h1]+b*xax[h2]
    return x

# define model coefficients
k11  = 1.1*10**(-5)        # 1/min
h11  = 0.5                 # 1/min
k10  = 3.3*10**(-4)        # 1/min
lk10 = 500.0               # 1/min          l for overline
h10  = 1.0                 # 1/min
k9   = 20.0                # 1/min
h9   = 0.2                 # 1/min
k89  = 100.0               # 1/(nM min)
h89  = 100.0               # 1/min
k8   = 1*10**(-5)          # 1/min
h8   = 0.31                # 1/min
k5   = 0.17                # 1/min
h5   = 0.31                # 1/min
k510 = 100.0               # 1/(nM min)
h510 = 100.0               # 1/min
k2   = 2.45                # 1/min
```

```
lk2   = 2000.0                       # 1/min
h2    = 1.45                         # 1/min
K2m   = 58.0                         # nM
lK2m  = 210.0                        # nM
T0    = 1400.0                       # nM
A     = T0                           # activation level at point x=0 of XIa

endp=True
if nreq == 1:
    m = 150
elif nreq == 2:
    m = 150
elif nreq == 6:
    m = 150
maxIter = 6000000
endpoint = 2
x = np.linspace(0,endpoint,num=m,endpoint=endp)
scaling = 100
deltat = 10**(-5)
Diter = 19

# define empty arrays for propagation speed and narrow reaction zone speed
propspeed = np.zeros((Diter,1))
NarrowReactionZoneSpeed = np.zeros((Diter,1))

# introducing nonlinear parts for the models (both dimensional and dimensionless)
if dimensions == 0:
    h = 0.5
    Dmin = 0.5*10**(-3)/h2
    Dmax = 5*10**(-3)/h2
    if nreq == 1:
        M1 = k9*k11/(h2*h9*h10*h11)
        M2 = k8*k89*lk10*T0/(h8*h89)
        M3 = lk2*k5*k510*T0/(h5*h510)
        b = M1*M2*M3
        def nonlin(tn,u,matrix,boundaryconditions,m):
            values = b*(u**3)*(1-u)-u
            return values

        c1 = k9*k11/(h9*h10*h11*h2)
        c2 = lk10*k89*k8*T0/(h89*h8)
        c3 = lk2*k510*k5*T0/(h510*h5)
        constant1 = c1*k10*k2-1
        constant2 = c1*c2*k2+c1*k10*c3-c1*k10*k2
        constant3 = c1*c2*c3-c1*c2*k2-c1*k10*c3
        constant4 = -c1*c2*c3
        def nonlin_2(tn,u,matrix,boundaryconditions,m):
            T1 = u
            T2 = T1*u.astype(np.float64)
            T3 = T2*u.astype(np.float64)
            T4 = T3*u.astype(np.float64)
            values = constant1*T1\
                     +constant2*T2\
                     +constant3*T3\
                     +constant4*T4
            return values
else:
    h = 700
    Dmin = 0.5*10**(-3)
    Dmax = 5*10**(-3)
    if nreq == 1:
        c1 = k9*k11/(h9*h10*h11)
        c2 = lk10*k89*k8/(h89*h8)
        c3 = lk2*k510*k5/(h510*h5)
        constant1 = c1*k10*k2-h2
        constant2 = c1*c2*k2+c1*k10*c3-c1*k10*k2/T0
        constant3 = c1*c2*c3-c1*c2*k2/T0-c1*k10*c3/T0
        constant4 = -c1*c2*c3/T0
        def nonlin(tn,u,matrix,boundaryconditions,m):
            T1 = u
            T2 = T1*u.astype(np.float64)
            T3 = T2*u.astype(np.float64)
            T4 = T3*u.astype(np.float64)
            values = constant1*T1\
                     +constant2*T2\
                     +constant3*T3\
```

```
                        +constant4*T4
            return values
    elif nreq == 2:
        c1 = k9/(h9*h10)
        c2 = lk10*k89*k8/(h89*h8)
        c3 = lk2*k510*k5/(h510*h5)
        def nonlin(tn,u,matrix,boundaryconditions,m):
            # linear part
            t1 = u[:m]
            u1 = u[m:]
            nonlinT = np.zeros(u.shape)
            nonlinT[:m] = c1*u1*(k10+c2*t1)*(k2+c3*t1)*(1-t1/T0)-h2*t1
            nonlinT[m:] = k11*t1-h11*u1
            return nonlinT
    elif nreq == 6:
        c1 = lk10*k89/h89
        c2 = lk2*k510/h510
        c3 = k2/T0
        c4 = c2/T0
        def nonlin(tn,u,matrix,boundaryconditions,m):
            # linear part
            t = u[:m]
            u5 = u[m:2*m]
            u8 = u[2*m:3*m]
            u9 = u[3*m:4*m]
            u10 = u[4*m:5*m]
            u11 = u[5*m:]
            nonlinT = np.zeros(u.shape)
            nonlinT[:m] = k2*u10+c2*u10*u5-c3*u10*t-c4*u10*u5*t-h2*t
            nonlinT[m:2*m] = k5*t-h5*u5
            nonlinT[2*m:3*m] = k8*t-h8*u8
            nonlinT[3*m:4*m] = k9*u11-h9*u9
            nonlinT[4*m:5*m] = k10*u9+c1*u9*u8-h10*u10
            nonlinT[5*m:] = k11*t-h11*u11
            return nonlinT

# introducing function to calculate the narrow reaction zone approximation of the propagation speed
def NarrowReactionZone(D,k9,b,sigma,T):
    A = 4*b*D*(T**2/4-T**3/5)
    speed = (A-2*D*sigma)/np.sqrt(2*A)
    return speed

# introducing function to calculate the piecewise linear approximation of the propagation speed
def PiecewiseLinearApproximation(D,k9):
    wstar = 0.00436951
    n=3
    sigma = 1
    alpha = -sigma
    beta = b*n*wstar**(n-1)-b*(n+1)*wstar**n-sigma
    r = b*wstar**(n+1)*(-n/2-b/(n+1))+b*wstar**(n+2)*((n+1)/2+1/(n+2))+sigma*wstar**2
    w0 = (-beta*wstar+np.sqrt((beta*wstar)**2-2*(alpha-beta)*r))/(alpha-beta)
    lw = w0/(w0-wstar)
    speed = np.sqrt(D[:,np.newaxis])*(alpha*lw**2-beta)/np.sqrt((lw-1)*(alpha*lw**2-beta*lw))
    return speed

# introducing counter to determine at which locations in arrays specific data needs to be saved
j=0

# calculate the model for different values of D
for D in np.linspace(Dmin,Dmax,num=Diter,endpoint=True):
    # define empty height and heightloc arrays to calculate the propagation speed
    height = np.zeros((int(maxIter/scaling)+1,2))
    heightloc = np.zeros((int(maxIter/scaling)+1,1))

    # initial distribution of w (for the different models)
    if nreq == 1:
        w = 0*np.ones(x.shape)
    elif nreq == 2:
        T = 0*np.ones(x.shape)                              #initial condition for T
        U11 = 0*np.ones(x.shape)                            #initial condition for U11
        w = np.concatenate((T,U11),axis=0)
    elif nreq == 6:
        T = 0*np.ones((1,m))                                #initial condition for T
        U5 = 0*np.ones((1,m))                               #initial condition for U5
        U8 = 0*np.ones((1,m))                               #initial condition for U8
        U9 = 0*np.ones((1,m))                               #initial condition for U9
```

```
        U10 = 0*np.ones((1,m))                                   #initial condition for U10
        U11 = 0*np.ones((1,m))                                   #initial condition for U11
        w = np.concatenate((T,U5),axis=1)
        w = np.concatenate((w,U8),axis=1)
        w = np.concatenate((w,U9),axis=1)
        w = np.concatenate((w,U10),axis=1)
        w = np.concatenate((w,U11),axis=1)  #w contains all initial conditions as a long vector
                                            #[T,U5,U8,U9,U10,U11]

    # calculate distance between two consecutive locations in space
    if endp == True:
        deltax = endpoint/(m-1)
    elif endp == False:
        deltax = endpoint/m

    # define array to save the concentrations of the different factors
    allw = np.zeros((int(maxIter/scaling)+1,nreq*m))
    allw[0] = w

    # define matrix and BC for the second order space-derivative
    if dimensions == 0:
        matrix,BC = der2_5pointScheme(deltax,D,1)
    else:
        matrix,BC = der2_5pointScheme(deltax,D,A)

    # print time it takes to calculate 1 minute
    print("min, calculation time")
    totaltime = 0
    start = time.time()
    for i in range(maxIter):
        # calculate next concentration
        allw[int(np.ceil((i+1)/scaling))]=Merson(fnonlin,i*deltat,deltat,allw[int(np.ceil(i/scaling))],mat

        # when 1 minute has passed: print calculation time and stop if all values are bigger than
        # given value h
        if i%int(1/deltat)==0:
            end = time.time()
            print(np.round(i*deltat),end-start)
            if np.any(allw[int(np.ceil((i+1)/scaling)),:m] >= h) == True:
                break
            start = time.time()
            totaltime = totaltime + (end-start)
        # if the concentrations are NaN: stop the calculations
        if np.all(allw[int(np.ceil((i+1)/scaling)),:m]!=allw[int(np.ceil((i+1)/scaling)),:m]):
            print("stop at",i)
            exit("No convergence")
            break

    # calculate the propagation speed
    for i in range(len(allw)):
        #height-vector
        height[i] = hval(allw[i][0:m],h) #locations between which value h is obtained
        heightloc[i] = exloc(allw[i][:,np.newaxis],height[i],h,x,m) #exact location where h is obtained

    propspeed[j] = (np.max(heightloc)-np.min(heightloc))/((np.where(heightloc!=0)[0][-1]-(np.where(heightl

    # calculate narrow reaction zone approximation of the propagation speed
    b=k9*k11/(h9*h10*h11)*lk10*k89*k8/(h89*h8)*lk2*k510*k5/(h510*h5)*T0**2
    NarrowReactionZoneSpeed[j] = NarrowReactionZone(D,k9,b,h2,1.24993)
    j = j+1

# determine whether we are working with D or \tilde{D}
if dimensions == 0:
    D = "Dt"
    propspeed = propspeed#*h2/T0
#    NarrowReactionZoneSpeed = NarrowReactionZoneSpeed*h2/T0
else:
    D = "D"

# make a plot of the propagation speed
plt.figure()
plt.plot(np.linspace(Dmin,Dmax,num=Diter,endpoint=True),propspeed)
#plt.plot(np.linspace(Dmin,Dmax,num=Diter,endpoint=True),NarrowReactionZoneSpeed)
plt.xlabel("{}, mm$^2$/min".format(D))
plt.ylabel("speed of the wave propagation, mm/min")
```

D.2 Script Periodic Boundary Conditions

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.sparse import diags
import scipy.sparse as sp

nreq = 1                          # determine which model we are considering (either 1,2 or 6)

# introducing 5-point scheme for space-derivative
def der2_5pointScheme_PeriodicBC(deltax):
    det2mat = diags([16,-1,-1,16,-30,16,-1,-1,16],[-m+1,-m+2,-2,-1,0,1,2,m-2,m-1],shape=(m,m))
    matrix = sp.kron(np.eye(nreq),det2mat)/(12*deltax**2)
    return matrix

# define model coefficients and initial distribution
endp=False
m = 100
maxIter = 1000000
endpoint = 2*np.pi
x = np.linspace(0,endpoint,num=m, endpoint=endp)
w = np.sin(x)
if endp == True:
    deltax = endpoint/(m-1)
elif endp == False:
    deltax = endpoint/m
deltat = 10**(-5)
D=10

# define array to save the concentrations of the different factors
allw = np.zeros((maxIter+1,m))
allw[0] = w

# define matrix and BC for the second order space-derivative
matrix = der2_5pointScheme_PeriodicBC(deltax)
antwoord = matrix*w

# results of Periodic Boundary Condition
fig,(ax0,ax1,ax2) = plt.subplots(ncols=3,constrained_layout = True)
fig.set_figheight(4)
fig.set_figwidth(12)
plt.suptitle('Periodic Boundary Conditions',fontsize=25)

# second derivative
ax0.set_title('Second Derivative')
ax0.set_ylabel('Concentration')
ax0.set_xlabel('Location x')
ax0.plot(x,w,label='$sin(x)$')
ax0.plot(x,antwoord,label=r'$\frac{\partial^2}{\partial x^2}sin(x)$')
ax0.legend(loc='upper left')

# error of the second derivative
ax1.set_title('Error in Second Derivative Sine')
ax1.set_ylabel('Error concentration')
ax1.set_xlabel('Location x')
ax1.plot(x,np.abs(w+antwoord))

# evolution over time with diffusion coefficient D=10 and forward Euler
ax2.set_title('Diffusion Equation')
ax2.set_ylabel('Concentration')
ax2.set_xlabel('Location x')
for i in range(maxIter):
    if np.log10(i)%1==0:
        ax2.plot(x,allw[i],label="time {}".format(i*deltat))
    allw[i+1] = allw[i]+D*deltat*der2_5pointScheme_PeriodicBC(allw[i].T,deltax)
ax2.legend(loc='upper right')
```

D.3 Script Homogeneous Dirichlet Boundary Conditions

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.sparse import diags
import scipy.sparse as sp

nreq = 1                          # determine which model we are considering (either 1,2 or 6)
```

```
# introducing 5-point scheme for space-derivative
def der2_5pointScheme_zeroBC(deltax):
    der2mat = diags([-1,16,-30,16,-1],[-2,-1,0,1,2],shape=(m,m))
    der2mat = sp.csr_matrix(der2mat)
    der2mat[0,0]=0
    der2mat[0,1]=0
    der2mat[0,2]=0
    der2mat[1,0]=12
    der2mat[1,1]=-24
    der2mat[1,2]=12
    der2mat[1,3]=0
    der2mat[m-2,m-1]=12
    der2mat[m-2,m-2]=-24
    der2mat[m-2,m-3]=12
    der2mat[m-2,m-4]=0
    der2mat[m-1,m-1]=0
    der2mat[m-1,m-2]=0
    der2mat[m-1,m-3]=0
    der2mat2 = sp.kron(np.eye(nreq),der2mat)/(12*deltax**2)
    return der2mat2

# define model coefficients and initial distribution
endp=True
m = 100
maxIter = 1000000
endpoint = 2*np.pi
x = np.linspace(0,endpoint,num=m,endpoint=endp)
w = np.sin(x)
if endp == True:
    deltax = endpoint/(m-1)
elif endp == False:
    deltax = endpoint/m
deltat = 10**(-5)
D=10

# define array to save the concentrations of the different factors
allw = np.zeros((maxIter+1,m))
allw[0] = w

# define matrix and BC for the second order space-derivative
matrix = der2_5pointScheme_zeroBC(deltax)
antwoord = matrix*w

# results of Homogeneous Dirichlet Boundary Condition
fig,(ax0,ax1,ax2) = plt.subplots(ncols=3,constrained_layout = True)
fig.set_figheight(4)
fig.set_figwidth(12)

# second derivative
ax0.set_title('Second Derivative')
ax0.set_ylabel('Concentration')
ax0.set_xlabel('Location x')
ax0.plot(x,w,label='$sin(x)$')
ax0.plot(x,antwoord,label=r'$\frac{\partial^2}{\partial x^2}sin(x)$')
ax0.legend(loc='upper left')

# error of the second derivative
ax1.set_title('Error in Second Derivative Sine')
ax1.set_ylabel('Error concentration')
ax1.set_xlabel('Location x')
ax1.plot(x,np.abs(w+antwoord))

# evolution over time with diffusion coefficient D=10 and forward Euler
ax2.set_title('Diffusion Equation')
ax2.set_ylabel('Concentration')
ax2.set_xlabel('Location x')
for i in range(maxIter):
    if np.log10(i)%1==0:
        ax2.plot(x,allw[i],label="time {}".format(i*deltat))
    allw[i+1] = allw[i]+D*deltat*matrix*allw[i]
ax2.legend(loc='upper right')
```

D.4 Script Neumann Boundary Conditions

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.sparse import diags
import scipy.sparse as sp

nreq = 1                          # determine which model we are considering (either 1,2 or 6)

# introducing 5-point scheme for space-derivative
def der2_5pointScheme(deltax,A):
    # boundary conditions
    BC = np.zeros((nreq,m))
    if nreq == 6:
        startBC0 = np.array([0,0,0,0,0,-28*deltat*D/deltax*A])
        startBC1 = np.array([0,0,0,0,0,2*deltat*D/deltax*A])
        endBC0 = np.array([0,0,0,0,0,0])
        endBC1 = np.array([0,0,0,0,0,0])
    elif nreq == 2:
        startBC0 = np.array([0,-28*deltax*A])
        startBC1 = np.array([0,2*deltax*A])
        endBC0 = np.array([0,0])
        endBC1 = np.array([0,0])
    elif nreq == 1:
        startBC0 = -28*A*deltax
        startBC1 = 2*A*deltax
        endBC1 = 0
        endBC0 = 0
    BC[:,0]=startBC0
    BC[:,1]=startBC1
    BC[:,-2]=endBC1
    BC[:,-1]=endBC0
    BC = BC/(12*deltax**2)

    # create matrix
    der2mat = diags([-1,16,-30,16,-1],[-2,-1,0,1,2],shape=(m,m))
    der2mat = sp.csr_matrix(der2mat)
    der2mat[0,1]=32
    der2mat[0,2]=-2
    der2mat[1,1]=-31
    der2mat[m-2,m-2]=-31
    der2mat[m-1,m-3]=-2
    der2mat[m-1,m-2]=32

    der2mat2 = sp.kron(np.eye(nreq),der2mat)/(12*deltax**2)

    BC = np.reshape(BC,(-1,1))
    return der2mat2,BC.T

# define model coefficients and initial distribution
endp=False
m = 100
maxIter = 1000000
endpoint = 2*np.pi
x = np.linspace(0,endpoint,num=m,endpoint=endp)
w = np.sin(x)
if endp == True:
    deltax = endpoint/(m-1)
elif endp == False:
    deltax = endpoint/m
deltat = 10**(-5)
D=10

# define array to save the concentrations of the different factors
allw = np.zeros((maxIter+1,m))
allw[0] = w

# define matrix and BC for the second order space-derivative
matrix,BC = der2_5pointScheme(deltax,-1)

# second order derivative of w with boundary conditions
antwoord = (matrix*w+BC)[0]

# results of Neumann Boundary Condition
fig,(ax0,ax1,ax2) = plt.subplots(ncols=3,constrained_layout = True)
fig.set_figheight(4)
fig.set_figwidth(12)
```

```
# second derivative
ax0.set_title('Second Derivative')
ax0.set_ylabel('Concentration')
ax0.set_xlabel('Location x')
ax0.plot(x,w,label='$sin(x)$')
ax0.plot(x,antwoord,label=r'$\frac{\partial^2}{\partial x^2}sin(x)$')
ax0.legend(loc='upper left')

# error of the second derivative
ax1.set_title('Error in Second Derivative Sine')
ax1.set_ylabel('Error concentration')
ax1.set_xlabel('Location x')
ax1.plot(x,np.abs(w+antwoord))

# evolution over time with diffusion coefficient D=10 and forward Euler
ax2.set_title('Diffusion Equation')
ax2.set_ylabel('Concentration')
ax2.set_xlabel('Location x')
for i in range(maxIter):
    if np.log10(i)%1==0:
        ax2.plot(x,allw[i],label="time {}".format(i*deltat))
    allw[i+1] = allw[i]+D*(matrix*allw[i]+BC)*deltat
ax2.legend(loc='upper right')
```

D.5 Script Merson's Time Integration

```
import numpy as np
import matplotlib.pyplot as plt

nreq = 1                          # determine which model we are considering (either 1,2 or 6)

# introducing Merson's coefficients
cM = np.array([0,1/3,1/3,1/2,1])
bM = np.array([1/6,0,0,2/3,1/6])
bM = bM[:,np.newaxis]
aM = np.array([[0,0,0,0,0],[1/3,0,0,0,0],[1/6,1/6,0,0,0],[1/8,0,3/8,0,0],[1/2,0,-3/2,2,0]])
s = 5 #order of Menson's Method

# introducing two different function on which we will apply time integration
def f(tn,yn,matrix,boundaryconditions):
    y=np.exp(tn)
    return y

def f2(tn,yn,matrix,boundaryconditions):
    y=1
    return y

# define model coefficients
k11  = 1.1*10**(-5)               # 1/min
h11  = 0.5                        # 1/min
k10  = 3.3*10**(-4)               # 1/min
lk10 = 500                        # 1/min           l for overline
h10  = 1                          # 1/min
k9   = 20                         # 1/min
h9   = 0.2                        # 1/min
k89  = 100                        # 1/(nM min)
h89  = 100                        # 1/min
k8   = 1*10**(-5)                 # 1/min
h8   = 0.31                       # 1/min
k5   = 0.17                       # 1/min
h5   = 0.31                       # 1/min
k510 = 100                        # 1/(nM min)
h510 = 100                        # 1/min
k2   = 2.45                       # 1/min
lk2  = 2000                       # 1/min
h2   = 1.45                       # 1/min
K2m  = 58                         # nM
lK2m = 210                        # nM
D    = 3.7*10**(-3)               # mm^2/min
T0   = 1400                       # nM
A    = T0                         # activation level at point x=0 of XIa


# introducing function to calculate Merson's time integration
def Merson(f,tn,deltat,w,matrix,i,boundaryconditions,m):
```

```
    kM = np.zeros((s,nreq*m))
    kM[0] = f(tn,w,matrix,boundaryconditions)
    kM[1] = f(tn+deltat*cM[1],w+deltat*aM[1,0]*kM[0],matrix,boundaryconditions)
    kM[2] = f(tn+deltat*cM[2],w+deltat*(aM[2,0]*kM[0]+aM[2,1]*kM[1]),matrix,boundaryconditions)
    kM[3] = f(tn+deltat*cM[3],w+deltat*(aM[3,0]*kM[0]+aM[3,1]*kM[1]+aM[3,2]*kM[2]),matrix,boundaryconditio
    kM[4] = f(tn+deltat*cM[4],w+deltat*(aM[4,0]*kM[0]+aM[4,1]*kM[1]+aM[4,2]*kM[2]+aM[4,3]*kM[3]),matrix,bo
    w = w+deltat*np.sum(bM*kM, axis=0)
    return w

# define model
endp=True
m = 1
maxIter = 1000000
endpoint = 2*np.pi
x = np.linspace(0,endpoint,num=m,endpoint=endp)
w = np.ones(x.shape)
deltat = 10**(-5)

# define arrays to save the concentrations of the different factors
allw = np.zeros((maxIter+1,m))
allw[0] = w

allw2 = np.zeros((maxIter+1,m))
allw2[0] = w

# define matrix and BC for the second order space-derivative (no space-integration)
matrix=0
BC=0

# calculate function values over time
for i in range(maxIter):
    allw[i+1]=Merson(f,i*deltat,deltat,allw[i],matrix,1,BC,m)
    allw2[i+1]=Merson(f2,i*deltat,deltat,allw2[i],matrix,1,BC,m)

# exact solutions over time
time = np.arange(maxIter+1)*deltat
exsol = np.exp(time)
exsol2 = 1+time

# results of Merson's time integration
fig,((ax0,ax1),(ax2,ax3)) = plt.subplots(ncols=2,nrows=2,constrained_layout = True)
fig.set_figheight(9)
fig.set_figwidth(12)

# time integration exponential function
ax0.set_ylabel('Value')
ax0.set_xlabel('Time t')
ax0.plot(time,exsol,label='Exact Solution $exp(t)$')
ax0.plot(time,allw,label='Estimated Solution')
ax0.legend(loc='upper left')

# error of time integration exponential function
ax1.set_title('Error in Merson Time Integration')
ax1.set_ylabel('Error')
ax1.set_xlabel('Time t')
ax1.plot(time,np.abs(exsol-allw.T[0]))

# time integration constant function
ax2.set_ylabel('Value')
ax2.set_xlabel('Time t')
ax2.plot(time,exsol2,label='Exact Solution $t+1$')
ax2.plot(time,allw2,label='Estimated Solution')
ax2.legend(loc='upper left')

# error of time integration constant function
ax3.set_title('Error in Merson Time Integration')
ax3.set_ylabel('Error')
ax3.set_xlabel('Time t')
ax3.plot(time,np.abs(exsol2-allw2.T[0]))
```