

Parametric Models for Clinical Measurements in Ophthalmology

Problem Presenter

Anna Kondic, Certara

Report Editor

David A. Edwards, University of Delaware

Thirty-Fifth Annual Workshop on Mathematical Problems in Industry
June 17–21, 2019
New Jersey Institute of Technology

Table of Contents

Preface	ii
Deterministic Modeling	
Modeling and Analytical Solutions <i>D. A. Edwards, B. Emerick, K. Kiradjiev, C. Raymond</i>	1
Analytical Models and Numerical Solutions <i>K. Kiradjiev</i>	29
Parameter Fitting the Continuous Model <i>L. Barnes, S. Hussung, D. Jonas, J. Slepoi</i>	34
The CMA Model <i>B. Emerick, C. Raymond</i>	70
Estimating Parameters of the Best Corrected Visual Acuity Discrete Map using Variational Data Assimilation <i>M. Moye</i>	74
Diffusion Model of Drug Delivery <i>M. Zyskin</i>	84
Stochastic Modeling	
Stochastic Modeling of Central Retinal Thickness <i>M. Sirlanci</i>	95
Statistical Analysis <i>S. Dhar</i>	102

Preface

At the 35th Annual Workshop on Mathematical Problems in Industry (MPI), Anna Kondic of Certara presented a problem concerning the understanding of clinical measurements in ophthalmology.

This manuscript is really a collection of reports from teams in the group working on several aspects of the problem. Here is a brief summary of each:

1. Edwards *et al.* present a general introduction of the problem. They introduce continuous and discrete models, and present some basic analytical solutions.
2. Kiradjiev presents some analytical solutions of the continuous model, and evaluates the effect of changing the parameters on their behavior.
3. Barnes *et al.* present detailed results from parameter fitting the continuous model to patient data. R and Matlab codes are included.
4. Emerick and Raymond present some results from parameter fitting the discrete model to patient data, with an eye to identifying commonalities in parameters in responsive and nonresponsive patients.
5. Moyer obtains some additional results using the discrete model and a variational method to incorporate experimental noise and parametric error.
6. Zyskin presents some brief results on expanding the continuous model to include diffusion.
7. Sirlanci uses an Ohrenstein-Uhlenbeck stochastic model for the retinal thickness, then a Bayesian optimization approach to estimate the parameters.
8. Dhar uses a statistical approach to fit the data involving generalized linear models.

In addition to the authors of these reports, the following people participated in the group discussions:

- Linda Cummings, New Jersey Institute of Technology
- Pavel Dubovski, Stevens Institute of Technology
- Sarafa Iyaniwura, University of British Columbia
- Davin Lutz, Oxford University
- Vrushaly Shinglot, University of Texas at Dallas

We particularly appreciate Lutz's contribution to the understanding of possible relationships between the maximum effectiveness and the tolerance. Special recognition is due to Lauren Barnes, Daniel Jonas, and Steve Hussung for making the group's oral presentations throughout the week.

Modeling and Analytical Solutions

Brooks Emerick, Kutztown University
David A. Edwards, University of Delaware
Kristian Kiradjiev, Oxford University
Christopher Raymond, University of Delaware

Section 1: Introduction

Age-related macular degeneration is a leading cause of blindness. In the wet variety (wAMD), abnormal blood vessels leak fluid into the subretinal space, which results in vision loss [7]. One treatment regimen for wAMD is to inject the eye at regular intervals with medication that stems this fluid loss. To determine the effectiveness of the regimen, two measurements are used. The first is the best-corrected visual acuity (BCVA) score, which is essentially the score determined from having the patient read an eye chart. The second is central sub-field (retinal) thickness (CSFT).

As the treatment regimen progresses, these two types of measurements are taken and plotted versus time. In a responsive patient, one would expect to see BCVA increase while CSFT decreases. The group's project was to create mathematical models of the response of these two measurements to the treatment regimen. These models involve tracking not only the concentration of the medication in the eye, but also the body's increasing tolerance to the medication as time increases.

Once the models were formulated, they were tested against the given data, with three long-term goals in mind:

- model validation,
- predictive value,
- establishing correlation between the fit parameters of certain patients and their clinical outcomes.

Section 2: Linear Continuous Models

We first present a continuous model for the response of the measurements to the treatment regimen. Our general approach is motivated by the work in [5], though the terms we use are different.

We begin by modeling the (mass) concentration $\tilde{C}(\tilde{t})$ of the medication in the eye. At time $\tilde{t} = 0$, an initial dosage \tilde{D} (in units of mass) is administered. This can then be converted to a dosage concentration \tilde{D} by dividing by the volume V of the vitreous fluid in the eye. (Parameter values may be found in the Appendix.)

We assume that the concentration of the medication decays away exponentially. Repeat dosages (all assumed to be of the same value) are administered at N subsequent discrete times t_i (so there are $N + 1$ injections total). These times are relatively standard from patient to patient, but are not evenly spaced. (The case where each dosage has a different value is a trivial extension to this work.)

Thus the model can be expressed in the following two equivalent forms:

$$\frac{d\tilde{C}}{d\tilde{t}} = -k_C\tilde{C} + \frac{\tilde{D}}{V} \sum_{i=1}^N \delta(\tilde{t} - \tilde{t}_i), \quad \tilde{C}(0) = \tilde{D}, \quad (2.1a)$$

$$\frac{d\tilde{C}}{d\tilde{t}} = -k_C\tilde{C} + \frac{\tilde{D}}{V} \sum_{i=0}^N \delta(\tilde{t} - \tilde{t}_i), \quad \tilde{C}(0) = 0, \quad \tilde{t}_0 = 0, \quad (2.1b)$$

where k_C is the rate constant for the medication decay. (Note the difference between the expressions in the lower limit of summation.) Mulyukov [5] indicates that the half-life of \tilde{C} seems to be the same in all patients, which means that k_C will be as well.

For the BCVA acuity measurement $\tilde{a}(\tilde{t})$, we assume that in the absence of medication, the measurement will decay exponentially to a nonzero value which is consistent with blindness. Hence we have

$$\frac{d\tilde{a}}{d\tilde{t}} = \tilde{k}_{\text{in}}[1 + M(T)E(\tilde{C})] - \tilde{k}_{\text{out}}\tilde{a}, \quad \tilde{a}(0) = \tilde{a}_0. \quad (2.2)$$

Here \tilde{k}_{out} is a rate constant for the decay of acuity in patients with wAMD, \tilde{k}_{in} represents some natural ‘‘inflow’’ of acuity which balances this decay, and the product $M(T)E(\tilde{C})$ is the *effectiveness factor* of the medication.

Note that from a purely biological perspective, it might make more sense to multiply \tilde{k}_{out} by the effectiveness factor, since the medication is designed to slow the decay rate. The mathematical problem with doing this is that if we wish to solve (2.2) using Laplace transform techniques, the coefficient of \tilde{a} must be a constant. Mulyukov [5] also uses the approach in (2.2), but seems to do so for mathematical reasons as well.

As noted above, the effectiveness factor is a product of two factors. First, there is a *maximum effectiveness factor* $M(T)$, where $T(\tilde{t})$ is the *tolerance* to the drug. Initially, the

patient has no tolerance to the drug, which corresponds to a maximum effectiveness level of M_∞ :

$$M(T = 0) = M_\infty, \quad T(\tilde{t} = 0) = 0. \quad (2.3)$$

Under this treatment regimen, the most rapid improvement typically occurs after the first few injections. As the patient receives more doses, the patient has an increased tolerance to the drug, so the maximum effectiveness of the drug goes down. Hence we have that

$$\frac{dM}{dT} < 0.$$

As described above, the presence of \tilde{C} reduces T . However, in the absence of \tilde{C} , the body will clear the remaining medication and T will return to 0. Therefore, the evolution of T can be modeled as follows:

$$\frac{dT}{d\tilde{t}} = \tilde{k}_T \tilde{C} - \tilde{k}_\infty T, \quad (2.4)$$

subject to the initial condition in (2.3). Here \tilde{k}_∞ is a constant whose interpretation will be discussed in more detail in the next section.

The simplest functional form for M is just to make M linear in T :

$$M_1(T) = M_\infty(1 - T), \quad (2.5a)$$

where the subscript “1” refers to “linear.” But for $T > 1$, M can go negative. Other forms without this problem include

$$M_a(T) = \frac{M_\infty}{1 + T}, \quad M_e(T) = M_\infty e^{-T}, \quad (2.5b)$$

where the subscripts “a” and “e” refer to algebraic and exponential decay, respectively. Note that all three forms have similar behavior as $T \rightarrow 0$.

The second component of the effectiveness factor is the *dosage effectiveness factor* $E(\tilde{C})$, which describes how much of the maximum effectiveness is achieved for a certain concentration \tilde{C} . In [5], the authors use a sigmoidal function:

$$E_s(\tilde{C}) = \frac{\tilde{C}}{C_* + \tilde{C}}, \quad (2.6a)$$

where C_* is the half-constant and the subscript “s” refers to “sigmoidal”. However, in order to obtain analytical results, we simplify the sigmoidal curve to a piecewise-constant function:

$$\begin{aligned} E_H(\tilde{C}) &= \begin{cases} 0, & \tilde{C} < C_*, \\ 1, & \tilde{C} > C_*, \end{cases} \\ &= H(\tilde{C} - C_*), \end{aligned} \quad (2.6b)$$

where $H(\cdot)$ is the Heaviside step function.

Note that in both cases, if $\tilde{C} \equiv 0$ for all time, then $E(\tilde{C}) \equiv 0$ as well, which forces the steady state of (2.2) to be $\tilde{k}_{\text{in}}/\tilde{k}_{\text{out}}$. Recall that acuity decreases as the disease progresses, so we must have that

$$\frac{\tilde{k}_{\text{in}}}{\tilde{k}_{\text{out}}} < \tilde{a}_0. \quad (2.7)$$

In Mulykov [5], the authors indicate that in the treatment regimen they are studying (which is similar to ours), E_C stays in the upper branch of the sigmoid function. Therefore, we are motivated to take C_* to be small enough that E_C doesn't drop to zero until regularly administered injections stop.

We scale these equations to make them dimensionless. For \tilde{a} , it is natural to normalize by the initial value a_0 . For \tilde{t} , the fact that the half-life is known for the medication (and doesn't vary between patients) makes it an obvious choice, and that half-life is related to k_C (see Appendix).

For \tilde{C} , there are two candidates. The first is \tilde{D}/V . Its advantage is that it is known for each patient, but the downside is that the dosage varies among patients. Therefore, instead we use the threshold C_* , which is also assumed to have no interpatient variability.

Thus we have

$$a(t) = \frac{\tilde{a}(\tilde{t})}{\tilde{a}_0}, \quad C(t) = \frac{\tilde{C}(\tilde{t})}{C_*}, \quad t = k_C \tilde{t}. \quad (2.8)$$

Substituting (2.8) into (2.1a), we obtain

$$\begin{aligned} \tilde{k}_C C_* \frac{dC}{dt} &= -k_C C_* C + \frac{\tilde{D}}{V} \sum_{i=1}^N \delta((t - t_i)/k_C), \quad C_* C(0) = \frac{\tilde{D}}{V}, \\ \frac{dC}{dt} &= -C + \frac{\tilde{D}}{V k_C C_*} \sum_{i=1}^N k_C \delta(t - t_i) \\ \frac{dC}{dt} &= -C + D \sum_{i=1}^N \delta(t - t_i), \quad C(0) = D, \quad D = \frac{\tilde{D}}{V C_*}. \end{aligned} \quad (2.9a)$$

(Note that for the medication to have any efficacy, D must be greater than 1.) Similarly, (2.1b) becomes

$$\frac{dC}{dt} = -C + D \sum_{i=0}^N \delta(t - t_i), \quad C(0) = 0, \quad t_0 = 0. \quad (2.9b)$$

Substituting (2.8) into (2.2), we have the following:

$$\begin{aligned} \tilde{a}_0 k_C \frac{da}{dt} &= \tilde{k}_{\text{in}} [1 + M(T)E(C)] - \tilde{k}_{\text{out}} \tilde{a}_0 a, \quad \tilde{a}_0 a(0) = \tilde{a}_0 \\ \frac{da}{dt} &= k_{\text{in}} [1 + M(T)E(C)] - k_{\text{out}} a, \quad a(0) = 1, \end{aligned} \quad (2.10a)$$

$$k_{\text{in}} = \frac{\tilde{k}_{\text{in}}}{\tilde{k}_C \tilde{a}_0}, \quad k_{\text{out}} = \frac{\tilde{k}_{\text{out}}}{\tilde{k}_C}. \quad (2.10b)$$

Note that (2.7) implies that $k_{\text{in}}/k_{\text{out}} < 1$, as desired. Substituting (2.8) into (2.4), (2.3), and (2.6), we obtain

$$k_C \frac{dT}{dt} = \tilde{k}_T C_* C - \tilde{k}_\infty T$$

$$\frac{dT}{dt} = k_T C - k_\infty T, \quad T(0) = 0, \quad k_T = \frac{\tilde{k}_T C_*}{k_C}, \quad k_\infty = \frac{\tilde{k}_\infty}{k_C}, \quad (2.11)$$

$$E_s(C) = \frac{C_* \tilde{C}}{C_* + C_* \tilde{C}} = \frac{C}{C + 1}, \quad (2.12a)$$

$$E_H(C) = H(C - 1). \quad (2.12b)$$

Upon examination of the data (after normalization), we saw similar behavior for both the acuity and thickness. Therefore, we can model the evolution of the retinal thickness CSFT measurement $\tilde{s}(\tilde{t})$ in a similar manner to (2.2). Essentially, all the parameters would have new values. In addition, since the thickness would tend to increase as a function of time in the absence of treatment, we would have a new bound

$$\frac{\tilde{k}_{\text{in}}}{\tilde{k}_{\text{out}}} > \tilde{s}_0 \quad (2.13)$$

to replace (2.7).

Section 3: Analytical Solutions

Given the amount of parameter fitting involved and the large number of data sets, it is convenient to find analytical solutions of these equations, at least to perform basic model validation. To do so, we define the Laplace transform in the usual way:

$$\hat{f}(s) = \int_0^{\infty} e^{-st} f(t) dt.$$

Taking the Laplace transform of (2.9b), we have

$$\begin{aligned} s\hat{C} &= -\hat{C} + D \sum_{i=0}^N e^{-st_i} \\ \hat{C} &= \frac{D}{s+1} \sum_{i=0}^N e^{-st_i}, \end{aligned} \quad (3.1a)$$

$$C(t) = D \sum_{i=0}^N H(t-t_i) e^{-(t-t_i)}, \quad (3.1b)$$

where the inverse Laplace transform can be found in [3, Tab. 6.2.1]. Since we can solve for C explicitly in terms of Heaviside functions, using (3.1b) directly may be more useful in Matlab than using (2.9), given the presence of the delta functions in the latter.

A graph of (3.1b) for the parameters in the Appendix is shown in Fig. 3.1. Note the high dosage compared to the threshold value 1.

Taking the Laplace transform of (2.11), we obtain

$$\begin{aligned} s\hat{T} &= k_T \hat{C} - k_{\infty} \hat{T} \\ \hat{T} &= \frac{k_T \hat{C}}{s + k_{\infty}}, \end{aligned} \quad (3.2a)$$

$$T(t) = k_T \int_0^t C(t') e^{-k_{\infty}(t-t')} dt', \quad (3.2b)$$

which could also have been derived from (2.11) directly using integrating factors.

Equations (3.1b) and (3.2b) hold for *any* choice of M and E , and hence can be used throughout this work. In the formulation (3.2b), it is easy to see that k_{∞} measures the memory effect of the accumulated concentration. If we want the tolerance to represent the accumulated concentration, we would take $k_{\infty} = 0$. However, such a choice has a particular deficiency: the body will never clear the medication and recover T to its original value. Therefore, for any reasonable biological application, $\tilde{k}_{\infty} \neq 0$. This will ensure that if $C = 0$ for some period of time, eventually T will return to the initial value 0 as $t \rightarrow \infty$.

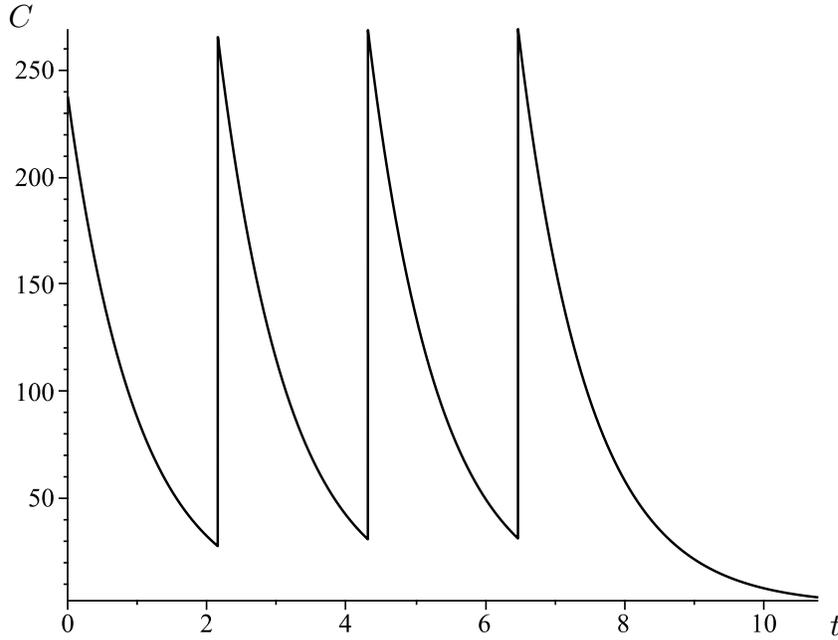


Figure 3.1. C vs. t . Here the dosage is given by the value for medication B every four weeks for four visits.

We may simplify \hat{T} with the help of (3.1a):

$$\begin{aligned}\hat{T} &= \frac{k_T D}{(s + k_\infty)(s + 1)} \sum_{i=0}^N e^{-st_i} \\ &= \frac{k_T D}{1 - k_\infty} \left(\frac{1}{s + k_\infty} - \frac{1}{s + 1} \right) \sum_{i=0}^N e^{-st_i},\end{aligned}\quad (3.3a)$$

$$T(t) = \frac{k_T D}{1 - k_\infty} \sum_{i=0}^N H(t - t_i) \left[e^{-k_\infty(t-t_i)} - e^{-(t-t_i)} \right]. \quad (3.3b)$$

If $k_\infty > 1$, the denominator is less than zero, but so is the bracketed term, yielding the positive value of $T(t)$ desired.

A graph of (3.3b) is shown in Fig. 3.2. We use the parameters in the Appendix, as well as the values computed by Barnes and Slepoi (see their chapter) for validation patient 12:

$$k_T = 0.1927075, \quad M_\infty = 0.001151316, \quad k_{\text{in}} = 0.32463696, \quad k_{\text{out}} = 0.290570446. \quad (3.4)$$

(We had to do a slight fudge and multiply T by M_∞ because of a difference in models.) Note that in this patient, the tolerance is quite small. We also see oscillations in the tolerance as the drug levels spike and then decay away.

To yield an analytical solution for a , we first choose to use the form M_1 given in (2.5a). To keep M positive, we must have that $T(t) < 1$ for all time. We may derive a rough

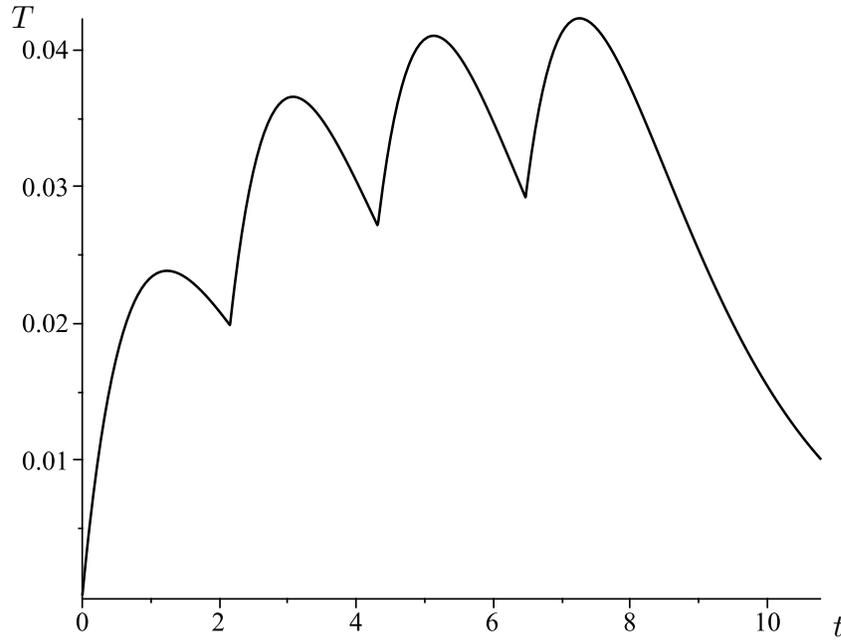


Figure 3.2. T vs. t . Here the dosage is given by the value for medication B every four weeks for four visits.

bound by noting from (3.3b) that

$$T(t) < \frac{k_T D(N+1)}{1-k_\infty} (e^{-k_\infty t} - e^{-t}). \quad (3.5)$$

Taking the derivative of the right-hand side of (3.5) equal to zero to find the position of the maximum, we have

$$\begin{aligned} -k_\infty e^{-k_\infty t} + e^{-t} &= 0 \\ -t &= -k_\infty t + \log k_\infty \\ t &= \frac{\log k_\infty}{k_\infty - 1} > 0, \end{aligned}$$

which we substitute back into the right-hand side of (3.5) to obtain

$$\begin{aligned} \frac{k_T D(N+1)}{1-k_\infty} \left[\exp\left(-\frac{k_\infty \log k_\infty}{k_\infty - 1}\right) - \exp\left(-\frac{\log k_\infty}{k_\infty - 1}\right) \right] &< 1 \\ \frac{k_T D(N+1)}{1-k_\infty} \left[k_\infty^{-k_\infty/(k_\infty-1)} - k_\infty^{-1/(k_\infty-1)} \right] &< 1 \\ k_T D(N+1) k_\infty^{-k_\infty/(k_\infty-1)} &< 1 \\ k_T &< \frac{k_\infty^{k_\infty/(k_\infty-1)}}{D(N+1)}. \end{aligned} \quad (3.6)$$

As D or N increases, the amount of medication in the eye increases, increasing tolerance. Hence the bound is tighter.

In order to obtain an analytical solution, the final step is to take $E(C)$ to be $E_H(C)$ as defined in (2.12b). Substituting M_1 and E_H into (2.10a), we have the following:

$$\frac{da}{dt} = k_{\text{in}}[1 + M_{\infty}(1 - T)H(C - 1)] - k_{\text{out}}a. \quad (3.7)$$

For simplicity, let τ_1 be the first (dimensionless) time for which $C(\tau_1) = 1$, and let $t_j < \tau_1 < t_{j+1}$. Then for the region $0 < t < \tau_1$, $E(C) = 1$, and so (2.10a) becomes

$$\begin{aligned} \frac{da}{dt} &= k_{\text{in}}(1 + M_{\infty} - M_{\infty}T) - k_{\text{out}}a \\ s\hat{a} - 1 &= k_{\text{in}} \left[\frac{1 + M_{\infty}}{s} - \frac{k_T D M_{\infty}}{1 - k_{\infty}} \left(\frac{1}{s + k_{\infty}} - \frac{1}{s + 1} \right) \sum_{i=0}^j e^{-st_i} \right] - k_{\text{out}}\hat{a} \\ \hat{a} &= \frac{1}{s + k_{\text{out}}} \left\{ 1 + k_{\text{in}} \left[\frac{1 + M_{\infty}}{s} - \frac{M_{\infty} k_T D}{1 - k_{\infty}} \left(\frac{1}{s + k_{\infty}} - \frac{1}{s + 1} \right) \sum_{i=0}^j e^{-st_i} \right] \right\} \\ &= \frac{1}{s + k_{\text{out}}} + \frac{k_{\text{in}}(1 + M_{\infty})}{k_{\text{out}}} \left(\frac{1}{s} - \frac{1}{s + k_{\text{out}}} \right) \\ &\quad - \frac{M_{\infty} k_T k_{\text{in}} D}{1 - k_{\infty}} \sum_{i=0}^j e^{-st_i} \times \\ &\quad \left[\frac{1}{k_{\text{out}} - k_{\infty}} \left(\frac{1}{s + k_{\infty}} - \frac{1}{s + k_{\text{out}}} \right) - \frac{1}{k_{\text{out}} - 1} \left(\frac{1}{s + 1} - \frac{1}{s + k_{\text{out}}} \right) \right] \Big\} \\ &= \frac{1}{s + k_{\text{out}}} \left[1 - \frac{k_{\text{in}}(1 + M_{\infty})}{k_{\text{out}}} \right] + \frac{k_{\text{in}}(1 + M_{\infty})}{k_{\text{out}} s} \\ &\quad - \frac{M_{\infty} k_T k_{\text{in}} D}{1 - k_{\infty}} \sum_{i=0}^j e^{-st_i} \left[\left(\frac{1}{k_{\text{out}} - 1} - \frac{1}{k_{\text{out}} - k_{\infty}} \right) \frac{1}{s + k_{\text{out}}} \right. \\ &\quad \left. + \frac{1}{(k_{\text{out}} - k_{\infty})(s + k_{\infty})} - \frac{1}{(k_{\text{out}} - 1)(s + 1)} \right], \end{aligned} \quad (3.8)$$

where we have used (3.3a). (Note the new upper index of summation.) Then inverting the transform, we have that

$$\begin{aligned} a(t) &= \left[1 - \frac{k_{\text{in}}(1 + M_{\infty})}{k_{\text{out}}} \right] e^{-k_{\text{out}}t} + \frac{k_{\text{in}}(1 + M_{\infty})}{k_{\text{out}}} \\ &\quad - \frac{M_{\infty} k_T k_{\text{in}} D}{1 - k_{\infty}} \sum_{i=0}^j H(t - t_i) \left[\left(\frac{1}{k_{\text{out}} - 1} - \frac{1}{k_{\text{out}} - k_{\infty}} \right) e^{-k_{\text{out}}(t-t_i)} \right. \\ &\quad \left. + \frac{e^{-k_{\infty}(t-t_i)}}{(k_{\text{out}} - k_{\infty})} - \frac{e^{-(t-t_i)}}{(k_{\text{out}} - 1)} \right]. \end{aligned} \quad (3.9)$$

Alternatively, the solution can be computed using integrating factors (see the chapter by Kiradjiev).

Once $a(\tau_1)$ has been computed, we can then solve (2.10a) in the region $t > \tau_1$ by remembering that in this region, $E(C) = 0$ and there is simple exponential decay. This behavior holds until t_{j+1} , at which point the new injection would force C above 1 again. Hence we would compute $a(t_{j+1})$ as the new initial condition for this problem. This process can be repeated across injections as needed, yielding oscillations.

However, there is excellent news. According to Mulyukov [5], C decays below 1 only if there are around three months between dosages. In the data provided, the furthest spacing between doses is two months. Hence we should never be in the regime where $C < 1$.

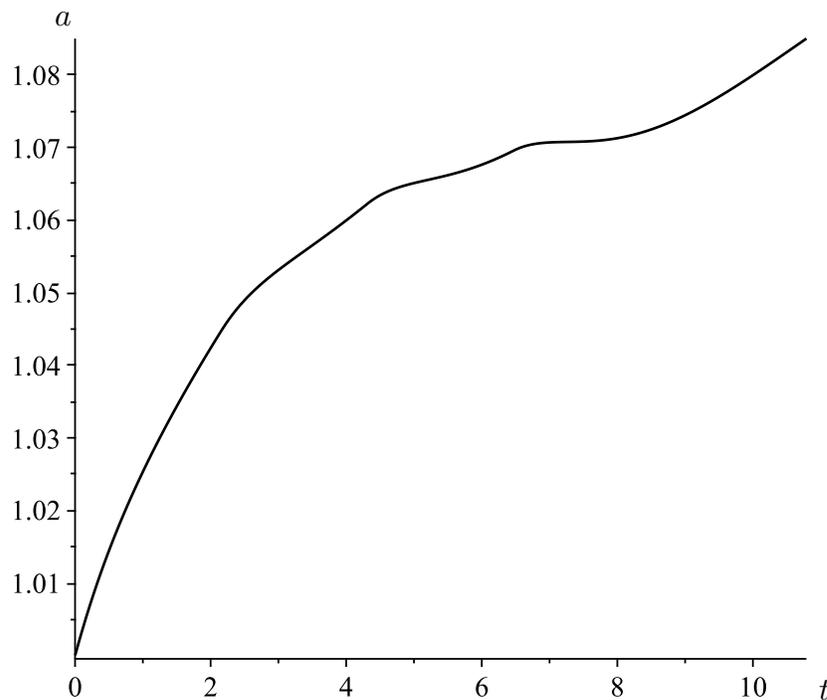


Figure 3.3. a vs. t . Here the dosage is given by the value for medication B every four weeks for four visits.

A graph of (3.9) is shown in Fig. 3.3, again using the parameters in the Appendix and (3.4). There is consistent improvement in acuity due to the low tolerance level. However, there are small oscillations corresponding to the variations in concentration and tolerance.

Given the form of M_1 in (2.5a), we still face the problem of whether T can ever be greater than 1. To retain analytical solutions while removing this problem, we can replace (2.5a) with

$$M(T) = M_\infty(1 - T)H(1 - T). \quad (3.10)$$

With this definition, we simply redefine τ_1 to be the first time at which $T(\tau_1) = 1$ and proceed as described above, since when $t > \tau_1$, $M(T) = 0$. T would then have to be tracked using (3.3b).

Section 4: Introducing a Map

Since we have data only at discrete times (patient office visits), which for the most part are evenly spaced, it may be appropriate to replace the continuous model with a discrete dynamical system.

Let C_n be the measurement of the medication at time n (and similarly for a_n and M_n). The first visit is a screening visit, and is given a negative time in the data. Therefore, we take

$$C_{-1} = 0, \quad (4.1)$$

since there is no medication administered at the initial screening. We work directly with the dimensionless versions of the continuous models; an appropriate discrete analog of (2.9a) is the following:

$$C_{n+1} = \kappa_C C_n + D\delta_{n+1,d}, \quad n \geq -1, \quad (4.2a)$$

$$\delta_{n+1,d} = \begin{cases} 1, & \text{dose given at time } n+1 \text{ (visit } n+3) \\ 0, & \text{else.} \end{cases} \quad (4.2b)$$

Without a time scale, we must retain the rate constant for the decay of C , which is now written with a positive sign. We replace k with κ to emphasize the structural difference between the two models. (More discussion of restrictions on κ_C will follow.)

The discrete analog of (2.10a) is given by

$$a_{n+1} = \kappa_{\text{in}}[1 + M(T_n)E(C_n)] + \kappa_{\text{out}}a_n, \quad a_{-1} = 1. \quad (4.3)$$

Again we remove all minus signs. The discrete analog of (2.11) is given by the following:

$$T_{n+1} = \kappa_T C_n + \kappa_{\infty} T_n, \quad T_{-1} = 0. \quad (4.4)$$

To gain insight into the system, we look at the fixed points in two cases: no dosing and regular dosing. In the case of no dosing, (4.2a) becomes homogeneous and has only the fixed point $C = 0$. Substituting the fixed point $C = 0$ into (4.4), we again obtain a homogeneous equation with only the fixed point $T = 0$. Lastly, substituting the given fixed points into (4.3), we have the equation

$$a_{n+1} = \kappa_{\text{in}} + \kappa_{\text{out}}a_n,$$

which has the single fixed point

$$a = \frac{\kappa_{\text{in}}}{1 - \kappa_{\text{out}}}. \quad (4.5)$$

To examine stability, we compute the Jacobian of the map:

$$J = \begin{pmatrix} \kappa_C & 0 & 0 \\ \kappa_T & \kappa_{\infty} & 0 \\ \kappa_{\text{in}}M(T)E'(C) & \kappa_{\text{in}}M'(T)E(C) & \kappa_{\text{out}} \end{pmatrix}. \quad (4.6)$$

But this matrix is triangular with eigenvalues κ_C , κ_∞ , and κ_{out} for *any* fixed point. Therefore, each of these parameters must be bounded between -1 and 1 for stability. And moreover, each must be positive for all the iterates to be positive.

Next we look at the case of regular dosing at each interval. In that case, (4.2a) becomes

$$C_{n+1} = \kappa_C C_n + D, \quad (4.7)$$

which has the fixed point

$$C = \frac{D}{1 - \kappa_C}. \quad (4.8)$$

Substituting this fixed point into (4.4), we have

$$T_{n+1} = \frac{\kappa_T D}{1 - \kappa_C} + \kappa_\infty T_n,$$

which has the fixed point

$$T = \frac{\kappa_T D}{(1 - \kappa_C)(1 - \kappa_\infty)}. \quad (4.9)$$

Substituting the steady states into (4.3), we obtain the following:

$$a_{n+1} = \kappa_{\text{in}}[1 + M(T)E(C)] + \kappa_{\text{out}} a_n,$$

where C and T are given in (4.8) and (4.9). Hence we have the fixed point

$$a = \frac{\kappa_{\text{in}}[1 + M(T)E(C)]}{1 - \kappa_{\text{out}}},$$

and again this fixed point is stable under the conditions listed above.

We also consider the case of doses being administered on alternate visits. Given that the dosage is always administered at $n = 0$, we can then rewrite (4.2) as the system

$$\begin{aligned} C_{2m} &= \kappa_C C_{2m-1} + D, \quad m \geq 0, \\ C_{2m+1} &= \kappa_C C_{2m}, \end{aligned}$$

which may be combined to yield

$$C_{2m+1} = \kappa_C^2 C_{2m-1} + \kappa_C D. \quad (4.10)$$

(Note that this is not an iterated map, since the dosage is given only once.) This new map has fixed point

$$C = \frac{\kappa_C D}{1 - \kappa_C^2};$$

a stability analysis in this case is beyond the scope of this manuscript.

In the case of intervals of regularly spaced injections, we may solve for C_n as follows. The general solution of (4.7) is given by

$$C_{n+1} = \frac{D}{1 - \kappa_C} + \alpha \kappa_C^n, \quad (4.11)$$

where α is an arbitrary constant. Suppose that injections are given during P injection periods at times $n \in [n_{p-}, n_{p+}]$ for $p = 1, 2, \dots, P$, and let $n_{1-} = 0$. Then using the general solution (4.11) for the region $n \in [0, n_{1+}]$, we have

$$C_n = \frac{D}{1 - \kappa_C} (1 - \kappa_C^{n+1}), \quad n \in [0, n_{1+}]. \quad (4.12)$$

(Note the extra 1 in the exponent.)

In general, during a dosing interval $[n_{p-}, n_{p+}]$, the contribution to the total C_n from the dosing term is given by the following:

$$\frac{D}{1 - \kappa_C} (1 - \kappa_C^{n+1-n_{p-}}), \quad n \in [n_{p-}, n_{p+}]. \quad (4.13)$$

After that interval, the contribution from that dosing interval is just reduced by the factor κ_C :

$$\frac{D}{1 - \kappa_C} (1 - \kappa_C^{n_{p+}+1-n_{p-}}) \kappa_C^{n-n_{p+}} = \frac{D}{1 - \kappa_C} (\kappa_C^{n-n_{p+}} - \kappa_C^{n+1-n_{p-}}), \quad n \geq n_{p+}. \quad (4.14)$$

Note there is no extra 1 in the first exponent.

Combining these terms for all dosing intervals yields the full solution in this region:

$$\begin{aligned} C_n &= \frac{D}{1 - \kappa_C} \sum_{p=1}^P H(n - n_{p+}) (\kappa_C^{n-n_{p+}} - \kappa_C^{n+1-n_{p-}}) \\ &\quad + [H(n - n_{p-}) - H(n - n_{p+})] (1 - \kappa_C^{n+1-n_{p-}}) \\ &= \frac{D}{1 - \kappa_C} \sum_{p=1}^P H(n - n_{p-}) (1 - \kappa_C^{n+1-n_{p-}}) - H(n - n_{p+}) (1 - \kappa_C^{n-n_{p+}}). \end{aligned} \quad (4.15a)$$

To find a more biologically pleasing form, we note that the last term is 0 at $n = n_{p+}$, so it can be shifted to obtain

$$C_n = \frac{D}{1 - \kappa_C} \sum_{p=1}^P H(n - n_{p-}) (1 - \kappa_C^{n+1-n_{p-}}) - H(n - (n_{p+} + 1)) \left[1 - \kappa_C^{n+1-(n_{p+}+1)} \right]. \quad (4.15b)$$

Equation (4.15b) can be explained as follows. The first term represents the solution for constant dosing, starting at n_{p-} . To shut the dosing off, we just subtract off another term for constant dosing, starting at $n = n_{p+} + 1$, which is the first period after the end of the real dosing.

Section 5: Considerations of Parameter Fitting

In this section we outline some of the issues that arose during parameter fitting. Actual results and more detailed discussions will be presented in later chapters.

Typically it is easier to fit the normalized data. That involves a preprocessing step of scaling the data by \tilde{a}_0 (or \tilde{s}_0). In the continuous case, the time is also scaled by k_C ; that step is unnecessary for the discrete model.

Values of D and k_∞ are listed in the Appendix; \tilde{a}_0 , \tilde{s}_0 , and the t_i can be determined from the data. Thus the parameters to be fit are k_{in} , k_{out} , k_T (or the corresponding κ), and M_∞ .

According to Mulyukov [5], all of these quantities can vary with the patient, perhaps depending on their age and sex. Since we did not have access to such variables in the data, we could not consider such a dependence. However, after making that statement Mulyukov [5] included variance of M_∞ only on age—hence we concluded that the other variations were not statistically significant.

There are several orders of magnitude between \tilde{D}/V and C_* . (There may be biological reasons for this; see the next section.) As we have scaled by the smaller one, that means that D (and hence C) will typically be quite large. Hence from (2.11) our parameter fit should have k_T (or κ_T) much less than 1 to prevent the tolerance becoming very large very quickly. (Note this is not a function of the particular choice of scaling; if we had reversed the scaling, there would have been small numbers in the normalized version of E_s , etc.)

One question we were asked to address was to see if there were certain clusters of patients with similar parameters, and whether those clusters corresponded with any particular clinical characteristics. One choice of clinical characteristic would be the improvement at the end of the trial, which could be measured by the ratio

$$\rho = \frac{\tilde{a}_{\text{last visit}}}{\tilde{a}_0}. \quad (5.1)$$

or the equivalent for r_0 . Then we could plot each of our parameters (as well as the error) *vs.* ρ to find trends.

Another question posed was the predictive value of the solutions. Typically the data runs over two years of visits and injections. Suppose that we fit the parameters to only the first year of data, and then use those parameters to predict the measurements in the second year. How well do our predictions match the actual data?

We have six different combinations of postulated functional forms of M and E in (2.5) and (2.12). [Here we ignore the additional cutoff linear model from (3.10).] These should be tested to see which works “best”. (This determination could be made by patient, by class, or globally.)

Section 6: Diffusion

In actuality, the medication is not injected directly into the target area (the center of the retina) due to the potential for damage. Instead, it is injected into the vitreous liquid inside the eye, from which it diffuses to the center of the retina. We may use dimensional analysis to estimate the time t_d needed for the medication to diffuse a distance equal to the radius R_e of the eye:

$$t_d = \frac{R_e^2}{4\tilde{\mathcal{D}}}, \quad (6.1)$$

where $\tilde{\mathcal{D}}$ is the diffusion coefficient in the vitreous fluid. Using the parameters in the Appendix, we have that this is on the order of weeks. Therefore, it seems important to include these dynamics.

To consider them, we take the simplest possible model that includes both decay and diffusion. Working in symmetric spherical coordinates, we replace (2.1a) by

$$\frac{\partial \tilde{C}}{\partial \tilde{t}} = -k_C \tilde{C} + \frac{\tilde{\mathcal{D}}}{\tilde{r}^2} \frac{\partial}{\partial \tilde{r}} \left(\tilde{r}^2 \frac{\partial \tilde{C}}{\partial \tilde{r}} \right), \quad (6.2a)$$

where \tilde{r} is the spherical coordinate. Note that when $\tilde{\mathcal{D}} = 0$, (6.2a) reduces to (2.1a) in the absence of the injection.

For simplicity, we take the coordinate system to be centered at the injection site. Therefore, we may model the injection by the following initial condition:

$$\tilde{C}(\tilde{r}, 0) = \tilde{D} \delta(\tilde{r}). \quad (6.2b)$$

The scalings for \tilde{C} and \tilde{t} are given by (2.8). For \tilde{r} , we use the distance R_d between the injection site and the center of the retina (see Fig. 6.1). Making these substitutions, we obtain

$$\begin{aligned} C_* k_C \frac{\partial C}{\partial t} &= -C_* k_C C + \frac{\tilde{\mathcal{D}}}{R_d^2 r^2} \frac{1}{R_d} \frac{\partial}{\partial r} \left(R_d r^2 C_* \frac{\partial C}{\partial r} \right), \\ \frac{\partial C}{\partial t} &= -C + \frac{\mathcal{D}}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial C}{\partial r} \right), \quad \mathcal{D} = \frac{\tilde{\mathcal{D}}}{R_d^2 k_C}, \end{aligned} \quad (6.3a)$$

$$\begin{aligned} C_* C(r, 0) &= \frac{\tilde{D}}{R_d^3} \delta(r) \\ C(r, 0) &= D_d \delta(r), \quad D_d = \frac{\tilde{D}}{R_d^3 C_*}, \end{aligned} \quad (6.3b)$$

where we note that when scaling the δ -function, we have to divide by the length three times since we are working in 3-space.

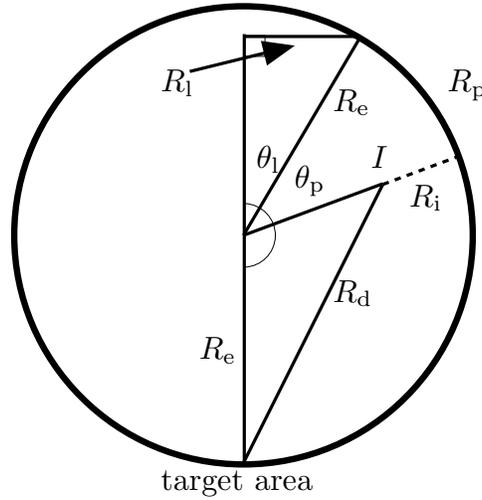


Figure 6.1. Cross-section of the eye through center of sphere (top view). R_l is the radius of the limbus. The dashed line represents the path of the needle, and R_d is the distance the medication must diffuse from the injection site I to the target area at bottom.

The decay in (3.1b) motivates the substitution

$$C(r, t) = e^{-t}u(r, t). \quad (6.4)$$

Substituting (6.4) into (6.3), we have the following:

$$e^{-t} \frac{\partial u}{\partial t} - e^{-t}u = -e^{-t}u + \frac{\mathcal{D}}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial (e^{-t}u)}{\partial r} \right)$$

$$\frac{\partial u}{\partial t} = \frac{\mathcal{D}}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial u}{\partial r} \right), \quad (6.5a)$$

$$u(r, 0) = D_d \delta(r). \quad (6.5b)$$

For simplicity, we solve in free space; the solution is then just the Green's function for the diffusion equation in 3-space, multiplied by the dosage strength D_d :

$$u(r, t) = \frac{D_d}{(4\pi \mathcal{D}t)^{3/2}} \exp\left(-\frac{r^2}{4\mathcal{D}t}\right),$$

$$C(r, t) = \frac{D_d}{(4\pi \mathcal{D}t)^{3/2}} \exp\left(-\frac{r^2}{4\mathcal{D}t} - t\right). \quad (6.6)$$

The relevant concentration is at the retina, which is at a dimensional length R_d from the source, which means that it corresponds to $r = 1$. Hence we compute

$$C(1, t) = \frac{D_d}{(4\pi \mathcal{D}t)^{3/2}} \exp\left(-\frac{1}{4\mathcal{D}t} - t\right). \quad (6.7)$$

To introduce the effects of multiple injections, we note that since the problem is linear, we may just shift time from t to $t - t_i$, and the solution remains the same. Hence in the full case we obtain

$$C(1, t) = \frac{D_d}{(4\pi\mathcal{D})^{3/2}} \sum_{i=0}^N \frac{H(t - t_i)}{(t - t_i)^{3/2}} \exp\left(-\frac{1}{4\mathcal{D}(t - t_i)} - (t - t_i)\right), \quad (6.8)$$

which would then replace (3.1b) in computations of T and a .

To conclude, we compute the distance R_d to be traversed. Per [4], the injection site is chosen to be a distance R_p (the subscript ‘‘p’’ is for ‘‘position’’) from the edge of the limbus, which is a structure near the iris with cross-sectional radius R_l (where the subscript ‘‘l’’ now stands for ‘‘limbus’’; see Fig. 6.1). The needle is injected a distance R_i (the subscript ‘‘i’’ is for ‘‘injection’’) into the eye and the medicine is released. In Fig. 6.1, we show the needle directed toward the center of the eye. It is also possible that the needle is directed more towards the retina [4], which would shorten the path. However, as we are looking to quantify the maximum effect of diffusion, we consider the route with the longest path.

The computation is a tedious exercise in trigonometry. In particular, from the right triangle we have

$$\cos \theta_1 = \frac{R_l}{R_e}. \quad (6.9a)$$

The definition of a radian yields

$$R_e \theta_p = R_p \quad \implies \quad \theta_p = \frac{R_p}{R_e}. \quad (6.9b)$$

Then by the Law of Cosines for the lower triangle, we have the following:

$$\begin{aligned} R_d^2 &= R_e^2 + (R_e - R_i)^2 - 2R_e R_i \cos(\pi - \theta_p - \theta_1) \\ &= R_e^2 + (R_e - R_i)^2 + 2R_e R_i \cos\left(\frac{R_p}{R_e} + \theta_1\right) \\ &= R_e^2 + (R_e - R_i)^2 + 2R_e R_i \left[\cos\left(\frac{R_p}{R_e}\right) \cos(\theta_1) - \sin\left(\frac{R_p}{R_e}\right) \sin(\theta_1) \right] \\ &= R_e^2 + (R_e - R_i)^2 + 2R_e R_i \left[\frac{R_l}{R_e} \cos\left(\frac{R_p}{R_e}\right) - \sqrt{1 - \frac{R_l^2}{R_e^2}} \sin\left(\frac{R_p}{R_e}\right) \right] \\ &= R_e^2 + (R_e - R_i)^2 + 2R_i \left[R_l \cos\left(\frac{R_p}{R_e}\right) - \sqrt{R_e^2 - R_l^2} \sin\left(\frac{R_p}{R_e}\right) \right]. \end{aligned} \quad (6.10)$$

A graph of (6.8) for the parameters in the Appendix is shown in Fig. 6.2. We note the following differences from Fig. 3.1:

- There is a slight delay between the injection time and the peak due to diffusion.
- The peak is an order of magnitude smaller than in the previous graph, since the medication must spread throughout the eye.
- Because of the smaller peak, the concentration is below the cutoff threshold $C = 1$ for about 1/3 of the time between doses.

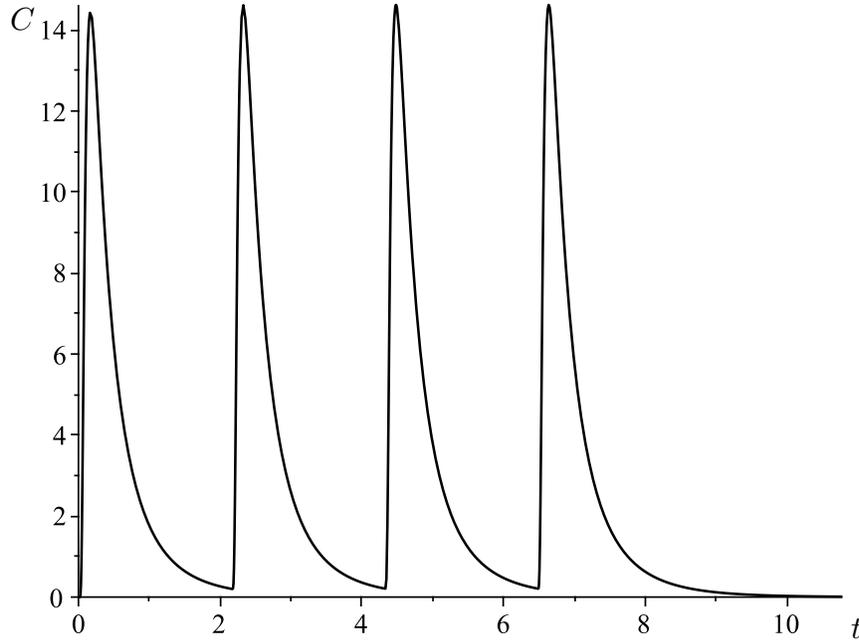


Figure 6.2. C vs. t . Here the dosage is given by the value for medication B every four weeks for four visits.

In actuality, we are probably underestimating the size of the peak because we did the calculation in free space. The full problem has a solution using separation of variables. In particular, we would solve the non-symmetric version of (6.5a):

$$\frac{\partial u}{\partial t} = \mathcal{D}\nabla^2 u, \quad 0 \leq r \leq \frac{R_e}{R_d} \quad (6.11)$$

subject to Neumann conditions on the surface:

$$\frac{\partial u}{\partial r} \left(\frac{R_e}{R_d}, \theta, \phi, t \right) = 0. \quad (6.12a)$$

The initial condition would then be given by

$$u(r, \theta, \phi, 0) = D_d \delta(\mathbf{r} - I), \quad (6.12b)$$

where I is the injection point identified in Fig. 6.1. This problem can be solved by separation of variables, and the δ function in (6.12b) would make the coefficients easy to compute. Then the concentration at the target point (which could be taken to have $\theta = 0$, $\phi = 0$ for simplicity) could be calculated and expressed as an infinite sum (see the chapter by Zyskin).

Section 7: Other Ideas

More Complicated Continuous Model

During the week, we also derived a more complicated continuous model. We were attempting to fix the deficiencies in the model in section 2: first, k_T must be tuned so that M doesn't go negative. Moreover, if $k_\infty = 0$, then the tolerance doesn't recover at all.

We hence postulated the following model directly for the maximum effectiveness, treating it is a function of \tilde{t} directly:

$$\frac{dM}{d\tilde{t}} = -\tilde{k}_T \tilde{C} M + \tilde{k}_\infty (M_\infty - M). \quad (7.1)$$

We may interpret (7.1) in light of (2.4) by noting that $M_\infty - M$ plays the role of T , but now there is an extra M multiplying the first term on the right-hand side. Introducing the scalings in (2.8), we immediately obtain

$$\frac{dM}{dt} = -k_T C M + k_\infty (M_\infty - M), \quad M(0) = M_\infty. \quad (7.2)$$

This equation then serves as an evolution equation for M , replacing (2.11) and one of the models for $M(T)$. This system seems like a prime candidate for numerical integration, but we first attempt to find an analytical solution.

Using a Laplace transform will fail due to the CM term on the right-hand side, since C is again given by (3.1b). However, since (7.2) is still linear and first order we may use the integrating factor technique:

$$\frac{d(M\mu)}{dt} = k_\infty M_\infty \mu, \quad \mu = \exp\left(\int_0^t k_T C(t') + k_\infty dt'\right), \quad (7.3a)$$

$$M(t) = \frac{M_\infty}{\mu} \left(1 + k_\infty \int_0^t \mu(t') dt'\right). \quad (7.3b)$$

Computing μ and its integral using (3.1b), we obtain the following:

$$\begin{aligned} \int_0^t k_T C(t') + k_\infty dt' &= Dk_T \int_0^t \sum_{i=0}^N H(t' - t_i) e^{-(t'-t_i)} dt' + k_\infty t \\ &= Dk_T \sum_{i=0}^N H(t - t_i) [1 - e^{-(t-t_i)}] + k_\infty t \\ \mu(t) &= \exp\left(Dk_T \sum_{i=0}^N H(t - t_i) [1 - e^{-(t-t_i)}] + k_\infty t\right), \end{aligned} \quad (7.4a)$$

$$\int_0^t \mu(t') dt' = \int_0^t \exp\left(Dk_T \sum_{i=0}^N H(t' - t_i) [1 - e^{-(t'-t_i)}] + k_\infty t'\right) dt'. \quad (7.4b)$$

When there is only one dose, then (7.4b) may be able to be written in terms of an incomplete gamma function. But when there is more than one dose, (7.4b) must be computed numerically.

We can also adapt this model in to a discrete map similar to our work in section 3. In particular, we have

$$M_{n+1} = M_\infty - \kappa_T C_n M_n + \kappa_\infty (M_n - M_\infty), \quad M_{-1} = M_\infty. \quad (7.5)$$

However, we abandoned this approach when it became clear that using T was a superior method.

Stochastic and Statistical Techniques

Others in the group are looking to analyze the data using a simple model for the acuity, with added parameter-related stochastic noise. (See the chapter by Serlanci.)

Others are using statistical techniques. In particular, \sqrt{a} was assumed to follow a normal distribution. (The transformation was used to eliminate skew to the left in the data.) The mean of \sqrt{a} was assumed to be written as

$$E(\sqrt{a}) = \exp(\text{linear profile}),$$

where the linear profile is a linear combination of the various characteristics of the data, including:

- visit number
- acuity
- dosage level at visit
- patient number

Similarly, $\log \tilde{s}$ was assumed to follow a normal distribution. Again, the transformation was used to remove skew in the right. It should be pointed out that this is also the scaling that was used in the initial presentation to illustrate the data in \tilde{s} .

Initial results indicated that the results did not depend on the actual dose administered, which is consistent with saturation at the lowest dosage. This agrees with the discussion in Mulyukov [5], where they indicate that the medication is at saturation for roughly 2–3 months after administration. As patients in this study typically received injections more often than that, they remained saturated. (For more details, see the chapter by Dhar.)

Miscellaneous

In the paper by Mulyukov [5], the authors say that their parameters could depend on the initial condition a_0 to some power. This argues for using some sort of nonlinear model, since in a linear model any parameters should be independent of the initial condition. However, in the paper they don't provide any parameter dependence for it, so perhaps when they fit their model, they found any relationship to be not significant.

There may be ways to use simpler phenomenological models, such as $a(t) = f(C/M)$, etc.

One could introduce an additional coefficient of T into the models (2.5). This extra parameter would have to be fit.

There was also talk of adding delay terms to the equation. If the delay is just in the effect of the concentration (similar to the diffusion effects discussed in section 6), then that would just add a shift to the forcing in the equation for a (or s), which is easy to treat mathematically. If the delay term is in a or s itself, it would be more difficult to treat mathematically, but it's not clear whether such terms have a biological basis.

Section 8: Conclusions

To treat wAMD, doctors inject medication into the eye that reduces fluid leakage. This in turn should cause increased visual acuity (reflected in the BCVA score) and decreased retinal thickness (reflected in the CSFT score). In this work, we have introduced models for the response of the BCVA score to the medication. We expect the CSFT score to respond similarly, though with different parameter values.

We first posited a linear continuous model. The concentration C is hypothesized to decay linearly over time, with periodic injections raising its value. Similarly, the tolerance T is assumed to follow a linear decay model with forcing proportional to C . This reflects the fact that the body acquires a tolerance to the drug as the concentration increases, but that tolerance will decay away again in the absence of injections.

The maximum effectiveness M of the medication is then a decreasing function of T : we hypothesized three different models for it. In this work, we solve the problem using the simplest: a linear model. However, the model has the disadvantage of going negative for large values of the tolerance, though in practice those values seem never to be reached. The effectiveness factor E , which determines how much of the maximum is achieved with each injection, has two models: a Heaviside-function model which leads to analytical solutions, and a more realistic sigmoid model, which has to be solved numerically.

The analytical solutions of the simplified model demonstrate oscillations in the concentration and tolerance consistent with clinical understanding. Moreover, the solution for a is qualitatively similar to the clinical data. More quantitative analysis is delayed to later chapters.

Given that injections and measurements occur at discrete times, it is natural to consider a discrete map instead of a continuous model. By using a discrete analog of the continuous decay equations for C and T , we were able to construct such a map. We solved the map for C in generic, medically relevant cases. We were also able to perform some basic stability analysis of the map under constant-dose and no-dose conditions.

Our solutions for C showed that the injection dosage was far higher than the half-constant at which we expect the effectiveness to saturate (see Fig. 3.1). This is partly because we ignored diffusion in the simplest model. In practice, the medication is delivered some distance from the target area in the retina. Hence the medication has to diffuse to the retina through the vitreous fluid. Not only is this a slow process, but as the medicine diffuses in all directions, it means the dosage that reaches the retina will be less than that injected.

We constructed a simple free-space model for the concentration that produced profiles with a much smaller peak at the target area (see Fig. 6.2). A more realistic spherical model is presented in a later chapter.

Nomenclature

Units are listed in terms of acuity (A), length (L), mass (M), and time (T). If a symbol appears both with and without tildes, the symbol with tildes has units, while the one without is dimensionless. Equation numbers where a variable is first defined is listed, if appropriate.

- $\tilde{a}(\tilde{t})$: acuity measurement, units A (2.2).
- $\tilde{C}(\cdot)$: medication concentration, units M/L^3 (2.1a).
- \tilde{D} : medication dosage, units M (2.1a).
- $\tilde{\mathcal{D}}$: diffusion constant in the vitreous fluid, units L^2/T (6.1).
- $E(\tilde{C})$: effectiveness factor (2.2)
- $f(\cdot)$: arbitrary function.
- I : injection site.
- i : indexing variable for injections (2.1a).
- J : Jacobian matrix (4.6).
- j : last visit for which $C(t_j) > 1$ (3.8).
- \tilde{k} : rate constant, variable units depending on context.
- $M(T)$: maximum effectiveness factor (2.2).
- m : indexing variable for discrete model.
- N : number of injections (2.1a).
- n : indexing variable for discrete model (4.2a).
- P : maximum number of injection periods.
- p : indexing variable for injection periods.
- \tilde{r} : radial variable, units L (6.2a).
- R : distance, units L (6.1).
- $\tilde{s}(\tilde{t})$: retinal thickness, units L .
- s : Laplace transform variable.
- $T(\tilde{t})$: tolerance (2.2).
- \tilde{t} : time, units T (2.1a).
- $u(r, t)$: scaled solution in the diffusion system (6.4).
- V : volume of eye, units L^3 (2.1a).
- \mathcal{Z} : the integers.
- α : arbitrary constant (4.11).
- θ : azimuthal angle measurement in eye (6.9a).
- κ : scaling factor in discrete models (4.2a).
- $\mu(t)$: integrating factor (7.3a).
- ρ : improvement factor (5.1).
- τ : cutoff time for model.
- ϕ : polar angle measurement in eye.

Other Notation

- A: as a subscript on \tilde{D} , used to indicate treatment A (A.7a).
- a: as a subscript on M , used to indicate algebraic decay (2.5b).
- B: as a subscript on \tilde{D} , used to indicate treatment B (A.8).
- C: as a subscript, used to indicate concentration (2.1a).
- d: as a subscript, used to indicate dosage (4.2a) or diffusion (6.1).
- e: as a subscript, used to indicate exponential decay (2.5b) or the eye (6.1).
- H: as a subscript on E , used to indicate a Heaviside function (2.6b).
- i: as a subscript on R , used to indicate injection.
- $i \in \mathcal{Z}$: index for injection (2.1a).
- in: as a subscript, used to indicate inflow (2.2).
- l: as a subscript on M , used to indicate a linear model (2.5a); otherwise used to indicate the limbus.
- out: as a subscript, used to indicate outflow (2.2).
- p: as a subscript, used to indicate position.
- s: as a subscript on E , used to indicate a sigmoidal function (2.6a).
- T: as a subscript, used to indicate tolerance (2.4).
- 0: as a subscript, used to indicate an initial value (2.1b).
- *: as a subscript on \tilde{C} , used to indicate the half-constant (2.6a).
- $\hat{\cdot}$: denotes the Laplace transform.
- ∞ : as a subscript, used to indicate a steady-state value (2.3).
- −: as a subscript on n , used to indicate the beginning of an injection period.
- +: as a subscript on n , used to indicate the end of an injection period.

References

- [1] Inessa Bekerman, Paul Gottlieb, and Michael Vaiman. Variations in eyeball diameters of the healthy adults. *Journal of Ophthalmology*, 2014, 2014.
- [2] Jan PG Bergmanson and Jesus G Martinez. Size does matter: What is the corneolimb diameter? *Clinical and Experimental Optometry*, 100(5):522–528, 2017.
- [3] W. E. Boyce and R. C. DiPrima. *Elementary Differential Equations*. John Wiley & Sons, 9th edition, 2009.
- [4] Ronald EP Frenkel, Shamim A Haji, Melvin La, Max PC Frenkel, and Angela Reyes. A protocol for the retina surgeon’s safe initial intravitreal injections. *Clinical ophthalmology (Auckland, NZ)*, 4:1279, 2010.
- [5] Zufar Mulyukov, Sebastian Weber, Etienne Pigeolet, Andreas Clemens, Thorsten Lehr, and Amy Racine. Neovascular age-related macular degeneration: A visual acuity model of natural disease progression and ranibizumab treatment effect. *CPT: Pharmacometrics & Systems Pharmacology*, 7, 07 2018.
- [6] Komsan Rattanakijsumtorn, Anita Penkova, and Satwindar S. Sadha. Mass diffusion coefficient measurement for vitreous humor using FEM and MRI. *IOP Conference Series: Materials Science and Engineering*, 297:012024, 2018.
- [7] Charles C Wykoff, W Lloyd Clark, Jared S Nielsen, Joel V Brill, Laurence S Greene, and Cherilyn L Heggen. Optimizing anti-vegf treatment outcomes for patients with neovascular age-related macular degeneration. *J Manag Care Spec Pharm*, 24(2-a Suppl):S3–S15, Feb 2018.

Appendix: Parameter Values

In [5], the authors state the half-life of the medication is 9 days. In our notation,

$$\begin{aligned} e^{-(k_C)(9 \text{ day})} &= \frac{1}{2} \\ (k_C)(9 \text{ day}) &= \log 2 \\ k_C &= \frac{\log 2}{9 \text{ day}} = \frac{7.70 \times 10^{-2}}{\text{day}}. \end{aligned} \quad (\text{A.1})$$

If the spacing of the visits is four weeks, that corresponds to a spacing in t of

$$(28 \text{ day}) \left(\frac{7.70 \times 10^{-2}}{\text{day}} \right) = 2.156. \quad (\text{A.2})$$

In addition, the authors state that \tilde{k}_∞ (in our notation) is given by

$$\tilde{k}_\infty = \frac{\log 2}{14 \text{ day}} = \frac{4.95 \times 10^{-2}}{\text{day}}, \quad (\text{A.3a})$$

from which we have that

$$k_\infty = \frac{4.95 \times 10^{-2}/\text{day}}{7.70 \times 10^{-2}/\text{day}} = 6.43 \times 10^{-1}. \quad (\text{A.3b})$$

To construct these constants for the map, we use the fact that the visits are spaced 4 weeks apart. So we have

$$C_{n+1} = 2^{-28 \text{ day}/9 \text{ day}} C_n \quad \implies \quad \kappa_C = 2^{-28/9} = 1.16 \times 10^{-1} \quad (\text{A.4})$$

and

$$\kappa_\infty = 2^{-28 \text{ day}/14 \text{ day}} = 2^{-2} = 0.25. \quad (\text{A.5})$$

In [5], they indicate that they were unable to estimate C_* well, since with monthly treatments, the dosage effectiveness was always near the saturation value. They end up quoting multiple values; the one in the table is

$$C_* = 2.1 \times 10^{-3} \frac{\text{mg}}{\text{mL}}. \quad (\text{A.6})$$

We have data for two treatment regimes. For the first, (treatment A), we have that the dosage is either of the following two values:

$$\tilde{D}_A = 3 \text{ mg or} \quad (\text{A.7a})$$

$$\tilde{D}_A = 6 \text{ mg.} \quad (\text{A.7b})$$

For treatment B, there is only one dosage:

$$\tilde{D}_B = 2 \text{ mg.} \quad (\text{A.8})$$

The volume of the vitreous fluid is [5]

$$V = 4 \text{ mL.} \quad (\text{A.9})$$

Then using (A.6), we have the normalized values

$$D_A = \frac{3 \text{ mg}}{(4 \text{ mL})(2.1 \times 10^{-3} \text{ mg/mL})} = 357 \text{ or} \quad (\text{A.10a})$$

$$D_A = \frac{6 \text{ mg}}{(4 \text{ mL})(2.1 \times 10^{-3} \text{ mg/mL})} = 714, \quad (\text{A.10b})$$

while

$$D_B = \frac{2 \text{ mg}}{(4 \text{ mL})(2.1 \times 10^{-3} \text{ mg/mL})} = 238. \quad (\text{A.11})$$

These numbers seem very large; do they make sense? In [5], they indicate that only after three months did the dosage decrease enough to fall noticeably beyond saturation. Three months (thirteen weeks) after dosage, under the discrete model with the maximum dosage we would have

$$C = 2^{-91/9}(714) = (9.04 \times 10^{-4})(714) = 0.646, \quad (\text{A.12})$$

which is less than the dimensionless threshold value 1. So this does agree.

The diffusion coefficient in the vitreous fluid can be roughly estimated using the data in [6] as

$$\tilde{D} = 2 \times 10^{-6} \frac{\text{cm}^2}{\text{s}} = 2 \times 10^{-4} \frac{\text{mm}^2}{\text{s}}. \quad (\text{A.13})$$

In Bekerman [1] they cite a value for the diameter of the eyeball of around 25 mm, which means that

$$R_e = 12.5 \text{ mm.} \quad (\text{A.14})$$

Therefore, the value of t_d as given in (6.1) can be computed as

$$\begin{aligned} t_d &= \frac{\pi^2(1.25 \text{ cm})^2}{4(2 \times 10^{-6}) \text{ cm}^2/\text{s}} = \left(\frac{15.4}{8} \times 10^6 \text{ s} \right) \left(\frac{\text{hr}}{3600 \text{ s}} \right) = \left(\frac{1.925}{3.6} \times 10^3 \text{ hr} \right) \left(\frac{\text{day}}{24 \text{ hr}} \right) \\ &= 22.3 \text{ day.} \end{aligned} \quad (\text{A.15})$$

In [2] they cite many measurements of the limbal diameter; using a rough average, we obtain

$$R_l = 6 \text{ mm.} \quad (\text{A.16})$$

We want to compute the maximum possible diffusion distance R_d , which occurs in pseudophakic or aphakic patients [4]. For those patients, they indicate that the injection site is at a distance R_p given in the following range:

$$3.0 \text{ mm} \leq R_p \leq 3.5 \text{ mm}, \quad (\text{A.17})$$

and the injection distance R_i is given in the following range:

$$4.0 \text{ mm} \leq R_i \leq 6.0 \text{ mm}. \quad (\text{A.18})$$

To obtain the maximal distance R_d , we will use the minimum values of each of these ranges.

Substituting these results into (6.10), we have

$$\begin{aligned} R_d^2 &= (12.5 \text{ mm})^2 + [(12.5 - 4.0) \text{ mm}]^2 \\ &\quad + 2(4.0 \text{ mm}) \times \\ &\quad \left[(6 \text{ mm}) \cos\left(\frac{3.0 \text{ mm}}{12.5 \text{ mm}}\right) - \sqrt{(12.5 \text{ mm})^2 - (6 \text{ mm})^2} \sin\left(\frac{3.0 \text{ mm}}{12.5 \text{ mm}}\right) \right] \\ &= \{156.25 + 72.25 + 8 [6 \cos(0.24) - \sqrt{156.25 - 36} \sin(0.24)]\} \text{ mm}^2 \\ R_d &= \left\{ 228.5 + 8 \left[6(0.971) - \sqrt{120.25}(0.238) \right] \right\}^{1/2} \text{ mm} \\ &= \{228.5 + 8 [5.826 - 11.0(0.238)]\}^{1/2} \text{ mm} = [228.5 + 8 (5.826 - 2.61)]^{1/2} \text{ mm} \\ &= [228.5 + 8(3.22)]^{1/2} \text{ mm} = [228.5 + 25.76]^{1/2} \text{ mm} = (254.26)^{1/2} \text{ mm} \\ &= 15.9 \text{ mm}, \end{aligned} \quad (\text{A.19})$$

which means that the dimensionless values become

$$\begin{aligned} \mathcal{D} &= \left(2 \times 10^{-4} \frac{\text{mm}^2}{\text{s}} \right) \frac{1}{(15.9 \text{ mm})^2 (7.70 \times 10^{-2} / \text{day})} \left(\frac{3600 \times 24 \text{ s}}{\text{day}} \right) \\ &= \frac{172.8 \times 10^1}{(252.81)(7.70)} = 8.88 \times 10^{-1}, \end{aligned} \quad (\text{A.20})$$

$$\begin{aligned} D_d &= \frac{2 \text{ mg}}{(15.9 \text{ mm})^3} \left(2.1 \times 10^{-3} \frac{\text{mg}}{\text{mL}} \right)^{-1} = \frac{.952 \times 10^3 \text{ mL}}{(1.59 \text{ cm})^3} = \frac{9.52 \times 10^2}{4.02} \\ &= 237, \end{aligned} \quad (\text{A.21})$$

where we use the value of \tilde{D} from administration B for comparison. Note the close agreement between (A.21) and (A.11).

However, there is still one discrepancy. If we compute the volume of a sphere using (A.14), we obtain

$$V = \frac{4\pi}{3} (1.25 \text{ cm})^3 = (4.19)(1.95) \text{ mL} = 8.17 \text{ mL}, \quad (\text{A.22})$$

which is twice the value in (A.9). Perhaps the vitreous fluid occupies just a portion of the interior of the eyeball.

Analytical Models and Numerical Solutions

Kristian Kiradjiev
Mathematical Institute, University of Oxford

August 1, 2019

1 Analysis on the Mathematical Model

We consider the following system of differential equations for the concentration, C , tolerance, T , and acuity, a .

$$\frac{dC}{dt} = -C + D \sum_{i=0}^N \delta(t - t_i), \quad (1)$$

$$\frac{dT}{dt} = k_T C - k_\infty T, \quad (2)$$

$$\frac{da}{dt} = k_{\text{in}}(1 + M(T)E(C)) - k_{\text{out}}a, \quad (3)$$

together with the initial conditions

$$C(0) = 0, \quad (4)$$

$$T(0) = 0, \quad (5)$$

$$a(0) = 1, \quad (6)$$

where $M(T)$ is the maximum effectiveness, and $E(C)$ is the effectiveness factor. In order to avoid negativity issues, we model $M(T) = M_\infty(1 - T)H(1 - T)$ and for the effectiveness we take $E(C) = H(C - 1)$. The first one says that once the tolerance becomes the maximum possible, i.e., 1, the maximum effectiveness drops to zero, while the second one says that once the concentration becomes below zero, the effectiveness factor is zero. Alternatively, one could impose a smooth convergence to zero and take $M(T) = M_\infty/(1 + T)$ and $E(C) = C/(C + 1)$.

We can solve the equation for C straight away to obtain

$$C(t) = D \sum_{i=0}^N H(t - t_i) e^{-(t-t_i)}. \quad (7)$$

Similarly, we can use an integrating factor to solve the equation for T ,

$$T(t) = \frac{k_T D}{k_\infty - 1} \sum_{i=0}^N H(t - t_i) [e^{-(t-t_i)} - e^{-k_\infty(t-t_i)}]. \quad (8)$$

Finally, we can turn to the equation for a and solve it again by an integrating factor. We note that in practical application $C > 1$, so $E(C) = 1$, and we will integrate our equation until time τ_1 when $T = 1$. This way, $M(T) = M_\infty(1 - T)$. We use an integrating factor of $e^{k_{\text{out}}t}$ for the equation for a and obtain

$$a(t) = \left[1 - \frac{k_{\text{in}}(1 + M_\infty)}{k_{\text{out}}} \right] e^{-k_{\text{out}}t} + \frac{k_{\text{in}}(1 + M_\infty)}{k_{\text{out}}} + \frac{M_\infty k_T k_{\text{in}} D}{k_\infty - 1} \times \sum_{i=0}^N H(t - t_i) \left[\frac{e^{-k_\infty(t-t_i)}}{k_{\text{out}} - k_\infty} - \frac{e^{-(t-t_i)}}{k_{\text{out}} - 1} + \left(\frac{1}{k_{\text{out}} - 1} - \frac{1}{k_{\text{out}} - k_\infty} \right) e^{-k_{\text{out}}(t-t_i)} \right]. \quad (9)$$

Criterion for Initial Effectiveness Initially we give a dose $D > 1$, so $E(C) = 1$, and since $T = 0$, then $M(T) = M_\infty$, and so from the equation for a , with $a(0) = 1$, we see that initially $\dot{a} \geq 0$ provided

$$k_{\text{in}}(1 + M_\infty) - k_{\text{out}}a(0) \geq 0, \quad \text{i.e.,} \quad \frac{k_{\text{in}}(1 + M_\infty)}{k_{\text{out}}} \geq 1. \quad (10)$$

2 Numerical Results

We now solve (1)–(6) numerically using *Mathematica*. We first consider the case when $M(T) = M_\infty(1 - T)H(1 - T)$ and $E(C) = H(C - 1)$. In figures 1–6, we show plots of

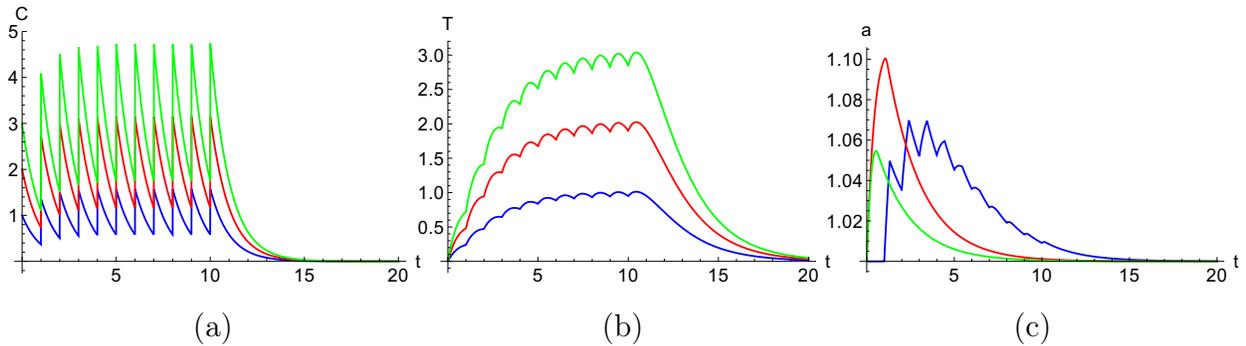


Figure 1: Plots of the (a) concentration, (b) tolerance, and (c) acuity varying d : 1 (blue), 3 (red), 5 (green).

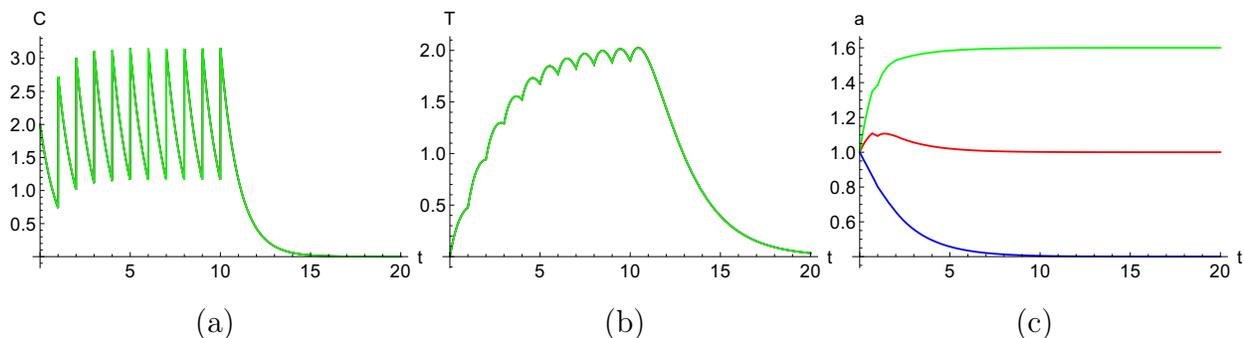


Figure 2: Plots of the (a) concentration, (b) tolerance, and (c) acuity varying k_{in} : 0.2 (blue), 0.5 (red), 0.8 (green).

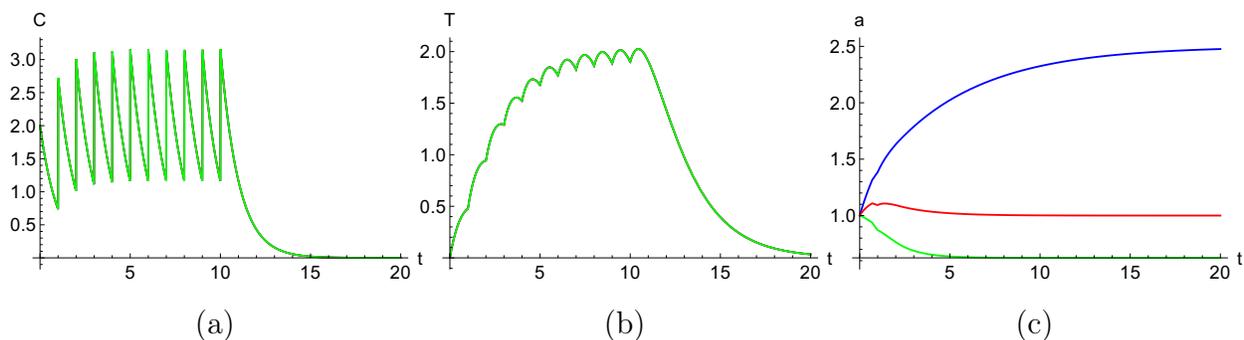


Figure 3: Plots of the (a) concentration, (b) tolerance, and (c) acuity varying k_{out} : 0.2 (blue), 0.5 (red), 0.8 (green).

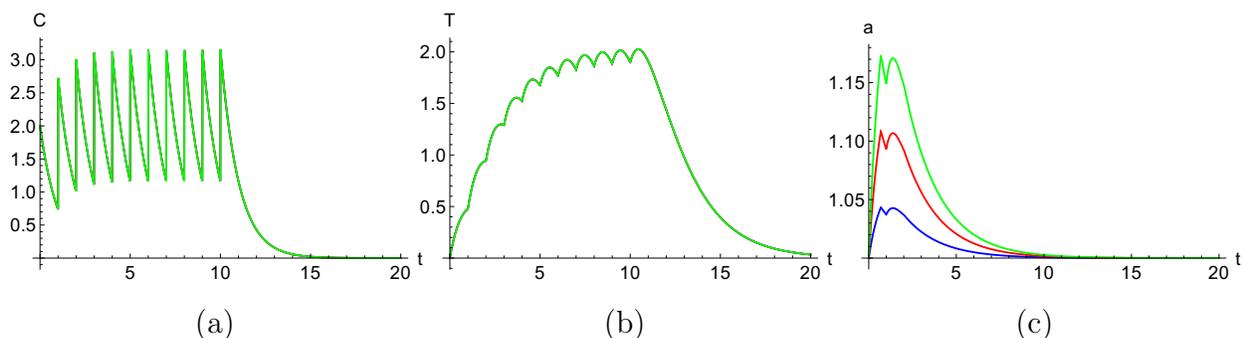


Figure 4: Plots of the (a) concentration, (b) tolerance, and (c) acuity varying M_{∞} : 0.2 (blue), 0.5 (red), 0.8 (green).

the concentration (figures (a)), tolerance (figures (b)), and acuity (figures (c)) for different values of the parameters d , k_{in} , k_{out} , M_{∞} , k_T , and k_{∞} , respectively. In all plots we give dosage at every dimensionless time unit until $t = 10$. We see that as we increase the dosage the peaks of the concentration increase as expected and tolerance increases as well. Acuity on the other hand first increases but then decreases as the high tolerance inhibits it. We see

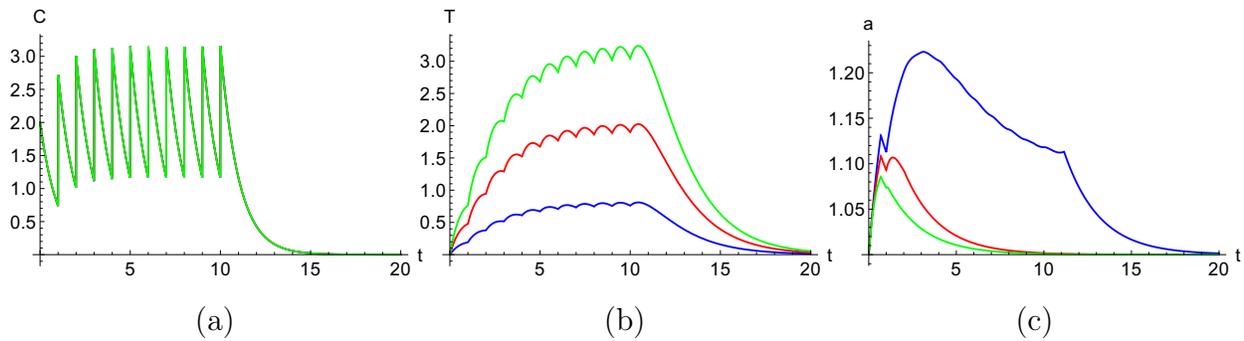


Figure 5: Plots of the (a) concentration, (b) tolerance, and (c) acuity varying k_T : 0.2 (blue), 0.5 (red), 0.8 (green).

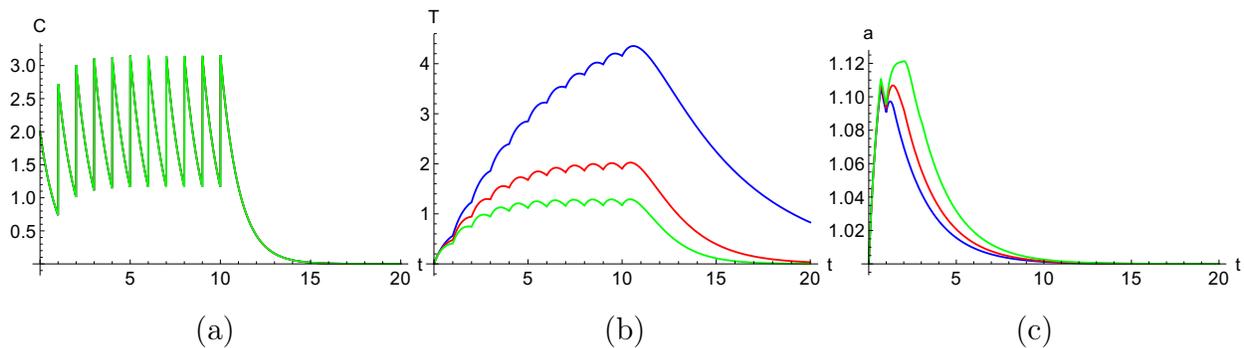


Figure 6: Plots of the (a) concentration, (b) tolerance, and (c) acuity varying k_∞ : 0.2 (blue), 0.5 (red), 0.8 (green).

how increasing k_T and k_∞ increases and decreases the tolerance, respectively. We see a very interesting effect on the acuity when we change k_{in} and k_{out} , as past a certain threshold, it changes from a decreasing to an increasing function even when we stop the dosage. When increasing M_∞ , acuity expectedly increases. When we increase k_T and k_∞ , acuity decreases and increases, respectively.

We note that when we plot the same results but using $M(T) = M_\infty/(1+T)$ instead (not shown), we get the same qualitative results but for different parameter values.

If we now use $E(C) = C/(1+C)$, we obtain the following plots for the acuity. We again see similar qualitative behaviour in figure 7, albeit more smoothed out, due to the sigmoid function. We also see a characteristic rise when changing k_∞ , because now $E(C)$ is not zero when C falls below 1, and before $T = 1$, it is positive.

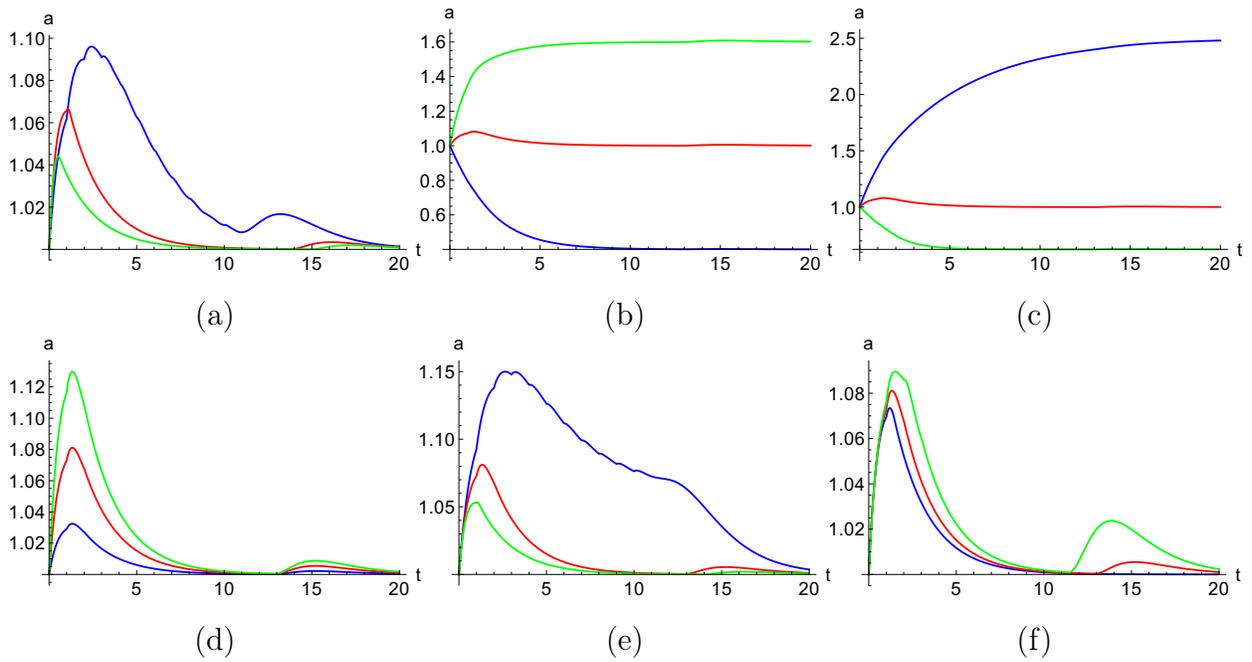


Figure 7: Plots of the acuity varying (a) d , (b) k_{in} , (c) k_{out} , (d) M_{∞} , (e) k_T , (f) k_{∞} . Parameter values are the same as in the corresponding figure 1–6.

Parameter Fitting the Continuous Model

Lauren Barnes (New Jersey Institute of Technology)
Jeff Slepoi (Stevens Institute of Technology)
Daniel Jonas (Colorado State University)
Steve Hussung (Indiana University)

June 2019

1 Introduction

We investigated the continuous model based on a system of three ordinary differential equations (ODEs). The states captured within the model included the Best-Corrected Visual Acuity score (BCVA) $a(t)$, immune system tolerance to the Neovascular Age-Related Macular Degeneration (nAMD) drug $T(t)$, and the drug concentration $C(t)$, at time t . The acuity differential equation (DE) is given by:

$$\dot{a} = k_{\text{in}}[1 + M(T)E(C)] - k_{\text{out}}a, \quad a(0) = 1, \quad (1)$$

where $E(C)$ is the dose effectiveness function, $M(T)$ tempers the drug-dose maximum effectiveness, $k_{\text{in}} \in \mathbb{R}_{>0}$ is the growth parameter, and $k_{\text{out}} \in \mathbb{R}_{>0}$ is the decay parameter. We considered two forms for the dose effectiveness:

$$E(C) = H(C - 1), \quad \text{or} \quad (2a)$$

$$E(C) = \frac{C}{1 + C}, \quad (2b)$$

where $H(C - 1) = \mathbb{1}(C)_{[1, \infty)}$ is the Heaviside function with step at $C = 1$; and the following three forms for the dose maximum effectiveness:

$$M(T) = M_{\infty}(1 - T), \quad \text{or} \quad (3a)$$

$$M(T) = \frac{M_{\infty}}{1 + T}, \quad \text{or} \quad (3b)$$

$$M(T) = M_{\infty}e^{-T}, \quad (3c)$$

where $M_{\infty} \in \mathbb{R}_{>0}$ is the maximum effectiveness parameter.

Every few weeks a patient is given an injection of the nAMD drug, which for simplicity is assumed to occur instantaneously, while the concentration of the medicine decays exponentially over time at the rate $k_C \in \mathbb{R}_{>0}$. The administration times form an increasing sequence of non-negative integers $\{t_i\}_{i=0}^{N_p}$, for each of $p \in \mathbb{N}$ patients, and the doses are of equal strength $D \in \mathbb{R}_{>0}$. The drug concentration is therefore modeled by the equation:

$$\dot{C} = -k_C C + D \sum_{i=0}^N \delta(t - t_i), \quad t_0 = 0, \quad C(0) = 0. \quad (4)$$

The Dirac delta function $\delta(t - t_i)$ was used to model the instantaneous injections at the times t_i . We point out that the value of k_C is known but the equation can be normalized using it by applying $t \rightarrow t/k_C$, effectively fixing its value at 1.

An individual's tolerance to the effectiveness of the drug increases with the concentration at the rate $k_T \in \mathbb{R}_{>0}$ and decreases to a baseline value in the absence of the drug at a rate $k_\infty = 6.43 \times 10^{-2}$. The tolerance DE is:

$$\dot{T} = k_T C - k_\infty T, \quad T(0) = 0. \quad (5)$$

For each patient, the dosage-sequence $\{t_i\}_{i=0}^{N_p}$ is taken directly from the data. The parameters k_{in} , k_{out} , M_∞ , and k_T need to be identified for each patient and were therefore estimated from the data.

2 Data

The format of the original data file was challenging and thus demanded a healthy dose of cleaning. In particular, the measured variables were lumped into a single column and needed to be extracted and positioned into columns of their own, aggregated by patient and sorted within by time. The R-script *Clean Certara Data* was written to reorganize the data in the needed way by extracting the BCVA and the Central Sub-Field Thickness (CSFT) values, as well as the corresponding dosage types and injection days, and producing a column for each. Note that this study primarily concerned itself with investigating the BCVA scores and will be extended to include CSFT dynamics. Different patient time series could easily be obtained using one of three output files: *BCVA_Data.csv*, *CSFT_Data.csv*, and *BCVA_CSFT_Equalized_Data.csv*. Note that the latter file is the result of taking the intersection of the former two files on the observation times, as both variables were not always measured in the same visit.

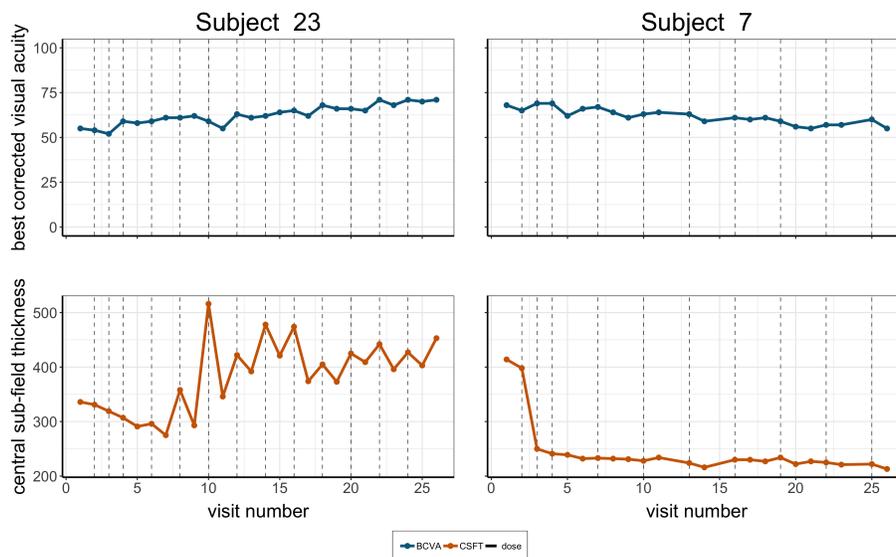


Figure 1: Data from example patients labeled Subject 23 and Subject 7. The BCVA score per visit corresponds to the solid blue curve, while the CSFT measurement per visit corresponds to the dashed orange curve. The vertical dotted lines represent visits in which a drug injection was administered. Subject 23 improves in terms of BCVA score, while Subject 7 worsens over time. The former's CSFT increases, while the latter's decreases suggesting a complicated relationship between BCVA scores and CSFT measurements, as lower CSFT values are typically associated with a healthier pair of eyes.

Pretty time series plots were then produced by the R-scripts *Visualizing Certara Data Overlaid Time Series* and *Visualizing Certara Data 2x2 Plots*. The first program visualizes the BCVA and CSFT data for each patient, along with

the visits in which they were given a dosage of the treatment (see Figure 3), allowing for an informal and preliminary comparison of the dynamics. The second organizes the time series of two patients' BCVA data into the first row of a 2×2 grid and the CSFT data into the second row, such that each column corresponds to a subject, as in Figure 1. These plots were designed to demonstrate complexities and similarities within the data across patients.

We also looked at simple time delay embeddings of the data via the R script *Visualizing Certara Data Time Delayed Embedding*. For the ordered time series $X = \{x_i\}_{i=1}^N$, we examined the 1-delay embedding (x_k, x_{k+1}) for $1 \leq k \leq N - 1$. Given the oscillatory nature of the BCVA data, we were hoping to see some telling structure after plotting the embedded points but the results varied wildly between patients and were difficult to interpret as in Figure 2. Perhaps further studies in this direction could be fruitful, but we opted to focus on other analyses.

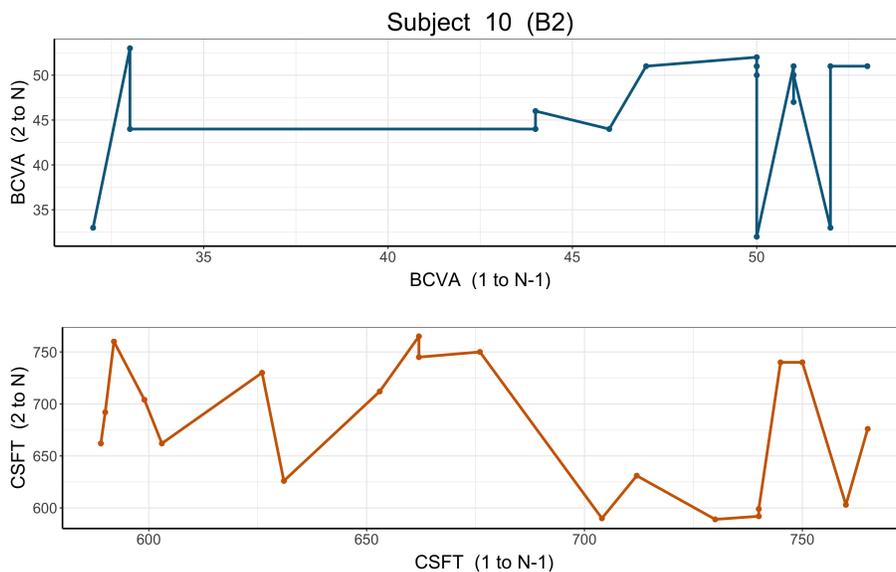


Figure 2: Time-delayed embeddings of the BCVA scores (top) and CSFT values (bottom) for the patient dubbed Subject 10. Both depict the result of a 1-delay embedding.

3 Parameter Estimation

3.1 Numerics

We developed a program to calculate concentration, tolerance, and acuity $(C(t), T(t), a(t))$ using a set of common baseline parameters. We used $k_T = 0.001$, $M_\infty = 0.5$, $k_{\text{in}} = 0.45$, and $k_{\text{out}} = 0.5$ to initially evaluate the model. The data was loaded into the two structures BCVA and DOSE which contained visits data and days of injection respectively. Dose of injection in the data file was stored as a number between 2 and 6 but represented the actual dosage of 238mg and 714mg; therefore we needed to use the factor of 119 to scale the dosage back to the actual value.

We numerically solved the system of ODEs in MATLAB utilizing the *ode45* numerical integration method. Since the Dirac delta function cannot be handled by *ode45*, we coded the concentration as $\dot{C} = -C$ and integrated on intervals between dosages determined by the sequence $\{t_i\}_{i=0}^{N_p}$. After the first interval $[t_0, t_1)$, we recalculated the initial condition to be the ending concentration plus the dose-value and integrated over the next interval $[t_1, t_2)$, until the next injection (see the MATLAB function *func_BCVA*). This process amounted to solving the system of ODEs comprised

of Equations (1), (4), and (5).

Please note that we also normalized the BCVA time series for each patient by dividing each value by the patient's score on the first visit. This preprocessing ensures that each patient's initial acuity score is 1, which agrees with the initial condition in Equation (1).

Once we were able to obtain numerical solutions for the system of ODEs, we moved on to finding the best set of parameters for each patient by utilizing the MATLAB least squares curve fit *lsqcurvefit*. Our goal was to estimate the parameters k_T , k_{in} , k_{out} , and M_∞ from the experimental data we were given. We applied the method to the vectors of values for $(C(t), T(t), a(t))$ resulting from numerically integrating Equations (1), (4), and (5).

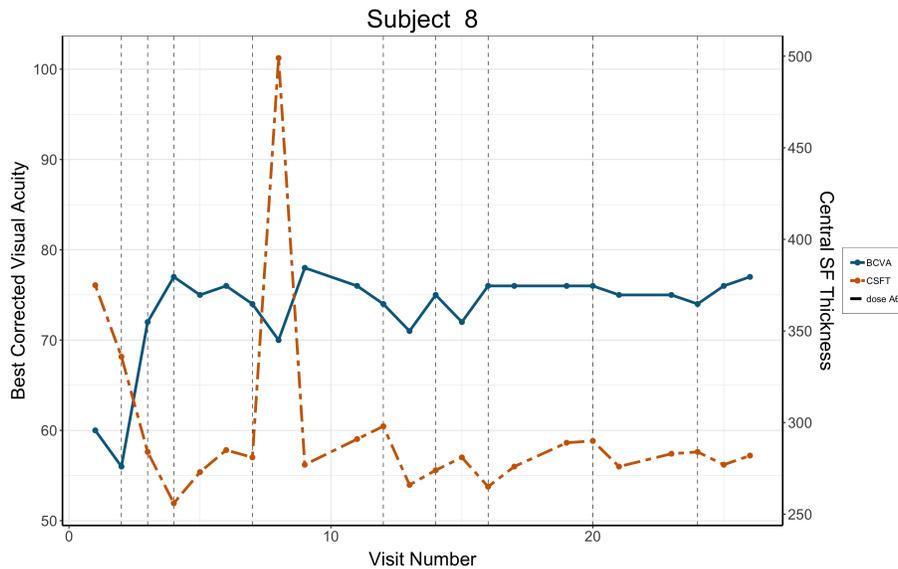


Figure 3: Data from an example patient labeled Subject 8. The BCVA score per visit corresponds to the solid blue curve, while the CSFT measurement per visit corresponds to the dashed orange curve. The vertical dotted lines represent visits in which a drug injection was administered.

To use the algorithm we first needed to modify the function *func_BVCA* to return only the values of the continuous acuity function $a(t)$ at the dosage times $\{t_i\}_{i=0}^{N_p}$, that is, the actual visits on which the BCVA of the patient was measured. The resulting new function is titled *model_BCVA*, which is called from the main routine script *Fit_contin*. We also assumed that each parameter should be within certain limits: $0 < k_T < 0.2$, $0 < M_\infty < 1000$, $0 < k_{in} < 1000$, and $0 < k_{out} < 1$ in our use of *lsqcurvefit*. The program returns the parameter estimates and then produces the ODE output trajectories with these best fit parameters and displays them in graphs, as seen in Figure 4.

Finally, we wanted to estimate the unknown parameters for all patients utilizing the six possible models produced by considering all combinations of $M(T)$ and $E(C)$. The program *Fit_contin_Loop* runs the function *func_Fit_contin*, which is based on the program *Fit_contin*, for each patient six times. The output is stored into a matrix and then a text file, populating the eight columns: patient number, version number for Maximum effectiveness (1, 2, or 3), version number for Dose effectiveness (1 or 2), residual norm returned by *lsqcurvefit*, k_T , M_∞ , k_{in} , and k_{out} . The results are stored in double precision format, while the text file contains the parameter values with eight digits, four after the decimal point (8.4f).

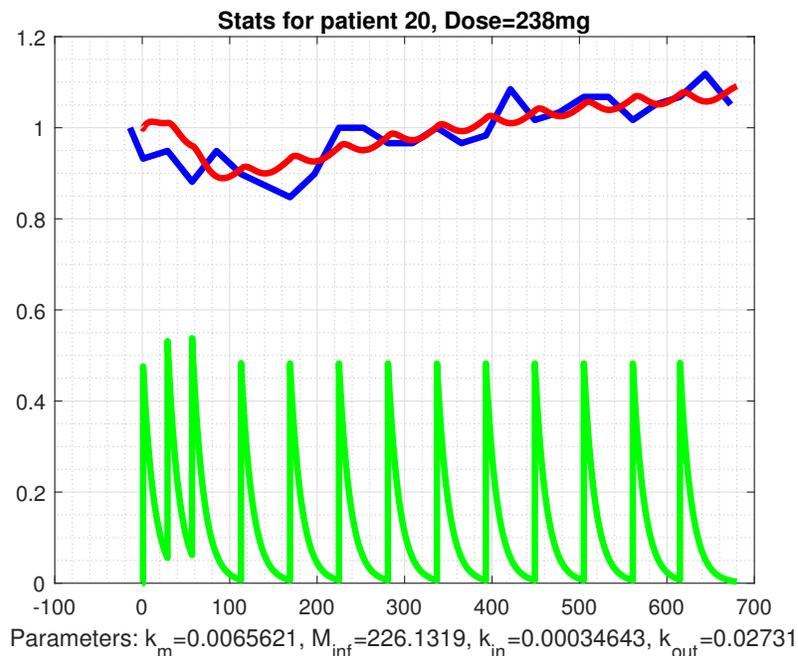


Figure 4: Model output with best-fit parameters generated for an example patient. Blue: Actual acuity data. Red: Best-fit curve. Green: Concentration.

The output from our model is given in section 3.3.

3.2 Comparison of the Models

We also compared results from the *lsqcurvefit* code for different combinations of the three different models for M and E . Using MATLAB, plots were made for each of the parameters (k_T , M_{∞} , k_{in} , and k_{out}). Each plot is a histogram representing, for a given parameter, the fitted values of that parameter (*i.e.*, the outputs of the MATLAB code using *lsqcurvefit*) for all 26 of the patients whose data was given. This was done for each of the 6 possible combinations of the models for dose effectiveness $E(C)$ (given in Eqs (2)a,b) and maximum effectiveness $M(T)$ (given in Eqs (3)a,b,c).

Looking at these histograms can not only assist us in making a more educated guess as to what values for parameters k_{in} , k_{out} , M_{∞} , and k_T should be used in a particular (specified $E(C)$ and $M(T)$) version of the model; it can also help us to compare and contrast all 6 possible combinations of $E(C)$ and $M(T)$ formulas in the model. While it won't tell us which combination is "best", it can at least give an indication of which one may be the most consistent for all 26 patients, in terms of the parameter values (as fitted by the MATLAB code).

The histograms are shown in Figures 5 (for k_{in}), 6 (for k_{out}), 7 (for M_{∞}), and 8 (for k_T).

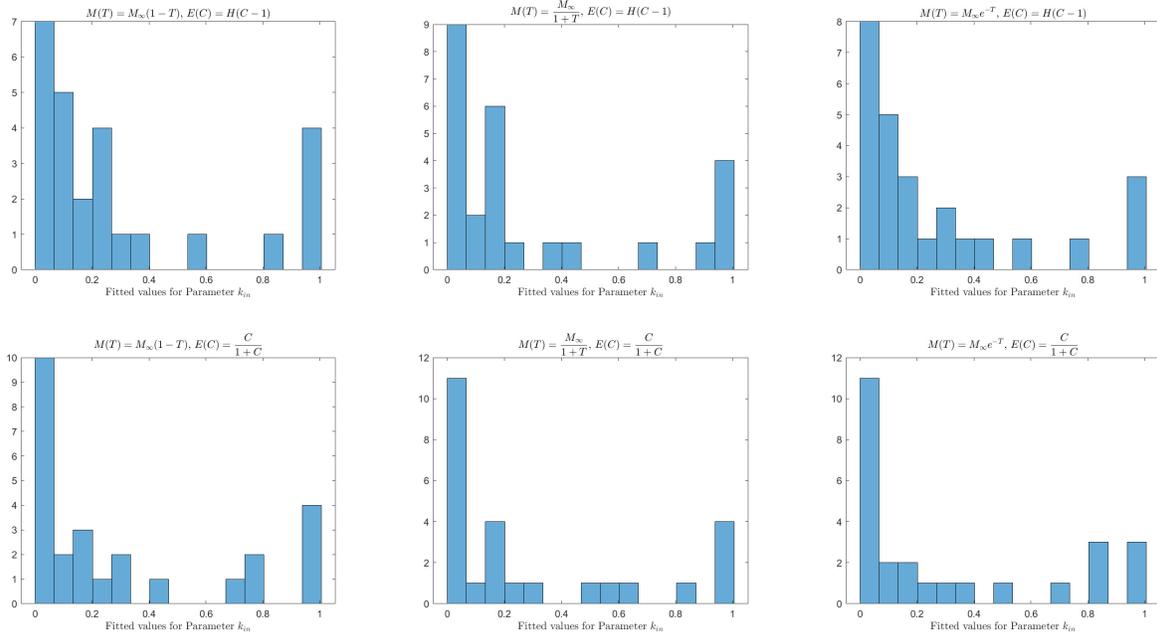


Figure 5: Histograms of the fitted values, given by the code, for the parameter k_{in} (taking into account all 26 patients)

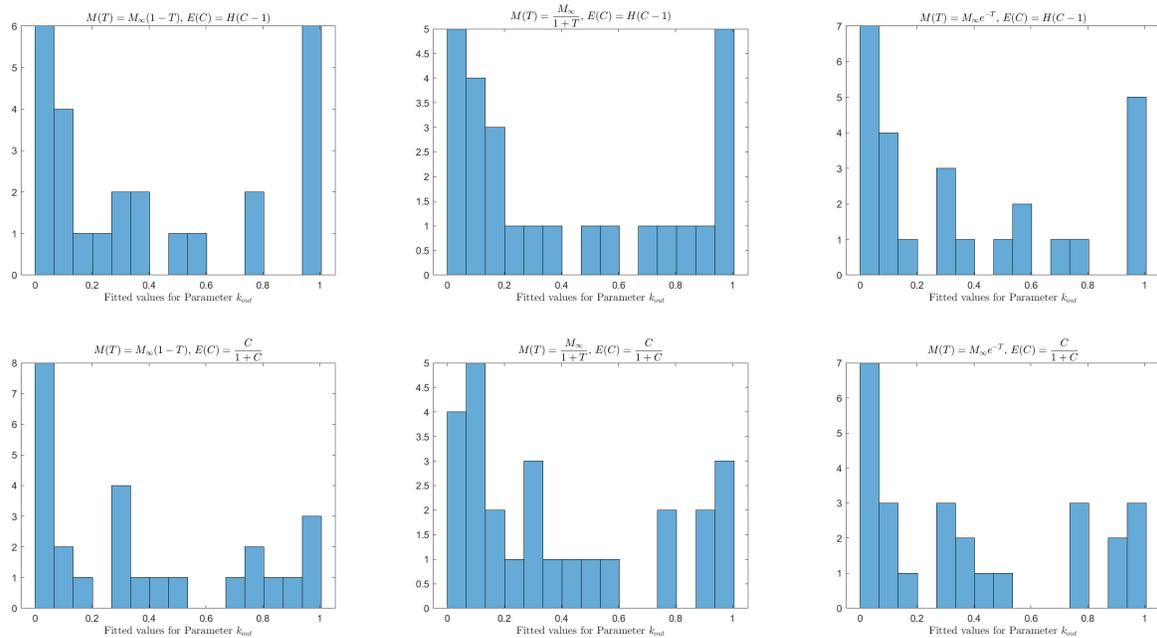


Figure 6: Histograms of the fitted values, given by the code, for the parameter k_{out}

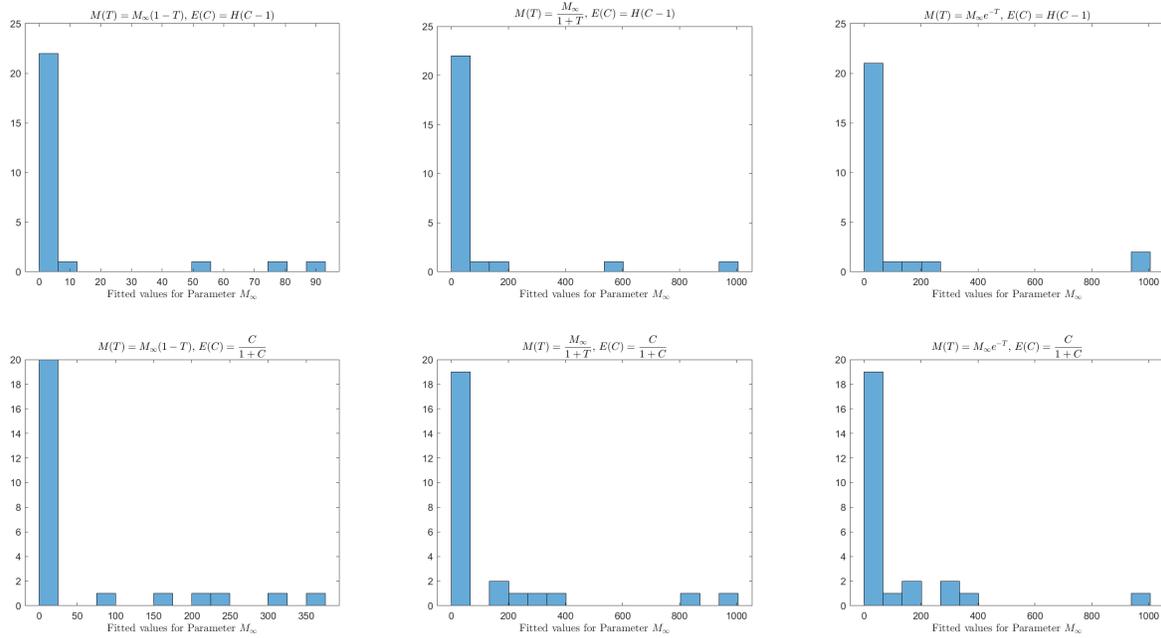


Figure 7: Histograms of the fitted values, given by the code, for the parameter M_∞

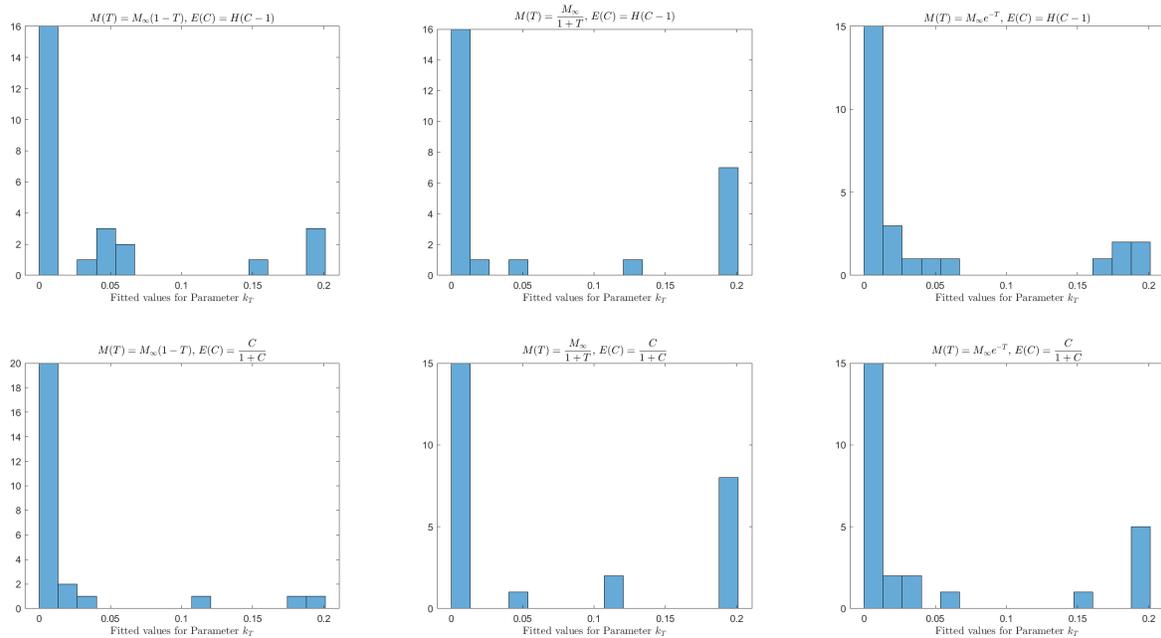


Figure 8: Histograms of the fitted values, given by the code, for the parameter k_T .

3.3 Code Output

We list below the output from the parameter-fitting code for 26 patients whose measurements come from the real data file provided to us. Models for M and E are reflected in equations (3) and (2) respectively. (NB: for patients 1,6 and

21 the data included unscheduled visits without the actual day of the visit. These unscheduled visits were ignored.)

Patient #	Model for M	Model for E	Residual Norm	k_T	M_∞	k_{in}	k_{out}
1	1	1	0.1424	0.0437	0.0379	0.0479	0.0397
1	1	2	0.1425	0.0244	0.0693	0.0513	0.0442
1	2	1	0.1424	0.0196	1.247	0.0282	0.0388
1	2	2	0.1422	0.0132	1.8973	0.0253	0.0435
1	3	1	0.1424	0.0167	0.7571	0.0345	0.0383
1	3	2	0.1423	0.0124	0.8658	0.0342	0.0409
2	1	1	0.0201	0	0.7016	0.0704	0.1086
2	1	2	0.018	0	14.523	0.0039	0.0489
2	2	1	0.0207	0	0.667	0.0737	0.1122
2	2	2	0.018	0	14.0965	0.0041	0.049
2	3	1	0.024	0	0.0634	0.3075	0.3124
2	3	2	0.018	0	15.1522	0.0038	0.0487
3	1	1	0.0391	0.0459	0	0.8365	0.7665
3	1	2	0.0391	0.0066	0	0.7952	0.7286
3	2	1	0.0391	0	0	0.8831	0.8093
3	2	2	0.0391	0.2	0	0.8585	0.7867
3	3	1	0.0391	0	0	0.8006	0.7336
3	3	2	0.0391	0.0073	0	0.8526	0.7813
4	1	1	0.0475	0	0	0.947	1
4	1	2	0.0475	0	0	0.947	1
4	2	1	0.0476	0	0	0.9422	0.995
4	2	2	0.0476	0	0	0.9462	0.9992
4	3	1	0.0476	0	0	0.9457	0.9987
4	3	2	0.0468	0.2	1.028	0.9449	1
5	1	1	0.1057	0	0.625	0.2057	0.3358
5	1	2	0.1042	0	0.9271	0.1516	0.2846
5	2	1	0.1044	0	0.6453	0.1939	0.3196
5	2	2	0.1042	0	0.9285	0.1512	0.2841
5	3	1	0.1051	0	0.6642	0.1844	0.3081
5	3	2	0.1042	0	0.9285	0.1512	0.2841
6	1	1	0.0252	0.0001	0.0965	0.2035	0.2064
6	1	2	0.0259	0	0.0167	0.2833	0.2693
6	2	1	0.0252	0.1999	0.6273	0.6923	0.6785
6	2	2	0.0259	0.2	0	0.3172	0.2976
6	3	1	0.0253	0.022	0.103	0.5687	0.5462
6	3	2	0.0259	0.1999	0	0.3155	0.296
7	1	1	0.0189	0	3.4421	0.2275	1
7	1	2	0.0238	0.0015	0.4888	0.7266	0.9998
7	2	1	0.0185	0	6.147	0.1414	0.9999
7	2	2	0.0235	0.0022	0.7124	0.6522	1
7	3	1	0.0187	0	4.3537	0.1888	1
7	3	2	0.0237	0.0018	0.6161	0.6813	1
8	1	1	0.059	0.2	0.0004	0.3922	0.3484
8	1	2	0.0589	0.1747	0.0007	0.464	0.4116
8	2	1	0.0592	0.2	0	0.3926	0.3505
8	2	2	0.0586	0.2	0.349	0.4837	0.4434

Patient #	Model for M	Model for E	Residual Norm	k_T	M_∞	k_{in}	k_{out}
8	3	1	0.0591	0.0641	0.1014	0.4266	0.3829
8	3	2	0.0582	0.0628	0.1893	0.4896	0.4425
9	1	1	2.4497	0	6.6254	0.1312	0.9999
9	1	2	2.3567	0	301.5364	0.0014	0.4012
9	2	1	2.4505	0	3.8491	0.2063	0.9999
9	2	2	2.3567	0	318.3283	0.0013	0.401
9	3	1	2.4512	0	2.6856	0.2714	0.9998
9	3	2	2.3567	0	323.6229	0.0013	0.401
10	1	1	0.2066	0	0	0	0.0077
10	1	2	0.2066	0	0	0	0.0077
10	2	1	0.2066	0.001	0	0	0.0077
10	2	2	0.2066	0.0005	0	0	0.0077
10	3	1	0.4044	0.1998	86.4827	0.0914	0.1295
10	3	2	0.2066	0.2	0	0	0.0077
11	1	1	0.0177	0.0001	0.0739	0.1832	0.1665
11	1	2	0.0174	0.0011	0.1625	0.1651	0.1581
11	2	1	0.0178	0	0.0506	0.1922	0.1713
11	2	2	0.0174	0.0011	0.1605	0.1632	0.1562
11	3	1	0.0177	0.0001	0.0743	0.1836	0.1669
11	3	2	0.0174	0.0012	0.1631	0.165	0.1581
12	1	1	0.0092	0.2	0.0011	0.3225	0.2886
12	1	2	0.0093	0.2	0.001	0.3151	0.2824
12	2	1	0.0097	0	0	0.188	0.1694
12	2	2	0.0095	0.2	0.4308	0.2319	0.2157
12	3	1	0.0061	0.1812	36.2021	0.0814	0.0767
12	3	2	0.0074	0.1905	1.985	0.3714	0.343
13	1	1	0.0003	0.0531	0.1946	0.0689	0
13	1	2	0.0002	0.0083	94.3315	0.001	0.0033
13	2	1	0.0007	0.2	145.6247	0.0098	0.0825
13	2	2	0.0006	0.1152	137.406	0.0075	0.09
13	3	1	0.0006	0.0412	165.5629	0.0051	0.0347
13	3	2	0.0006	0.0146	189.9139	0.0014	0.0584
14	1	1	0.2757	0.0301	0	0.0566	0.0425
14	1	2	0.2723	0	229.5346	0.0002	0.0387
14	2	1	0.2759	0	0.0001	0.0607	0.0459
14	2	2	0.2723	0	158.1318	0.0003	0.0388
14	3	1	0.2772	0	0.0057	0.0713	0.0551
14	3	2	0.2723	0	84.2667	0.0007	0.0389
15	1	1	0.7897	0.001	0.8546	0.9997	1
15	1	2	0.8075	0.0024	0.7221	1	0.8559
15	2	1	0.7891	0.0013	0.8658	1	1
15	2	2	0.8009	0.0048	0.8941	1	0.8898
15	3	1	0.7894	0.0011	0.8597	1	1
15	3	2	0.8039	0.0033	0.8016	1	0.8738
16	1	1	0.0034	0	0.0426	0.0838	0.0756
16	1	2	0.0033	0	0.0741	0.0723	0.0661
16	2	1	0.0038	0.2	0.0001	0.1536	0.1375
16	2	2	0.004	0.2	0.0055	0.1722	0.1547

Patient #	Model for M	Model for E	Residual Norm	k_T	M_∞	k_{in}	k_{out}
16	3	1	0.0033	0.17	7.5866	0.0952	0.0851
16	3	2	0.0035	0.1589	0.1138	0.1095	0.0968
17	1	1	0.2754	0.1901	0.0236	0.1296	0.0737
17	1	2	0.2756	0.0048	11.7322	0.0122	0.0809
17	2	1	0.28	0.2	102.6384	0.0112	0.0748
17	2	2	0.2719	0.2	248.4826	0.0067	0.0925
17	3	1	0.2847	0.0097	21.9226	0.0064	0.0617
17	3	2	0.2786	0.0081	155.457	0.0013	0.0888
18	1	1	0.2218	0.0594	0.2176	0.0624	0
18	1	2	0.2198	0.0105	170.315	0.0005	0
18	2	1	0.3433	0.1285	999.851	0.0007	0.0645
18	2	2	0.3439	0.1191	999.9889	0.0008	0.0733
18	3	1	0.307	0.0348	992.609	0.0005	0.0381
18	3	2	0.3057	0.033	993.7528	0.0005	0.0411
19	1	1	0.0344	0.0005	0.0001	0.5743	0.5871
19	1	2	0.0343	0.0159	0.0001	0.7728	0.7899
19	2	1	0.0344	0	0.264	0.4346	0.5615
19	2	2	0.0344	0.0004	0	0.5534	0.5658
19	3	1	0.0344	0	0.4182	0.3946	0.5716
19	3	2	0.0342	0.0358	0.0433	0.8573	0.879
20	1	1	0.0465	0.0067	92.3081	0.0007	0.0246
20	1	2	0.0463	0.0066	204.7767	0.0004	0.0273
20	2	1	0.0573	0.0433	566.442	0.0005	0.0618
20	2	2	0.0593	0.0436	827.076	0.0004	0.0725
20	3	1	0.0507	0.0154	219.799	0.0007	0.0404
20	3	2	0.0507	0.0156	351.591	0.0005	0.0445
21	1	1	0.2982	0	0	0.9957	0.9403
21	1	2	0.2982	0	0	0.9697	0.9157
21	2	1	0.2982	0	0	0.9694	0.9154
21	2	2	0.2982	0	0	0.9673	0.9135
21	3	1	0.2004	0.1831	1000	0.0083	0.016
21	3	2	0.2985	0	0.0002	0.8088	0.7643
22	1	1	0.048	0.0012	2.3802	0.2486	0.5277
22	1	2	0.0424	0.0022	2.545	0.2631	0.5147
22	2	1	0.0484	0.0012	4.8577	0.1449	0.5297
22	2	2	0.041	0.0036	6.1713	0.1423	0.517
22	3	1	0.0482	0.0012	3.1826	0.2019	0.5285
22	3	2	0.0417	0.0028	3.8128	0.2021	0.5162
23	1	1	0.0668	0.0627	0.038	0.1767	0.1167
23	1	2	0.067	0.0353	0.0692	0.1689	0.1157
23	2	1	0.0668	0.2	9.066	0.086	0.1067
23	2	2	0.0663	0.2	13.1006	0.0784	0.1086
23	3	1	0.0677	0.0093	1.2161	0.09	0.1117
23	3	2	0.0672	0.008	2.0107	0.0758	0.118
24	1	1	0.0579	0.0022	77.8407	0.0048	0.2822
24	1	2	0.0644	0.0038	2.2857	0.1333	0.3001
24	2	1	0.0585	0.0041	60.1047	0.0063	0.2679
24	2	2	0.0641	0.0071	43.2297	0.0132	0.3214

Patient #	Model for M	Model for E	Residual Norm	k_T	M_∞	k_{in}	k_{out}
24	3	1	0.0582	0.0029	66.8002	0.0056	0.274
24	3	2	0.0641	0.0047	8.125	0.0557	0.3133
25	1	1	0.1963	0.1552	0.001	1	0.7617
25	1	2	0.1963	0.1119	0.0014	1	0.7627
25	2	1	0.1973	0.1999	0	1	0.771
25	2	2	0.1973	0.1998	0	1	0.771
25	3	1	0.1971	0.2	2.3492	1	0.7718
25	3	2	0.1965	0.2	0.637	1	0.7723
26	1	1	1.8586	0	51.2106	0.0256	1
26	1	2	1.6654	0	363.6016	0.004	1
26	2	1	1.8586	0	45.8	0.0285	1
26	2	2	1.6654	0	345.3206	0.0042	1
26	3	1	1.8586	0	57.7365	0.0227	1
26	3	2	1.6654	0	275.2923	0.0052	1

4 R Code

4.1 Code for *Clean Certara Data.R*

```
## Set the working directory to the source file location.
## NOTE: please use your desired directory.
setwd("~/Documents/Workshops/MPI/2019/Code")

## Load packages.
library(dplyr)

## Clear all variables.
rm(list=ls())

#####
## Clean the Certara Data and Extract BCVA CSFT Data Into Columns
#####

## -----
## Load the Data
## -----
## NOTE: please use your desired directory.
data <- read.csv("~/Documents/Workshops/MPI/2019/Data/workshop_100_corrected.csv")

## -----
## Subset the Data and Build Datasets that are in a More Useful Format
## -----
## Extract the unique character strings from the names column and store in "names".
names <- as.character(unique(data$name))

## Subset only by Dose A.
subA <- subset(data, usubjid %in% unique(data[which(data$name == names[4]), 1]))
```

```

## Subset Dose A by Dose A3.
subA3 <- subset(subA,usubjid %in% unique(subA[which(subA$avalc == 3),1]))
## Subset Dose A by Dose A6.
subA6 <- subset(subA,usubjid %in% unique(subA[which(subA$avalc == 6),1]))

## Subset only by Dose B.
subB2 <- subset(data,usubjid %in% unique(data[which(dataa$name == names[3]),1]))

## Create a new column that contains the dosage codes.
subA3$dose <- 3
## Create a new column that contains the dosage codes.
subA6$dose <- 6
## Create a new column that contains the dosage codes.
subB2$dose <- 2

## Subset by BCVA score and concatenate into one data set.
BCVA <- rbind(subset(subA3,name == names[1]),
              subset(subA6,name == names[1]),
              subset(subB2,name == names[1]))

## Subset taking only the user ID, visit number, value, and dose columns.
bcva <- BCVA[,c(1,4,6,7)]
## Change the name of the value column to "BCVA" and reset the numbering of the rows.
names(bcva)[3] <- "BCVA"
rownames(bcva) <- NULL

## Subset by CSFT score and concatenate into one data set.
CSFT <- rbind(subset(subA3,name == names[2]),
              subset(subA6,name == names[2]),
              subset(subB2,name == names[2]))

## Subset taking only the user ID, visit number, value, and dose columns.
csft <- CSFT[,c(1,4,6,7)]
## Change the name of the value column to "CSFT" and reset the numbering of the rows.
names(csft)[3] <- "CSFT"
rownames(csft) <- NULL

## Create a column in both datasets that will indicate which visits the patients
## received a dose.
bcva$dvisit <- 0
csft$dvisit <- 0

## Loop through each subject and determine the visit numbers that correspond to
## a dosing visit. Then populate the corresponding rows of the BCVA and CSFT
## datasets with a 1 at each dosing visit under the dvisit column.
for (i in unique(data$usubjid)) {

  dvis <- subset(data,usubjid == i & name %in% c(names[3:4]))$visitnum

  bcva[which(bcva$usubjid == i & bcva$visitnum %in% dvis),]$dvisit <- 1
  csft[which(csft$usubjid == i & csft$visitnum %in% dvis),]$dvisit <- 1
}

```

```

}

## Remove the dummy variables.
remove(i, dvis)

## Take only the rows in "bcva" that it has in common with "csft" and vice-versa.
bcvaC <- semi_join(bcva, csft)
csftC <- semi_join(csft, bcva)

## Create a dataset that contains both measured values, the dose type, the dose dates;
## aggregated by subject and ordered there by visit number.
dats <- cbind(bcvaC[, c(1, 2, 3)], csftC[, c(3, 4, 5)])

## -----
## Write the Datasets to CSV Files
## -----
## Choose TRUE to write the data files.
toWrite = FALSE

if (toWrite == TRUE) {

  write.csv(dats, file="BCVA_CSFT_Equalized_Data.csv", row.names=FALSE)
  write.csv(bcva, file="BCVA_Data.csv", row.names=FALSE)
  write.csv(csft, file="CSFT_Data.csv", row.names=FALSE)

}

```

4.2 Code for Visualizing Certara Data Overlaid Time Series.R

```

## Set the working directory to the source file location.
## NOTE: please use your desired directory.
setwd("~/Documents/Workshops/MPI/2019/Code")

## Load packages.
library(reshape2)
library(ggplot2)
library(grid)
library(gridExtra)

## Clear all variables.
rm(list=ls())

#####
## Visualize the Certara Data
#####

## -----
## Load the Data
## -----
## NOTE: please use your desired directory.

```

```

data <- read.csv("~/Documents/Workshops/MPI/2019/Data/workshop_100_corrected.csv")

## Extract the unique subject identifiers. These will be looped over during the
## image generation. The vector is "ids".
ids <- unique(data$usubjid)

## -----
## Plot the BCVA and CBFT Data Together Over the Visits for Each Patient
## -----
## Preallocate an empty list called "tsplots" for plot output.
tsplots <- vector(mode="list",length=(length(ids)))

## Populate "tsplots" with data from each patient in "ids".
for (i in 1:length(ids)) {

  # Extract the measured quantities' names.
  type <- unique(subset(data,usubjid==ids[i])$name)

  # Subset the data by the first type, BCVA.
  subtype1 <- subset(data,usubjid==ids[i] & name==type[1])
  # Subset the data by the second type, CSFT.
  subtype2 <- subset(data,usubjid==ids[i] & name==type[2])
  # Subset the data by the third type, dosage.
  subdose <- subset(data,usubjid==ids[i] & name==type[3])

  # Create a data frame "subdata1" that will hold the visit number (h-axis)
  # and the BCVA (v-axis).
  subdata1 <- data.frame(visitnum=subtype1$visitnum,name1=subtype1$avalc)

  # Create a data frame "subdata1" that will hold the visit number (h-axis)
  # and the CSFT (v-axis).
  subdata2 <- data.frame(visitnum=subtype2$visitnum,name2=subtype2$avalc)

  # For plotting the second (right-hand), independent vertical axis, we must
  # first scale the data. We will multiply it by the average of the BCVA
  # range divided by the CSFT range, as the latter values have greater
  # magnitude.
  rangescale <- mean(range(subdata1$name1)/range(subdata2$name2))

  # If the patient receives drug A, then follow the next steps.
  if (as.character(unique(subdose$name)) == "Dose administration Drug A") {

    #If drug A is of type 3, then follow the next steps.
    if (unique(subdose$avalc) == 3) {

      # Create the plot, specifying first the BCVA data first.
      ggplot(subdata1,aes(x=visitnum,y=name1,group=1,color="name1")) +
        # Create vertical lines for the dosing visits.
        geom_vline(data=subdose,aes(xintercept=visitnum,group=2,color="dose A"),
                  linetype="dashed",size=0.25,show.legend=FALSE) +
        # Include data points.

```

```

geom_point(size=2) +
# Connect the points with lines.
geom_line(size=1.25) +
# Plot the CSFT data using lines.
geom_line(data=subdata2, aes(x=visitnum, y=name2*rangescale, group=1,
color="name2"),
          linetype="twodash", size=1.25) +
# Plot the CSFT data points.
geom_point(data=subdata2, aes(x=visitnum, y=name2*rangescale, group=1,
color="name2"),
          size=2) +
# Create the second (right-hand), independent vertical axis for the CSFT data.
# We must undo the data scaling that we imposed before and
# specify the axis name.
scale_y_continuous(sec.axis=sec_axis(~./rangescale,
name="Central SF Thickness")) +
# Choose data colors, order the legend items, label the legend items, specify
# the legend item line types and point shapes. Note that "dose A3" is used.
scale_colour_manual(values=c("black", "deepskyblue4", "darkorange3"),
                    breaks=c("name1", "name2", "dose A"),
                    labels=c("BCVA", "CSFT", "dose A3"),
                    guide=guide_legend(override.aes=list(
                      linetype=c("solid", "twodash", "dashed"),
                      shape=c(19, 19, NA)))) +
# Change the plot theme to black-and-white.
theme_bw() +
# Alter various plot theme values such as axis font size, axis title size and
# position, axis borders, plot title font size and position. Set a box around
# the legend and remove the legend title.
theme(axis.text.x=element_text(size=14),
      axis.text.y=element_text(size=14),
      axis.title.x=element_text(size=18, vjust=-1),
      axis.title.y=element_text(size=18, vjust=2),
      axis.title.y.right=element_text(size=18, vjust=2),
      axis.line=element_line(color="black", size=0.75, linetype="solid"),
      plot.title=element_text(hjust=0.5, size=24),
      legend.box.background = element_rect(colour="black"),
      legend.title = element_blank()) +
# Specify the horizontal axis label.
xlab("Visit Number") +
# Specify the vertical axis label.
ylab("Best Corrected Visual Acuity") +
# Specify the title label as "Subject X" and store the plot into the list.
ggtitle(paste0("Subject ", ids[i])) -> tsplots[[i]]

# Title the "tsplots" list entry as "subjectX".
names(tsplots)[i] <- paste0("subject", ids[i])

# If drug A is of type 6, then follow the next steps. Note that "dose A6" is used.
# The plotting comments are the same as above.
} else {

```

```

ggplot(subdata1, aes(x=visitnum, y=namel, group=1, color="namel")) +
  geom_vline(data=subdose, aes(xintercept=visitnum, group=2, color="dose A"),
            linetype="dashed", size=0.25, show.legend = FALSE) +
  geom_point(size=2) +
  geom_line(size=1.25) +
  geom_line(data=subdata2, aes(x=visitnum, y=name2*rangescale, group=1,
                              color="name2"),
            linetype="twodash", size=1.25) +
  geom_point(data=subdata2, aes(x=visitnum, y=name2*rangescale, group=1,
                              color="name2"),
            size=2) +
  scale_y_continuous(sec.axis=sec_axis(~./rangescale,
                                       name="Central SF Thickness")) +
  scale_colour_manual(values=c("black", "deepskyblue4", "darkorange3"),
                     breaks=c("name1", "name2", "dose A"),
                     labels=c("BCVA", "CSFT", "dose A6"),
                     guide = guide_legend(override.aes = list(
                       linetype = c("solid", "twodash", "dashed"),
                       shape = c(19, 19, NA)))) +
  theme_bw() +
  theme(axis.text.x=element_text(size=14),
        axis.text.y=element_text(size=14),
        axis.title.x=element_text(size=18, vjust=-1),
        axis.title.y=element_text(size=18, vjust=2),
        axis.title.y.right=element_text(size=18, vjust=2),
        axis.line=element_line(color="black", size=0.75, linetype="solid"),
        plot.title=element_text(hjust=0.5, size=24),
        legend.box.background = element_rect(colour="black"),
        legend.title = element_blank()) +
  xlab("Visit Number") +
  ylab("Best Corrected Visual Acuity") +
  ggtitle(paste0("Subject  ", ids[i])) -> tsplots[[i]]

names(tsplots)[i] <- paste0("subject", ids[i])
}

# If the patient receives drug B, then follow the next steps. Note that "dose B2"
# is used. The plotting comments are the same as above.
} else {

ggplot(subdata1, aes(x=visitnum, y=namel, group=1, color="namel")) +
  geom_vline(data=subdose, aes(xintercept=visitnum, group=2, color="dose B"),
            linetype="dashed", size=0.25, show.legend = FALSE) +
  geom_point(size=2) +
  geom_line(size=1.25) +
  geom_line(data=subdata2, aes(x=visitnum, y=name2*rangescale, group=1,
                              color="name2"),
            linetype="twodash", size=1.25) +
  geom_point(data=subdata2, aes(x=visitnum, y=name2*rangescale, group=1,
                              color="name2"),
            size=2) +
  scale_y_continuous(sec.axis=sec_axis(~./rangescale,
                                       name="Central SF Thickness")) +
  scale_colour_manual(values=c("black", "deepskyblue4", "darkorange3"),
                     breaks=c("name1", "name2", "dose B"),
                     labels=c("BCVA", "CSFT", "dose B6"),
                     guide = guide_legend(override.aes = list(
                       linetype = c("solid", "twodash", "dashed"),
                       shape = c(19, 19, NA)))) +
  theme_bw() +
  theme(axis.text.x=element_text(size=14),
        axis.text.y=element_text(size=14),
        axis.title.x=element_text(size=18, vjust=-1),
        axis.title.y=element_text(size=18, vjust=2),
        axis.title.y.right=element_text(size=18, vjust=2),
        axis.line=element_line(color="black", size=0.75, linetype="solid"),
        plot.title=element_text(hjust=0.5, size=24),
        legend.box.background = element_rect(colour="black"),
        legend.title = element_blank()) +
  xlab("Visit Number") +
  ylab("Best Corrected Visual Acuity") +
  ggtitle(paste0("Subject  ", ids[i])) -> tsplots[[i]]

names(tsplots)[i] <- paste0("subject", ids[i])
}

```

```

color="name2"),
      size=2) +
scale_y_continuous(sec.axis=sec_axis(~./rangescale,
name="Central SF Thickness")) +
scale_colour_manual(values=c("black", "deepskyblue4", "darkorange3"),
                    breaks=c("name1", "name2", "dose B"),
                    labels=c("BCVA", "CSFT", "dose B2"),
                    guide = guide_legend(override.aes = list(
                      linetype = c("solid", "twodash", "dashed"),
                      shape = c(19,19,NA)))) +
theme_bw() +
theme(axis.text.x=element_text(size=14),
      axis.text.y=element_text(size=14),
      axis.title.x=element_text(size=18,vjust=-1),
      axis.title.y=element_text(size=18,vjust=2),
      axis.title.y.right=element_text(size=18,vjust=2),
      axis.line=element_line(color="black",size=0.75,linetype="solid"),
      plot.title=element_text(hjust=0.5,size=24),
      legend.box.background = element_rect(colour="black"),
      legend.title = element_blank()) +
xlab("Visit Number") +
ylab("Best Corrected Visual Acuity") +
ggtitle(paste0("Subject  ",ids[i])) -> tsplots[[i]]

names(tsplots)[i] <- paste0("subject",ids[i])

}

}

# Remove the dummy variables.
remove(i, subdata1, subdata2, subdose, subtype1, subtype2, type)

## -----
## Export the Time-Series Images
## -----
## Choose TRUE to produce plots.
toPlot = FALSE

if (toPlot == TRUE) {

# Loop through "tsplots" and export the time series data.
for (i in 1:length(tsplots)) {

# Write the tsplots. We specify the file name, image type, dimensions, units, and
# resolution.
ggsave(
  paste0(names(tsplots[i]), "_time_series", ".png"),
  plot=tsplots[[i]],
  height=7.5,
  width=12,

```

```

        units="in",
        dpi=300
    )
}

# Remove the loop variable.
remove(i)

}

```

4.3 Code for Visualizing Certara Data 2x2 Plots.R

```

## Set the working directory to the source file location.
## NOTE: please use your desired directory.
setwd("~/Documents/Workshops/MPI/2019/Code")

## Load packages.
library(reshape2)
library(ggplot2)
library(grid)
library(gridExtra)

## Clear all variables.
rm(list=ls())

#####
## Visualize the Certara Data
#####

## -----
## Load the Data
## -----
## NOTE: please use your desired directory.
data <- read.csv("~/Documents/Workshops/MPI/2019/Data/workshop_100_corrected.csv")

## Extract the unique subject identifiers. These will be looped over during the
## image generation. The vector is "ids".
ids <- unique(data$usubjid)

## -----
## Plot the BCVA and CBFT Data in 2x2 Panels for Specific Patients
## -----
## Choose the desired patients and store their ids in "pats".
pats <- c(23,24,32,70)

# For setting the CSFT plot axis ranges, we find the min and max CSFT values
# across all the patients in "pats".
mini <- NULL
maxi <- NULL

for (i in 1:length(pats)) {

```

```

mini <- c(mini,
          min(subset(data,usubjid == pats[i] &
                    name == "Central SF Thickness")$avalc))

maxi <- c(maxi,
          max(subset(data,usubjid == pats[i] &
                    name == "Central SF Thickness")$avalc))

}

## Remove the loop variable.
remove(i)

## Function to extract legend taken from:
## https://github.com/hadley/ggplot2/wiki/Share-a-legend-between-two-ggplot2-graphs
g_legend<-function(a.gplot){
  tmp <- ggplot_gtable(ggplot_build(a.gplot))
  leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
  legend <- tmp$grobs[[leg]]
  return(legend)}

## Specify the patient numbers in "pats".
pats <- c(23,7,32,70)

## Calculate the min and max for the vertical axis of the CSFT plots.
mini <- NULL
maxi <- NULL

for (i in 1:length(pats)) {

  mini <- c(mini,
            min(subset(data,usubjid == pats[i] &
                      name == "Central SF Thickness")$avalc))
  maxi <- c(maxi,
            max(subset(data,usubjid == pats[i] &
                      name == "Central SF Thickness")$avalc))

}

## Remove the loop variable.
remove(i)

## Preallocate an empty list called "gridplots" for plot output.
gridplots <- vector(mode="list",length=(length(pats)))

## Populate "gridplots" with plots for each patient in "pats" that alternates
## between the BCVA and CSFT time series, along with dosing times.
for (i in 1:length(gridplots)) {

  if ((i %% 2) != 0) {

```

```

# Extract the measured quantities' names.
type <- unique(subset(data,usubjid==pats[i])$name)

# Subset the data by the first type, BCVA.
subtype1 <- subset(data,usubjid==pats[i] & name==type[1])
# Subset the data by the second type, CSFT.
subtype2 <- subset(data,usubjid==pats[i] & name==type[2])
# Subset the data by the third type, dosage.
subdose <- subset(data,usubjid==pats[i] & name==type[3])

# Create a data frame "subdata1" that will hold the visit number (h-axis)
# and the BCVA (v-axis).
subdata1 <- data.frame(visitnum=subtype1$visitnum,name1=subtype1$avalc)

# Create a data frame "subdata1" that will hold the visit number (h-axis)
# and the CSFT (v-axis).
subdata2 <- data.frame(visitnum=subtype2$visitnum,name2=subtype2$avalc)

# Create the plot, specifying first the BCVA data.
ggplot(subdata1,aes(x=visitnum,y=name1,group=1,color="name1")) +
  # Create vertical lines for the dosing visits.
  geom_vline(data=subdose,aes(xintercept=visitnum,group=2,color="dose"),
             linetype="dashed",size=0.25,show.legend=FALSE) +
  # Include data points.
  geom_point(size=2,show.legend=TRUE) +
  # Connect the points with lines.
  geom_line(size=1.25,show.legend=TRUE) +
  #
  geom_blank(aes(color="name2")) +
  # Change the plot theme to black-and-white.
  theme_bw() +
  # Alter various plot theme values such as axis font size, axis title size and
  # position, axis borders, plot title font size and position. Set a box around
  # the legend and remove the legend title.
  theme(axis.text.x=element_blank(),
        axis.text.y=element_text(size=14),
        axis.title.x=element_text(size=18,vjust=-1),
        axis.title.y=element_text(size=18,vjust=2),
        axis.line=element_line(color="black",size=0.75,linetype="solid"),
        plot.title=element_text(hjust=0.5,size=24),
        legend.box.background=element_rect(colour="black"),
        legend.title=element_blank(),
        legend.position="bottom") +
  # Choose data colors, order the legend items, label the legend items, specify
  # the legend item line types and point shapes. Note that "dose A3" is used.
  scale_colour_manual(values=c("black","deepskyblue4","darkorange3"),
                     breaks=c("name1","name2","dose"),
                     labels=c("BCVA","CSFT","dose"),
                     guide=guide_legend(override.aes=list(
                       linetype=c("solid","solid","dashed"),

```

```

      shape=c(19,19,NA))) +
guides(size = "legend", group = "none") +
# Specify horizontal axis tick mark number.
scale_x_continuous(breaks=scales::pretty_breaks(n=5)) +
# Set the vertical axis range.
ylim(0,100) +
# Specify the horizontal axis label.
xlab(" ") +
# Specify the vertical axis label.
ylab("best corrected visual acuity") +
# Specify the title label as "Subject X" and store the plot into the list.
ggtitle(paste0("Subject  ",pats[i])) -> gridplots[[i]]$BCVA

# Create the plot, specifying first the CSFT data.
ggplot(subdata2, aes(x=visitnum, y=name2, group=1, color="name2")) +
# Create vertical lines for the dosing visits.
geom_vline(data=subdose, aes(xintercept=visitnum, group=2, color="dose"),
           linetype="dashed", size=0.25, show.legend=FALSE) +
# Include data points.
geom_point(size=2) +
# Connect the points with lines.
geom_line(size=1.25) +
# Change the plot theme to black-and-white.
theme_bw() +
# Alter various plot theme values such as axis font size, axis title size and
# position, axis borders, plot title font size and position. Remove the legend.
theme(axis.text.x=element_text(size=14),
      axis.text.y=element_text(size=14),
      axis.title.x=element_text(size=18,vjust=-1),
      axis.title.y=element_text(size=18,vjust=2),
      axis.line=element_line(color="black", size=0.75, linetype="solid"),
      plot.title=element_text(hjust=0.5, size=24),
      legend.position="none") +
# Choose data colors.
scale_colour_manual(values=c("black", "darkorange3")) +
# Specify horizontal axis tick mark number.
scale_x_continuous(breaks=scales::pretty_breaks(n=5)) +
# Set the vertical axis range.
ylim(min(mini), max(maxi)) +
# Specify the horizontal axis label.
xlab("visit number") +
# Specify the vertical axis label.
ylab("central sub-field thickness") +
# Specify the title label as a blank.
ggtitle(paste0(" ")) -> gridplots[[i]]$CSFT

## Name the list entry.
names(gridplots)[i] <- paste0("subject", ids[i])

} else {

```

```

# Extract the measured quantities' names.
type <- unique(subset(data,usubjid==pats[i])$name)

# Subset the data by the first type, BCVA.
subtype1 <- subset(data,usubjid==pats[i] & name==type[1])
# Subset the data by the second type, CSFT.
subtype2 <- subset(data,usubjid==pats[i] & name==type[2])
# Subset the data by the third type, dosage.
subdose <- subset(data,usubjid==pats[i] & name==type[3])

# Create a data frame "subdata1" that will hold the visit number (h-axis)
# and the BCVA (v-axis).
subdata1 <- data.frame(visitnum=subtype1$visitnum,name1=subtype1$avalc)

# Create a data frame "subdata1" that will hold the visit number (h-axis)
# and the CSFT (v-axis).
subdata2 <- data.frame(visitnum=subtype2$visitnum,name2=subtype2$avalc)

# Create the plot, specifying first the BCVA data.
ggplot(subdata1,aes(x=visitnum,y=name1,group=1,color="name1")) +
  # Create vertical lines for the dosing visits.
  geom_vline(data=subdose,aes(xintercept=visitnum,group=2,color="dose"),
            linetype="dashed",size=0.25,show.legend=FALSE) +
  # Include data points.
  geom_point(size=2,show.legend=TRUE) +
  # Connect the poits with lines.
  geom_line(size=1.25,show.legend=TRUE) +
  #
  geom_blank(aes(color="name2")) +
  # Change the plot theme to black-and-white.
  theme_bw() +
  # Alter various plot theme values such as axis font size, axis title size and
  # position, axis borders, plot title font size and position. Set a box around
  # the legend and remove the legend title.
  theme(axis.text.x=element_blank(),
        axis.text.y=element_blank(),
        axis.title.x=element_text(size=18,vjust=-1),
        axis.title.y=element_text(size=18,vjust=2),
        axis.line=element_line(color="black",size=0.75,linetype="solid"),
        plot.title=element_text(hjust=0.5,size=24),
        legend.box.background=element_rect(colour="black"),
        legend.title=element_blank(),
        legend.position="bottom") +
  # Choose data colors, order the legend items, label the legend items, specify
  # the legend item line types and point shapes. Note that "dose A3" is used.
  scale_colour_manual(values=c("black","deepskyblue4","darkorange3"),
                    breaks=c("name1","name2","dose"),
                    labels=c("BCVA","CSFT","dose"),
                    guide=guide_legend(override.aes=list(
                      linetype=c("solid","solid","dashed"),
                      shape=c(19,19,NA)))) +

```

```

guides(size = "legend", group = "none") +
# Specify horizontal axis tick mark number.
scale_x_continuous(breaks=scales::pretty_breaks(n=5)) +
# Set the vertical axis range.
ylim(0,100) +
# Specify the horizontal axis label.
xlab(" ") +
# Specify the vertical axis label.
ylab(" ") +
# Specify the title label as "Subject X" and store the plot into the list.
ggtitle(paste0("Subject  ",pats[i])) -> gridplots[[i]]$BCVA

# Create the plot, specifying first the CSFT data.
ggplot(subdata2, aes(x=visitnum, y=name2, group=1, color="name2")) +
# Create vertical lines for the dosing visits.
geom_vline(data=subdose, aes(xintercept=visitnum, group=2, color="dose"),
           linetype="dashed", size=0.25, show.legend=FALSE) +
# Include data points.
geom_point(size=2) +
# Connect the points with lines.
geom_line(size=1.25) +
# Change the plot theme to black-and-white.
theme_bw() +
# Alter various plot theme values such as axis font size, axis title size and
# position, axis borders, plot title font size and position. Remove the legend.
theme(axis.text.x=element_text(size=14),
      axis.text.y=element_blank(),
      axis.title.x=element_text(size=18, vjust=-1),
      axis.title.y=element_text(size=18, vjust=2),
      axis.line=element_line(color="black", size=0.75, linetype="solid"),
      plot.title=element_text(hjust=0.5, size=24),
      legend.position="none") +
# Choose data colors.
scale_colour_manual(values=c("black", "darkorange3")) +
# Specify horizontal axis tick mark number.
scale_x_continuous(breaks=scales::pretty_breaks(n=5)) +
# Set the vertical axis range.
ylim(min(mini), max(maxi)) +
# Specify the horizontal axis label.
xlab("visit number") +
# Specify the vertical axis label.
ylab(" ") +
# Specify the title label as a blank.
ggtitle(paste0(" ")) -> gridplots[[i]]$CSFT

## Name the list entry.
names(gridplots)[i] <- paste0("subject", ids[i])

}

}

```

```

## Remove the dummy variables.
remove(i, subdata1, subdata2, subdose, subtype1, subtype2, type, maxi, mini)

## Extract the legend from one of the plots.
mylegend <- g_legend(gridplots[[1]]$BCVA)

## Save the 2x2 grid plots in a list called "squareplots".
squareplots <- vector(mode="list", length=length(gridplots)/2)

for (i in 1:(length(gridplots)/2)) {

  # For each patient, save a 2x2 grid of the BCVA and CSFT plots with patients at
  # the columns. Note that each individual plot's legend is suppressed, but then
  # a single common legend is added as the second row of a two-row grid-plot,
  # where the top row contains the 2x2 image.
  arrangeGrob(arrangeGrob(gridplots[[2*i-1]]$BCVA + theme(legend.position="none"),
                          gridplots[[2*i]]$BCVA + theme(legend.position="none"),
                          gridplots[[2*i-1]]$CSFT + theme(legend.position="none"),
                          gridplots[[2*i]]$CSFT + theme(legend.position="none"),
                          nrow=2),
              mylegend, nrow=2, heights=c(10,1)) -> squareplots[[i]]

  names(squareplots)[i] <- paste0("subjects_", pats[2*i-1], "_", pats[2*i])

}

## Remove the loop variable.
remove(i)

## -----
## Export the 2x2 Grid Images
## -----
## Choose TRUE to produce plots.
toPlot = FALSE

if (toPlot == TRUE) {

  # Export the grid images.
  for (i in 1:length(squareplots)) {

    # Write the gridplots. We specify the file name, image type, dimensions, units, and
    # resolution.
    ggsave(
      paste0("Subjects_", pats[1], "_", pats[2], "_time_series_grid", ".png"),
      plot=squareplots[[i]],
      height=7.5,
      width=12,
      units="in",
      dpi=300
    )
  }
}

```

```

}

# Remove the loop variable.
remove(i)

}

```

4.4 Code for Visualizing Certara Data Time Delayed Embedding.R

```

## Set the working directory to the source file location.
## NOTE: please use your desired directory.
setwd("~/Documents/Workshops/MPI/2019/Code")

## Load packages.
library(reshape2)
library(ggplot2)
library(grid)
library(gridExtra)

## Clear all variables.
rm(list=ls())

#####
## Visualize the Certara Data
#####

## -----
## Load the Data
## -----
## NOTE: please use your desired directory.
data <- read.csv("~/Documents/Workshops/MPI/2019/Data/workshop_100_corrected.csv")

## Extract the unique subject identifiers. These will be looped over during the
## image generation. The vector is "ids".
ids <- unique(data$usubjid)

## -----
## Ready the Data for Plotting a Time-Delayed Embedding
## -----
## Preallocate an empty list called "tdplots" for plot output.
tdplots <- vector(mode="list",length=(length(ids)))

## Populate "tdplots" with data from each patient in "ids".
for (i in 1:length(ids)) {

  # Extract the measured quantities' names.
  type <- unique(subset(data,usubjid==ids[i])$name)

  # Subset the data by the first type, BCVA.
  subtype1 <- subset(data,usubjid==ids[i] & name==type[1])

```

```

# Subset the data by the second type, CSFT.
subtype2 <- subset(data,usubjid==ids[i] & name==type[2])
# Subset the data by the third type, dosage.
subdose <- subset(data,usubjid==ids[i] & name==type[3])

# Specify the embedding delay.
delay <- 1

frst1 <- subtype1$avalc[1:(length(subtype1$avalc)-delay)]
last1 <- subtype1$avalc[(delay+1):length(subtype1$avalc)]
frst2 <- subtype2$avalc[1:(length(subtype2$avalc)-delay)]
last2 <- subtype2$avalc[(delay+1):length(subtype2$avalc)]

# Create a data frame "subdata1" that will hold the first n-delay BCVA values
# (h-axis) and the last n-delay BCVA values (v-axis).
subdata1 <- data.frame(name11=frst1,name12=last1)

# Create a data frame "subdata1" that will hold the first n-1 CSFT values
# (h-axis) and the last n-1 CSFT values (v-axis).
subdata2 <- data.frame(name21=frst2,name22=last2)

# If the patient receives drug A, then follow the next steps.
if (as.character(unique(subdose$name)) == "Dose administration Drug A") {

  #If drug A is of type 3, then follow the next steps.
  if (unique(subdose$avalc) == 3) {

    # Create the plot, specifying first the n-1 BCVA data first.
    ggplot(subdata1,aes(x=name11,y=name12,group=1,color="name1")) +
      geom_point(size=2) +
      # Connect the poits with lines.
      geom_line(size=1.25) +
      # Change the plot theme to black-and-white.
      theme_bw() +
      # Manually set the data color.
      scale_colour_manual(values=c("deepskyblue4")) +
      # Alter various plot theme values such as axis font size, axis title size and
      # position, axis borders, plot title font size and position.
      theme(axis.text.x=element_text(size=14),
            axis.text.y=element_text(size=14),
            axis.title.x=element_text(size=18,vjust=-1),
            axis.title.y=element_text(size=18,vjust=2),
            axis.line=element_line(color="black",size=0.75,linetype="solid"),
            plot.title=element_text(hjust=0.5,size=24),
            legend.position="none") +
      # Specify the horizontal axis label.
      xlab(paste0("BCVA (1 to N-",delay,")")) +
      # Specify the vertical axis label.
      ylab(paste0("BCVA (",delay+1," to N)")) +
      # Specify the title label as "Subject X" and store the plot into the list.
      # Note that "A3" is used.

```

```

ggtitle(paste0("Subject ",ids[i]," (A3)")) -> tdplots1

# Create the plot, specifying first the n-1 CSFT data first.
ggplot(subdata2, aes(x=name21,y=name22,group=1,color="name1")) +
  geom_point(size=2) +
  # Connect the points with lines.
  geom_line(size=1.25) +
  # Change the plot theme to black-and-white.
  theme_bw() +
  scale_colour_manual(values=c("darkorange3")) +
  # Alter various plot theme values such as axis font size, axis title size and
  # position, axis borders, plot title font size and position.
  theme(axis.text.x=element_text(size=14),
        axis.text.y=element_text(size=14),
        axis.title.x=element_text(size=18,vjust=-1),
        axis.title.y=element_text(size=18,vjust=2),
        axis.line=element_line(color="black",size=0.75,linetype="solid"),
        plot.title=element_text(hjust=0.5,size=32),
        legend.position="none") +
  # Specify the horizontal axis label.
  xlab(paste0("CSFT (1 to N-",delay,")")) +
  # Specify the vertical axis label.
  ylab(paste0("CSFT (",delay+1," to N)")) +
  ggtitle(paste0(" ")) -> tdplots2

tdplots[[i]] <- arrangeGrob(tdplots1,tdplots2)

# Title the "tdplots" list entry as "subjectX".
names(tdplots)[i] <- paste0("subject",ids[i],"_delay",delay)

# If drug A is of type 6, then follow the next steps. Note that "A6" is used.
# The plotting comments are the same as above.
} else {

ggplot(subdata1, aes(x=name11,y=name12,group=1,color="name1")) +
  geom_point(size=2) +
  geom_line(size=1.25) +
  theme_bw() +
  scale_colour_manual(values=c("deepskyblue4")) +
  theme(axis.text.x=element_text(size=14),
        axis.text.y=element_text(size=14),
        axis.title.x=element_text(size=18,vjust=-1),
        axis.title.y=element_text(size=18,vjust=2),
        axis.line=element_line(color="black",size=0.75,linetype="solid"),
        plot.title=element_text(hjust=0.5,size=24),
        legend.position="none") +
  xlab(paste0("BCVA (1 to N-",delay,")")) +
  ylab(paste0("BCVA (",delay+1," to N)")) +
  ggtitle(paste0("Subject ",ids[i]," (A6)")) -> tdplots1

ggplot(subdata2, aes(x=name21,y=name22,group=1,color="name1")) +

```

```

geom_point(size=2) +
geom_line(size=1.25) +
theme_bw() +
scale_colour_manual(values=c("darkorange3")) +
theme(axis.text.x=element_text(size=14),
      axis.text.y=element_text(size=14),
      axis.title.x=element_text(size=18,vjust=-1),
      axis.title.y=element_text(size=18,vjust=2),
      axis.line=element_line(color="black",size=0.75,linetype="solid"),
      plot.title=element_text(hjust=0.5,size=32),
      legend.position="none") +
xlab(paste0("CSFT (1 to N-",delay,")")) +
ylab(paste0("CSFT (",delay+1," to N)")) +
ggtitle(paste0(" ")) -> tdplots2

tdplots[[i]] <- arrangeGrob(tdplots1,tdplots2)

names(tdplots)[i] <- paste0("subject",ids[i],"_delay",delay)

}

# If the patient receives drug B, then follow the next steps. Note that "B2"
# is used. The plotting comments are the same as above.
} else {

ggplot(subdata1,aes(x=name11,y=name12,group=1,color="name1")) +
  geom_point(size=2) +
  geom_line(size=1.25) +
  theme_bw() +
  scale_colour_manual(values=c("deepskyblue4")) +
  theme(axis.text.x=element_text(size=14),
        axis.text.y=element_text(size=14),
        axis.title.x=element_text(size=18,vjust=-1),
        axis.title.y=element_text(size=18,vjust=2),
        axis.line=element_line(color="black",size=0.75,linetype="solid"),
        plot.title=element_text(hjust=0.5,size=24),
        legend.position="none") +
  xlab(paste0("BCVA (1 to N-",delay,")")) +
  ylab(paste0("BCVA (",delay+1," to N)")) +
  ggtitle(paste0("Subject ",ids[i]," (B2)")) -> tdplots1

ggplot(subdata2,aes(x=name21,y=name22,group=1,color="name1")) +
  geom_point(size=2) +
  geom_line(size=1.25) +
  theme_bw() +
  scale_colour_manual(values=c("darkorange3")) +
  theme(axis.text.x=element_text(size=14),
        axis.text.y=element_text(size=14),
        axis.title.x=element_text(size=18,vjust=-1),
        axis.title.y=element_text(size=18,vjust=2),
        axis.line=element_line(color="black",size=0.75,linetype="solid"),

```

```

        plot.title=element_text(hjust=0.5,size=32),
        legend.position="none") +
xlab(paste0("CSFT (1 to N-",delay,")") +
ylab(paste0("CSFT (",delay+1," to N)") +
ggtitle(paste0(" ")) -> tdplots2

tdplots[[i]] <- arrangeGrob(tdplots1,tdplots2)

names(tdplots)[i] <- paste0("subject",ids[i],"_delay",delay)

}

}

## Remove the dummy variables.
remove(i,subdata1,subdata2,subdose,subtype1,
        subtype2,frst1,last1,frst2,last2,tdplots1,tdplots2,type)

## -----
## Export the Time-Delayed Embedding Images
## -----
## Choose TRUE to produce plots.
toPlot = FALSE

if (toPlot == TRUE) {

# Loop through "tsplots" and export the time series data.
for (i in 1:length(tdplots)) {

# Write the tsplots. We specify the file name, image type, dimensions, units, and
# resolution.
ggsave(
    paste0(names(tdplots[i]),".png"),
    plot=tdplots[[i]],
    height=7.5,
    width=12,
    units="in",
    dpi=300
)

}

# Remove the loop variable.
remove(i)

}

```

5 MATLAB Code

5.1 Code for function *func_conc*

```
function accu = func_conc(t,y,Kt,Minf,Kin,Kout, Meff_flg, E_flg)
```

```

Kinf=0.643;
accu=zeros(3,1);
accu(1) = -y(1); % concentration ODE
accu(2) = Kt*y(1)-Kinf*y(2); % tolerance ODE
switch Meff_flg
    case 2
        MT=Minf/(1+y(2)); % M(T)=M_inf/(1+T)
    case 3
        MT=Minf*exp(-y(2)); % M(T)=M_inf*exp(-T)
    otherwise
        MT=Minf*(1-y(2)); % M(T)=M_inf*(1-T)
end
if E_flg == 2
    EC=y(1)/(1+y(1));
else
    EC=heaviside(y(1)-1);
end
accu(3) = Kin*(1+MT*EC)-Kout*y(3); % accuity ODE

```

5.2 Code for function *func_BCVA*

```

function [actT, Accty, Conc, Toler]=func_BCVA(pno, P, Meff_flg, E_flg, RA_timing)
%pno=1; km=0.001; Minf=0.5; kin=0.45; kout=0.5;
Kt=P(1); Minf=P(2); Kin=P(3); Kout=P(4);
global Val_DOSE
kc=0.077; % time is scaled by kc (mult by kc)
dose_scale=119; % 2 -- 238; 6 -- 714
%RA_timing=BCVA(pno).nomtime'; % get visit days for the patient
timing=[]; % initialize timing array
TIME=[]; % initialize intervals b/w shots
timing=Val_DOSE(pno).dose_nomtime'; % days to administer the dose
no_visits = length(RA_timing); % number of actual visits
tmax=RA_timing(no_visits)+7;% last week of timing scale in case shot was administered
% at last visit
no_steps=length(timing); % number of shots
TIME(1)=timing(1); % first interval = first # of days
for i=2:no_steps
    TIME(i)=timing(i)-timing(i-1); % interval b/w shots
end
TIME(no_steps+1)=tmax-timing(no_steps); % last interval
Dose=max(Val_DOSE(1).bcva_value)*dose_scale; % dose for the patient
y0=[0 0 1]; % Initial conditions (concentr=0 tolerance=0)
% acuity=1
Y=[];
actT=[];
aT=0;
for i=1:no_steps+1
    t=0:0.01:TIME(i)*kc; % run until next dose
    % t = t*kc;
    [t1,ytemp] = ode45(@(t,y) func_conc(t,y,Kt,Minf,Kin,Kout, Meff_flg, E_flg),t,y0);

```

```

t2=t1+aT;
actT=[actT;t2];           % extend time scale for plotting
Y=[Y;ytemp];             % extend results scale
yneg=ytemp(end,:);      % collect ode solution at the tail
y0=yneg;                 % new IC for the next loop
y0(1)=y0(1)+Dose;       % add dose to concentration in IC
aT=aT+TIME(i)*kc;
end
actT=actT/kc;           % return time points
Accty=Y(:,3);           % return calculated accuity
Conc=Y(:,1);            % return calculated concentration
Toler=Y(:,2);           % return calculated tolerance

```

5.3 Code for *Fit_contin*

```

%% Load data
clear all
clc
global Val_BCVA Val_DOSE

load('Val_DOSE')
load('Val_BCVA')

%% Model setup
prompt='Input Patient #: ';
pno=input(prompt);           % patient # input
prompt='Which Max effectiveness version? 1=>Minf(1-T); 2=>Minf/(1+T);
3=>Minf*e^(-T) : ';
Meff_flg=input(prompt);
prompt='Which Dosage effectiveness version? 1=>H(C-1); 2=>C/(1+C) : ';
E_flg=input(prompt);

tic;
pno=55;                      % patient number
RA_timing=Val_BCVA(pno).nomtime'; % get visit days for the patient
RA_accuity=Val_BCVA(pno).value'; % get actual accuity for the patient
RA_scale=RA_accuity(1);      % initial accuity of the patient
solution_a = RA_accuity/RA_scale; % scale accuity to match a(0)=1
solution_t =RA_timing;      % setup time scale for actual accuity
dose_scale =119;            % dose=2 equiv to dose=238
Dose=max(Val_DOSE(1).bcva_value)*dose_scale; % real dose

%Initial Parameter Guesses
Kt=0.001; Minf=0.5; Kin=0.45; Kout=0.5;
%Kt=0.2; Minf=0.8; Kin=0.8; Kout=0.9;
%Kt=0.002; Minf=0.9; Kin=0.2; Kout=0.3;

P_0 = [Kt Minf Kin Kout]; % vector of initial parameters

% Run ODE Model with Original Parameters

```

```

% [actT, Orig_Accty, Conc, Toler0]=func_BCVA(pno, P_0, Meff_flg, E_flg, RA_timing);
% Create anonymous function for curve fitting
model_loaded = @(P, RA_timing) model_BCVA(pno, P, Meff_flg, E_flg, RA_timing);
lb=[0 0 0 0]; % lower bounds for parameters
ub=[0.2 1000 1000 1]; % upper bound for parameters

%% Run model
options = optimoptions('lsqcurvefit','MaxFunctionEvaluations',2000, 'Display', 'off');

[best_param, resnorm, residual, exitflag] = ...
    lsqcurvefit(model_loaded, P_0, solution_t, solution_a,lb,ub, options)

toc
%% Run BCVA function with found parameters
[actT, Cont_Accty, Conc, Toler]=func_BCVA(pno, best_param, Meff_flg, E_flg, RA_timing);

return
%% Display
plot(solution_t,solution_a,'b','LineWidth',3)
grid on
grid minor
hold on
plot(actT,Cont_Accty,'r','LineWidth',3)
plot(actT,Toler,'m','LineWidth',3)
xlabel(['Parameters: ','k_m=',num2str(best_param(1)),'...',
    M_{inf}=',num2str(best_param(2)),'...',
    ', k_{\rm in}=',num2str(best_param(3)),' ', k_{\rm out}=',num2str(best_param(4))])
title(['Stats for patient ',num2str(pno), ', Dose=',num2str(Dose),'mg']);
legend('Actual Accuity','Accuity',...
    'Tolerance','location','East')

%% Display results in one graph
plot(actT,Conc/500,'g','LineWidth',3)
grid on
grid minor
hold on
xlabel(['Parameters: ','k_m=',num2str(best_param(1)),'...',
    M_{inf}=',num2str(best_param(2)),'...',
    ', k_{\rm in}=',num2str(best_param(3)),' ', k_{\rm out}=',num2str(best_param(4))])
plot(solution_t,solution_a,'b','LineWidth',3)
title(['Stats for patient ',num2str(pno), ', Dose=',num2str(Dose),'mg']);
plot(actT,Cont_Accty,'r','LineWidth',3)
plot(actT,Orig_Accty,'k','LineWidth',3)
plot(actT,Toler,'m','LineWidth',3)
plot(actT,Toler0,'y','LineWidth',3)
legend('Concentration', 'Real Accuity','Calc Accuity', 'Accuity w Orig Param',...
    'Tolerance',...

```

```

        'Orig. Tolerance', 'location', 'East')
return
%% Display results in 2 panes
subplot(1,2,1)
plot(actT, Conc/500, 'g', 'LineWidth', 3)
grid on
grid minor
hold on
plot(solution_t, solution_a, 'b', 'LineWidth', 3)
xlabel(['Parameters: ', 'k_m=', num2str(P_0(1)), ', M_{inf}=', num2str(P_0(2)), ...
        ', k_{\rm in}=', num2str(P_0(3)), ', k_{\rm out}=', num2str(P_0(4))])
plot(actT, Orig_Accty, 'k', 'LineWidth', 3)
plot(actT, Toler0, 'y', 'LineWidth', 3)
title(['Stats for patient, orig params ', num2str(pno), ', Dose=', num2str(Dose), 'mg']);
legend('Concentration', 'Actual Accuity', 'Accuity', ...
        'Tolerance', 'location', 'East')

subplot(1,2,2)
plot(solution_t, solution_a, 'b', 'LineWidth', 3)
grid on
grid minor
hold on
plot(actT, Cont_Accty, 'r', 'LineWidth', 3)
plot(actT, Toler, 'm', 'LineWidth', 3)
xlabel(['Parameters: ', 'k_m=', num2str(best_param(1)), ', M_{inf}=', ...
num2str(best_param(2)), ...
        ', k_{\rm in}=', num2str(best_param(3)), ', ...
        k_{\rm out}=', num2str(best_param(4))])
title(['Stats for patient ', num2str(pno), ', Dose=', num2str(Dose), 'mg']);
legend('Actual Accuity', 'Accuity', ...
        'Tolerance', 'location', 'East')

```

5.4 Code for *model_BCVA*

```

function [Accty]=model_BCVA(pno, P, Meff_flg, E_flg, RA_timing)
Kt=P(1); Minf=P(2); Kin=P(3); Kout=P(4);
global Val_DOSE
kc=0.077; % time is scaled by kc (mult by kc)
dose_scale=119; % 2 -- 238; 6 -- 714
%RA_timing=BCVA(pno).nomtime'; % get visit days for the patient
timing=[]; % initialize timing array
TIME=[]; % initialize intervals b/w shots
timing=Val_DOSE(pno).dose_nomtime'; % days to administer the dose
no_visits = length(RA_timing); % number of actual visits
tmax=RA_timing(no_visits)+7;% last week of timing scale in case shot was administered
% at last visit
no_steps=length(timing); % number of shots
TIME(1)=timing(1); % first interval = first # of days
for i=2:no_steps
    TIME(i)=timing(i)-timing(i-1); % interval b/w shots

```

```

end
TIME(no_steps+1)=tmax-timing(no_steps); % last interval
Dose=max(Val_DOSE(1).bcva_value)*dose_scale; % dose for the patient
y0=[0 0 1]; % Initial conditions (concentr=0 tolerance=0)
% acuity=1
Y=[];
actT=[];
aT=0;
for i=1:no_steps+1
    t=0:0.01:TIME(i)*kc; % run until next dose
    % t = t*kc;
    [t1,ytemp] = ode45(@(t,y) func_conc(t,y,Kt,Minf,Kin,Kout, Meff_flg, E_flg),t,y0);
    t2=t1+aT;
    actT=[actT;t2]; % extend time scale for plotting
    Y=[Y;ytemp]; % extend results scale
    yneg=ytemp(end,:); % collect ode solution at the tail
    y0=yneg; % new IC for the next loop
    y0(1)=y0(1)+Dose; % add dose to concentration in IC
    aT=aT+TIME(i)*kc;
end
% Get only time points and calculated acuity that match original data
i=1; Accty=[]; Tm=[];
if RA_timing(1)< 0
    Tm(1)=RA_timing(1);
    Accty(1)=1;
    fst_j=2;
else
    fst_j=1;
end
for j = fst_j:no_visits
    while RA_timing(j)*kc > actT(i)
        i = i+1;
    end
    Tm(j)=round(actT(i)/kc);
    Accty(j)=Y(i,3);
    i = i+1;
end
end

```

5.5 Code for function *func_fin_contin*

```

function [best_param, resnorm, residual, exitflag]=Fit_continFUN(pno,Meff_flg,E_flg)
global Val_BCVA Val_DOSE
RA_timing=Val_BCVA(pno).nomtime'; % get visit days for the patient
RA_acuity=Val_BCVA(pno).value'; % get actual acuity for the patient
RA_scale=RA_acuity(1); % initial acuity of the patient
solution_a = RA_acuity/RA_scale; % scale acuity to match a(0)=1
solution_t =RA_timing; % setup time scale for actual acuity
dose_scale =119; % dose=2 equiv to dose=238
Dose=max(Val_DOSE(1).bcva_value)*dose_scale; % real dose

```

```

%Initial Parameter Guesses
Kt=0.001; Minf=0.5; Kin=0.45; Kout=0.5;

P_0 = [Kt Minf Kin Kout];           % vector of initial parameters

% Run ODE Model with Original Parameters
% [actT, Orig_Accty, Conc, Toler0]=func_BCVA(pno, P_0, Meff_flg, E_flg, RA_timing);
% Create anonymous function for curve fitting
model_loaded = @(P, RA_timing) model_BCVA(pno, P, Meff_flg, E_flg, RA_timing);
lb=[0 0 0 0];                       % lower bounds for parameters
ub=[0.2 1000 1000 1];               % upper bound for parameters

%% Run model

options = optimoptions('lsqcurvefit','MaxFunctionEvaluations',2000 , 'Display', 'off');
[best_param, resnorm, residual, exitflag] = ... %best_param(1)=km, best_param(2)=
    lsqcurvefit(model_loaded, P_0, solution_t, solution_a,lb,ub,options);

end

```

5.6 Code for the loop - all patients *Fin_contin_Loop*

```

clc
clear all
global Val_BCVA Val_DOSE
load('Val_DOSE')
load('Val_BCVA')
%make a loop that loops through patients (load BCVA.value)
tic;
NumPatients=26;
best_param=zeros(NumPatients,4); %4 is the number of parameters
resnorm=zeros(NumPatients,1);
%residual=zeros(i,24);
Matrix=zeros(NumPatients.*6,8); %first four columns will be best parameters;
% 5th column will be residual norm,
% 6th column will be which MaxEffVersion is used, and
% 7th column will be which DosEffVersion is used.
j=1;
fileID = fopen('Matrix.txt','wt');
for i=1:NumPatients %goes through all the patients%
    for MaxEffVersion=1:3
        for DosEffVersion=1:2
            [best_param, resnorm, ~, ~]=func_Fit_contin(i,MaxEffVersion,DosEffVersion);
            Matrix(j,1)=i;
            Matrix(j,2)=MaxEffVersion;
            Matrix(j,3)=DosEffVersion;
            Matrix(j,4)=resnorm;
            Matrix(j,5:8)=best_param(1:end);
            j=j+1
            fprintf(fileID, '%d %d %d %8.4f %8.4f %8.4f %8.4f %8.4f \n',i,...

```

```
        MaxEffVersion, DosEffVersion, resnorm, best_param(1:end));  
    end  
end  
end  
fclose(fileID);  
toc
```

The CMA Model

Brooks Emerick
Kutztown University

Chris Raymond
University of Delaware

July 17, 2019

Discrete CMA Map

We developed a discrete dynamical system to study the behavior of three concentrations over time. Time, n , is measured in weeks, where each value of n corresponds to a monthly visit. We consider three variables: concentration of the drug in the eye, denoted by c_n ; maximum effectiveness of the drug, denoted by m_n ; and best corrected visual acuity (BCVA) level, denoted by a_n . The non-dimensional CMA model is

$$c_{n+1} = k_c c_n + D \delta_{n+1,d} \tag{1}$$

$$m_{n+1} = m_\infty - k_m c_n + k_\infty (m_n - m_\infty) \tag{2}$$

$$a_{n+1} = k_{\text{in}} [1 + m_n \cdot H(c_n - 1)] + k_{\text{out}} a_n, \tag{3}$$

where $H(\cdot)$ is the Heaviside function, together with the following initial conditions:

$$c_0 = 0, \quad m_0 = m_\infty, \quad a_0 = 1. \tag{4}$$

The parameter k_c represents the natural decay of the concentration in the body, and for c_n to remain positive, we must have $0 < k_c < 1$. The parameter D is the dosage level, and $\delta_{n+1,d}$ is the delta function, which is one if a dose was administered and zero otherwise. The value d represents the week at which a dose was administered, and the dose is immediately added to the concentration on the same week it is administered. The value m_∞ is the maximum effectiveness of the drug, k_m is the rate at which the maximum effectiveness decreases due to drug tolerance, and k_∞ is the rate of natural rebound of drug effectiveness. Finally, k_{in} measures the strength of the response to the drug on the overall acuity and k_{out} is a measure of the decay of acuity due to no treatment. Since k_{out} is a natural decay of the acuity level, we have $0 < k_{\text{out}} < 1$.

The model assumes that a positive response always occurs when a dose is administered. This response, however, is not always the same, as the body will build a tolerance to the

drug, suggesting that maximum effectiveness decays with regular doses. The data is rather noisy and in some instances we don't always see a positive response to the drug. In fact, a patient may continually get worse even though he/she is receiving regular doses. We were able to estimate three of the seven parameter values (see caption of Figure 1 for values). The specific values for m_∞ , k_m , k_{in} , and k_{out} are thought to be patient dependent. Our goal was to optimize the four parameters of this model to each of the 103 test patients. After the parameters were fit for each patient, we also sought to determine if any values of k_{in} or k_{out} were characteristic of responsive or non-responsive patients.

Numerical Results and Parameter Fitting

We run this model using the dosage schedule of Patient 6 with a demonstrative set of parameter values. A typical trajectory of the concentrations of the drug, c_n ; maximum effectiveness, m_n ; and acuity, a_n is shown in Figure 1. We can see that the concentration of

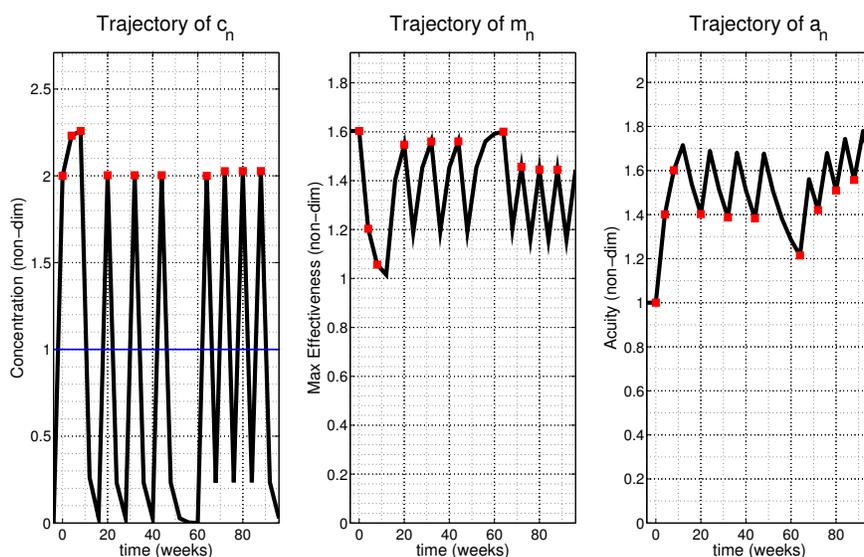


Figure 1: Patient 6 trajectory of c_n , m_n , and a_n with fixed parameters $k_c = .1157$, $D = 2$, and $k_\infty = .25$. To demonstrate the output of the model, we choose the unknown parameters to be $k_{in} = .25$, $k_{out} = .75$, $k_m = .2$, and $m_\infty = 1.6031$.

the drug is oscillatory in nature due to the frequency of the dosages. The red blocks on the output represent when a dose was administered. When a dose occurs, the concentration is at a maximum. The concentration, then, declines in the four weeks between visits. The blue line corresponds to the threshold value of the Heaviside function, and so if the concentration is below that line, the drug will not contribute to an increase in acuity. In the plot of m_n , we can see that the maximum effectiveness decreases after every dose, suggesting that a tolerance is taking effect and limiting the body's response to the drug. Finally, we can see

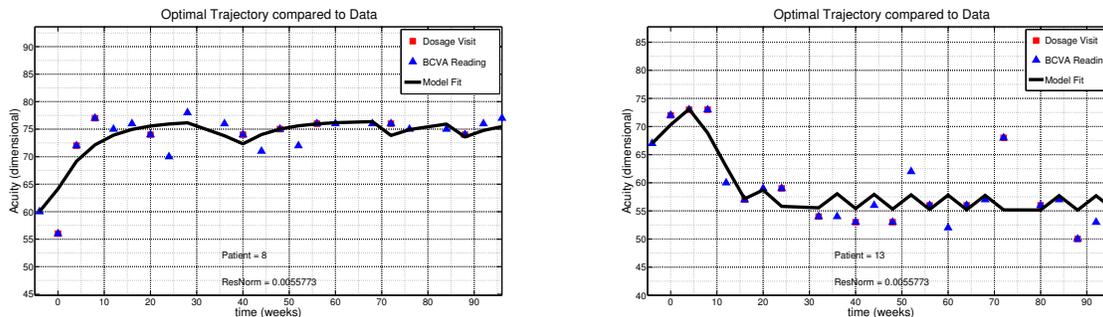


Figure 2: Optimized trajectory for acuity of Patients 8 and 13. The model does well to fit both responsive and non-responsive patients.

that acuity increases after every dose. Furthermore, we can see that between weeks 44 and 60 (when the patient missed a dose), the acuity declines more dramatically over that interval.

In general, the model does well to capture the patient data. In Figure 2, we optimize the four unknown parameters using MATLAB’s built-in `lsqnonlin` function for the acuity levels of Patient 8 and Patient 13. Because this solver inputs an initial parameter set, we also implemented MATLAB’s global optimization function `multistart` with 100 other initial conditions.

We optimized the four parameters for every patient in the given data set. We performed a statistical analysis on the set of optimized parameters to determine if non-responsive patients were more likely to have a value of k_{in} closer to 0, which would suggest that the drug has much less effect. Figure 3 shows the value of k_{in} and k_{out} for patients that are responsive (positive effect, blue) and non-responsive (negative effect, red). The measure of responsiveness is calculated from the data by considering the ratio of final acuity to initial acuity. If this value is greater than one, the drug had a positive effect on the patient; otherwise, the patient was non-responsive and the drug had a negative effect. From the figure, we see that there is slight clustering of the the non-responsive patients in the top-left corner, suggesting the $k_{in} \approx 0$ and $k_{out} \approx 1$. This confirms our suspicion for non-responsive patients. However, the responsive patients have quite a range of parameter values, suggesting that more statistical analysis should be done to elucidate any correlations between parameters and responsiveness.

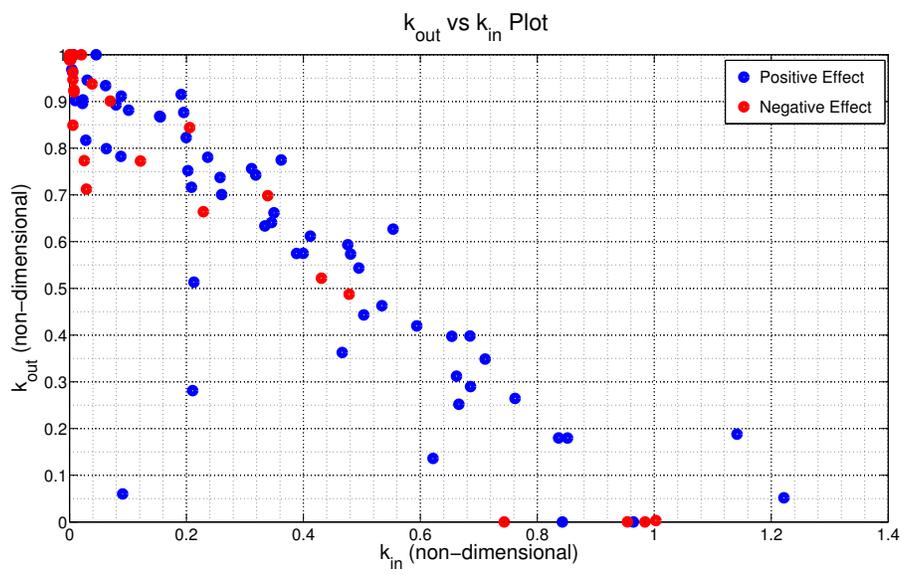


Figure 3: Scatterplot of optimized values k_{in} and k_{out} , separated by patient responsiveness.

Estimating Parameters of the Best Corrected Visual Acuity Discrete Map using Variational Data Assimilation

Matthew Moye
Department of Mathematical Sciences
New Jersey Institute of Technology

August 9, 2019

1 The Map

The discrete representation for the best corrected visual acuity (BCVA) measurement can take on the following form:

$$C_{n+1} = \kappa_C C_n + D\delta_{n+1,d} \quad (1)$$

$$T_{n+1} = \kappa_T C_n + \kappa_\infty T_n \quad (2)$$

$$a_{n+1} = \kappa_{\text{in}} [1 + M(T_n)E(C_n)] + \kappa_{\text{out}} a_n \quad (3)$$

where

$$\delta_{n+1,d} = \begin{cases} 1, & \text{dose given at time } n + 1 \\ 0, & 0 \text{ else.} \end{cases} \quad (4)$$

Here, C_n represents the dimensionless concentration of the medication, T_n the drug tolerance, and a_n the acuity.

We have explored modeling the dosage effectiveness, $E(C)$, and the maximum effectiveness, $M(T)$, in a variety of ways:

1. $E(C) = H(C - 1)$

2. $E(C) = \frac{C}{C+1}$

A. $M(T) = M_\infty(1 - T)$

B. $M(T) = \frac{M_\infty}{1+T}$

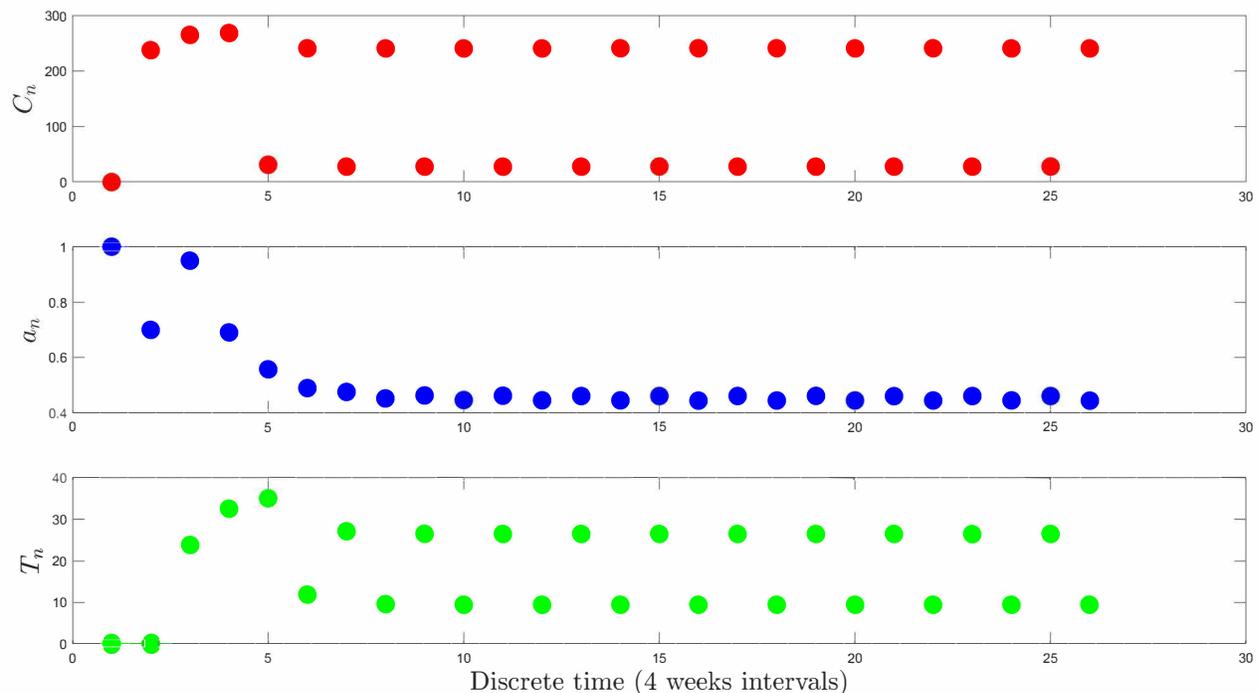


Figure 1: One run of the map. Here, $\kappa_{\text{in}} = .2$, $\kappa_{\text{out}} = .5$, $\kappa_T = .1$, $M_\infty = 2$.

C. $M(T) = M_\infty e^{-T}$.

We will begin by focusing on the formulation B.1, where $H(C - 1)$ is a Heaviside function.

Figure 1 shows a typical iteration of this system for each of the variables. The system can exhibit stable cycles or approach a stable fixed point.

We have a variety of known information at our disposal, namely the dosage schedule and dose, D and the rate constants κ_∞, κ_C . Additionally, we have each of the the initial conditions: $C_0 = 0$, $T_0 = 0$, $a_0 = 1$ reflecting the lack of concentration of the drug prior to injection, no accumulated tolerance, and a normalized acuity at the start. The other four parameters remain unknown, but have physiological bounds given in Table 1.

Table 1: Parameter bounds for discrete map

Parameter	Lower bound	Upper bound
κ_T	0	1
κ_{in}	0	∞
κ_{out}	0	1
M_∞	0	∞

The variables in our discrete map are non-dimensionalized, but their domains are $(0, \infty)$. The acuity, a_n , is roughly $O(1)$ after normalizing based on a baseline acuity reading to obtain a_0 .

Our goals will be to:

1. Obtain parameter estimates for each patient
2. Recognize any patterns amongst parameter clusters
3. Determine predictability of the model for patient acuity outcomes

Naturally, to answer #3, we would likely need longer time windows so that we could split up each patient's history into training data and testing data. With only a couple dozen data points at our disposal per patient, doing so at the moment is not necessarily feasible. However, this demands further investigation going forward to verify our (and any) useful model.

2 Variational Data Assimilation

To address Goal 1, we could use any number of parameter estimation methods, such as evolutionary algorithms, Markov chain Monte Carlo algorithms, simplex algorithms, etc. Our approach is a variational method derived from application to weather prediction. Data assimilation has been previously utilized for long-term patient treatments [1]. The following paragraphs will explain the methodology, but for brevity one could skip to the results in the following section.

This description is adapted from [4]. We will represent the evolution of the variables by:

$$x_{n+1} = f(x_n) + \omega_{n+1}, \quad \omega_n \in R^L \tag{5}$$

$$y_{n+1} = h(x_{n+1}) + \eta_{n+1}, \quad \eta_{n+1} \in R^M, \tag{6}$$

where $L = 3$ and $M = 1$ for the acuity as our only measured variable of the system.

In this state-space representation, x_n is interpreted as the state of the system at each time t_n , and y_n are our observations. f is our discrete model which maps states x_n between time points t_n and t_{n+1} . h is our observation operator which describes how we connect our observations y_{n+1} to our state space at t_{n+1} . We take $t_{n+1} - t_n = 4$ weeks. The random variables ω_{n+1} and η_{n+1} represent model error and measurement error respectively. We will assume our measurements are riddled with Gaussian noise, and that any errors in our model can be approximated by Gaussian noise as well. Then $\omega_n \sim \mathcal{N}(0, Q_n)$ and $\eta_n \sim \mathcal{N}(0, R_n)$, where Q_n is our model error covariance matrix and R_n is our measurement error covariance matrix.

We now have defined a stochastic discrete dynamical system where we have characterized the evolution of our states and observations therein based upon assumed error statistics. The goal of doing so is that we can find state estimates at each time point and parameter estimates

when the data is noisy and our model may be slightly inaccurate. The data assimilation method we use specifically is referred to as “4D-Var”.

Specifically, “strong 4D-Var” involves solving the following cost function:

$$C(x) = \frac{1}{2} \sum_{k=0}^N \|y_n - h(x_n)\|_{R_n}^2 \quad (7)$$

$$= \frac{1}{2} \sum_{k=0}^N R_n^{-1} (y_n - a_n)^2 \quad (8)$$

subject to

$$x_{n+1} = f(x_n), \quad x_n = (C_n, a_n, T_n). \quad (9)$$

We will use our estimates of the covariances of the measurement error and model error as weighting factors in the final formulated Lagrangian. Given that we have no prior knowledge of the measurement fidelity and precisely how accurate our model will be, for the rest of the chapter, we will set $R^{-1} = 1000$ and the weight associated with each dynamic constraint is $g_w = (.01, 1, .01)$. These will be additional multipliers onto the Lagrange multipliers in the problem. The weight for a_n is 100 times greater to reflect the difference in its intrinsic absolute variance compared to T_n and C_n and to keep each constraint similarly scaled.

We will solve this system using CasADi in MATLAB interfaced with IPOPT, an interior point optimizer [3]. CasADi will perform automatic differentiation (AD), and these derivatives will be passed to our solver [2]. For our problem, we take what is akin to a “multiple shooting” approach from trajectory optimization. Namely, we are trying to not only estimate the parameters, but each of the state variables at each time point. Therefore, we have an $NL + P$ -dimensional optimization problem. The size remains manageable for the present amount of data, ≈ 100 , but it is not uncommon for continuous dynamical systems to result in $10^k < NL + P$ variables for $k > 4$. For many problems, including this one, the Hessian of the Lagrangian will be sparse, so the technique scales quite well. Each run of the optimization takes less than a second to obtain a final estimate on a typical laptop.

We set the upper bounds for M_∞ and κ_{in} at 100, and the bounds for each of our variables somewhat arbitrarily high at 10^5 . Each state variable at every time point and each parameter is initialized randomly between its lower and upper bounds as our initial guess \mathbf{x}_0 of the system. Over the course of the optimization cycle, the solver will eventually reach a local minimum. Due to the nonconvexity associated with this nonlinear map, there may be multiple local minima emerging. To find our best fit for each patient, we will initialize the system randomly and solve it for some number of times, in this case $N_{init} = 25$, and take the solution which has the lowest value of the objective function if it suitably matched the other optimality criterion.

3 Results

3.1 Simulated data experiment

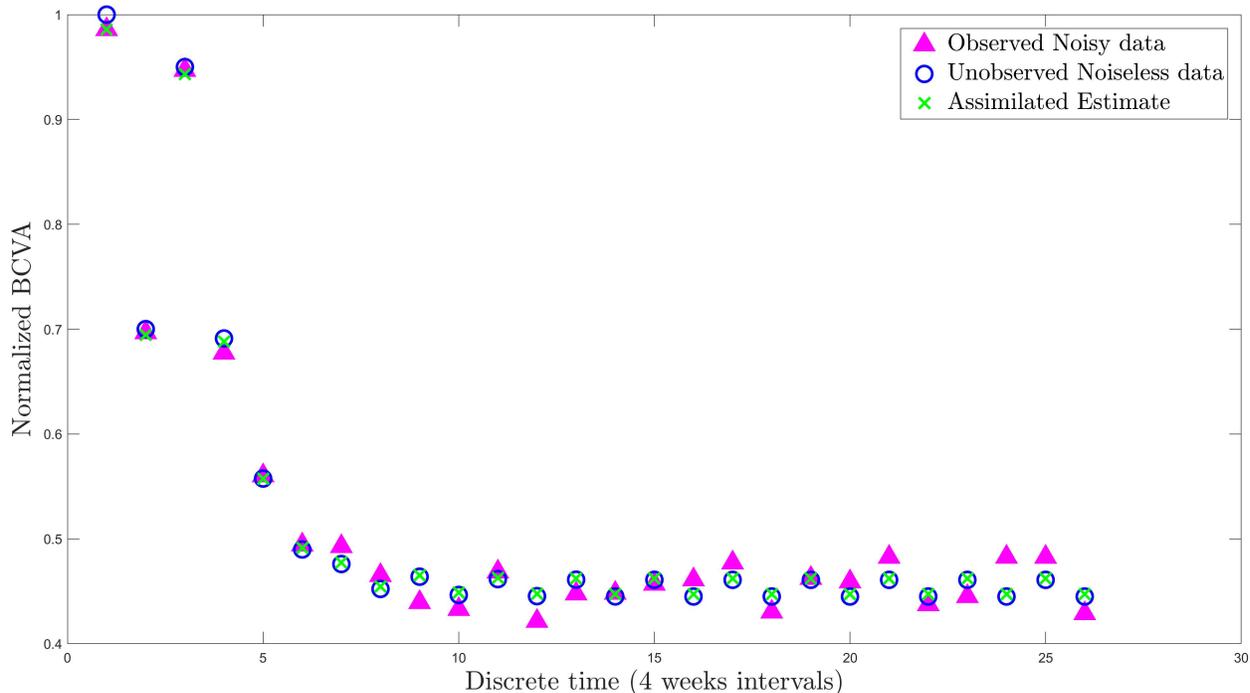


Figure 2: Discrete map twin experiment. The method is presented with the noisy data. The resulting estimate closely matches the unobserved, clean data.

Prior to exploring real data, we check that our methodology works for acuity data generated from the map, like that from row 2 of Figure 1. In the geosciences, this is referred to as a “twin experiment”, where one simulates data, then has the assimilation technique try to estimate the parameters which generated that data. Here, we can match the estimated parameters against their true values to validate performance; when working with real data, we do not have such knowledge available.

As an example consider Figure 2. Here, we perturbed the signal so that our observation

$$y_n = a_n + \epsilon_n \sqrt{\text{Variance}(\mathbf{a})} \quad (10)$$

for $\mathbf{a} = [a_0, a_1, \dots, a_N]$ and $\epsilon_n \sim \mathcal{N}(0, .01)$. Despite our reasonable perturbation from the clean signal with our noisy observations, the DA technique is able to sufficiently recover the clean dynamics.

As an exercise, we can use our created data from say B.1 and try to estimate assuming the model structure is A.1. Given that A.1 follows from an expansion of B.1., we would hope

that we could find similar parameters. Indeed we do, as the parameter estimates are given in Table 2.

Table 2: Parameter estimates for twin experiment. The true model for generating the data is given by formulation B.1. Observation generated from clean signal as in equation (10).

Parameter	True	Est. using A.1.	Est. using B.1.
κ_T	.2	0.2047	.2037
κ_{in}	.5	0.4968	.4978
κ_{out}	.1	0.1080	.1066
M_∞	2	1.9306	1.9329

3.2 Patient data

As we have checked our methodology on simulated data, we hope to now be able to do so with the the patient data. We conducted this for 103 patients whose BCVA data we had. We assumed that the dose volume was the same for each patient when they receive treatment. Following the estimations, 23 patients were retained in the sample given that the estimated parameters didn't merely hug one of their boundaries, and the value of the cost function at the optimal solution was sufficiently small. This doesn't mean that the solver was inappropriate for these patients, but may mean that further algorithmic parameter tuning would be required to get satisfactory results, *e.g.*, tolerances, bounds, etc.

The following results were generated assuming model formulation B.1, but similar results were obtained using A.1.

One hypothesis was that the predictive power of the model could be encapsulated into a simple metric; if $\kappa_{\text{in}} < \kappa_{\text{out}}$, acuity would improve over the course of the dosing period. However, as shown in Figure 3, there are a significant portion of the patients whose $\frac{\kappa_{\text{in}}}{\kappa_{\text{out}}} > 1$, but the blue \times 's indicate that these patients' acuity increased overall with the dosages. One such example is shown in Figure 8. Here, each iterate a_n^{est} generated by iterating the map with the estimated parameters is shown. Additional patients' estimates are given in Figures 5–7.

More consideration is required in how to cluster patients through parameter relationships, or a Bayesian estimator should be utilized to uncover parameter covariates. Full histograms for these patients are shown in Figure 4.

These techniques can naturally be extended to more complex, continuous models. Any interest in the code, or supplementary plots, feel free to contact me at mjm83@njit.edu.

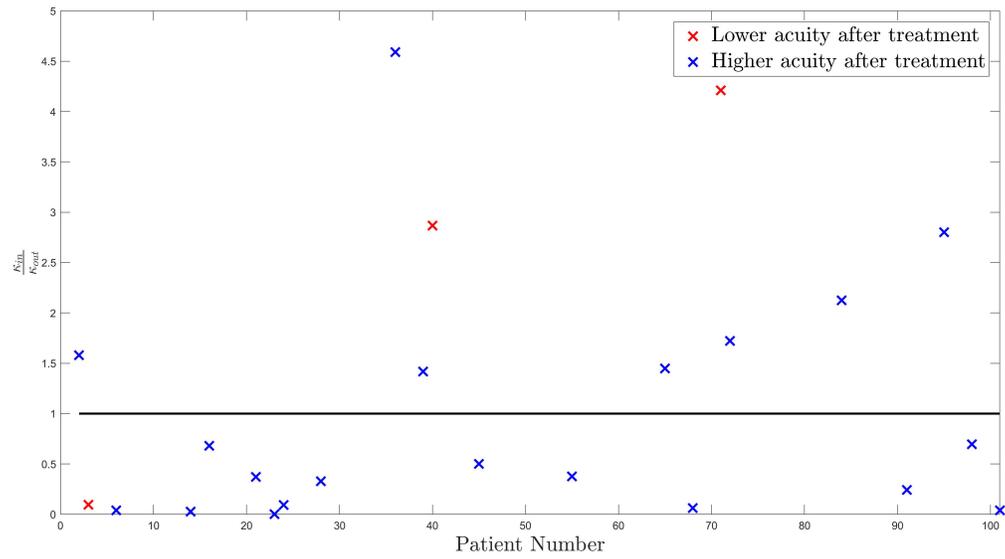


Figure 3: Ratio of κ_{in} , κ_{out} is shown. Blue \times 's have increased visual acuity after the dosing schedule. Red \times 's have decreased acuity after the dosing schedule.

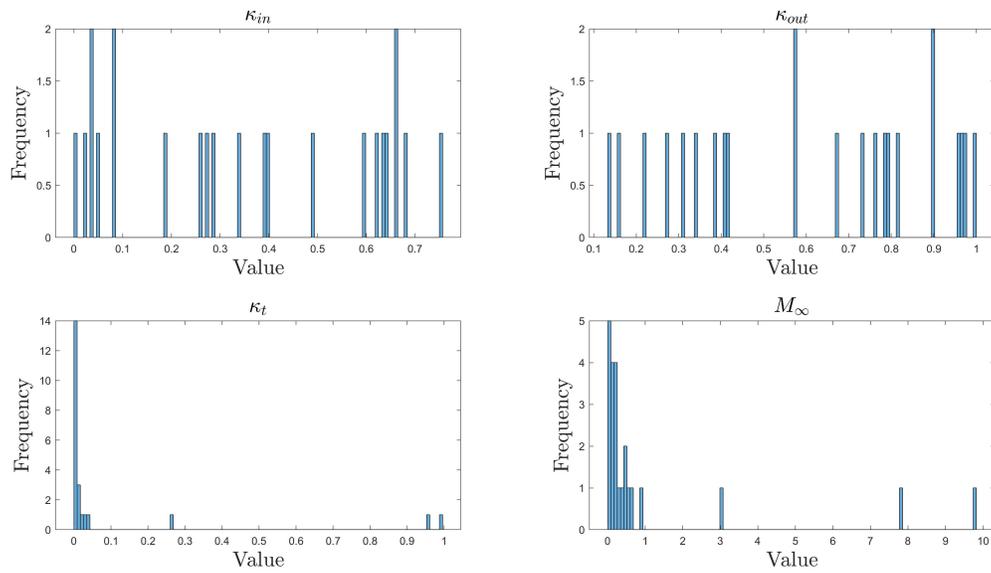


Figure 4: Parameter histograms. Assumed model formulation was B.1. Data was generated from 23 patients in total.

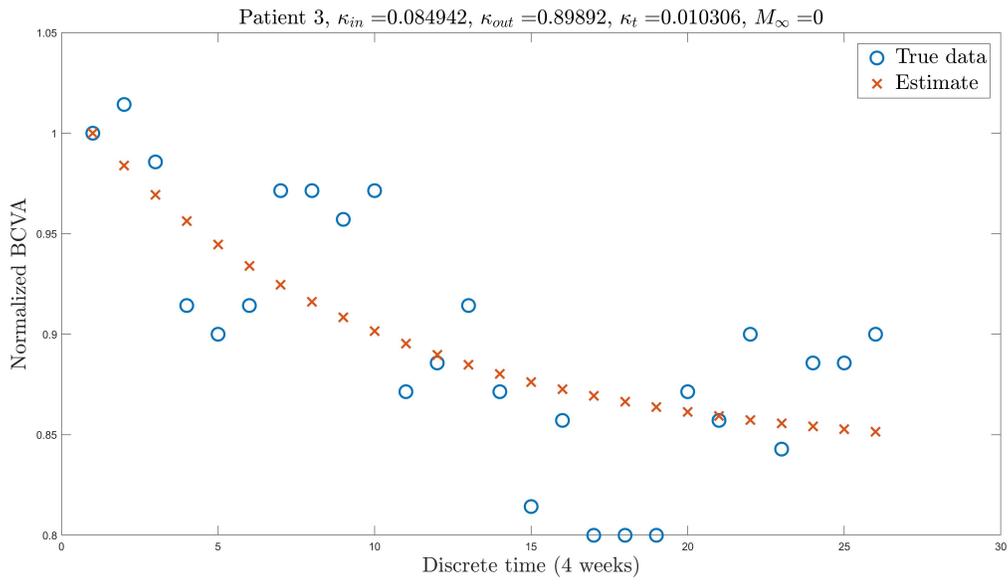


Figure 5: Estimated model for Patient 3.

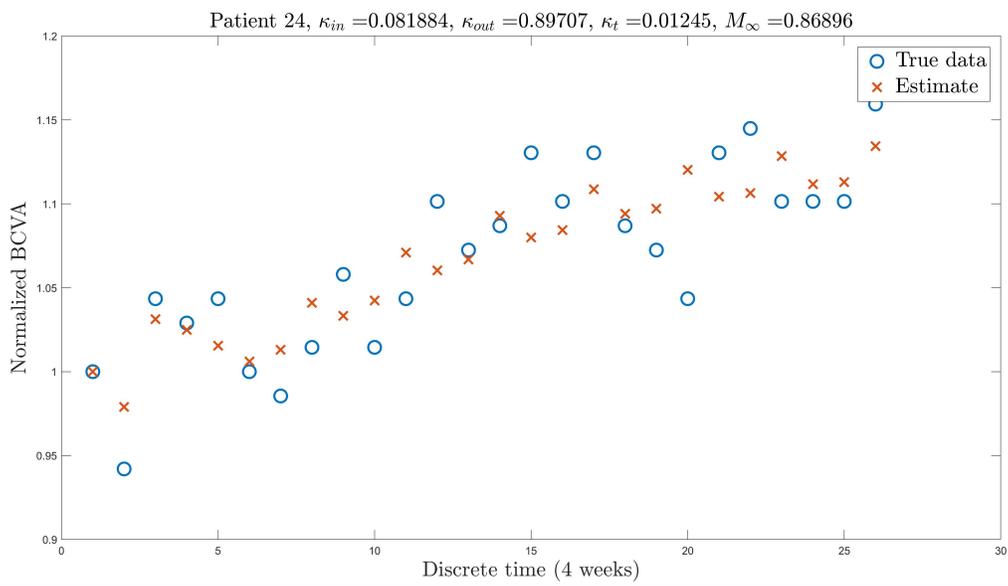


Figure 6: Estimated model for Patient 24.

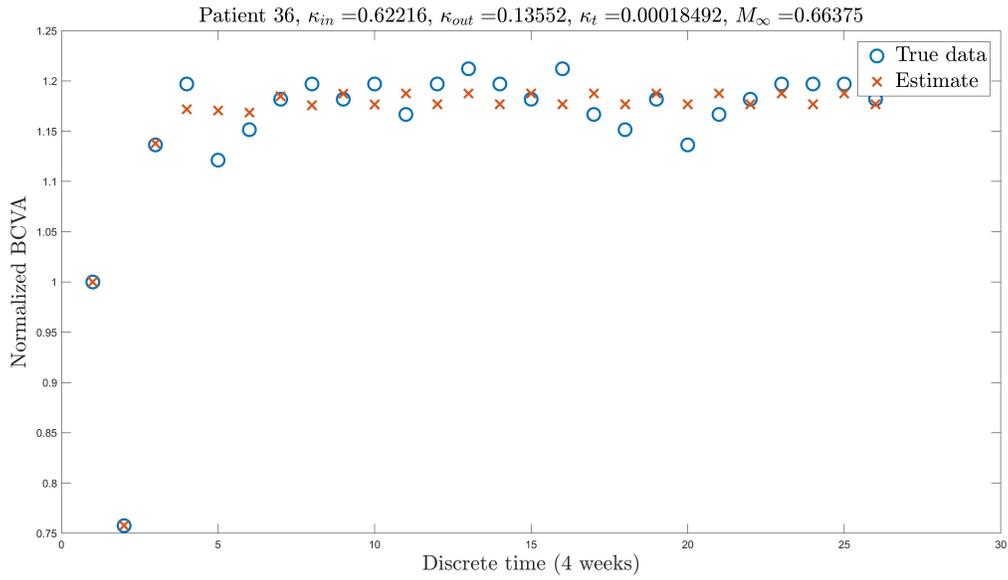


Figure 7: Estimated model for Patient 36.

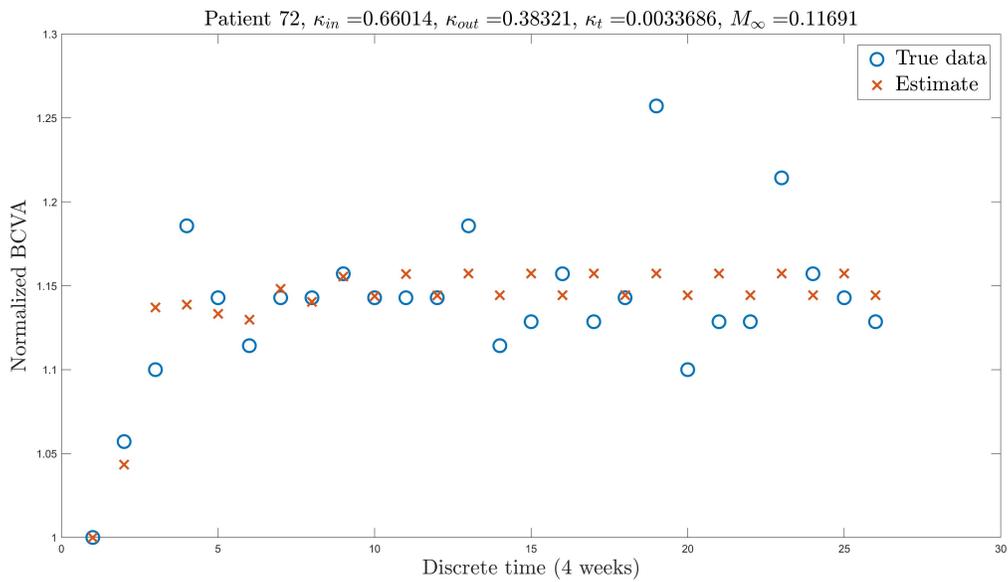


Figure 8: Estimated model for Patient 72.

References

- [1] David J. Albers, Matthew E. Levine, Andrew Stuart, Lena Mamykina, Bruce Gluckman, and George Hripcsak. Mechanistic machine learning: How data assimilation leverages physiologic knowledge using Bayesian inference to forecast the future, infer the present, and phenotype. *Journal of the American Medical Informatics Association*, 25(10):1392–1401, 2018.
- [2] Joel A.E. Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, Mar 2019.
- [3] W Andreas. Short tutorial: Getting started with IPOPT in 90 minutes. 2009.
- [4] Matthew J Moye and Casey O Diekman. Data Assimilation Methods for Neuronal State and Parameter Estimation. *The Journal of Mathematical Neuroscience*, 8(1):11, Dec 2018.

Diffusion model of drug delivery

Maxim Zyskin
University of Oxford

August 1, 2019

1 Introduction

We describe here a physical, diffusion-based model of medication delivery. We assume that medication is injected into the eyeball at the corner of the eye lid, and has to travel to the retina (at the back of the eyeball) by a diffusion mechanism inside a 3-dimensional ball. We describe an approach to model such a diffusion by mostly analytical means based on an eigenfunction expansion the Laplace operator, with some numerics required to compute the appropriate eigenvalues and to plot the results.

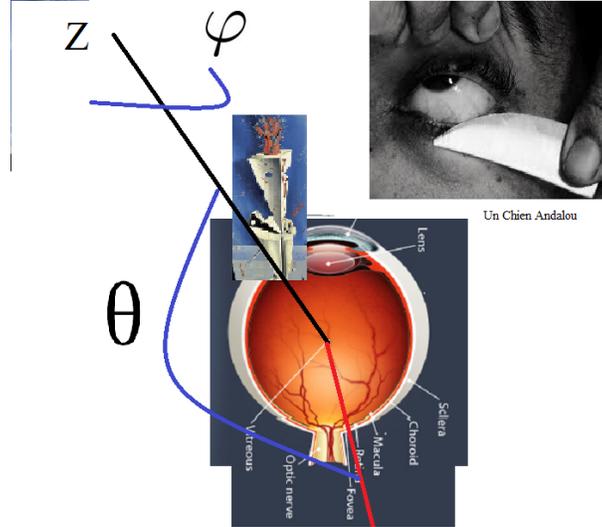
We discuss two types of boundary conditions:

- (a) Neumann boundary conditions prevent any leakage of medicine outside of the eyeball, so that all medicine delivered remains in the eyeball. Further modeling is then needed to account for any medication tolerance, which reducing its effect.
- (b) To account for possible medicine leakage, we add an extra thin outer layer to the eyeball region. We take the outer boundary condition as Dirichlet, which ensures that there is no medication outside of that added extra layer. We take the diffusion coefficient in the outer layer as a parameter, which is smaller then diffusion coefficient in the eyeball itself. This accounts for the fact that leakage is assumed to be quite small, compared to diffusion in the eyeball itself delivering the medicine. On the interface between the eyeball and the outer layer we require that the concentration of medicine is continuous, and that the flux (*i.e.*, the normal derivative weighted by the diffusion coefficient) is continuous when computed from the two sides of the eyeball-extra layer interface.

We describe some of the computational matters encountered, the conclusions, and possible directions of further research.

2 Eyeball diffusion Model

Our model aims to model the concentration $c(\mathbf{x}, t)$ of medicine, which satisfies a diffusion equation in a ball in \mathbb{R}^3 , with appropriate boundary conditions. The initial conditions are



chosen to model drug delivery by injections, which are highly localized in space and time, into a small volume inside the ball, quite close to the boundary. The target region, the retina, is also located on the boundary of the eyeball, and is nearly diametrically opposite to the injection area.

By choosing a coordinate system and rescaling (\mathbf{x}, t) appropriately, we can take the eye as a unit ball B_1 centered at the origin, and the diffusion coefficient in the eyeball to be one (the latter is fixed by selecting an appropriate time scale). The drug is delivered at a point on the z -axis $\mathbf{x}_0 = (0, 0, \zeta)$, where $0 < \zeta < 1$ and $1 - \zeta \ll 1$, at times $\{t_i\}$:

$$\begin{aligned} \frac{\partial c(\mathbf{x}, t)}{\partial t} &= \Delta c(\mathbf{x}, t), \mathbf{x} \in B_1, t_{i-1} < t < t_i; \\ c(\mathbf{x}, t_{i-1} + \varepsilon) &= c(\mathbf{x}, t_{i-1} - \varepsilon) + d_i \delta(\mathbf{x} - \mathbf{x}_0), \\ c(\mathbf{x}, 0) &= 0. \end{aligned} \quad (1)$$

Boundary conditions. (a) In the leakless model, the boundary conditions are the Neumann conditions,

$$\nabla_n c|_{\partial B} = 0. \quad (2)$$

(b) The leakage is modeled by adding a thin spherical layer, $1 < r < 1 + h$, $0 < h \ll 1$ in the spherical coordinates (r, θ, ϕ) . The diffusion coefficient d in the layer is taken smaller than the diffusion coefficient in the eyeball, so $d < 1$. The concentration and the flux are assumed continuous at the eyeball-extra layer interface, and with Dirichlet boundary conditions on the boundary of the extended region B_{1+h} :

$$\begin{aligned} c|_{\partial B_{1+\varepsilon}} &= c|_{\partial B_{1-\varepsilon}}, \\ d \nabla_n c|_{\partial B_{1+\varepsilon}} &= \nabla_n c|_{\partial B_{1-\varepsilon}}, \\ c|_{\partial B_{1+h}} &= 0. \end{aligned} \quad (3)$$

Here $B_{1\pm\varepsilon}$ refers to the eye-layer interface as approached from the eye ($-$) or layer ($+$).

3 Eigenfunction expansion

Recall that on \mathbb{R} ,

$$\begin{aligned} c(x, t) &= \int \frac{e^{ikx}}{\sqrt{2\pi}} \frac{e^{-iky}}{\sqrt{2\pi}} e^{-k^2 t} c(y, 0) dk dy = \int \frac{e^{-\frac{(x-y)^2}{4t}}}{2\sqrt{\pi t}} c(y, 0) dy, \\ \psi_k &= \frac{e^{ikx}}{\sqrt{2\pi}}, \Delta \psi_k = -k^2 \psi_k, \\ \int \overline{\psi_{k_1}(x)} \psi_{k_2}(x) dx &= \delta(k_1 - k_2). \end{aligned}$$

Similarly on B ,

$$\begin{aligned} c(\mathbf{x}, t) &= \int_B d\mathbf{y} \sum_{\mathbf{n}} \psi_{\mathbf{n}}(\mathbf{x}) \psi_{\mathbf{n}}(\mathbf{y}) e^{-\lambda_{\mathbf{n}}(t-t_{i-1})} c(\mathbf{y}, t_{i-1} + \varepsilon), \quad t_{i-1} < t < t_i, \\ \Delta \psi_{\mathbf{n}} &= -\lambda_{\mathbf{n}} \psi_{\mathbf{n}}, \\ \int_B \psi_{\mathbf{n}} \psi_{\mathbf{n}'} d\mathbf{y} &= \delta_{\mathbf{n}, \mathbf{n}'}, \end{aligned} \tag{4}$$

where $B = B_1$ or $B = B_{1+h}$, and the eigenfunctions satisfy boundary conditions (2) or (3) in cases (a) or (b), accordingly.

$$\nabla_n \psi_{\mathbf{n}}|_{\partial B} = 0, \tag{5a}$$

$$\psi_{\mathbf{n}}|_{\partial B_{1+\varepsilon}} = \psi_{\mathbf{n}}|_{\partial B_{1-\varepsilon}}, \tag{5b}$$

$$\begin{aligned} d\nabla_n \psi_{\mathbf{n}}|_{\partial B_{1+\varepsilon}} &= \nabla_n \psi_{\mathbf{n}}|_{\partial B_{1-\varepsilon}}, \\ \psi_{\mathbf{n}}|_{\partial B_{1+h}} &= 0. \end{aligned} \tag{6}$$

We note that those boundary conditions are linear and homogeneous, so that linear combinations of functions which satisfy them also satisfy the same boundary conditions. (That would not be the case for example if a flux is prescribed on the boundary, in which case the method needs to be modified, for example by expanding a difference with a particular solution).

Thus the function defined by (4) obviously satisfies the equation and the boundary conditions. Now taking $t = t_{i-1}$, it follows from orthogonality and normalization of the eigenfunctions that

$$\int_B d\mathbf{x} \psi_{\mathbf{m}}(\mathbf{x}) \left[\int_B d\mathbf{y} \sum_{\mathbf{n}} \psi_{\mathbf{n}}(\mathbf{x}) \psi_{\mathbf{n}}(\mathbf{y}) \right] = \psi_{\mathbf{m}}(\mathbf{y}). \tag{7}$$

Thus if the collection of eigenfunctions is complete, the expression in brackets is a δ -function, and so the initial conditions are satisfied by (4) as well.

Eigenvalues of the Laplacian on a ball can be given as products of spherical functions accounting for the angular part, and spherical Bessel functions to account for the radial part

(such expansions are used in textbooks on quantum mechanics). Our model has a symmetry which is rotation about the z -axis (the axis of the injections), and we seek a solution which is symmetric with respect to such rotations. For that reason we do not need to include the azimuthal angle ϕ dependence, and the spherical functions needed are the Legendre polynomials of $\cos(\theta)$.

Thus in the case (a),

$$\begin{aligned} \psi_{\mathbf{n}} = \psi_{l,i} = \psi_{l,i}(r, \theta) &= j_l(\kappa_{l,i}r)P_l(\cos \theta), \quad l = 0, 1, \dots, \lambda_{l,i} = \xi_{l,i}^2, \quad n \neq 0; \\ \left. \frac{d}{dr} j_l(\kappa_{l,i}r) \right|_{r=1} &= 0, \\ \int_0^1 j_l(\kappa_{l,i}r)r^2 dr &= 1. \end{aligned} \quad (8)$$

where P_l are Legendre polynomials, the j_l are spherical Bessel- J_l functions scaled to have the unit norm, $\kappa_{l,i}$, $i=1,2,\dots$ are zeroes of the derivative of $j_l(x)$. In addition, a constant is an allowed eigenfunction:

$$\psi_0 = \frac{1}{\sqrt{V_B}}, \quad V_B = \frac{4\pi}{3}, \quad \lambda_0 = 0. \quad (9)$$

In the case (b),

$$\begin{aligned} \psi_{l,i}(r, \theta) &= f_l(\kappa_{l,i}r)P_l(\cos \theta), \quad \lambda_{l,i} = \xi_{l,i}^2 \neq 0, \\ f_l(\kappa_{l,i}r) &= \begin{cases} \alpha_{l,i}j_l(\kappa_{l,i}r) + \beta_{l,i}y_l(\kappa_{l,i}r) & 1 < r < 1+h \\ \gamma_{l,i}j_l(\kappa_{l,i}r) & 0 < r < 1 \end{cases} \end{aligned} \quad (10)$$

where j_l , y_l are the spherical Bessel- J_l and Bessel- Y_l functions. Here $\kappa_{l,i}$, $i = 1, 2, \dots$ are zeroes of the determinant, and $v_{l,i} = (\alpha_{l,i}, \beta_{l,i}, \gamma_{l,i})^T$ are vectors in the null space of the following linear system:

$$M(\kappa_{l,i})v_{l,i} = 0, \quad \det M(\kappa_{l,i}) = 0, \quad M(\kappa) = \begin{pmatrix} j_l(\kappa_{l,i}(1+h)) & y_l(\kappa_{l,i}(1+h)) & 0 \\ d j_l'(\kappa_{l,i}) & d y_l'(\kappa_{l,i}) & -j_l'(\kappa_{l,i}) \\ j_l(\kappa_{l,i}) & y_l(\kappa_{l,i}) & -j_l(\kappa_{l,i}) \end{pmatrix}, \quad (11)$$

where d is the outer layer diffusion coefficient, and f' stands for the derivative of a function. The required normalization is dictated by the condition that the Laplace operator with our boundary conditions is self-adjoint on B_{1+h} :

$$d \int_1^{1+h} |\alpha_{l,i}j_l(\kappa_{l,i}r) + \beta_{l,i}y_l(\kappa_{l,i}r)|^2 r^2 dr + \int_0^1 |\gamma_{l,i}j_l(\kappa_{l,i}r)|^2 r^2 dr = 1. \quad (12)$$

4 Finding eigenvalues and investigating solutions

Our adopted method requires finding eigenvalues and normalization constants of the eigenfunctions. The latter are expressed in terms of spherical Bessel functions; and in a nutshell

we need to find zeroes of certain special functions of a single variable. Those functions are well studied, yet finding zeroes is a transcendental problem which requires numerical work. Those zeroes can be found numerically with any precision using standard methods for finding zeroes numerically. The version of Mathematica we possess does not appear to have a program specifically for finding zeroes of the derivative of spherical Bessel functions (while computing spherical Bessel functions themselves, as well as Legendre polynomials, are included). However using the standard `NSolve` command in Mathematica, supplemented with bounds on eigenfunctions, works quite well (and may be further tested by plotting those functions of which we need zeroes). For example, finding zeroes of the derivative of a spherical Bessel J function may be accomplished with a one-liner like

```
f = D[SphericalBesselJ[0, x], x]; NSolve[f == 0 && x < 200, x]
```

We note that since the injection is described by a δ -function, for a single injection of medicine, the integration in (4) is not needed to find the solution. Instead, we take the appropriate $\psi_n(\mathbf{y})$ at the injection point (\mathbf{x}_0). For multiple injections we can use linearity and the Markovian property to find the solution:

$$c(\mathbf{x}, t) = \sum_{i: t_i \leq t} c_1(\mathbf{x}, t - t_i) \quad (13)$$

Thus after finding eigenvalues and normalizations, we need just to sum over contributions of various well-known functions involving spherical Bessel functions and Legendre polynomials. That task becomes complicated when time t is slightly greater than the injection time, as exponential damping $\exp(-\kappa^2(t - t_i))$ is suppressed. Hence we need to include contributions of many (ideally ‘all’) eigenfunctions to accurately represent the solution (which will have a sharp peak at the injection point \mathbf{x}_0). However when $t - t_i$ is not small, the contribution from modes with large eigenvalues is strongly suppressed, and we need to include just a few modes: those with smallest eigenvalues.

In the version (a), the constant part of the solution corresponding to the ψ_0 eigenfunction always remains present at all times, while all other modes vanish with time. Hence the total amount of medicine $\int_B c(\mathbf{x}, t) d\mathbf{x}$ equals the total dosage delivered in injections.

In the version (b), all modes ‘leak out’, the amount of delivered medicine is not preserved, and without injections the solution approaches $c = 0$ (imposed by the outer Dirichlet condition) at an exponential rate, with the slowest decaying mode corresponding to the lowest eigenvalue.

In certain cases, other eigenvalues are well separated from the lowest eigenvalue. (In particular, we observed that happening when the outer layer diffusion coefficient is small and the problem approaches the no-leakage case (a), which has a persistent zero eigenvalue.) In that case, if time intervals between injections are large enough for higher modes to decay, but not so for the lowest eigenvalue, we may approximate the solution by that single lowest mode with eigenvalue $\lambda_1 = \min \{\kappa_{l,i}^2\}$. The solution (as well as any quantity computed from it, such as total amount of medicine, or concentration at a particular location) is then,

assuming all injection amounts and placements are the same,

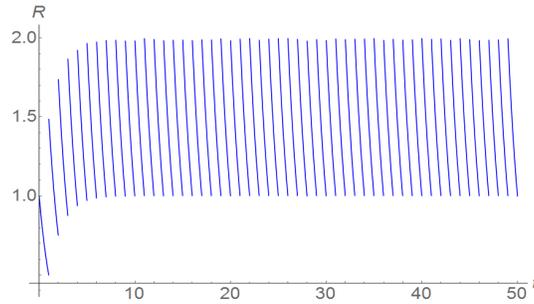
$$c(x, t) = c_1(x, 0) \sum_{i: t_i \leq t} \exp(-\lambda_1(t - t_i)), \quad (14)$$

where $c_1(x, 0)$ is the lowest eigenmode part for a single injection solution.

If there have been n periodic injections prior to time t at times $t_i = it_{\text{per}}$, where t_{per} is the period, our solution in such an approximation is, by using a geometric sum formula,

$$\begin{aligned} c_n(x, t) &= c_1(x, 0)R_n(t), \\ R_n(t) &= \exp(-\lambda_1(t - nt_{\text{per}})) \left(\frac{1 - q^{n+1}}{1 - q} \right), \\ q &= \exp(-\lambda_1 t_{\text{per}}) < 1, \\ \frac{q(1 - q^{n+1})}{1 - q} &\leq R_n(t) \leq \frac{1 - q^{n+1}}{1 - q}, \quad nt_{\text{per}} \leq t \leq (n + 1)t_{\text{per}}, \\ \liminf_{n \rightarrow \infty} R_n(t) &= \frac{q}{1 - q}, \quad \limsup_{n \rightarrow \infty} R_n(t) = \frac{1}{1 - q}. \end{aligned} \quad (15)$$

Below is a plot of $R(t)$ for $\lambda = \ln 2$, $t_{\text{per}} = 1$ for illustration.



We have computed solutions in detail with boundary conditions (a) and (b), including a number of low-lying eigenvalues in the expansion, and varying intervals between injections, for a small number of injections. We searched through l -modes up to 20 (traditionally $l = 0$ is called the s -mode, $l = 1$ the p -mode, etc; such terminology is used in chemistry); consecutive eigenvalues with the same l are labeled by i . We computed eigenvalues lower than 200 using `NSolve` in Mathematica, and selected lowest-lying ones from those.

In the lossless problem we found that the lowest-lying nonzero eigenvalues are ordered as follows:

$$\begin{aligned} \{i, l, \kappa_{l,i}\} &= \{1, 1, 2.08158\}, \{1, 2, 3.34209\}, \{1, 0, 4.49341\}, \{1, 3, 4.5141\}, \\ &\{1, 4, 5.6467\}, \{2, 1, 5.94037\}, \{1, 5, 6.75646\}, \{2, 2, 7.28993\}, \dots \end{aligned} \quad (16)$$

There is also a constant mode with zero eigenvalue for those boundary conditions. We plot consecutive snapshots of the lossless solution at times $t < 6$. Since there is no azimuthal

angle dependence, we plot c in a cross section of a plane through the origin and the z -axis; the domain is thus a circle $x^2 + y^2 < 1$. Injections are delivered at $(x, y, z) = (0, 0, 0.9)$ as δ -functions of weight one (in units of mass) centered at this point. Hence if this medicine is uniformly distributed in the unit ball, the concentration will be $\frac{3}{4\pi} \approx 0.239$. We cannot accurately plot the solution right after the injection as it has a sharp peak and, with finite number of modes included, oscillates a lot; however it can be accurately plotted after some delay after the injection, as high modes become exponentially suppressed.

In Figure 1, we plot snapshots of the lossless solution (case a) for a unit time delay between injections; in Figure 2, the time delay between injections is 2.

In Figure 3, we plot snapshots for a low-loss solution for a unit and double time delay between injections. We model the leakage by an outer layer of height $h = 0.15$, and a diffusion coefficient $d = 0.001$, with zero Dirichlet condition outside of that layer. In this case, the zero eigenvalue is not present, and the low-lying eigenvalues are arranged as follows (with the lowest one missed by Mathematica in searching for $\kappa < 200$, and spotted on a plot and by `NSolve` after a lower upper bound was tried):

$$\{i, l, \kappa_{l,i}\} = \{1, 0, 0.15153\}, \{1, 1, 2.08829\}, \{1, 2, 3.34686\}, \{2, 0, 4.49488\}, \\ \{1, 3, 4.51797\}, \{1, 4, 5.65002\}, \{2, 1, 5.94142\}, \{1, 5, 6.75937\} \dots \quad (17)$$

The volume-averaged concentration of the lowest mode of the solution right after a unit dose injection is 0.0648 (corresponding to the mass of 0.2716 which goes into that mode, of the total amount of 1 distributed between the modes). This is roughly the concentration which survives a few units of time after the injection, as this mode decays slowly while other modes are exponentially suppressed after that amount of time. If another injection is delivered before the lowest mode leaks, it will roughly double the concentration surviving after higher modes become suppressed.

We have also looked at solutions with the same h and $d = 0.01$ and $d = 0.1$. Those behaved as expected, with more leak for higher diffusion in the outer layer. We do not provide those details here.

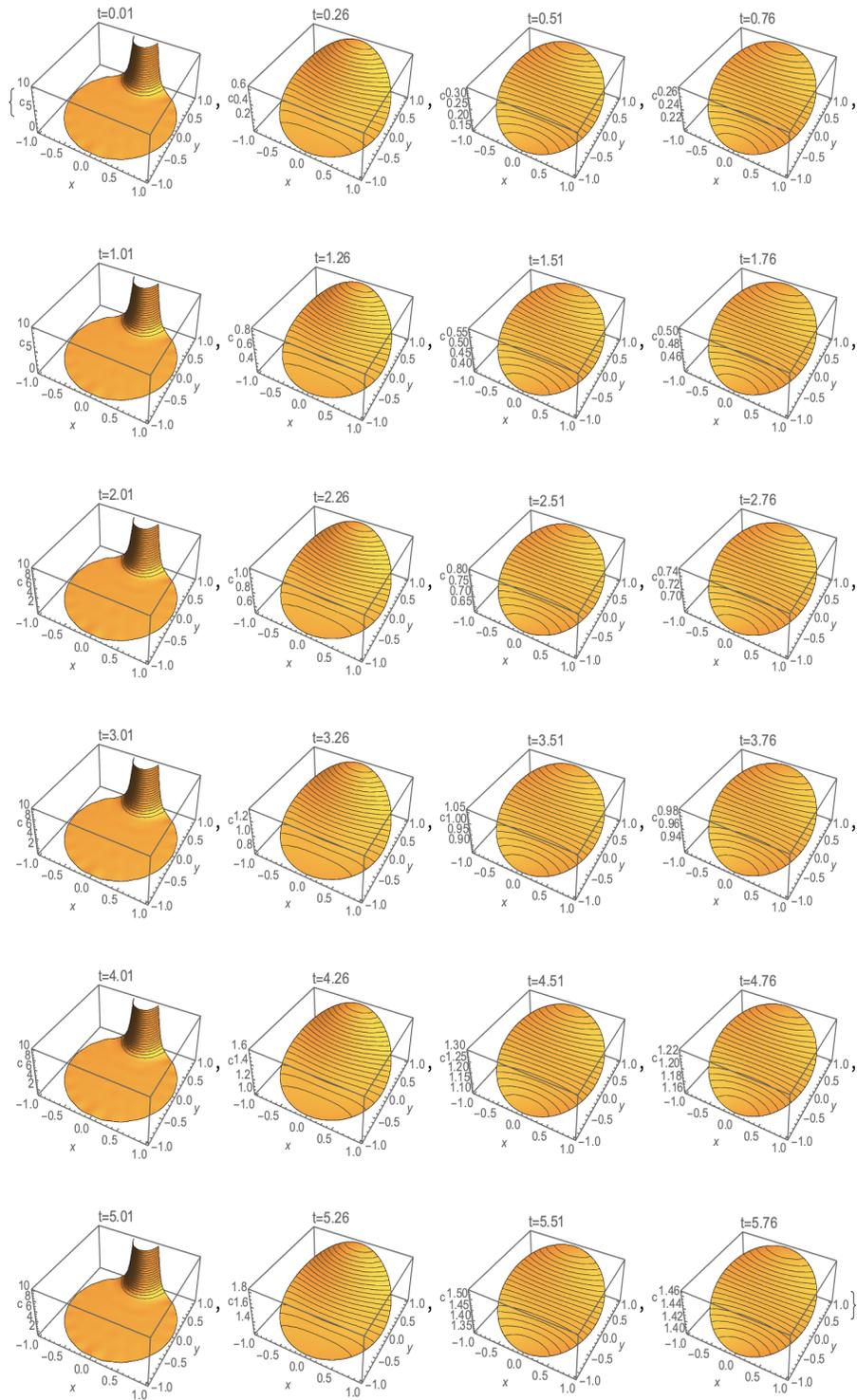


Figure 1. Insulated eyeball, unit time delay between injections.

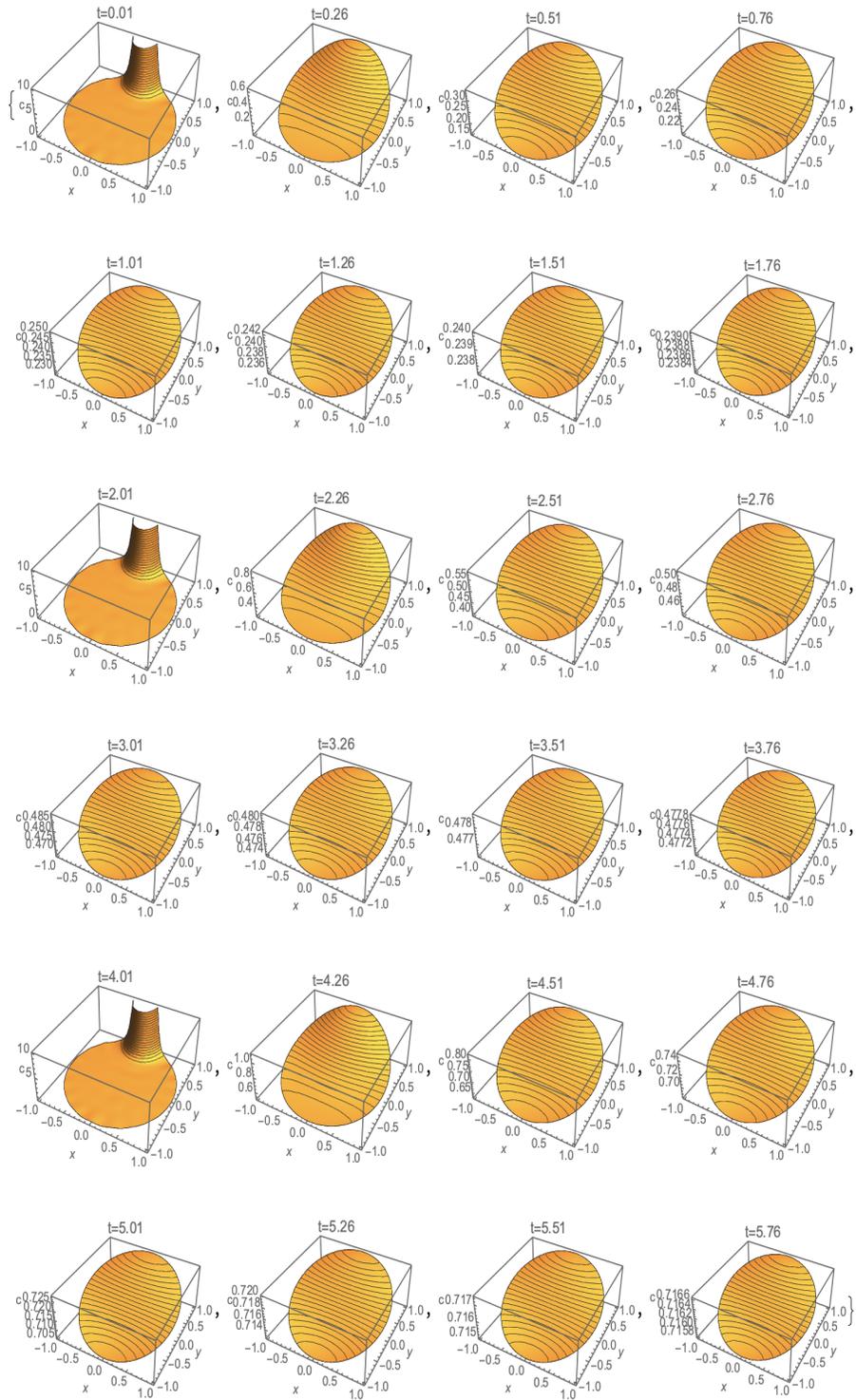


Figure 2. Insulated eyeball, double time delay between injections.

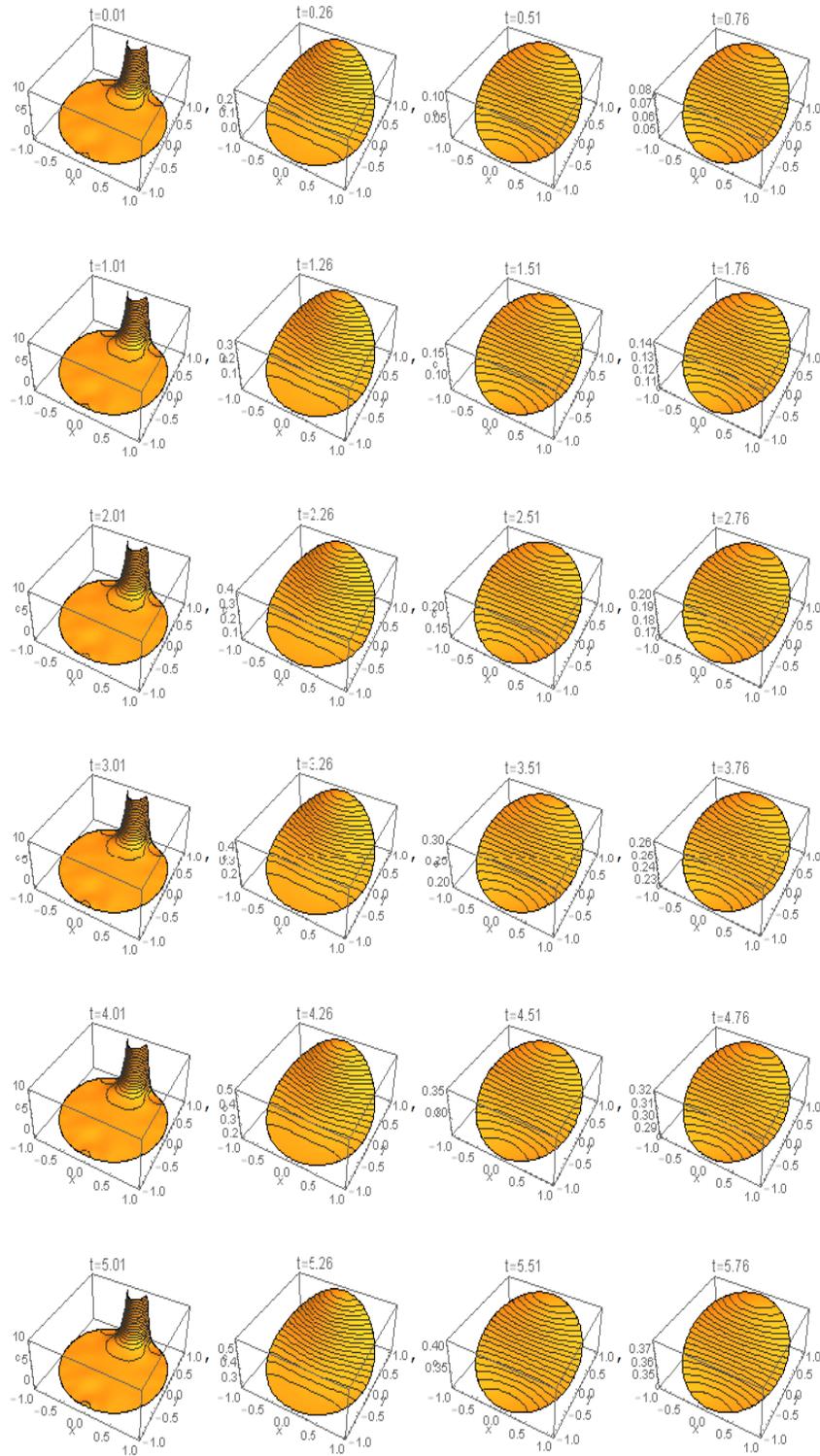


Figure 3. Low-leaky eyeball, $d = 0.001, h = 0.15, t_i = 1.0 * i$.

5 Conclusions and possible further research

We investigated solutions of the diffusion equation in a ball and with boundary conditions modeling different rates of leakage of medicine out of the eyeball. We used eigenfunction expansions, which give good insight at sufficiently large times, but require large number of eigenmodes to resolve short-time delays when the solution has a sharp peak at the injection spot.

Other interesting areas of exploration include:

1. looking into performance of other methods (*e.g.*, adapting the Fokas method, direct numerics, etc)
2. extending the model to a moving boundary problem describing retina growth (though the mechanism of such growth is not clear to us).
3. linking this part of the work with other modeling approaches and data analysis performed on this problem. However due to time constraints, we kept our part as simple as possible, without complicating details included.

Methods used here may have applications elsewhere: *e.g.*, to describe diffusion in small particles, in chemistry problems based on eigenvalues, etc.

Stochastic Modeling of Central Retinal Thickness

Melike Sirlanci
California Institute of Technology

July 31, 2019

1 Model Development

The graphics of central retinal thickness (CRT) measurements for each patient shows some similar behaviour: exponential decay with oscillations. The decay is assumed to be due to a combination of absorption of and tolerance to the injected drug; however the cause of the oscillations remains unknown. So, to model the CRT, we use the Ornstein-Uhlenbeck model whose drift term will be modeling the exponential decay and the diffusion term will capture the amplitude of the oscillations. Note that a similar approach is used in [1] to model glucose-insulin dynamics in humans.

Consider the following model,

$$\begin{aligned} \dot{r}(t) &= -\gamma(t)(r(t) - r_a) + \sqrt{2\gamma(t)\sigma^2}\dot{W}(t), \\ r(0) &= r_0, \end{aligned} \tag{1}$$

where $\gamma(t)$ is the decay function that represents the rate of decay in retinal thickness $r(t)$ and will be defined in detail below, r_a is the value to which $r(t)$ decays, and σ is the quantity representing the magnitude of the fluctuations with $W(t)$ being the Brownian motion.

The stochastic differential equation in (1) is analytically solvable. Because of the properties of the Brownian motion, it turns out that the solution is a normally distributed random variable. Since we will use this model on a discretely collected dataset, we first need to obtain the discrete version of the solution. For this purpose, let $\{t_k\}_{k=0}^N$ represent the union of all times at which retinal thickness is measured and/or drug injected. In this setting, we will assume that the decay rate over an interval $[t_k, t_{k+1}]$, $k = 0, 1, \dots, N - 1$, will be modeled as proportional to the dose amount, D_k , at time t_k . So, if there was no injection at time t_k , then $D_k = 0$. So, we have the decay function $\gamma(t)$ as

$$\gamma(t) = \sum_{k=0}^{N-1} \gamma D_k \chi_{[t_k, t_{k+1})}(t), \tag{2}$$

where γ is a proportionality constant.

Integrating (1) over $[t_k, t_{k+1})$ and representing $r_k := r(t_k)$, $\gamma_k := \gamma D_k$, and $h_k := t_{k+1} - t_k$, $k = 0, 1, \dots, N - 1$, we state the solution as

$$r_{k+1} = r_a + e^{-\gamma_k h_k} (r_k - r_a) + \sigma_k \xi_k, \quad k = 0, 1, \dots, N - 1, \quad (3)$$

where $\sigma_k := \sqrt{\sigma^2(1 - e^{-2\gamma_k h_k})}$, $k = 0, 1, \dots, N - 1$. Also, ξ_k , $k = 0, 1, \dots, N - 1$ are independent and identically distributed standard normal random variables. Using the recurrence relation in (3), we can write each r_{k+1} as follows

$$r_{k+1} = r_a + e^{-\sum_{i=0}^k \gamma_i h_i} (r_0 - r_a) + \sum_{i=0}^k e^{-\sum_{j=i+1}^k \gamma_j h_j} \sigma_i \xi_i, \quad k = 0, 1, \dots, N - 1. \quad (4)$$

Now, letting $\mathbf{r} := [r_k]_{k=0}^N$, we have $\mathbf{r} \sim N(\mathbf{m}, \mathbf{C})$, where \mathbf{m} and \mathbf{C} are the mean vector and covariance matrix that are functions of the parameters γ, r_a, σ , and could be computed from (4).

2 Parameter Estimation

With this modeling approach, we have stated the retinal thickness at time points $\{t_k\}_{k=0}^N$ as multivariate normal random variables. In reality, any sort of measurement is made due to some measurement noise. To account for this fact, we state the following observation equation

$$y_k = r_k + \varepsilon_k, \quad k = 0, 1, \dots, N. \quad (5)$$

where $\varepsilon_k \sim N(0, \delta^2)$, $k = 0, 1, \dots, N$ are independent random variables representing the measurement noise. Therefore, letting $\mathbf{y} := [y_k]_{k=0}^N$, we obtain $\mathbf{y} \sim N(\mathbf{m}, \mathbf{\Sigma})$, where $\mathbf{\Sigma} = \mathbf{C} + \mathbf{E}$ and $\mathbf{E} = \delta I$ with I is the $(N + 1) \times (N + 1)$ identity matrix.

To state the parameter estimation problem, we will follow a Bayesian approach. Let \mathbf{y} represent the observation and let θ represent the model parameters γ, r_a, σ . Then, using Baye' theorem, we can state

$$\mathbb{P}(\theta|\mathbf{y}) = \frac{\mathbb{P}(\mathbf{y}|\theta)\mathbb{P}(\theta)}{\mathbb{P}(\mathbf{y})}. \quad (6)$$

Some possible approaches at this point include using MCMC to sample from the posterior distribution, etc. However, we will start with an optimization approach: that is, we will find the parameters that maximize the posterior distribution, $\mathbb{P}(\theta|\mathbf{y})$. The work above provides us the distribution for $\mathbb{P}(\mathbf{y}|\theta)$, which is the normal distribution with the mean and covariance as stated above. On the other hand, to stay as general as possible, for the prior distribution, $\mathbb{P}(\theta)$, we will assume initially that the parameters are independent and uniformly distributed over some reasonable regions. So, we set

$$\mathbb{P}(\theta) := \frac{1}{|\Theta|} \chi_{\Theta}(\theta),$$

where Θ is the set that the parameters θ take values from.

These reasonable regions can be obtained by simulating the retinal thickness with various values of the parameters and deciding the range that produces realistic simulations. Therefore, since the denominator, $\mathbb{P}(\mathbf{y})$, is independent of θ and our goal is to maximize the posterior distribution with respect to θ , we state the parameter estimation problem as follows:

$$\begin{aligned} \theta^* &= \arg \max_{\theta} \mathbb{P}(\mathbf{y}|\theta)\mathbb{P}(\theta) \\ &= \arg \max_{\theta \in \Theta} \frac{1}{(2\pi)^{-\frac{(N+1)}{2}} \det(\Sigma)^{-\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{y}-\mathbf{m})^T \Sigma^{-1}(\mathbf{y}-\mathbf{m})} \\ &= \arg \min_{\theta \in \Theta} \frac{1}{2}(\mathbf{y}-\mathbf{m})^T \Sigma^{-1}(\mathbf{y}-\mathbf{m}) + \ln(\det(\Sigma)), \end{aligned} \quad (7)$$

where \mathbf{m} and Σ are functions of the unknown model parameters θ .

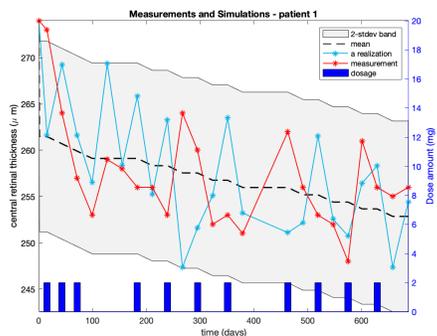
3 Numerical Results

In this section, we present numerical results using the stochastic model and parameter estimation scheme as described above. Since each patient reacts differently, we perform the parameter estimation scheme specifically for each patient, and use the estimated parameters to evaluate the efficacy of our approach.

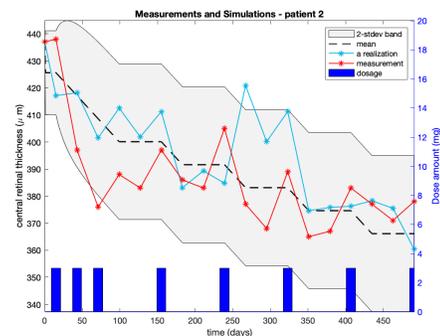
The parameters that we want to estimate based on patients' CRT measurement and dose amount of drug injections are γ , r_a , and σ . We will set the standard deviation of the measurement noise, δ to be some constant value that exhibits the magnitude of measurement noise in reality. Here, letting \hat{m} to be the average of all CRT for a particular patient, we set $\delta = 0.02 * \hat{m}$ and remain with the four parameters above to be estimated.

By following the parameter estimation scheme described above, we estimate the parameters and get simulations for the CRT with the estimated parameters and compare our simulation results with the actual measurements.

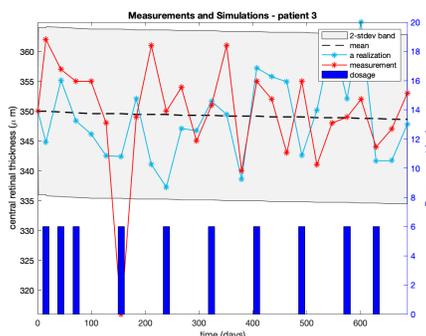
In each sub-figure of Figure 1, the dashed black curve represents the mean of CRT obtained with simulating the stochastic model with the estimated parameters. In each of them, the gray region is the 2-stdev band around the mean, and the green curve is one realization of CRT, once again obtained with the estimated parameters, since with this approach we consider CRT as a random process. Also, the blue bars show the time and amount of the dose of injections. In each figure, we see that the dashed curve captures the essence of the behaviour of the actual measurements, as being the mean of the random process. In addition, in each figure, the 2-stdev region captures almost all of the measurements. This feature shows the potential of this approach for predicting future CRT values based on dose amount. Finally, we see that the green curve, which represents an arbitrary realization of the simulated random process, behaves in a similar way as the measurements. Moreover, we can achieve the same accuracy with a smaller dataset as seen on the lower right panel of Figure 1, which is obtained for patient 4.



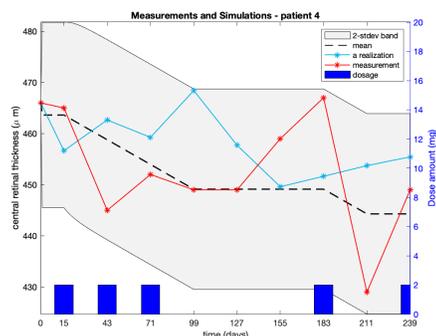
(a) Patient 1



(b) Patient 2



(c) Patient 3



(d) Patient 4

Figure 1: Simulated CRT with optimally estimated parameters for different patients

Finally, there are also patients who are non responsive to treatment and their CRT values can even increase. This model can also capture this type of behavior of CRT. In the two panels of Figure 2, we see such situations and the simulations with the estimated parameters in each case can capture this behaviour.

4 Discussion and Future Directions

1. **Limitations:** Figure 1 shows cases in which this model works well. However, this was unfortunately not the case for all patients. One reason for this is the size of the dataset. Even though it is known that patients may develop resistance to the drug over the time course of treatment and this slows down the decay rate of CRT, we had to assume the decay rate is constant over the treatment time due to computational reasons. This causes missing changes in the decay rate. Figure (3) shows a case in which the actual decay rate is high in the beginning; however it gets much smaller very quickly. Thus, in such cases, because of assuming the decay rate to be constant, we obtain an optimal parameter that is between the small and high values in order to

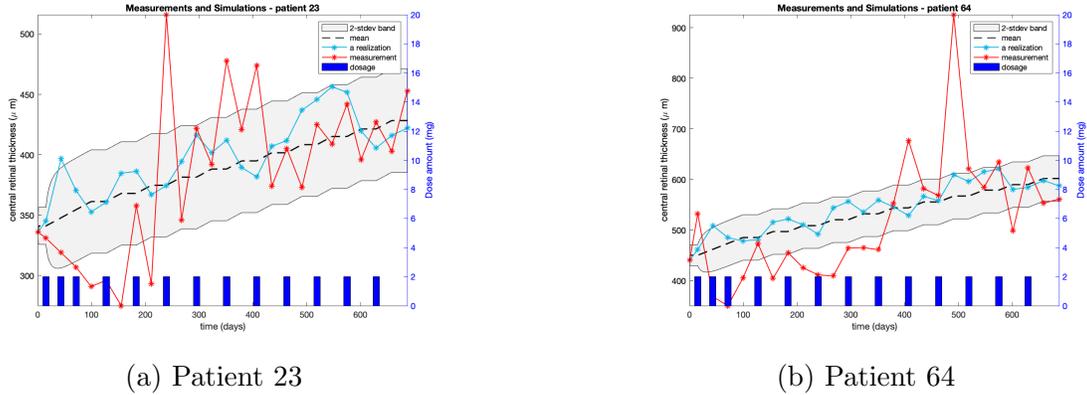


Figure 2: Simulated CRT with optimally estimated parameters for non-responsive patients

compensate for both cases (small and high decay rate during the treatment time). This situation is reflected in the simulations (see Figure 3). Even though the measurement and simulation converges to somewhat close values, we are not able to capture the exact behavior of the measurement. One solution approach for this could be to use a slightly more general $\gamma(t)$ function than the one in (2), such as

$$\gamma(t) = \sum_{k=0}^{N-1} \gamma_k D_k \chi_{[t_k, t_{k+1})}(t).$$

In this case, we may be able to capture the changes in the decay rate. Note smaller γ_k values means smaller decay rate while larger values mean higher decay rate. We were not able to use the $\gamma(t)$ function in (1) because of having only two measurements between two injections in most cases. If it is possible to obtain more measurements besides estimating the decay rate more accurately, we can also develop some understanding for patients' resistance to the drug over the treatment process.

Moreover, in this model, CRT is driven by the drug injections. It basically assumes that whenever there is an injection, there will be some decrease in CRT. However, we are not sure if this is in accordance with reality. Depending on suggestions from physicians, we may need to incorporate this knowledge into the model.

Also, currently measurement noise is set to be 2% of the average of all measurements of a patient. This decision should be revisited and this amount should be decided in consultation with the physicians.

2. **Other Possible Approaches for Parameter Estimation:** In this report, all the parameter estimation results are obtained with an optimization approach by maximizing the posterior distribution of the parameters. Another approach could be to use Markov Chain Monte Carlo techniques to sample from the posterior distribution. Hence, we can obtain the conditional (on data) distribution of the parameters for un-

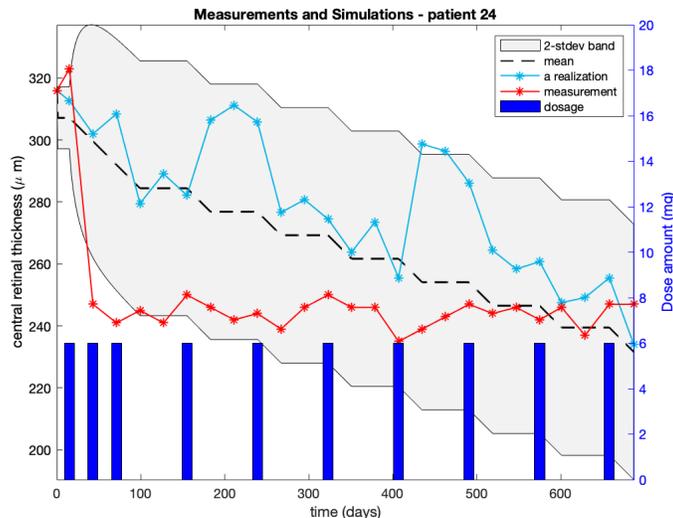


Figure 3: Simulation of CRT

certainty quantification, which could provide valuable information on how likely are the outcomes obtained with the model.

3. **Classifying Patients with respect to their Estimated Parameters:** In this approach, as stated above, the parameter r_a represents the value CRT decays to, and decay in CRT is positively correlated with the achievement of the treatment. Therefore, the patients for which $r_a > r_0$, where r_0 is the CRT value at the initial time, can be classified to be non-respondent to treatment (see Figure 2). On the other hand, having $r_a < r_0$ does not always mean that the patient is respondent. For this, r_a should be significantly smaller than r_0 . Therefore, depending on the insight of expert physicians in this area, we can decide a threshold value, ϵ and classify the patients for which $r_a - r_0 < \epsilon$ to be respondent and the ones with $r_a - r_0 \geq \epsilon$ to be non-respondent.
4. **Future Directions:** Besides improving the current limitations of the model there are also some other aspects of the problem we can work on by using this model. One possibility is to use the patient-specific model, which is the model formed by the estimated parameters for each patient, to forecast future CRT levels based on the dose amounts. Another interesting problem might be to estimate the required dose amount for a desired CRT level. This problem is a typical control problem in which we need to estimate the input (dose amount) of the system to achieve desired output (CRT). So, we will need to state and solve the control problem by using a stochastic differential equation model as in (1).

References

- [1] M. Sirlanci, M. Levine, A. Stuart, and D. Albers. A simple modeling framework for short-time prediction in the human glucose-insulin system. In preparation.

Statistical Analysis

Sunil Dhar, New Jersey Institute of Technology

Introduction

Two clinical measurements, Best-Corrected Visual Acuity (BCVA) measurements and Central Sub-Field retinal thickness (CSFT), are used as biomarkers to evaluate disease progression of Age-related Macular Degeneration (AMD) and measure the effect of treatment variables. These biomarker variables are weakly correlated. Therefore, for the sake of developing simple and easily interpretable models that answer clinical and pharmaceutical questions, these response variables are separately analyzed. Models like these help to predict the patterns of the response variables' oscillations and their amplitude over time. Also, these models answer how much of the variability in the response variables can be explained by significant covariates. Statistical Analysis Software (SAS) was used on all patient data simultaneously to arrive at these models and also confirm them based on validation data.

The different covariates that were scrutinized in this study were as follows. The covariate "time of treatment" is defined as days starting at Day "-14" to the "end of treatment day." Patient ID (identification) is a unique identification number associated with each patient. Treatment indicator variables specify if Dose A was used at 3mg or 6mg, represented by treatA1 and treatA2, respectively. For example, treatA1 = 0 and treatA2 = 0 represents Dose B was being administered. The variable "dose administered" is defined as a binary, indicating "yes" when the day's dose is administered to the patient and indicating "no", otherwise. The variable "visit" is

defined as the visit number of the patient and perfectly correlates with the time of treatment. Therefore, time of treatment is used to develop the models.

Model Description:

The models used to fit the data are a class of Generalized Linear Models (GLM) with a natural log link function. The response variable is either the square-root transformation of BCVA measurements or the natural log transformation of CSFT. These transformations are due to the fact that BCVA distribution is left-skewed and the CSFT data is positively skewed. These transformations help make the response variables bell-shaped. The response variable is denoted by $y_i(t)$, where t represents time in days and i denotes a fixed patient's ID ($i = 1, \dots, n$). Also, \mathbf{y}_i represents the column vector over time t for the i^{th} patient ID. Furthermore, $\mathbf{y}_i, i = 1, \dots, n$ are taken to be mutually independent and multivariate normal with mean $E(\mathbf{y}_i)$ and variance-covariance Σ . The data represents patient response to treatment. Here, the patient's previous reading would have an impact on their current reading, so an Autoregressive (1) structure is assumed for Σ . A repeated measure design, repeated on the time variable, is implemented in a GLM framework. The sample size n in the original data was 103 patients. All parameters, including the coefficients of the linear regression part in the GLM, are derived based on the maximum likelihood method. Here, the conditional expectation of the response variable is described by:

$$E[y_i(t)] = f \left[\begin{array}{l} \beta_0 + \sum_{i=1}^n \beta_{1i} I_{\{\text{ID}=i\}}, t, X_3(t), X_4(t), \sum_{i=1}^n \beta_{5i} X_3(t) I_{\{\text{ID}=i\}}, \sum_{i=1}^n \beta_{6i} t I_{\{\text{ID}=i\}}, \\ \sum_{i=1}^n \beta_{7i} I_{\{\text{ID}=i\}} X_3(t) \end{array} \right]$$

$$= \exp \left\{ \begin{array}{l} \beta_0 + \sum_{i=1}^n \beta_{1i} I_{\{\text{ID}=i\}} + \beta_2 t + \beta_3 X_3(t) + \beta_4 X_4(t) + \\ \sum_{i=1}^n \beta_{5i} X_3(t) I_{\{\text{ID}=i\}} + \sum_{i=1}^n \beta_{6i} t I_{\{\text{ID}=i\}} + \sum_{i=1}^n \beta_{7i} I_{\{\text{ID}=i\}} X_3(t) t \end{array} \right\}.$$

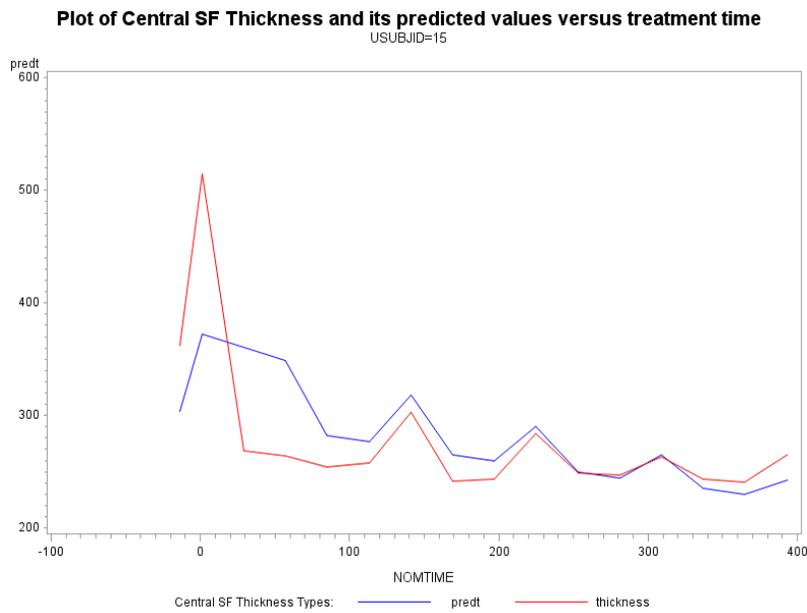
The first argument of the function f represents the intercept plus the effect of Patient ID's. Next, $X_3(t)$ is 'no' if a dose is not administered at time t or 'yes' if a dose is administered at time t . Further, $X_4(t)$ is $X_3(t)$ multiplied by t , describing the interaction effect between dose and time t . The fifth argument of the function f represents the interaction effect of the dose administered and patient ID. The sixth argument of the function f represents the interaction effect of patient ID with time t . The seventh argument of the function f represents the interaction effect among three variables: dose administered, patient ID, and time t . Note that $I_A = 1$ if event A is true and 0, otherwise. Justification for the validity of the SAS based models is justified by convergence at all levels of the algorithms.

Model Validation

The models derived in the Model Description section are based on an original sample of 103 patients and the models derived in this section are based on an additional data of 26 new patients for validation. The models were redeveloped by taking all the important covariates into consideration as described in the Introduction section. The resulting models are similar to the ones described in the Model Description section. The regression term is described by the full model in terms of the variables *time of treatment*, *time of dose*, and the *patient ID*. The same transformations of the biomarkers were applied as before to define the response variables. The model fit criterion used was QIC and QICu. QICu is smaller than QIC because the former accounts for the penalization for the number of parameters used in the model. A smaller criterion value indicates a better fitting model (Pan, 2001). The sample size of 103 patients is larger than

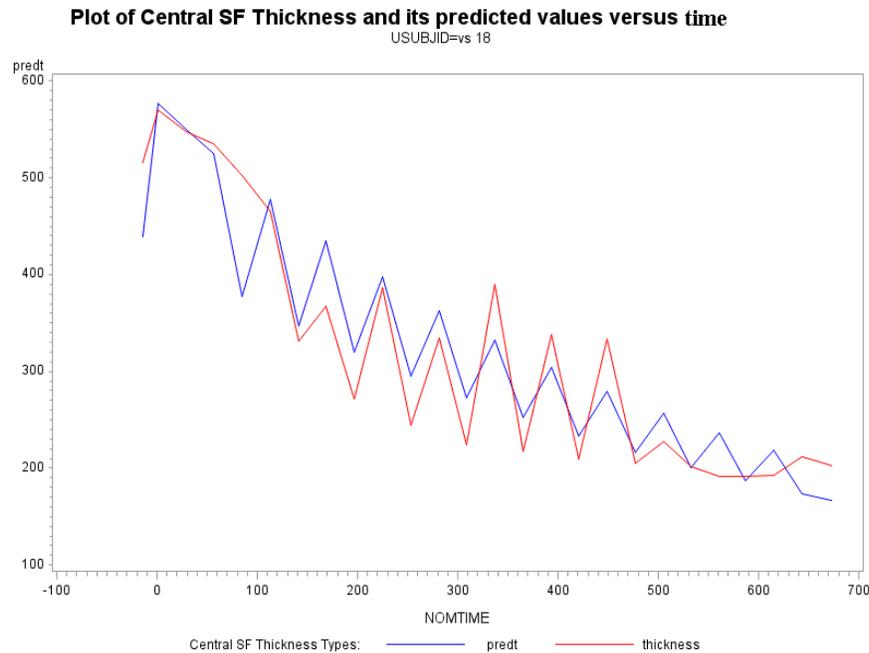
that of 26; however, the corresponding fit criterion for the model based on smaller sample size turns out to be smaller. This indicates a better fit when the sample size is smaller as is depicted in the graphs below. In the case of CSFT, the fit for the larger sample size is also very good, capturing the crest and fall of the curves fairly accurately. A typical fit (for Patient 15) is shown in Figure 1.

Figure 1.



The following Figure 2 is the fit for a typical patient from CSFT data of Patient 18.

Figure 2.



The models are built on the entire data. Using the model fit criterion is a better way of comparing these models. In the case of CSFT, $QIC = 2039$ and $QICu = 2861$ for the original data, and $QIC = 506$ and $QICu = 714$ for the validation data. These values show that the validation models have a better fit overall. Not only are the model variables and interaction terms identical for the two GLM models, but the point estimates for the corresponding parameter of the covariates are also similar (see Tables 1 and 2).

Table 1. CFST, for sample size 103.

Parameter	Estimate	Standard Error	95% Limits	Confidence	Z	Pr > Z
Intercept	1.7599	0.0000	1.7599	1.7599	1.1E10	<.0001
NOMTIME	-0.0001	0.0000	-0.0001	-0.0001	-2.2E8	<.0001
DOSE no	-0.0055	0.0000	-0.0055	-0.0055	-3.09E6	<.0001
NOMTIME*DOSE no	-0.0000	0.0000	-0.0000	-0.0000	-	<.0001
					995460	

Table 2. CFST, for sample size 26.

Parameter		Estimate	Standard Error	95% Limits	Confidence	Z	Pr > Z
Intercept		1.7492	0.0000	1.7492	1.7492	1.315E8	<.0001
NOMTIME		-0.0001	0.0000	-0.0001	-0.0001	-2.68E6	<.0001
DOSE	no	-0.0034	0.0000	-0.0034	-0.0034	-1.55E6	<.0001
NOMTIME*DOSE	no	0.0000	0.0000	0.0000	0.0000	895885	<.0001

Since the patient profiles vary, the parameter coefficients relating to covariates with Patient ID in them in the two models have not been reported here for the sake of brevity. BCVA reports similar results. However, in the context of 103-patient data, although the BCVA model captures the trend in the data when compared to the CFST model, it does not as accurately capture the oscillations. However, the BCVA model does accurately capture the oscillations for the corresponding validation data.

Conclusions

The models for both CFST and BVCA show that the only variables that play a role in explaining the GLM model are the main effect variables *Patient ID*, *time of treatment* in days, *time of dose*, and their two-way and three-way interactions. This automatically excludes variables like the type of medicine and patient dose levels in milligrams. Interestingly, validation

data arrived at the exact same models, confirming findings consistent with other research in this field.

References

Pan W. (2001) Akaike's information criterion in generalized estimating equations. *Biometrics*. 57, 120-125.