Parameter Estimation for Tumor-Immune ODE System *

L.G. de Pillis and A.E. Radunskaya

August 22, 2002

* This work was supported in part by a grant from the W.M. Keck Foundation

PARAMETER ESTIMATION

Overview

1. Gathering Data
2. Fitting Curves to Data
3. Calculating Curves
4. Function Minimization
5. Root Finding
6. Example with MATLAB
7. Estimating s and d using measured steady-state values.
8. Parameter estimation without an explicit solution.
9. Demonstration of the procedure, and results.

Parameter Estimation

Gathering Data

- Parameters in the model: $s, p, g, m, d, a, b,$ and n.
- To find values for these eight parameters, compare the output from the model equations with curves generated from experimental data.

Step One: Gather appropriate data.

On the next slide is a graph showing the growth of tumor cells in a control group of mice with no bone marrow, and hence no immune response. We can use these data to estimate tumor growth parameters

______________________________(1).

Parameter Estimation

Notes for Gathering Data slide:

Answers:

(1) a and b.

Notes: If time allows, a discussion on what parameters might be measured experimentally should precede this slide. How might the different growth terms and competition terms be measured experimentally? Is it actually possible to isolate the effects of the different cell types? What assumptions that were made in the construction of the model should be questioned?

Parameter Estimation

Mouse Data

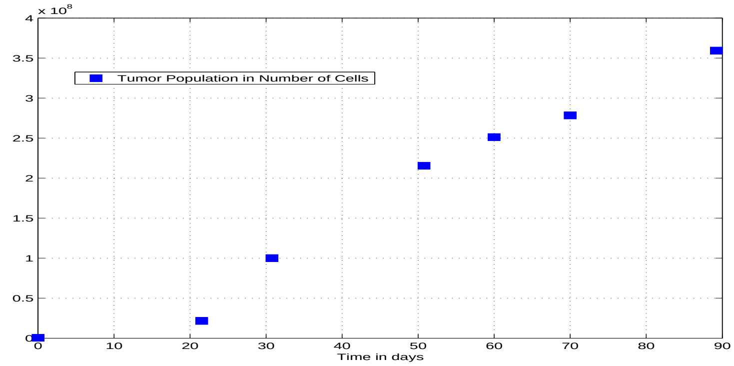


Figure 1: *Tumor cell growth in mice with bone marrow destroyed*

Parameter Estimation

Curve through Mouse Data

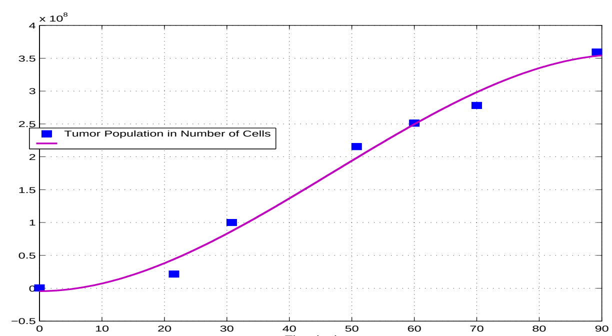


Figure 2: *What curve best fits the data?*

Parameter Estimation

Notes for Mouse Data slide:

Note: The previous slide shows the cubic polynomial which best fits the data. (*This curve was generated using MATLAB's "Basic Fitting" tool in the pull-down menu of the Figure window*). The students should discuss what type of curve might fit the data, with justification for their answers. Some students may recognize the data as having the S-shaped form characteristic of logistic growth. (In this case, you may assure the students that the logistic differential equation is solved explicity a later in this module.)

4-1

Parameter Estimation

Fitting the Curve

Step Two: **Fitting the Curve to the Data**

Main Idea: Minimize the total distance from the model curve to the data points.

Collected Data: $d_1, ..., d_n$ at times $t_1, ..., t_n$

In our example these are the values of

______________________________ (1)

Model Solution: $\vec{x}(t)$ or $\{\vec{x}(t_i)\}, i = 1, \ldots n.$

Goal: Minimize

$$D = \sum_{i=1}^{n} (x(t_i) - d_i)^2$$

(This is called a"least squares fit.")

Parameter Estimation

Notes for Fitting the Curve slide:

Answers:

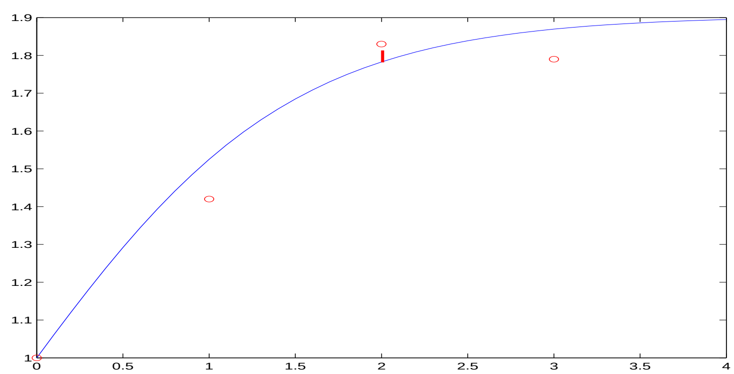
(1) the number of tumor cells at the 7 different times.

Note: There are two points to be made here:

- What is an appropriate measure of “goodness of fit”? The students should come up with a few different ideas here. For example, should we minimize the *average* distance, should data points later in time be weighted more heavily, and so forth. (*The next slide shows graphically what is meant by the “distance” to the data*).
- How can we find the solution to the model equations? What equation are we trying to solve here? Since there are no immune cells, the model reduces to a one-dimensional logistic equation, which *can* be solved explictly. (*This is worked out in the following notes page*). Numerical methods are discussed in the module on Numerical Methods.

Parameter Estimation

Distance to Data



Parameter Estimation

Finding the Curve

The $x(t_i)$'s are determined:

- by solving the D.E. analytically (in which case we have $x(t)$ for all values of t) or
- numerically (which gives $x(t)$ only for the t-values we specify.)

The minimization can also be done analytically (in very simple cases) by

____(1)

or numerically.

7

Parameter Estimation

Notes for Finding the Curve slide:

Answers:

(1) differentiating the function D. *See the next slide*.

Note:

1. The differential equation we are solving here is:

$$x'(t) = ax(1-bx)$$

This equation may already be familiar to some students, and can be solved by separating variables and using partial fractions:

$$\begin{aligned} \frac{1}{x(1-bx)}\,dx &= a\,dt \Rightarrow \int \frac{1}{x(1-bx)}\,dx = \int a\,dt \\ \Rightarrow \int \frac{1}{x} + \frac{b}{1-bx}\,dx &= \ln\left(\frac{x}{1-bx}\right) = at + C \end{aligned}$$

Solving for x in terms of t, and writing the constant of integration in terms of the initial value of x gives:

$$\begin{aligned}
\frac{x}{1-bx} = Ce^{at} \quad &\Rightarrow \quad x = Ce^{at} - bCe^{at}x \\
&\Rightarrow \quad x(1+bCe^{at}) = Ce^{at} \\
&\Rightarrow \quad x = \frac{Ce^{at}}{1+bCe^{at}} = \frac{1}{Ce^{-at}+b}
\end{aligned}$$

Letting $x(0) = x_0$:

$$x_0 = \frac{1}{C+b} \Rightarrow C = \frac{1}{x_0} + b$$

2. You can also use a weighted least squares fit, or any other norm, as your criterion. See [MT73, Section 10.2] for more details.

Parameter Estimation

Function Minimization

- Analytically we find a minimum by *differentiating* D with respect to the *parameters*, finding the zeroes, and determining which zeroes correspond to minima of D.

  For example, suppose the model solution were: $x(t) = at + b$ and the data points were: $\{t_i\} = \{1, 3, 5\}, \{x_i\} = \{2, 5.1, 7.8\}$.
  The parameters which give the least squares distance are: ________ (1) and ________ (2).

- There are many numerical methods for function minimization. Most mathematical software packages have built-in routines to do this. The numerical method is often specific to whether the function is linear or non-linear, and whether there are additional constraints.

NB: A search for “minimize” usually produces available routines.

Parameter Estimation

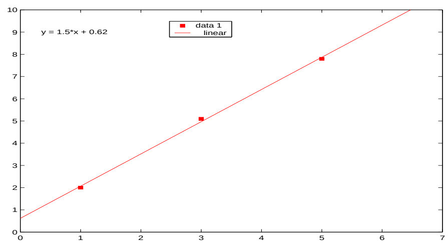
Notes for Function Mimimization slide:

Answers:

(1) a=1.45

(2) b=.6167

Note: In MATLAB, once data points are plotted, you can click on the Tools pull-down menu button in the Figure window, then choose “Basic Fitting”, and then “quadratic” in the pop-up menu. Here is the answer given by MATLAB’s Basic Fitting with a quadratic fitting routine:



The formula used is:

$$D = \sum_{i=1}^{n}(at_i + b - x_i)^2 \Rightarrow \frac{\partial D}{\partial a} = 2\left(\sum_{i=1}^{n} at_i^2 + \sum_{i=1}^{n}(bt_i - x_i t_i)\right)$$

Setting this equal to zero and solving for a gives:

$$a = \frac{\sum(x_i t_i) - b\sum t_i}{\sum t_i^2}$$

Computing $\frac{\partial D}{\partial b}$ and setting the result equal to zero gives

$$a\sum_{i=1}^{n} t_i = \sum_{i=1}^{n}(x_i - b)$$

Simultaneously requiring both partial derivatives to be zero by substitution gives, for example (substituting the value for a from the second equation into the first):

$$b = \frac{\sum t_i \sum x_i t_i - \sum x_i \sum(t_i^2)}{(\sum t_i)^2 - n\sum(t_i^2)}$$

This formula, with $n = 3$ and $\vec{x} = [2, 5.1, 7.8], \vec{t} = [1, 3, 5]$, gives the desired values: $a = 1.45$, and $b = .6167$. In this example, we obtain only one possible value for the minimizing parameters. We should check that we have indeed found a minimum by, for example, checking the second-order condition, i.e. by computing the Hessian:

$$H(a,b) = \begin{bmatrix} \frac{\partial^2 D}{\partial a^2} & \frac{\partial^2 D}{\partial a \partial b} \\ \frac{\partial^2 D}{\partial b \partial a} & \frac{\partial^2 D}{\partial b^2} \end{bmatrix} = \begin{bmatrix} 2(\sum t_i^2) & 2\sum t_i \\ 2\sum t_i & 2n \end{bmatrix}$$

The determinant of this matrix is $4(n \sum t_i^2 - (\sum t_i)^2)$ which is strictly positive (this can be shown by induction, for example). Furthermore, both diagonal entries are positive. So we see that we have indeed found a minimum.

It is rare that one can analytically find a minimum of the least squares problem. As an illustrative, but perhaps painful, exercise, the students might try to write out the function D where $\vec{x}(t)$ is the solution to the logistic differential equation, take its derivatives with respect to the parameters a and b, and set them equal to zero. As in the simpler linear case, they will get two equations which must be solved simultaneously, and which will contain the data, $\{t_i\}$ and $\{x_i\}$. Here is an outline of the procedure:

$$D = \sum_{i=1}^{n} (x(t_i) - x_i)^2$$

Taking the derivative of D with respect to a, for example, gives:

$$\begin{aligned} \frac{\partial D}{\partial a} &= 2\sum_{i=1}^{n} \left(\frac{1}{(1/x_0 - b)e^{-at_i} + b} - x_i \right) \frac{\partial x(t_i)}{\partial a} \\ &= 2\sum_{i=1}^{n} \left(\frac{1}{(1/x_0 - b)e^{-at_i} + b} - x_i \right) \left(\frac{t_i(1/x_0 - b)e^{-at_i}}{[(1/x_0 - b)e^{-at_i} + b]^2} \right) \end{aligned}$$

The equation for $\frac{\partial D}{\partial b}$ is obtained in a similar manner, and is equally complicated. By this point, however, the students should be convinced that numerical methods are at least preferable, and, in most cases, essential.

Parameter Estimation

Root Finding

Usually parameter estimation requires ____________$_{(1)}$ a function, which in turn requires finding the ____________$_{(2)}$ of its ____________$_{(3)}$.

This can be difficult to do by hand!

Fortunately, there are ____________$_{(4)}$ algorithms for finding the zeroes of functions. For example:

NEWTON'S METHOD: To find a zero of the function f, iterate the equation

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

starting at some initial guess for the zero, x_0.

9

Parameter Estimation

Notes for Root Finding slide:

Answers:

(1) minimizing
(2) zeroes
(3) derivative(s)
(4) numerical

Parameter Estimation

Root Finding Example

Applying this algorithm to the function $f(x) = 4x\sin(x)(3-2x)$ with two different starting values gives the following sequences:

$$1.2, 1.6737, 1.5137, 1.5001, 1.5000; \quad \text{and} \quad 3, 3.1662, 3.1421, 3.1416, \ldots$$

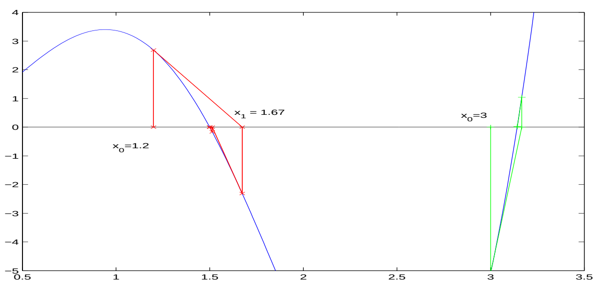


Figure 3: Note the importance of the initial guess!

Parameter Estimation

Notes for Root Finding Example slide:

Notes: Newton's method may be discussed at greater length here, or the previous two slides may be omitted altogether. Alternatively, or in addition, an exercise which uses Newton's method or other minimizing routines could be assigned for in-class or at-home work.

A root-finding demo using Newton's method can be found in the MATLAB appendix.
MATLAB demo code: See *ParDemo1* scripts.

Parameter Estimation

MATLAB Curve Fitting: Solve ODE

As in Newton's method, most numerical minimizing routines require an initial "guess" - *The value of this initial guess is usually very important.*

In this example, we use MATLAB's "fminsearch" routine to fit the tumor growth data to the function.

Step A: Find an explicit formula for $T(t)$ by solving the appropriate differential equation. Use the first point in the data set as your initial condition. The initial value problem we are solving is (remember, we are assuming that there are no immune cells): ______________________________ (1)

The solution is

$$T(t) = \frac{1}{Ce^{at} + b} \quad \text{where} \quad C = \frac{1}{T_0} - b$$

(*You should have gotten this!*).

Parameter Estimation

Notes for MATLAB Curve Fitting Solve ODE slide:

Answers:

(1)

$$\frac{dT}{dt} = aT(1 - bT), \quad T_0 = T(0)$$

Notes: This is the logistic equation which can be solved by partial fractions. See the notes after slide 7.

Parameter Estimation

MATLAB Curve Fitting: Choose a Metric

Step B: Write the function to be minimized as a MATLAB M-file. This function should return the sum of the squares of the distances of the solution to the data:

$$D(a,b) = \sum_{i=1}^{n} (x_{(a,b)}(t_i) - d_i)^2$$

where the **input** to the function is ________________$_{(1)}$, and the function takes as additional arguments: ______________________________$_{(2)}$.

12

Parameter Estimation

Notes for MATLAB Curve Fitting Choose a Metric slide:

Answers:

(1) the parameters a and b. Note:In the demonstration code included in the appendix, a and b are components of one input vector.

(2) the data points $\{(t_i, d_i)\}$, and the solution to the DE $x(t)$.

Parameter Estimation

MATLAB Curve Fitting: Example

To calculate the distance between our DE solution evaluated with given parameters a and b and the data points, we call with MATLAB syntax:

distance_function([*parameters*],@*function_name*,data)

where

- *distance_function* calculates ________$_{(1)}$.
- *[parameters]* is the ________________$_{(2)}$.
- *function_name* is a routine that ________________________________$_{(3)}$.
- “data” is a ________________$_{(4)}$ array of data points

13

Parameter Estimation

Notes for MATLAB Curve Fitting Example slide:

Answers:

(1) D Note: See previous slide.

(2) vector $[a, b]$.

(3) calculates the solution to the DEs, $x(t)$.

(4) $2 \times n$

Parameter Estimation

MATLAB Curve Fitting: Use FMINSEARCH

MATLAB's "**fminsearch**" function will estimate the unknown parameters in the differential equation by ________________(1) between the solution and the actual data points.

MATLAB syntax: [p,fval] = **fminsearch**(@*distance_function*,[initial guess],[],data)

fminsearch takes as input

- *distance_function*, the name of the function to be minimized
- [initial guess], an initial guess for the ________________(2)
- data, the array of actual data or any argument needed by *distance_function*.

Note: There is a placeholder, [], that can contain special options.

fminsearch returns

- p, a vector containing ________________(3)
- fval, the ________________(4) evaluated at the minimizing values in p.

Parameter Estimation

Notes for MATLAB Curve Fitting Use FMINSEARCH slide:

Answers:

(1) minimizing the distance
(2) minimizing parameter values
(3) the minimizing parameter values
(4) distance function

Notes:

The MATLAB routine takes as arguments the name of the function to be minimized, distance_function, (the first argument of this function must be the unknown parameters), an initial guess, in this case a **vector** [initial guess], and any additional values which are required by the function to be minimized, data. The routine outputs the minimizing values in the vector p, as well as the value of the function itself *evaluated* at those minimizing values, fval. The empty vector [ ] is a place-marker, but can contain special *options* to be sent to the routine. See **fminsearch** in the MATLAB Help menu for details.

Parameter Estimation

MATLAB Curve Fitting: Program Flow

Program Flow: The syntax will be different in other languages but the procedure remains roughly the same.

1. Store data in an array: $\begin{bmatrix} t_1 & t_2 & t_3 & \dots & t_n \\ x_1 & x_2 & x_3 & \dots & x_n \end{bmatrix}$

2. Store the solution function as a separate routine (*logisticsolution*)
Arguments: a, b, T_0, t; Output: $T(t)$. (Could require an ODE solver.)

3. Store the distance function as a separate routine (*logisticdist*)
Arguments: a, b, data array; Output: $D(a, b)$. Calls *logisticsolution* with arguments a, b, the first data entry ($x_1 = T_0$), and $\vec{t}$ containing the first row of the data.

Programming Note: It is more efficient to write the solution function in vector form: if it can take a *vector* as its variable, then all of T-values can be computed at once. Otherwise, a FOR-loop is necessary.

Parameter Estimation

MATLAB Curve Fitting: Program Flow (continued)

4. Call fminsearch
Arguments: the name of the distance function, a vector containing the initial guesses for the minimizing values of a and b, and the data; Output: the minimizing values of a and b in p, and the associated minimum distance.

5. Plot the data and the solution $T(t)$ evaluated using the computed “best” parameter values for visual comparison.

Parameter Estimation

Notes for MATLAB Curve Fitting: Program Flow slides:

Notes: A demo of the software used in the course might be appropriate here. If a computer-classroom is available, one of the exercises for this section could be used as an in-class project.

MATLAB demo code: see *ParDemo2*.

Parameter Estimation

MATLAB Curve Fitting: Graphical Output

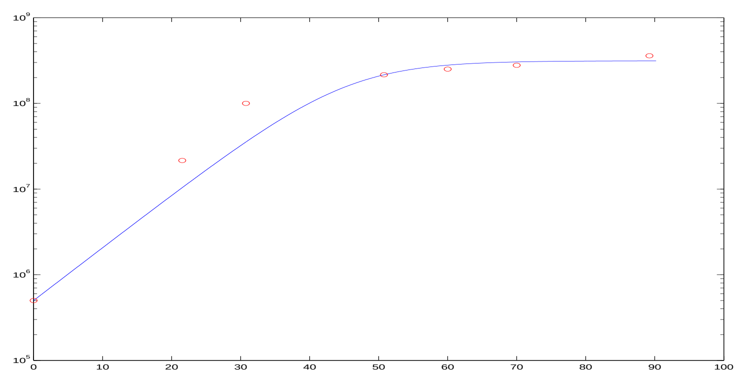


Figure 4: Comparison of estimated parameters with data

Parameter Estimation

Notes for MATLAB Curve Fitting Graphical Output slide:

Note: The results are plotted on a logarithmic scale because the number of tumor cells is so large, and also because the data are given this way in the paper.

Running the MATLAB routines will give the estimated values of a and b in the vector p to be $a = .14$ and $b = 3.1810^{-9}$. Ideally, the students will be able to see this generated real-time in class.

17-1

Parameter Estimation

Estimating the Other Parameters

The output of our minimization routine gives:

$a =$ ______ $_{(1)}$ and $b =$ ______ $_{(2)}$.

How can we find the remaining parameters?

Measurements in mice *without* tumors show $\longrightarrow$

- The spleen contains $\approx 10^8$ immune cells.
- Reacting CTL's (______ $_{(3)}$) comprise $\approx .32\%$ of the immune cells.
- $\Rightarrow$ The *number* of CTL's in the spleen $\approx$ ______ $_{(4)}$

Parameter Estimation

Notes for Estimating the Other Parameters slide:

Answers:

(1) .14 day^{-1}

(2) 3.1810^{-9}cell^{-1}

Notes Make note of the *units* here, as a reminder of the role the parameters play in the model. Also, we point out that the results here differ from the values given in [KMTP94]. This is due to the fact that the data we used were obtained by reading off values from the rather small figures in the paper, while the authors of the paper presumably used numerical data obtained from experiments. In the later sections, we will revert back to the parameter values given in the article, since we feel that the data they used are more accurate. However, the demonstration is useful in order to illustrate the *process* of parameter estimation.

(3) Cytotoxic T-Lymphocytes Note: These are part of the 'Effector' or E-population in our model.

(4) 3.2×10^8 cells $= .0032 \times 10^8$ cells

Parameter Estimation

Estimating s and d

From our earlier analysis of the model, we found that without any tumor, the number of immune cells approaches the ________________ (1) at $E_{\text{ss}} =$ ____________ (2).

Other experiments show that the average lifetime of a lymphocyte is 24.25 days, giving a death *rate* of $d =$ ____________ (3).

If we assume that the measured number of CTL's is the steady state value, we can estimate:

$$s = E_{\text{ss}} d \approx \text{\_\_\_\_\_\_\_\_\_\_\_\_}_{(4)}.$$

Parameter Estimation

Notes for Estimating s and d slide:

Answers:

(1) stable equilibrium (the tumor-free equilibrium)

(2) s/d, (we solved for this equilibrium by setting $T = 0$, and $dE/dt = 0$).

(3) $\frac{1}{24.25} \approx .0412 \text{day}^{-1}$.

(4) $1.3 \times 10^4 \frac{\text{cells}}{\text{day}} = .0412 \text{day}^{-}1 \times 3.2 \times 10^5$ cells

19-1

Parameter Estimation

Incorporating the Immune Parameters

To find the remaining parameters: __________ (1), __________ (2), __________ (3), and __________ (4), we need to repeat the ______________________________ (5) procedure, using the following data:

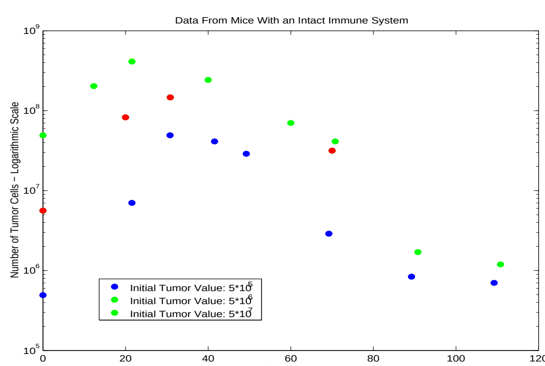


Figure 5: Tumor volumes over time with three initial conditions

Parameter Estimation

Notes for Incorporating the Immune Parameters slide:

Answers:

(1) p, (2) g, (3) m, (4) n.

(5) data fitting

Note: It might be worth recalling the biological meaning of these parameters here, to tie the equations back to the 'real world':

$$\begin{aligned} p &= \text{maximimum immune response rate}, \\ g &= \text{steepness of immune response}, \\ m &= \text{fraction of immune cells inactivated in interactions with tumor cells} \\ n &= \text{fraction of tumor cells killed in interactions with immune cells} \end{aligned}$$

Note also that these data have a different shape from the data for the chimeric mice: point out that the tumor grows and then shrinks, due to the response from the immune system. Clinically, these tumors are called "immunogenic".

20-1

Parameter Estimation

Parameter Estimation Without an Explicit Solution

Again, we need to minimize the ____________ $_{(1)}$ function $D(p, g, m, n)$ with respect to the unknown ____________ $_{(2)}$.

However, we no longer have a *formula* for $T(t)$. We need to compute it by ____________ $_{(3)}$ the system of differential equations.

The procedure is the same:

1. Gather the ____________ $_{(4)}$ in an array.
2. Numerically evaluate $T(t)$ at the ____________ $_{(5)}$ given in the data.
3. Calculate the distance between the ____________ $_{(6)}$ and the ____________ $_{(7)}$.
4. Minimize this distance over all possible ________ $_{(7)}$ values.

Parameter Estimation

Notes for Parameter Estimation Without an Explicit Solution slide:

Answers:

(1) distance Recall: The distance function measures the distance between the computed solution and the data. We've used a sum of squares formula in our routines, but other distances, or norms, may also be used.

(2) parameters

(3) numerically integrating

(4) data: $\{(t_i, d_i)\}$

(5) times, t_i

(6) computed values $T(t_i)$

(7) data, d_i.

(8) parameter

MATLAB demo code: see *ParDemo3.* This demo estimates the parameters p, g, m and n using the data given in Figures 1 a), b) and c) of [KMTP94].

21-1

Parameter Estimation

Results: Estimates of Immune Response Parameters

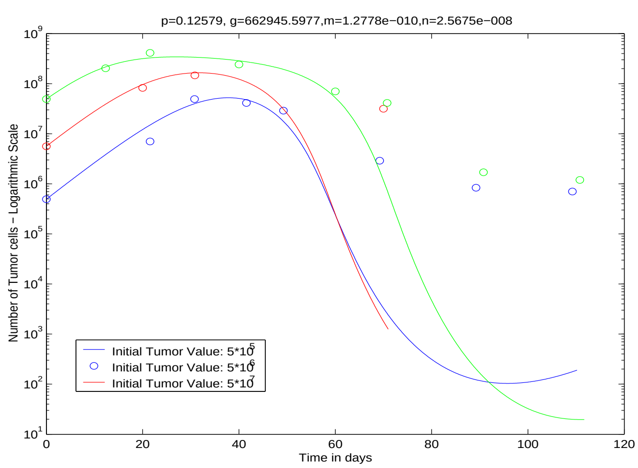


Figure 6: The parameters are estimated to be: $p = .1258, g = 6.62 \times 10^5, m = 1.27 \times 10^{-10}, n = 2.57 \times 10^{-8}$

Parameter Estimation

Notes for Results Estimates of Immune Response Parameters slide:

Notes: Point out that the tumor values are plotted on a logarithmic scale, to conform with the figures in the article [KMTP94]. Thus, the discrepancies at lower tumor values look larger than they are, relative to the higher values. On the other hand, we have not made any attempts to refine the parameter estimation procedure for this demo, preferring a straight-forward approach. The estimated parameters give graphs which do not match the data as well for the later time values. This suggests the following:

Questions for discussion:

1. In what way could the parameter estimation procedure be modified in order to better match the data points at the later time values? Suggestion: Give more weight to distances between computed values and data values for the later time points, by multiplying the squared differences by an increasing function of time. It should be noted, also, that in our estimation we *force* the initial values to match, perhaps thereby encouraging stronger agreement with the data for small time values.

2. Is it likely that the differences between computed values and the data are due to experimental error? How could we test this? Suggestion: The data are an

average of several experiments: the first and last sets are an average of two individual experiments, while the second set is the result of a single experiment. We are using all of the data together, and assuming that the parameters are the same for all of the experimental subjects. What is known about the variation in immune response between the different mice? Perhaps a large sample is needed to get parameters which are optimal for all of the experiments.

3. How bad is this fit? Is it bad enough to require an adjustment in the model equations? If so, what adjustments do the errors suggest? Suggestion: This is a tough question. Kuznetsov et al. argue that the qualitative results, in particular the regrowth of the tumor shown in the computed graphs, has been observed clinically. We contend that the sample size is too small, and that parameter values vary too widely over the different mice. A larger sample size would be needed to reach a definitive answer as to whether this model adequately mirrors reality. In general, we can only hope to get qualitative information from a model which only uses two cell types from the entire organism.

Note: In the analysis of the module, we will use the parameter estimates from the article [KMTP94] as the 'normal' values, rather than our own estimates. We do this for two reasons: we assume that the authors of the article had access to more precise data, and we feel that using different parameter values might be confusing to the student who is reading the orginal article as she works through the module.

We collect here for completeness the parameter estimates from [KMTP94]:

$$\begin{aligned} a &= .18\,\text{day}^{-1}, \quad b = 2.0 \times 10^{-9}\text{cells}^{-1}, \quad d = .0412\,\text{day}^{-1} \\ p &= 0.1245\text{day}^{-1}, \quad g = 2.019 \times 10^{7}\text{cells}, \quad s = 1.3 \times 1-^{4}\,\text{cells day}^{-1} \\ m &= 3.422 \times 10^{-10}\text{day}^{-1}\text{cells}^{-1}, \quad n = 1.101 \times 10^{-7}\text{day}^{-1}\text{cells}^{-1} \end{aligned}$$

References

[KMTP94] Vladmir A. Kuznetsov, Iliya A. Makalkin, Mark A. Taylor, and Alan S. Perelson. Nonlinear dynamics of immunogenic tumors: Parameter estimation and global bifurcation analysis. *Bulletin of Mathematical Biology*, 56(2), 1994.

[MT73] Daniel P. Maki and Maynard Thompson. *Mathematical Models and Applications*. Prentice-Hall, Inc., 1973.