Proposed Procedures: Continuous Analysis of a Model for T Cells in Autoimmune Diabetes

David Li

Materials

Estimated Total Cost.....$0

1. Computer (x86-64 architecture running a Linux distribution; no need to purchase).....$0
   The exact specifications do not matter, nor should the operating system, as the software used runs on all platforms. A relatively recent (past 5 years or so) computer will absolutely suffice.
2. Computer Software.....$0
   - Python 3.3.2 with `mpmath` (http://www.python.org, http://www.mpmath.org). This will be used to calculate the data for the plots.
   - XPP AUTO 7.0 (http://www.math.pitt.edu/~bard/xpp/xpp.html). This will be used to calculate the data for the bifurcation diagrams.
   - ConTEXt 2013.05.28 (http://wiki.contextgarden.net)[1]. This will be used to generate the plots and bifurcation diagrams (explained below).
   - Gnuplot 4.6.4 (http://www.gnuplot.info), also used to generate plots.

High-Level Procedural Overview

Background Details

See the "Detailed Procedures" appendix on page 4 for more details.

This experiment centers on the bifurcation analysis of the model designed by Mahaffy and Keshet as described at [1]. Their model describes the level of T cells reactive to a peptide found on pancreatic beta cells. These immune system cells destroy the insulin-producing cells, leading to type 1 diabetes. The level of T cells in the weeks before the onset of diabetes (the appearance of symptoms) oscillates, as demonstrated by previous experiments [[2] cited in [1]]. One of the parameters in the model[2] is the level of beta cells, which slowly decreases over time as the disease progresses; after a certain point, symptoms begin to appear. The original analysis centered on indirectly varying the level of beta cells (via another parameter, the *peptide clearance rate* δ_p) to look for *bifurcations*, or qualitative changes in the model's behavior. In particular, the authors considered the long-term behavior of the model system at multiple constant values of the parameter [1].

This experiment applies the research of Dr. Baer, who found that when such parameters are slowly and continuously varied, the qualitative nature of the system differs from when the parameter is statically set as described before [3]. This change more accurately reflects what actually occurs, as the original paper explicitly states the parameter should continuously slowly fall[1]. Thus, this experiment will help scientists and medical professionals better understand and apply the theoretical results of Mahaffy and Edelstein-Keshet's work as well as the experimental results they cited.

To summarize: for the original *static* analysis, the authors re-ran the model multiple times, each time setting the parameter to a fixed value. For the *continuous* analysis conducted here, the parameter starts at a given value and then continuously decreases over time.

---

[1] Inkscape may also be required to be able to generate SVG diagrams.

[2] The original paper contains multiple variants of the model; we are focusing on the *scaled reduced model* here.

Procedural Overview

XPP AUTO, the same software used in the original experiment, is used to compute the solution of

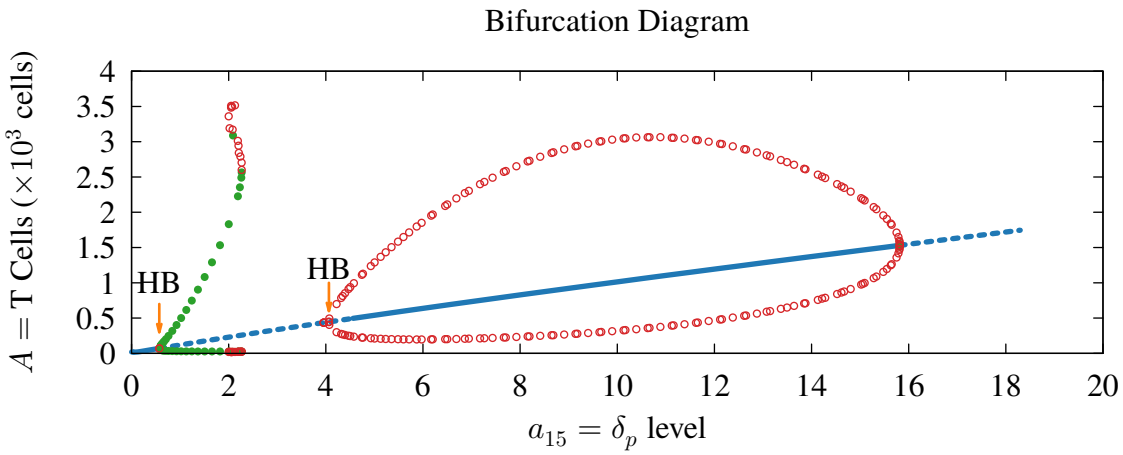
$$p = \frac{a_{14}}{a_{15}} EB \qquad f_1 = \frac{p^{a_1}}{{a_2}^{a_1} + p^{a_1}} \qquad f_2 = \frac{a_4 {a_5}^{a_3}}{{a_5}^{a_3} + p^{a_3}}$$

$$\frac{dA}{dt} = (a_6 + a_7 M) f_1(p) - a_8 A - a_9 A^2$$
$$\frac{dM}{dt} = a_{10} f_2(p) A - f_1(p) a_7 a_{16} M - a_{11} M$$
$$\frac{dE}{dt} = a_{12}(1 - f_2(p)) A - a_{13} E$$

with $a_{15} = \delta_p = \text{constant}$ for the static analysis or $\frac{da_{15}}{dt} = \frac{d\delta_p}{dt} = \text{rate of change}$ for the continuous analysis[3]. Here, A represents the level of activated T cells, B the level of pancreatic beta cells, E the effector T cells, M the memory T cells, and p the peptide level. (The fluctuation of activated T cells matters the most for the onset of diabetes.) In the model, pancreatic beta cells that undergo apoptosis (cell death) generate peptide; in lymph nodes, T cells become activated and "recognize" this particular peptide. These cells then become either effector cells, which destroy more pancreatic cells, thus producing more peptide, or memory cells, which encounter the peptide and trigger an immune response.

Because B and δ_p are constant, an increase in the latter is equivalent to a decrease in the former (and vice-versa), so the latter was increased in Mahaffy's analysis, which is equivalent to a decrease in beta cell levels [1]. Additionally, a 4$^{\text{th}}$ order Runge-Kutta ODE solver written in Python is also used to compute the numerical solution in the second case. Runge-Kutta is a standard algorithm for solving ODEs; it is used here because Dr. Baer's work noted that the continuous analysis could lead to numerical precision errors, and the Python implementation allows for arbitrary-precision computations [3][4][5].

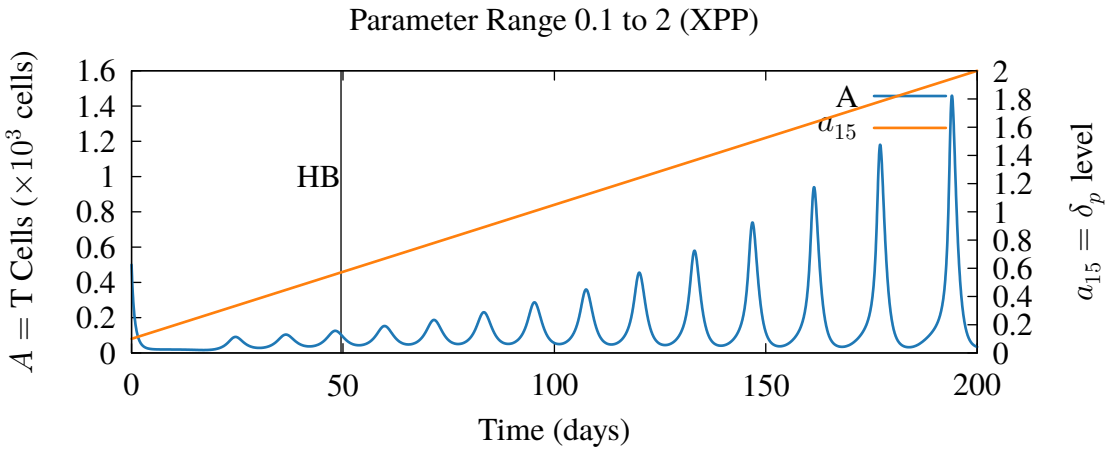
After simulation, the experiment will result in a set of bifurcation diagrams much like this one[4]:



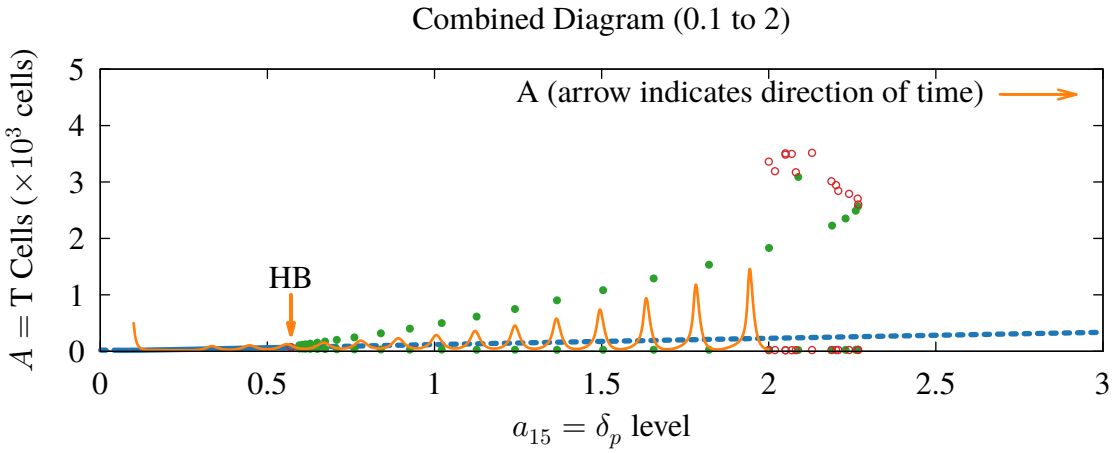
as well as model-over-time diagrams such as this one:

[3] The rate of change depends on the parameter range in consideration (see the appendix).

[4] These diagrams were generated by beta runs of the eventual experimental software.



The bifurcation diagram shows the *maximum and minimum values of A over time for a trial run of the model with the particular parameter level indicated on the x-axis*. Thus, it shows at what *static* parameter levels diabetes can occur. For this experiment, a third diagram will be generated overlaying the model-over-time diagram for the *continuous* system onto the bifurcation diagram:



Since the parameter will continuously and linearly vary over time, a correspondence between the points of the two graphs exists (i.e. for the model-vs-time graph, at a certain time, the parameter will be at a unique particular value in the X-range of the bifurcation diagram), and therefore the two modes of simulation can be compared – the goal of this experiment. (See the appendix for the implementation details of processing the data and generating the diagrams.)

Additionally, this experiment will compare the static bifurcation diagrams of various other parameters when δ_p is analyzed statically and continuously, in order to find out how these parameters interact with the continuous model versus the static model. $a_9 = \varepsilon$ controls the competition between T cells, and $a_5 = k_2$ represents the level of peptide at which f_2, the maximum number of activated T cells, is at half its maximum level.

Appendix A. Detailed Procedures

Note: familiarity with the Linux command line is assumed.

1. Download the necessary experiment software and experiment files (location TBD).
2. Extract the experiment files to a directory.
3. Open a shell and `cd` to the directory containing the experiment files.
4. To generate plots of the system over time:
   a. Run `python3 rk4_ode.py` to generate the data via the Runge-Kutta method.
   b. Run `./time_diagrams_xppaut.sh` to generate the data via XPP AUTO.
   c. Run `context time_diagrams.tex` to generate a PDF containing the plots.
5. Then, to generate the plots of the system vs the bifurcation parameters (the peptide clearance rate, δ_p; fraction of memory T cells produced, a; T cell competition parameter ε, and peptide level for ½ max memory cells, k_2):
   a. Run `../path_to_xpp/xppaut -runnow mk_static.ode` to launch XPP AUTO. Bring up AUTO and save the points for the complete bifurcation diagram in `mk_static.dat`. If other parameters are to be analyzed, also save the diagrams for those cases.
      i. Press `f a` to bring up AUTO.
      ii. Press `f l` and choose `mk_static.ode.auto` to load the parameters for the bifurcation diagram.
      iii. Press `r s` to generate the diagram.
      iv. Press `g`, `<TAB>` over to the bifurcation point (labeled `HB` on the bottom), press `<ENTER>` to select it, and then `rp` to generate the branch.
      v. Repeat for each bifurcation point.
      vi. Press `f a` and save the points in `mk_static.dat`.
      vii. Close XPP AUTO (`Control-C` the program in the terminal).
   b. Repeat step a for `mk_static_epsilon.ode` with `mk_epsilon.ode.auto` and `mk_static_k2.ode` with `mk_k2.ode.auto`, and then for `mk_continuous_epsilon.ode` and `mk_continuous_k2.ode` with the corresponding `.auto` files. Name the result file after the parameter and `.ode` file (e.g. `mk_continuous_epsilon.dat`, `mk_static_k2.dat`).
   c. Run `python3 bifurcation_diagrams.py` to process the plot data.
   d. Run `context bifurcation_diagrams.tex` to generate a PDF containing the plots.

Parameters for XPP AUTO and Python

The parameters in the following table are *not* parameters for the model, but rather for the programs used to generate the model data. All parameters not listed were left at default values.

For the parameter δ_p:

Parameter	**Value**
Integration time (the time the system will be run for)	200 days
Integration step (RK4, both Python and XPP AUTO)	0.05 days
`mpmath` Decimal Precision (`mpmath.mp.dps`)	200
AUTO Par 1	`a15`
AUTO Hi-Lo Y-axis	`A`
AUTO Hi-Lo Main Parameter	`a15`
AUTO Hi-Lo ranges	X: $[0, 18]$ Y: $[0, 3]$
AUTO Par Max	18

For the parameter ε (assume other parameters are the same as before):

AUTO Par 1	`a9`
AUTO Hi-Lo Main Parameter	`a9`
AUTO Hi-Lo ranges	X: $[0, 5]$ Y: $[0, 3]$
AUTO Par Max	5
AUTO Ds	-0.02

For the parameter k_2 (assume other parameters are the same as before):

AUTO Par 1	`a5`
AUTO Hi-Lo Main Parameter	`a5`
AUTO Hi-Lo ranges	X: $[0, 2.5]$ Y: $[0, 3]$
AUTO Par Max	2
AUTO Ds	-0.02

Initial Conditions

For the model when processed over time:

$$A = 0.5 \quad M = 0 \quad E = 1 \quad B = 1 \quad a_{15} = \text{See below}$$

For the model when generating bifurcation diagrams for the parameter a_{15}:

$$A = 0.01961524 \quad M = 2.802597 \times 10^{-45} \quad E = 0.006538414 \quad B = 1 \quad a_{15} = 0$$

Parameter Ranges

As the total time is 200 days, the parameter will be varied at a rate of $(\delta_{p_2} - \delta_{p_1})/200$ per day. These ranges were chosen based on a reading of Mahaffy's paper [1]. Although the authors did not call out explicit parameter ranges, the ranges here represent either the domains of graphs given or are near bifurcation points they noted. Also, they tended to include a parameter value of 0, which has been adjusted to 0.1 here to avoid a division-by-zero error. XPP AUTO does not return an error when this occurs, but Python does, so the adjustment avoids conflicts and undocumented behaviors. This will not adversely affect the impact of this experiment; because the original analysis used static parameter values, the overall range of values does not matter as long as it includes the range used in the continuous analysis to be done here for comparison.

Parameter	**Ranges**
δ_p, the peptide clearance rate	0.5 to 2.3
(continuous, others are static)	0.1 to 2
	2 to 0.1
ε, the T cell competition parameter	5 to 0
k_2, the half-maximal peptide level	2 to 0

Appendix B. Bibliography

[1]: Mahaffy, Joseph M. & Edelstein-Keshet, L. (2007). Modeling Cyclic Waves of Circulating T Cells in Autoimmune Diabetes. *SIAM Journal on Applied Mathematics, 67*.

[2]: Trudeau, J. D. & Kelly-Smith, C. & Verchere, C. B. & Elliott, J. F. & Dutz, J. P. & Finegood, D. T. & Santamaria, P. & Tan, R. (2003). Prediction of spontaneous autoimmune diabetes in NOD mice by quantification of autoreactive T cells in peripheral blood. *Journal of Clinical Investigation, 111*.

[3]: Baer, S. & Erneux, T. & Rinzel, J. (1989). The Slow Passage Through a Hopf Bifurcation: Delay, Memory Effects, and Resonance. *SIAM Journal on Applied Mathematics, 49*.

[4]: Johansson, F. (2011). Precision and representation issues. *mpmath v0.17 documentation*. Retrieved on September 14, 2013 from http://mpmath.googlecode.com/svn/trunk/doc/build/technical.html.

[5]: Gonsalves, R. J. (2009). Runge-Kutta Methods for ODE Systems. *Computational Physics*. Retrieved on October 24, 2013 from http://www.physics.buffalo.edu/phy410-505-2009/topic3/lec-3-2.pdf.