Enabling Physiologically Representative Simulations of Pancreatic Beta Cells Imbedded in an Islet

Team members: Sidafa Conde[1], Teresa Lebair[2], Christopher Raastad[3], Virginia Smith[4];
Graduate assistants: Kyle Stern[5], David Trott[5]; Faculty mentor: Matthias K. Gobbert[5];
Clients: Bradford E. Peercy[5], Arthur Sherman[6]

[1]Department of Mathematics, University of Massachusetts Dartmouth
[2]Department of Mathematics, Saint Vincent College
[3]Department of Mathematics, Department of Computer Science, University of Washington
[4]Department of Mathematics, University of Virginia
[5]Department of Mathematics and Statistics, University of Maryland, Baltimore County
[6]Laboratory of Biological Modeling, National Institutes of Health

Abstract

Diabetes is a collection of diseases marked by high levels of glucose in the blood. The condition results from defects in insulin production or function, which are activities performed by the pancreas. Within the endocrine system of the pancreas lie clusters of cells called islets. Each islet is composed of four different cells, the most prevalent of which being the beta cell. The main function of beta cells is to secrete insulin in response to blood glucose levels. As a result, the behavior of these cells is an issue of ongoing interest in diabetes research.

Our research aims to take the next step in implementing the mathematical model governing beta cells by continuing the development of a computational islet. The mechanisms of insulin secretion within beta cells can be modeled with a set of deterministic ordinary differential equations. Considering cell dynamics of a cube of individual heterogeneous cells, the key parameters influencing the time evolution include ionic fluxes, calcium handling, metabolism, and electrical coupling. Capturing sudden changes of cell properties on a millisecond time scale requires the use of a stiff ODE solver. The computational complexity makes the simulation of islet behavior difficult and inefficient without sophisticated software built with careful consideration of robust mathematical numerical techniques.

Our research focuses on creating an extensible, efficient, and functional computational beta cell software to aid current and future research in beta cell dynamics. In particular, we adapt existing glycolytic oscillator Matlab code into a numerically robust, modular set of Matlab files. By developing in Matlab, we create code that remains easily modifiable by mathematical biologists for a broad range of future applications. Studies on the cluster tara in the UMBC High Performance Computing Facility demonstrate that simulations up to the desired resolution are now practical. Application simulations of the beta cell islet model led to an unexpected discovery that warrants further study: For certain intermediate values of the coupling strength, a small increase in the number of fast cells acts by first increasing the burst period, before falling into the pattern of reducing the burst period with larger proportions of fast cells again.

Keywords: Beta cell model, computational islets, glycolytic oscillations

1 Introduction

In the REU Site: Interdisciplinary Program in High Performance Computing in the Department of Mathematics and Statistics at the University of Maryland, Baltimore County (UMBC), during the summer of 2010, our team of four students Sidafa Conde, Teresa Lebair, Christopher Raastad, and Virginia Smith, researched ways of enabling more effective ways of implementing physiologically representative models of pancreatic beta cell islets, under the guidance of Dr. Matthias Gobbert. This project was proposed to us by Dr. Bradford Peercy and Dr. Arthur Sherman.

Diabetes is a disease distinguished by an abnormally high concentration of glucose in the blood stream. Approximatly 23.6 million children and adults in the United States, 7.8% of the population, suffer from the desease (American Diabetes Association, [1]). The condition results either from defects in insulin production within the pancreas or from defects in functionality as the insulin interacts with cells throughout the body. Within the endocrine system of the pancreas lie clusters of cells called the islets of Langerhans. Each islet is composed of four cell types, the most prevalent of which is the beta cell.

The individual beta cells within an islet of Langerhans are believed to be coupled, that is joined together by proteins, called gap junctions. These gap junctions create pores that allow the passage of ionic and small molecular material between two adjacent cells. Therefore, our model takes into account nearest neighbor coupling between cells and for ease we take these cells to exist on a cubic lattice of $N \times N \times N$ cells. Physiologically relevant results require at least the resolution of $N = 5$, but higher resolutions up to $N = 10$ are desirable.

A recently proposed dual oscillator model combines electrical and metabolic mechanisms to explain the pulsing activity of the insulin secretion from beta cells. This model is given by a set of seven ordinary differential equations for each cell. Individual beta cells can be classified by their electrical patterns as either slow or fast bursting cells. The insulin secretion of each of these cell types follows the electrical oscillatory pattern: a fast cell secretes insulin at a relatively constant elevated level having smoothed out the relatively fast electrical oscillations, while a slow cell secretes insulin with a period of roughly 4 to 6 minutes. The islets of Langerhans contain both slow and fast bursting beta cells, although the proportion and distribution of slow and fast cells within the islets is currently unknown.

In the eight weeks of the program, we accomplished the following:

- The implementation of the computational model for an islet of beta cell was improved in efficiency: Key modifications to Matlab's ODE solver `ode15s` reduced the memory usage dramatically and enabled fast simulations for physiologically relevant resolutions of $N = 5$ and made simulations up to $N = 10$ feasible, instead of previously only $N = 2$ or 3. The research was truly interdisciplinary: For instance, it was close interaction with the application researchers that led to a more efficient design of the simulations by reducting the number of trials for proportions of fast-slow cells to no more than 11 for larger N. Simulations with $N = 10$ still take 18 hours, but using 11 nodes of the cluster tara in the UMBC High Performance Computing Facility (`www.umbc.edu/hpcf`) simultaneously accomplishes all desired trials at the same time, thus the complete set of simulations is accomplished in the time it takes one simulation to run.

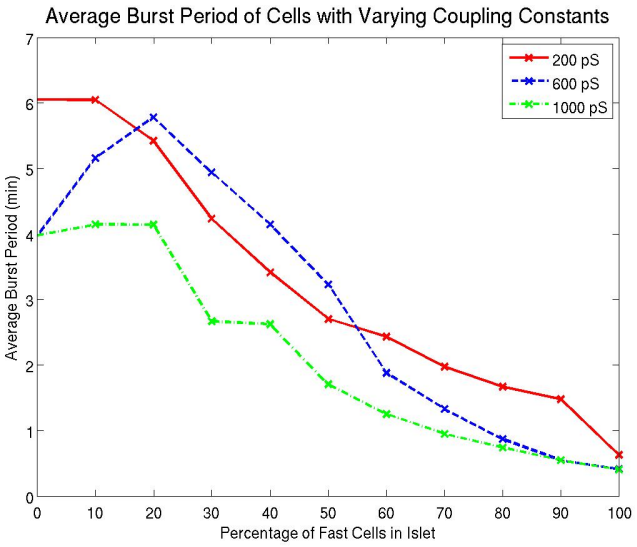
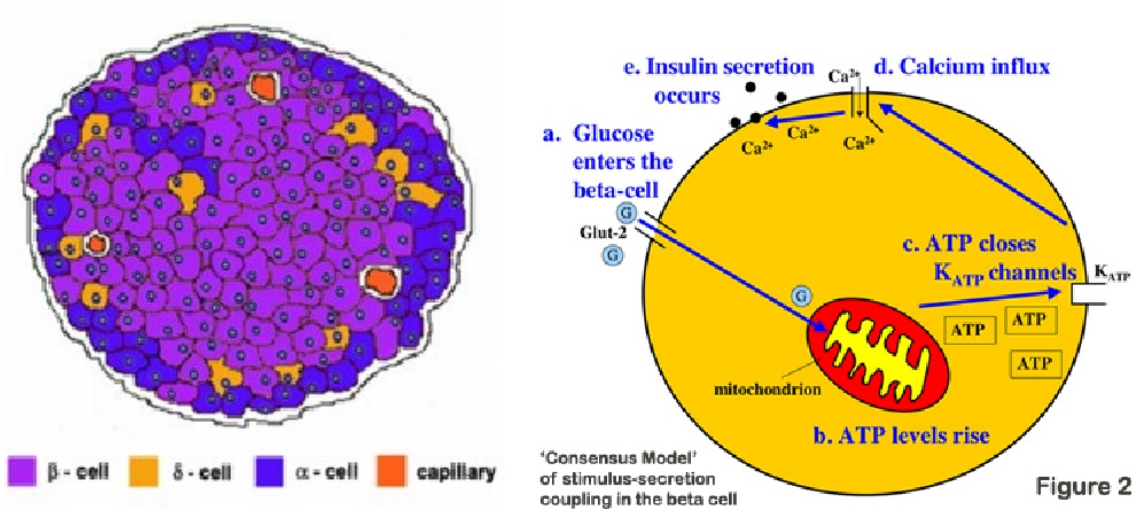


Figure 1.1: Average burst period of cells in units of minutes vs. percentage of fast cells for cell coupling strength 200, 600, 1000 pS.

- The implementation of the computational model for an islet of beta cell was also extended in functionality: We extended the implementation of the model by implementing a beta cell islet with random coupling strengths. By running simulations with varying coupling strength, we simulated the impact of coupling within the islets. In particular, we ran simulations for different coupling strengths on a variety of islets, each with a different combination of slow and fast cells.

Application simulations of the beta cell islet model led to a potentially transformative discovery: Figure 1.1 displays the average burst period of cells in an islet with $5 \times 5 \times 5$ cells, with varying proportions of fast and slow cells. The plot is shown for three different strengths of coupling: 200, 600, and 1000 pS. In the extreme cases of weak and of strong coupling, an increase of the proportion of fast cells lessens the average burst period. However, for intermediate values of the coupling strength, a small increase in the number of fast cells acts by first increasing the burst period, actually slowing down the oscillations, before falling into the pattern of reducing the burst period with larger proportions of fast cells again. This result is counter-intuitive and warrants further study.

This paper is organized as follows: In Section 2, we descibe the biological background of beta cell islets and glycolytic oscillations. Included in Section 3 is a summary of the deterministic mathematical model of a bimodal glycolytic oscillators. In Section 4, we give an overview of the numerical implementation of the mathematical model of the beta cell and an analysis of an adapted Matlab `ode15s` fitted for the problem. Section 5 summarizes the performance and application results of experimental runs of the Matlab simulation. We discuss the technical improvements of our implementation of the simulation in Section 5.1 and qualify the validity of our computational experiments in a scientific context in Section 5.2.



(a) (b)

Figure 2.1: (a) Islet of Langerhans. (b) Consensus model.

2 Background

Diabetes mellitus, more commonly known as diabetes, is a set of diseases distinguished by an abnormally high concentration of glucose in the blood stream. Approximatly 23.6 million children and adults in the United States, 7.8% of the population, suffer from the desease (American Diabetes Association, [1]). The condition results either from defects in insulin production within the pancreas or from defects in functionality as the insulin interacts with cells throughout the body. Ideally, the concentration of glucose in the blood stream is well-regulated by insulin secreted from the endocrine system within the pancreas. However, this is not the case for diabetics, whose pancreases are unable to produce an adequate amount of functional insulin or whose bodies are unable to respond properly to the insulin released.

Within the endocrine system of the pancreas lie clusters of cells called the islets of Langerhans. Islands of cells within the pancreas containing beta cells, δ-cells, α-cells and capillaries as shown in Figure 2.1 (a) of which the most prevalent is the beta cell. The metabolic and electrical activities of these islets are closely related with the islets' insulin secretion. The beta cells' ability to secrete insulin is a function of the amount of glucose in the blood stream. Figure 2.1 (b) depicts the consensus model of the stimulus-secretion coupling in the beta cell.

The amount of insulin collectively secreted by the islets is intricately related to the electrical and metabolic activity of these cells [7]. Thus, modeling the electrical and metabolic activities of these islets is of paramount importance to consider when studying beta cell insulin secretion and more generally, diabetes. The depolarization of the beta cell occurs in reponse to an ATP increase closing ATP-sensitive K^+-channels. Calcium entry during depolarization causes exocytosis of insulin containing vesicles. The change in voltage of the beta cell often occurs in bursts repeating in seconds or minutes.

The pulsatility of individual beta cells' voltage patterns can be used to categorize beta

cells as either slow or fast bursting cells. The voltage of each of these cell types follows an oscillatory pattern: a fast cell bursts with an oscillatory pattern lasting tens of seconds long, while a slow cell bursts with an oscillatory pattern lasting 4 to 6 minutes [7]. There exist both slow and fast bursting beta cells in the islet due to variations in channel distribution although the proportion and distribution of the cell types is unknown and varies.

The individual cells within an islet of Langerhans are coupled, joined together by proteins, called gap junctions, spanning neighboring cells membranes to create a molecular pore. In a nearest neighboor coupling, these gap junctions allow the passage of ionic and molecular material between two adjacent cells. As a result of this coupling, the pulsatile activity of the islet itself resembles that of the individual beta cells within the islet [7]. We consider numerous heterogenous cells, on a cubic lattice, of on average 1000 heterogeneous cells with key parameters including ionic fluxes, calcium handling, metabolism, and electrical coupling.

Individual beta cell insulin secretion depends upon the cytosolic calcium concentrations derived from both electrical and metabolic activity in each individual beta cells influences how insulin is secreted. A recently proposed mechanism of insulin secretion amalgamates glycolytic and electrical activity models to explain the pulsatile activity of the secreted insulin of the cell in the dual oscillator model. The dual oscillator model accounts for many more hypotheses and explains more trends in a plethora of data gathered in the past few decades than either the electrical or glycolytic models separately [3]. This model is defined by seven deterministic ordinary differential equations that have been used for sometime in literature [2, 3]. In this set of ODEs, 4 of the equations model the electrical system (membrane potential, activation of delayed rectifier, free cytosolic calcium concentration, concentration of free calcium in the endoplasmic reticulum) while the remaining 3 pertains to glycolysis (cytosolic ADP concentration, fructose 1,6-bisphosphate concentration, and glucose 6-phosphate) with critical feedback from ADP to the electrical subsystem and calcium to metabolism of ADP. Cell dynamics are then observed on a cubic lattice of on average 1000 heterogeneous cells, coupled according to different inter-facial conductance values.

The sheer number of degrees of freedom makes implementing an islet model with individual beta cell dynamics computationally complex. In addition, the rapid changes in beta cell behavior influences the need for simulation on a millisecond time scale for durations of minutes or more on the laboratory time scale. This requires the use of a stiff ODE solver. As a result, simulations are difficult and/or inefficient without sophisticated software built with careful consideration of robust numerical and efficient programming techniques. Codes in the past have offered solutions, but are ineffective at providing the necessary physiologically representative models of beta cells due to memory and time usage.

Developing code to implement models of the chemical and electrical activities of the cells within an islet of Lagerhans that account for various porportion of slow/fast beta cells, the concentration of a variety of chemical species, and different coupling between the cells will enhance the current understanding of whole islet function and, ideally, of diabetes as well. To this end, we develop a memory and time efficient, extensible code, loosely based on preexisting code by Dr. Bradford Peercy and Kaung San (UMBC) (unpublished), in Matlab, to allow modeling of the electrical and metabolic activity of cubical large islets.

3 Mathematical Model

We simulate the glycolytic oscillator model of the beta cell using a deterministic model developed in [8]. Bertram and Sherman [2] coupled this original model to the same model developed by Chay and Keizer [5] as updated by Sherman [7], including voltage data by Rorsman and Trube [6].

In the dual oscillator model, the dynamics of a single beta cell are govered by the system of time dependent ordinary differential equations

$$\frac{dV}{dt} = \frac{-(I_{\mathrm{K}} + I_{\mathrm{Ca}} + I_{\mathrm{K\,(Ca)}} + I_{\mathrm{K\,(ATP)}})}{C_m}, \tag{3.1}$$

$$\frac{dn}{dt} = \frac{(n_\infty - n)}{\tau_n}, \tag{3.2}$$

$$\frac{d[\mathrm{Ca}]}{dt} = f_{\mathrm{cyt}}\,(J_{\mathrm{mem}} + J_{\mathrm{er}}), \tag{3.3}$$

$$\frac{d[\mathrm{Ca_{er}}]}{dt} = -\sigma_V\, f_{\mathrm{er}}\, J_{\mathrm{er}}, \tag{3.4}$$

$$\frac{d[\mathrm{ADP}]}{dt} = \frac{[\mathrm{ATP}] - [\mathrm{ADP}]\,\exp\left\{(r+\gamma)\left(1 - \frac{[\mathrm{Ca}]}{r_1}\right)\right\}}{\tau_a}, \tag{3.5}$$

$$\frac{d[\mathrm{G6P}]}{dt} = \kappa\,(R_{\mathrm{GK}} - R_{\mathrm{PFK}}), \tag{3.6}$$

$$\frac{d[\mathrm{FBP}]}{dt} = \kappa\,(R_{\mathrm{PFK}} - 0.5\,R_{\mathrm{GPDH}}). \tag{3.7}$$

The 7 independent variables are V, n, [Ca], [$\mathrm{Ca_{er}}$], [ADP], [G6P], [FBP], where [XX] denotes concentration of compound XX in mols per liter. Equations (3.1)–(3.4) describe the electrical activity in the cell while (3.5)–(3.7) describe the metabolic activity in the cell.

In (3.1), V describes membrane electric potential in mV. The quantities $I_{\mathrm{K}}, I_{\mathrm{Ca}}, I_{\mathrm{K\,(Ca)}}$, and $I_{\mathrm{K\,(ATP)}}$ represent ionic currents, where

$$\begin{aligned}
I_{\mathrm{K}} &= g_{\mathrm{K}}\, n\,(V - V_{\mathrm{K}}),\\
I_{\mathrm{Ca}} &= g_{\mathrm{Ca}}\, m_\infty\,(V - V_{\mathrm{Ca}}),\\
I_{\mathrm{K\,(Ca)}} &= g_{\mathrm{K\,Ca}}\,(V - V_{\mathrm{K}}),\\
g_{\mathrm{K\,Ca}} &= \frac{\overline{g_{\mathrm{K\,Ca}}}}{1 + \left(\frac{K_d}{[\mathrm{Ca}]}\right)^2},\\
I_{\mathrm{K\,(ATP)}} &= g_{\mathrm{K\,ATP}}\,(V - V_{\mathrm{K}}),\\
g_{\mathrm{K\,ATP}} &= \overline{g_{\mathrm{K\,ATP}}}\, o_\infty.
\end{aligned}$$

In (3.2), the independent variable n is the open fraction of voltage-gated K^+ channels. Equation (3.2) along with (3.1) contribute to create the spikes in traces during active phases of glucose oscillation.

In (3.3), [Ca] is the concentration of free intracellular [Ca] ions.

In (3.4), [$\mathrm{Ca_{er}}$] is the concenration of [Ca] ions in the endoplasmic reticulum.

In (3.5), [ADP] is the concenration of Adenosine Diphosphate compound.

In (3.6), [G6P] is the concenration of Glucose 6-Phosphate compound.

In (3.7), [FBP] is the concenration of Fructose 1, 6-Bisphosphate.

In order to simulate the electrical and metabolic behaviour in an islet of Langerhan, we discretize the islets into an $N \times N \times N$ cube of beta cells, where N is the number of beta cells along one side of the cube. We attribute each cell's state for n_s species, the number of indpendent variables attached to that particular cell. In the simulations, we maintain $n_s = 7$ independent variables, $V, n, [\mathrm{Ca}], [\mathrm{Ca_{er}}], [\mathrm{ADP}], [\mathrm{G6P}]$ and $[\mathrm{FBP}]$.

One of the most important parameter of interest for each simulation trial is the distribution of slow/fast cells in the islet. Each cell is assigned S or F for the three parameters $R_{\mathrm{GK}}, \overline{g_{\mathrm{K\,ATP}}}$, and $\overline{g_{\mathrm{K\,Ca}}}$, describing the secretition rate as slow or fast cells respectively. Specifically we assign

$$\begin{aligned} S &= (R_{\mathrm{GK}} = 0.2,\ \overline{g_{\mathrm{K\,ATP}}} = 27\,000,\ \overline{g_{\mathrm{K\,Ca}}} = 100) \\ F &= (R_{\mathrm{GK}} = 0.4,\ \overline{g_{\mathrm{K\,ATP}}} = 25\,000,\ \overline{g_{\mathrm{K\,Ca}}} = 600) \end{aligned}$$

Another important parameter influencing the solution for the simulation is species coupling and coupling strength. For simplicity, we assume nearest neighbor coupling for all beta cell interactions, that is, the state of directly nearest neighbors, only in the immediete i, j, k direction (no diagonal) influences the state of particular cell. Mathematically, consider an N^3 grid of beta cells, each coupled to its nearest neighbors. Figure 3.1 shows the arrangement of N^3 cells on a three-dimensional lattice for $N = 5$. Such a lattice arrangement of cells is appropriate for this research, since the focus is on studying the influence of the values of the physiological parameters, not on the islet geometry. For a given N and species s, nearest neighbor coupling is represented by an $N^3 \times N^3$ adjacency matrix C. Let l represent the current variable in the full set of S variables, such that the system may be split into coupled and uncoupled variables (S/l). The (i, j, k) position in $\mathbb{R}^3$ is found by:

$$\begin{aligned} c_l^{i,jk} &= f_l(c^{i,j,k}; p^{i,j,k}) + g_j^A(c^{i+1,j,k} - c^{i,j,k}) + g_j^B(c^{i-1,j,k} - c^{i,j,k}) + g_j^C(c^{i,j+1,k} - c^{i,j,k}) \\ &\quad + g_j^D(c^{i,j-1,k} - c^{i,j,k}) + g_j^E(c^{i,j,k+1} - c^{i,j,k}) + g_j^F(c^{i,j,k-1} - c^{i,j,k}) \\ c_{S/l}^{i,j,k} &= f_{S/l}(c^{i,j,k}; p^{i,j,k}) \end{aligned}$$

This is simplified by defining a nearest neighbor coupling matrix, C, such that for a given cell,

$$x_l^{i,j,k} = f(x^{i,j,k}; p^{i,j,k}) + C\,y$$

with y representing the solution vector.

We assume no interaction between cell variables of different species, so only the state of a given variable in neighboring cells influences that variable. Thus, allowing the user to vary coupling constants between neighboring cells for each species and observed the variation and influence in the islet.

We can then index each cell from the (i, j, k) position in the cube by an array index $a(i, j, k) = i + N\,j + N^2\,k$. Then the (i, j) entry of C indicates the coupling strength between cell a_i and cell a_j, where a_m is the m-th indexed cell in array indexing notation. We assume that cell coupling is equally symmetric, such that the coupling between a_i to a_j is equivalent

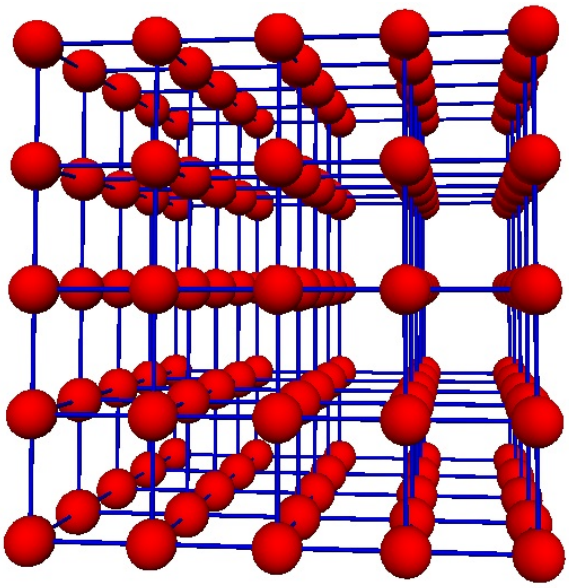


Figure 3.1: Beta cell connectivity in the 3-D computational islet. Beta cells (spheres) are coupled electrically (rods) to neighbors.

to the coupling between a_j to a_i, hence C is symmetric since $C(i,j) = C(j,i)$ for all entries.

4 Numerical Method

The origin of our implementation of the glycolytic oscillator mathematical model was formally described and published as a `.ode` file by Bertam, Satin, Zhang, Smolen, and Sherman in 2004 [4]. Using this description, Kaung San implemented a first version of the model in Matlab in a handful of Matlab files as a research project under the direction of Dr. Peercy in Summer 2009. We used San's Matlab implementation as starting point for our implementation of the glycolytic oscillator simulation. What follows is a complete summary of the simulation, key sets of parameters, extensibility, and functionality.

In order to implement our model numericaly we first vectorize the system of ordinary differential equations (ODEs) in a form of a statndard initial value problem

$$\frac{dy}{dt} = f(t,y) + Gy, \qquad 0 < t \leq t_f, \quad y(0) = y_0$$

We expand the solution vectors as follows

$$y = \big(V, n, [\mathrm{Ca}], [\mathrm{Ca_{er}}], [\mathrm{ADP}], [\mathrm{G6P}], [\mathrm{FBP}]\big)^T$$

with

$$V = \big(V_1, V_2, \ldots, V_{N^3}\big)^T,$$

$$
\begin{aligned}
n &= \left(n_1, n_2, \ldots, dn_{N^3}\right)^T, \\
[\mathrm{Ca}] &= \left([\mathrm{Ca}]_1, [\mathrm{Ca}]_2, \ldots, [\mathrm{Ca}]_{N^3}\right)^T, \\
[\mathrm{Ca_{er}}] &= \left([\mathrm{Ca_{er}}]_1, [\mathrm{Ca_{er}}]_2, \ldots, [\mathrm{Ca_{er}}]_{N^3}\right)^T, \\
[\mathrm{ADP}] &= \left([\mathrm{ADP}]_1, [\mathrm{ADP}]_2, \ldots, [\mathrm{ADP}]_{N^3}\right)^T, \\
[\mathrm{G6P}] &= \left([\mathrm{G6P}]_1, [\mathrm{G6P}]_2, \ldots, [\mathrm{G6P}]_{N^3}\right)^T, \\
[\mathrm{FBP}] &= \left([\mathrm{FBP}]_1, [\mathrm{FBP}]_2, \ldots, [\mathrm{FBP}]_{N^3}\right)^T,
\end{aligned}
$$

and

$$
f(t,y) = \left(f_V(t,y), f_n(t,y), f_{[\mathrm{Ca}]}(t,y), f_{[\mathrm{Ca_{er}}]}(t,y), f_{[\mathrm{ADP}]}(t,y), f_{[\mathrm{G6P}]}(t,y), f_{[\mathrm{FBP}]}(t,y)\right)^T
$$

The terms of $f(t,y)$ are the right hand side of the equations defined above for appropriate terms of $\frac{dy}{dt}$. The matrix G is the generalized system coupling matrix and Gy gives us the coupling values for a solution vector y. Notice for a given N, this amounts to solving $7 \times N^3$ degrees of freedom. Table 4.1 lists the degrees of freedom for desired values of N.

Table 4.1: System complexity DOF vs. N.

N	2	3	4	5	6	7	8	9	10
DOF	56	189	448	875	1512	2401	3584	5103	7000

Moreover, we find that we have a *stiff* problem, this means the number of necessary timesteps to approximate the solution is not driven by the error tolerances, but rather by the stability of the problem, hence we need a stiff ODE solver to correctly handle the system. In our case this arises because we wish to look at a solution interval on laboratory timescale of about 8 minutes while interested in activity system activity with 1 ms resolution. To solve this vectorized ODE system in Matlab, we assign a function handle $F(t,y) = f(t,y) + Gy$ and subsequently call Matlab's `ode15s`, a stiff ODE solver.

We observe that for $N = 2$ on an interval of 0 to 500 000 ms the solution matrix contains about 40,000 to 60,000 entries. Apparently, a default call to Matlab's `ode15s` is intended for much smaller systems of equations with maybe 10 to 100 equations. Running a Matlab profiler on the code, we found two issues in `ode15s` affecting performance. We proceeded to modify the `ode15s` function to fit and optimize the performance of our simulation.

We discovered a particular line in `ode15s` relating to memory management and its particularly inefficiency for our simulation. We call this the inadequate `chunk` size mistake. The particular line is:

```
chunk = min(max(100,50*refine), refine+floor((2^11)/neq));
```

This `chunk` is used immediately after in allocating the output variables `tout`, `yout`, and `kvec` as in

```
tout = zeros(1  ,chunk);
yout = zeros(neq,chunk);
kvec = zeros(1  ,chunk);
```

and more importantly used to reallocate these vectors

```
tout = [tout, zeros(1   ,chunk)];
yout = [yout, zeros(neq,chunk)];
kvec = [kvec, zeros(1   ,chunk)];
```

These lines are called at every timestep when the currently allocated solution vector size has been filled. Here `neq` is the number of equations and `refine` is either 1 or a value passed into `ode15s` using the `'Refine'` parameter for `odeset`, a function used to set parameters to a general Matlab ODE solver.

For any typical user, passing this `refine` seems highly obscure, unlikely and its use not emphasized in either the help page or comments of `ode15s`. Hence, we assume `refine=1` for all practical usage. We see $2^{11} = 2048$. So putting it all together, `max(100,50*refine)` always evaluates to 100 and `refine+floor((2^11/neq))` evaluates to 37 for $N = 2$, 10 for $N = 3$, 4 for $N = 4$, 2 for $N = 5$, and finally 1 for $N \geq 6$. Hence, this is a poor memory allocation scheme when our expected solution vectors are on the order of 40,000 to 60,000 entries. For $N \geq 6$ we are unnecessarily reallocating memory at every timestep! A quick fix we implemented, just assign `chunk=65536` or some similarly power of 2 so memory is allocated once or twice in total.

Another particular performance issue is `ode15s` saving the gradient approximations of the system at every timestep. This is completely unnecessary in our simulation since we have absolutely no intention of using the approximated gradients in post-processing. The problematic line is:

```
dif3d = cat(3,dif3d, zeros(neq,maxk+2,chunk));
```

Notice that `dif3d` is a three dimensional array. Recall that `neq` is $7N^3$, which is 875 for $N = 5$. Note that `maxk` is the maximum order of the solution, so usually 5 for `ode15s`. Moreover, the allocation of `dif3d` is iterative with chunk size, hence suffers from the same problems mentioned above. For $N = 10$ then `dif3d` contains $7{,}000 \times 7 \times 50{,}000 = 2{,}450{,}000{,}000$ doubles. Multiplying by 8 bytes and dividing by 2^{30} bytes per GB, we find that this is about 18.25 GB of memory to store this one variable we never use! This could only possibly run on some high performance machine like the tara cluster, where each node has 24 GB of memory. Hence, we simply eliminated all occurrences of `dif3d` completely from our modified `ode15s` function. But, one must be careful as this quantity is passed into the `odefinalize` helper function on two occurrences on a line like this:

```
olver_output = odefinalize(solver_name, sol,...
                           outputFcn, outputArgs,...
                           printstats, [nsteps, nfailed, nfevals,...
                                        npds, ndecomps, nsolves],...
                           nout, tout, yout,...
                           haveEventFcn, teout, yeout, ieout,...
                           {kvec,idxNonNegative});
                          %{kvec,dif3d,idxNonNegative});
```

The last line in this call, which is now commented out, is the original one, which was replaced by the second-to-last line in our code. This demonstrates which line to modify in `ode15s` to eliminate the use of `dif3`. The two references in `odefinalize` to be commented out look like:

```
%[kvec,dif3d,idxNonNegative] = deal(interp_data{:});
...
%sol.idata.dif3d = dif3d(:,1:maxkvec+2,1:nout);
```

These two simple fixes are greatly beneficial for running the simulation for large systems, i.e., $N > 5$.

5 Numerical Results

5.1 Performance Study

Following modfications to Matlab's `ode15s`, we ran a performance study comparing the simulation before and after the memory management modfications for a single trial run. A trial solving the system of differential equations from $t_0 = 0$ ms to $t_f = 500{,}000$ ms with a 50% slow and 50% fast cell distribution and resolutions from $N = 2$ to $N = 10$ is used. Table 5.1 lists the results of the performance study before and after memory modifications. The results indicate approximately 30% to 50% decrease in wall clock time of the `ode15s` method for the simulation for every N in the study. These studies were performed on the cluster tara in the UMBC High Performance Computing Facility (`www.umbc.edu/hpcf`). The tara cluster consists of 82 compute nodes, each with two quad-core Intel Nehalem processors and 24 GB of memory.

The parameter studies necessary for the application simulations require several runs of the code. For small values of N with run times of minutes, one Matlab code with a for loop is a sensible way to accomplish the entire parameter study. But for larger values of N, when each run takes over an hour, it is effective to take advantage of the fact that the cluster tara consists of 82 compute nodes, thus allowing for several runs to take place simultaneously. At the same time, communications with the clients brought out that it is sufficient to consider 11 trials for the larger values of N, thus making runs even for $N = 10$ feasible if 11 nodes in tara are available.

5.2 Application Simulations

The results in this section were obtained using a $5 \times 5 \times 5$ cells, in an islet containing a varying fraction of fast cells and differing levels of coupling from uncoupled through weak coupling to strong coupling.

Slow and Fast Bursting Cells Figures 5.1 (a) and (b) show the trace of voltage over time in a slow and fast bursting beta cell, respectively. The slow cell has a longer burst plateau duration with rapidly varying voltage and longer interburst intervals between burst

Table 5.1: Run time in HH:MM:SS for original vs. modified code.

N	DOF	Original	Modified
2	56	00:00:51	00:00:51
3	189	00:01:37	00:01:18
4	448	00:05:41	00:03:19
5	875	00:21:32	00:13:07
6	1512	01:02:44	00:34:31
7	2401	02:40:18	01:25:38
8	3584	06:22:46	03:46:10
9	5103	14:09:50	08:37:59
10	7000	27:36:36	18:32:40

plateaus which combine for an average burst period of 3.98 minutes, whereas the fast cell exhibits shorter burst plateau durations, shorter interburst intervals, and an average burst period of 0.39 minutes (24 seconds). These are the dynamics for slow and fast cells regardless of the fraction of fast cells in the islet because the islet cells are uncoupled.

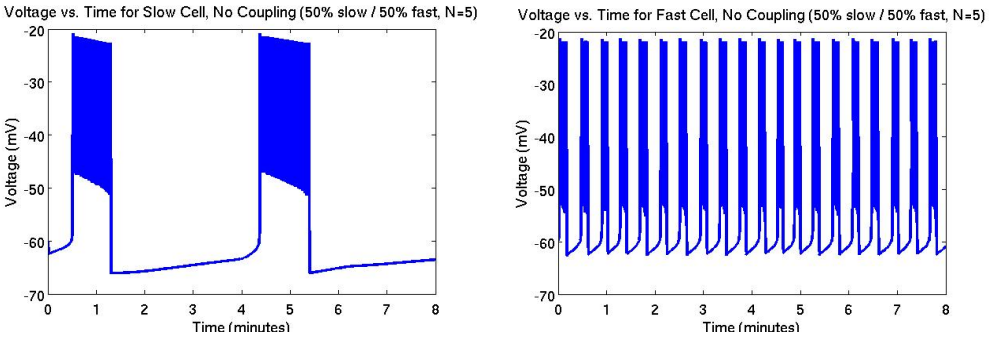


Figure 5.1: Islet with no cell coupling and 50% slow and 50% fast cells. Voltage vs. time of (a) a slow cell, (b) a fast cell.

Coupled vs. Uncoupled Cells Weakly coupling the beta cells in the islet creates a modest impact on the voltage traces of equal numbers of fast and slow cells. In Figures 5.2 (a) and (b), 50% slow and fast cells are coupled with a strength of 100 pS. The slow cell has a shorter interburst intervals but similar burst plateau durations leading to shorter burst periods. The burst plateau has a noticeably different structure with decaying amplitude of the oscillations. The fast cell has a dramatic change in both its burst plateau duration and interburst interval yielding much longer burst periods compared to Figure 5.1 (b). However, the nature of the fast cell occasionally comes through with a short burst (e.g., at minutes 1 and 6) and even a blunted burst just after 6 minutes. The amplitude of the burst oscillations in the fast cell grows during the burst in opposition to the that of the slow cell.

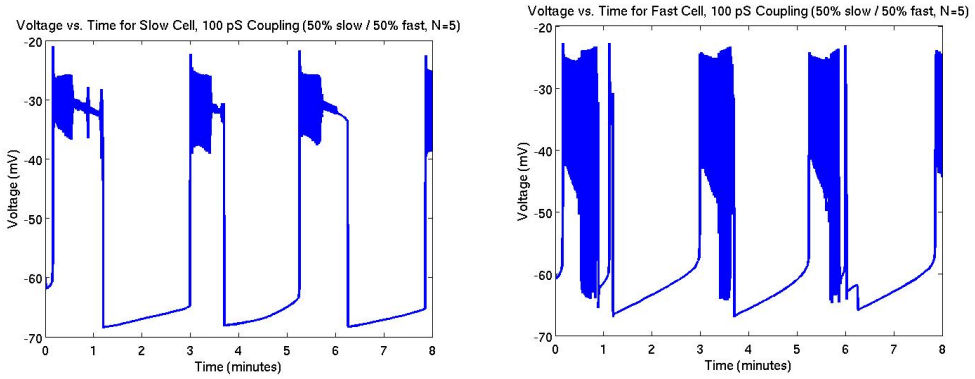


Figure 5.2: Islet with 100 pS cell coupling and 50% slow and 50% fast cells. Voltage vs. time of (a) a slow cell, (b) a fast cell.

Bursting Period vs Fast/Slow Distribution With stronger coupling, the voltage traces of slow and fast cells become effectively identical, but the fraction of fast cells effects the burst periods. In Figure 5.3 (a), we show the voltage trace of a cell from an islet with equal fraction of slow and fast cells. The coupling of 1000 pS is strong enough that all cells are synchronized. The burst period is shorter than that of the weak coupling case, showing that the fast cell has increased its influence in the islet over the weak coupling case where both slow and fast cell burst period appear closer to the uncoupled slow cell burst period. When we decrease the fraction of fast cells in the strongly coupled islet the cells remain synchronized but the average burst period for an islet cell drops, signifying the influence of the greater proportion of slow cells present in the islet. This is shown in Figure 5.3 (b) for an islet with 30% fast cells.

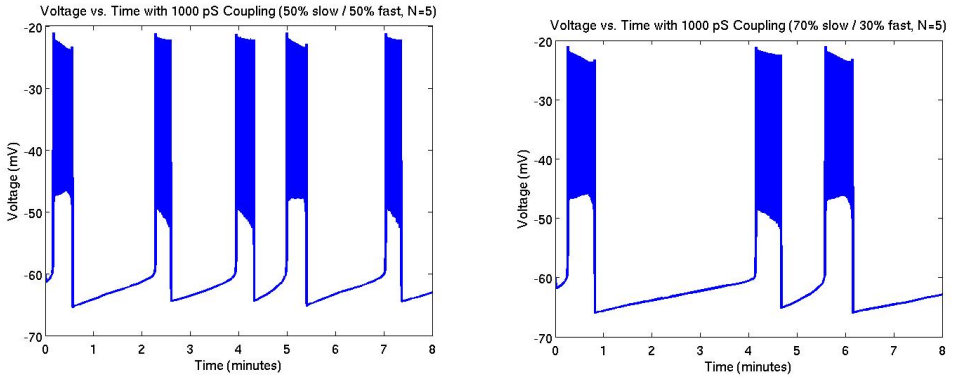


Figure 5.3: Islet with 1000 pS cell coupling. Voltage vs. time for (a) 50% slow and 50% fast cells, (b) 30% fast and 70% slow cells.

Burst Period vs. Fast Cells vs. Coupling However, decreasing fraction of fast cells does not always yield a slower average burst period. For intermediate coupling there exists a maximum average burst period for a nonzero fraction of fast cells. In Figure 1.1 for weak

(200 pS) and strong (1000 pS) coupling, we see a monotonic decrease in average burst period with increasing fast cell fraction. However, with intermediate coupling of 600 pS, we see a maximal average burst period at about 20% fast cells This result is counter-intuitive and warrants further study.

Acknowledgments

This research was conducted during Summer 2010 in the REU Site: Interdisciplinary Program in High Performance Computing (`www.umbc.edu/hpcreu`) in the UMBC Department of Mathematics and Statistics, funded by the National Science Foundation (grant no. DMS–0851749). This program is also supported by UMBC, the Department of Mathematics and Statistics, the Center for Interdisciplinary Research and Consulting (CIRC), and the UMBC High Performance Computing Facility (HPCF). The computational hardware in HPCF (`www.umbc.edu/hpcf`) is partially funded by the National Science Foundation through the MRI program (grant no. CNS–0821258) and the SCREMS program (grant no. DMS–0821311), with additional substantial support from UMBC.

References

[1] American Diabetes Association. Facts of diabetes. `http://forecast.diabetes.org/webfm_send/103`, 2010.

[2] Richard Bertram and Arthur Sherman. A calcium-based phantom bursting model for pancreatic islets. *Bulletin of Mathematical Biology*, vol. 66, pp. 1313–1344, 2004.

[3] Richard Bertram, Arthur Sherman, and Leslie S. Satin. Metabolic and electrical oscillations: partners in controlling pulsatile insulin secretion. *Am. J. Physiol. Endocrinol. Metab.*, vol. 293, no. 4, pp. E890–900, 2007.

[4] Richard Bertram, Min Zhang, Paul Smolen, and Arthur Sherman. Bj_04a.ode. `http://www.math.fsu.edu/~bertram/software/islet/BJ_04a.ode`, 2004.

[5] Teresa Ree Chay and Joel Keizer. Minimal model for membrane oscillations in the pancreatic beta-cell. *Biophysics Journal*, vol. 42, no. 2, pp. 181–190, 1983.

[6] P. Rorsman and G. Trube. Calcium and delayed potassium currents in mouse pancreatic beta-cells under voltage-clamp conditions. *The Journal of Physiology*, vol. 374, pp. 531–550, 1986.

[7] A. Sherman. Contributions of modeling to understanding stimulus-secretion coupling in pancreatic beta-cells. *Am. J. Physiol. Endocrinol. Metab.*, vol. 271, no. 2, pp. E362–372, 1996.

[8] Paul Smolen. A model for glycolytic oscillations based on skeletal muscle phosphofructokinase kinetics. *Journal of Theoretical Biology*, vol. 174, no. 2, pp. 137–148, 1995.