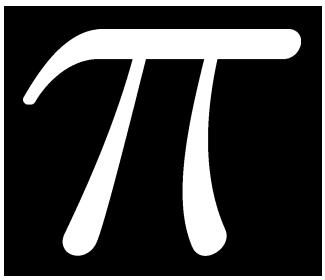


The
Pacific
Institute
for the Mathematical Sciences

http://www.pims.math.ca
pims@pims.math.ca

Report of the first

GRADUATE INDUSTRIAL MATH MODELING CAMP

PIMS MMC 1

Co-sponsored by:

The Natural Science and Engineering Research Council of Canada

and

The British Columbia Information, Science and Technology Agency

Editor: Huaxiong Huang, Pacific Institute for the Mathematical Sciences and University of British Columbia

FOREWORD BY THE PIMS DIRECTOR

Over the past two years, the Pacific Institute for the Mathematical Sciences (PIMS) has worked to provide new and innovative ways to further its goals, and high on its priority list has been the training of young mathematical scientists whether they are pursuing careers in academia or in industry. The **Graduate Industrial Mathematics Modeling Camp (GIMM)** was conceived to provide graduate students with learning opportunities and extensive training in the modern methods of applied mathematics, in order to prepare them for the PIMS annual **Industrial Problem Solving Workshop**.

The first GIMM was held at Simon Fraser University, May 24 - May 29, 1998. Forty Canadian graduate and near-graduate students came to Simon Fraser University to work with five mentors on various techniques for modeling industrial problems. The following report describes the proceedings of that innovative workshop. The students came from 12 universities across Canada: SFU, UBC, UCalgary, UAlberta, UVictoria, UToronto, McGill, UWO, UWaterloo, UQuebec at Sherbrook, UManitoba and Queens University. I am pleased to announce that the program was an UNQUALIFIED success. The mentors —as well as many who were around the SFU Math, Stats and Computer Science departments— surely felt and commented on the excitement and the impressive amount of energy and commitment of the students in the group.

Following the workshop, PIMS arranged for all the students and most mentors to join the second PIMS Industrial Problem Solving Workshop which was held this year at the University of Calgary, June 1 - June 6. Together with a mix of faculty from various Canadian and international universities, the students had a chance to try out their newly acquired skills on industrial problems ranging from detecting land mines to modeling the finances of building planes. The level of energy and preparedness of the graduate students from the GIMM was widely remarked by many of the experienced European participants.

Workshops like this one are only possible through the very hard and selfless work of many PIMS scientists. The concept of the training camp became a reality thanks to the efforts and determination of Dr. Keith Promislow from the SFU mathematics department and Dr. H. Huang, the PIMS industrial facilitator for British Columbia. I want to express my appreciation to both of them and to all the colleagues who have helped along the way. The next Industrial Mathematics Modeling Camp will be held in the Spring of 1999 at the University of Alberta, to be followed by the 3rd Industrial Problem Solving Workshop to be held at the University of Victoria. We will be looking forward for the continuation of this story of dedication and success.

Dr. Nassif Ghoussoub, Director
Pacific Institute for the Mathematical Sciences

EDITOR'S PREFACE

The inaugural **Graduate Industrial Mathematical Modeling Camp (GIMM)** of the Pacific Institute for the Mathematical Sciences (PIMS) was held at Simon Fraser University from May 24 - May 29, 1998. Close to 40 graduate and soon to be graduate students from universities across Canada participated in the modeling camp. They were guided by five mentors from industry and universities in Canada, the United States and the United Kingdom.

The objective of the camp is to provide a stimulating environment and a valuable learning experience in innovative techniques for approaching hard industrial mathematical problems for the students prior for the students priori to their participation in the annual PIMS **Industrial Problem Solving Workshop (PIMS-IPS)**. This year 27 of the participating students went on to participate in PIMS-IPS in Calgary, June 1 - June 6, 1998.

Format of the Modeling Camp

The format of the modeling camp followed the five-day format of PIMS-IPS so that students would know what to expect the subsequent week. On the first morning, the students were introduced to five problems presented by the mentors. The students then selected a problem they wanted to concentrate on for the remainder of the week. By the afternoon of the first day, these teams were working intensively to understand the problem. For the next three days, under the guidance of their mentors, the students devised models for attacking the problem, wrote computer programs, spent many hours in the library and often had impromptu meetings late into the night working with their teams.

The teams had a chance to exchange ideas at various breaks throughout the day. In the middle of the week, the students heard two invited presentations on industrial modeling. Les Scovell (Simons International Corporation) talked about the hard optimization problems their firm encounters when designing and building large installations (factories, mines, etc). Dr. Anthony Pierce, (Math, UBC) gave an overview of PIMS-IPS 1 and told the students what they could expect in PIMS-IPS 2. On the final day each group presented their results.

The culmination of the week's work by each team is found in this proceedings. In many cases, teams continued working on their problem after the camp; these efforts are also reported here.

Acknowledgments

PIMS is indebted to a number of individuals and organizations for the time and energy they put in to make the camp such a success.

- The main driving force behind the modeling camp was the organizing committee. This was the first such camp in Canada and required a huge commitment. In particular Dr. Keith Promislow (Math & Stats, SFU) who chaired the organizing committee was the real driving force behind the workshop. His dedication and organizational skills ensured that the workshop ran smoothly.
- The five mentors did a wonderful job selecting appropriate problems for the students. They would often stay up late with the students working into the night and definitely infected their teams with their energy and enthusiasm.
- I wish to thank the people who helped me produce these proceedings during and after the workshop. PIMS is thankful to all teams for putting considerable efforts into writing their reports.

- The PIMS office staff at UBC and SFU sites (Emma MacEntee, Tania Carpentier, Shannon Phillips, Thomas Uphill and Dorota Rygiel) and Michele Titcombe (Math, UBC) provided wonderful administrative assistance.
- PIMS acknowledges the Faculty of Science, the Department of Math & Stats Computer, the School of Computing Science, and the Institute of Applied Algorithms and Optimization Research all at SFU for providing many resources for the camp.

Structure of this Proceedings

We intend to use this proceedings to document the first PIMS graduate industrial mathematics modeling camp as well as to provide some insights on mathematical modeling in a workshop setting. A short introduction of the workshop, given by K. Promislow, is followed by five separate reports. Each report gives a statement of the problem and then the various approaches taken in attacking the problem. A list of the participants is included at the end of this proceedings. I would like to apologize in advance for any errors that have crept into any report.

Dr. Huaxiong Huang, Editor
Pacific Institute for the Mathematical Sciences

Contents

Chapter 1

Introduction

Mathematical modeling is the process of breaking apart a physical problem or process into its irreducible parts, making a mathematical cast of each piece, and attempting to reconstruct the resultant odd bits into a whole with some functionality. It is by nature a trial-and-error, iterative, brainstorming, sociable, frustrating, and at times, gratifying task. The goal of this camp was to provide graduate students in the mathematical sciences with a taste of this peculiar elixir; not unlike an intoxicating first visit to the country whose native language and culture one has studied from afar.

A total of 36 graduate and near-graduate students from 12 Universities across Canada participated in this inaugural event. Mentors from Kodak, University of Minnesota, University of Southampton, UBC, and SFU each presented a favorite modeling problem to the students, who through self-selection were divided into groups of seven to ten. For five intensive days, under the gentle guidance of their mentor, each team broke down and built up models of their process. The event culminated with the oral presentations and a written report from each group.

The modeling camp was one part of the PIMS Industrial Forum, and as such the students immediately continued their mathematical modeling efforts at the PIMS Industrial Problem Solving Workshop at the Univeristy of Calgary, June 1-6 (PIMSIPS 2). Together with a mix of faculty from various universities they worked in teams on problems straight from industry.

As a whole the program was highly successful. The group had great excitement and energy; remarkable even to several long-time participants in such events. Each of the mentors went out of their way to remark on the considerable progress their team had made. Indeed, Chris Budd – an an member of the original Oxford Study Group, now the preeminent European Study Group and the organizer of a recent meeting in Bath, England – indicated that they may adapt the PIMS Forum style (mentored graduate student camp as warm-up for a mixed faculty-grad student industrial session) to their ongoing effort.

The written reports are compiled in the sequel. It is hoped that they will provide some insight into the process of model building and guidance for those who endeavor to establish a mathematical modeling camp or classroom course.

Keith Promislow
Chair, Organizing Committee
Department of Mathematics and Statistics
Simon Fraser University

Chapter 2

Traffic Signals

Participants: L. Goddyn (Mentor), A. Amariei, D. Gaur, K. Hare, K. Kwok, J. Madden, S. Mitrovic-Minic, R.W. Tse

PROBLEM STATEMENT: The problem is one of sequencing the cycle of a traffic light at a large intersection. Certain pairs of traffic movements are forbidden to occur simultaneously (eg. East-North left-hand turn time interval should be disjoint from the West straight through time interval). The problem is to find a sequencing of the lights of shortest possible duration avoiding such forbidden conflicts. This problem is closely related to the computation of a difficult and unexplored graph parameter called "star-chromatic number".

2-1 The Problem

The problem is one of specifying an efficient sequence for traffic lights at a major intersection. We make the following simplifying assumptions regarding the traffic flow at this intersection: that it is stable and that the intersection is isolated from other major intersections. These assumptions eliminate from consideration intersections at which traffic flow changes a great deal over a short period of time and those intersections whose traffic flow is strongly influenced by the traffic flow at nearby intersections. A reasonable example of such an intersection would be an intersection of suburban arterial roads, which tend to have few signals on them except where they intersect with other arterial roads. We would like to determine a cyclic patter for the traffic lights that maximizes throughput of cars and minimizes wait time, while disallowing combinations of traffic flows that would tend to cause accidents due to conflict of right-of-way.

We construct a graph-theoretic model for the problem. Each vehicle enters an intersection from one of k directions (e.g., **N**orth, **S**outh, **E**ast, **W**est), and wishes to leave in one of these k directions. We disallow U-turns, so there are $k(k-1)$ such pairs. Each of these pairs (e.g.:N-W, E-W, E-S) is called a *transition*. Certain pairs of transitions are forbidden to occur simultaneously, so as to avoid conflict of right-of-way. For example, an E-S left-hand turn would be forbidden during a S-N green light. We represent the forbidden-pair information with an undirected graph, where transitions correspond to vertices and two vertices are joined by an edge if and only if the corresponding transitions are not allowed to occur at the same time. When formulated in this fashion, the problem is closely related to the problem of coloring vertices in a graph, where two adjacent vertices (ones which are connected by an edge) are not allowed to be assigned the same color.

If we assume that the traffic flows from each approaching road are uniform, the problem reduces to determining the *star-chromatic* number of a graph. In the case of non-uniform traffic flows, we associate a weight with each node, corresponding to (for example) the number of cars per cycle which pass through the intersection on the associated path. The problem is now one of arranging the arcs on a circle of minimum radius such that arcs corresponding to the adjacent nodes do not intersect.

In Vancouver, only a restricted set of transition sequences are seen in practice. Generally, a left-turn signal appears before the straight-through green light. However, in our problem we make no restrictions on the ordering of transitions.

We will also impose more simplifying (and less realistic) assumptions. We suppose that there are no sensors in the road way to test for the presence of cars, so that the traffic signals' pattern will not change in response to current conditions. All allowed transitions are explicitly signaled (so that, for example, we do not allow right turns into traffic when the signal is red). We further decree that the signals shall change according to a fixed cyclic sequence having a fixed period.

In each cycle, each transition v shall be allotted one contiguous time interval $I(v)$ of duration $P(v)$, during which the traffic signal permitting v is green. We may view $I(v)$ as an arc of length $P(v)$ in a circle S. Thus our problem can be formulated as follows:

Given a graph G with Vertex weights P, our problem is to find a mapping I from the vertices $V(G)$ to the set of intervals of a circle S, such that

1. *the interval $I(v)$ has length $P(v)$,*
2. if *uv is an edge of G then $I(u)$ and $I(v)$ must be disjoint,*
3. *the circumference of S is as small as possible.*

We shall call the minimum possible circumference of S the *weighted-star-chromatic number* of (G, P). Some study has been made of the star-chromatic number in the case where P is uniform. However, we are not aware of any algorithm for estimating the star-chromatic number, even for this special case.

This report is organized as follows. In section 2-2, we will describe the methods which we have used to solve the problem. In section 2-3, we will present and discuss some results for the techniques used. In section 2-4 we will present our conclusions, and in section 2-5 we will comment on the ways in which this work could be extended to make it more applicable to real-world problems.

2-2 Methods

We investigated three different approaches, as follows:

- Using local search to improve an initial feasible solution. Most of the work in this approach was in the formulation of the linear program which computes the next move to be made.
- Two slightly different greedy algorithms.
- Calculating the optimal solution using a commercial C++ optimization library. We modeled the problem as a set of the disjunctive constraints which have to be satisfied.

The first two approaches try to find the circle of minimum circumference, whereas the greedy algorithm tries to fit the arcs on the shortest possible interval. These two approaches correspond to calculating the weighted star chromatic number and calculating the weighted chromatic number.

The reason for investigating approaches other than calculating the optimal value directly is that calculating the chromatic number is known to be NP-hard and thus the number of steps required to calculate the optimal solution goes up exponentially with the size of the problem. Since calculating the star-chromatic and weighted star-chromatic numbers are clearly problems that are at least as hard as calculating the chromatic number, it is not considered feasible in general to find optimal solutions. Thus, while we calculated optimal solutions where possible so as to determine the effectiveness of our heuristic approaches, these heuristics are not expected to find the optimal answer in general.

2-2.1 Local search using Linear Programming

The problem of minimizing the signal plan cycle (and maximizing intersection throughput) can be solved by finding the weighted star-chromatic number of the conflict graph. In this section we present a local search heuristic for calculating the star-chromatic number of a graph.

Given (G, P) and any orientation $\vec{G}$ there is a linear program LP (to be described later) whose solution is an upper bound of the star-chromatic number of G.

Our goal is to find an orientation of $\vec{G}$ or G which minimizes or nearly minimizes the upper bound of the star-chromatic number. The LP is feasible provided $\vec{G}$ has no directed circuits. Thus an initial solution is obtained by ordering the vertices of G and orienting all edges toward the larger endpoint. Upon solving $LP_{\vec{G}}$, another orientation $\vec{G}'$ is obtained by reversing some arc of $\vec{G}$. This arc is chosen from a list of candidate arcs either randomly or via a meta-heuristic such as tabu search (see below). The list of candidate arcs is obtained by analyzing the dual optimum of $LP_{\vec{G}}$. (Any arc e whose

dual variable for $w_e \leq x_e$ is non-zero is a candidate for reversing.) The new LP is obtained from $LP_{\vec{G}}$ by multiplying the appropriate column of M by -1.

The heuristic can be described as follows:

- Step 1: Find an initial solution, i.e., an initial graph orientation $\vec{G}$.
- Step 2: Find an upper bound of the star-chromatic number by using the linear program corresponding to $\vec{G}$.
- Step 3: Reverse the orientation of one edge in $\vec{G}$. Go to Step 2.

The choice of the edge to be reversed in Step 3 can be made in several different ways. One possibility is a simple random selection, after which the algorithm will continue from this new solution regardless of the solution's optimal value. The randomness of the choice can be constrained to get better solutions. Another possibility is to implement a tabu search algorithm which will guide the moves from a solution to another from $\vec{G}$ to $\vec{G}'$ in the following way:

- Calculate the upper bounds of the star-chromatic number (solve LPs) for all graphs each of which is made by reversing the orientation of a single edge from the list of candidate edges. These graphs comprise the neighborhood of the current solution $\vec{G}$.
- Move from the current solution $\vec{G}$ to the best neighbor $\vec{G}'$, i.e., to the neighbor with the smallest upper bound for the star-chromatic number.

The procedures were coded in C, using the `lp_solve` package at
`http://ucsu.colorado.edu/~xu/software/lp/lp_solve.html` with a *Maple* front end. For graphs having fewer than a few dozen edges, one thousand iterations of the process seems to be sufficient to produce an optimal or nearly optimal solution.

Linear Program for calculating an upper bound of the star-chromatic number

Given (G, P) we compute edge weights $w_{uv} = [P(u) + P(v)]/2$. For each orientation $\vec{G}$ of G we consider the LP with variables t and $\{x_e : e \in E(\vec{G})\}$:

$LP_{\vec{G}}$: minimize t subject to
(1) $w_e \leq x_e \leq t - w_e$, for each arc e of $\vec{G}$,
(2) $\sum_{e \in C} \mathrm{sgn}(C, e) x_e = 0$, for each circuit C of G.

Here $\mathrm{sgn}(C, e) = \pm 1$ depends on whether e is directed in the same or opposite direction as (a fixed orientation of) C.

Only a *flow-space* basis of $m - n + 1$ circuits C need appear in (2), where n and m are the numbers of vertices and edges of G. (The set of fundamental circuits of any spanning tree of G will suffice.) Hence the LP has $3m - n + 1$ constraints on $m + 1$ variables. Any feasible solution (x, t) of $LP_{\vec{G}}$ corresponds to a legal sequencing $y : V(G) \to S$ of green lights over a time circle S of circumference t. We obtain y form x by solving $yM = x$, where M is the constraint matrix of (2), and reducing y modulo t.

Any solution x of (2) is called a *tension* or *potential*. Kirchoff and others studied potentials in the context of electrical networks. One can derive from their work that the minimum possible

circumference for a given (G, P) is exactly

$$\min_{\vec{G}} \min\{t : (t, x) \text{ satisfy (1) and (2)}\}$$

Our strategy here is to seek an orientation $\vec{G}$ of G which minimizes or nearly minimizes the value of $LP_{\vec{G}}$. The method is described in the previous section.

2-2.2 Greedy Algorithm

The basic idea of these heuristics is to pick the 'most problematic' vertex to add to the partial solution at all times. The idea is that the vertices which have the greatest number of conflicts, largest weights, etc. should be added early rather than late. If they are added late in the process, the program may not be able to overlap the vertices' associated segments with any other, and thus large intervals will be appended to the end of the interval. A sketch of the algorithms follows: Pick the first vertex with (1) largest weight and (2) highest degree.

The difference between the two greedy algorithms is in the method of selecting the node to be placed at each iteration. The priorities for vertex selection for Method 1 are as follows:

1. Vertex with the largest number of conflicts with vertices already placed
2. Vertex with the largest weight
3. Vertex with the highest degree

Method 2 is similar to Method 1, the only difference being that the priorities used in Method 1 are ordered 2, 1, 3.

Once a vertex has been selected, its corresponding line segment then must be placed on the interval. We want to do this in such a way as to minimize the total length of the interval. Thus, we place the start of the vertex's segment as close to zero as possible, and allow the segment to overlap with any non-conflicting segments.

To do this, we must avoid conflicts with segments that have already been placed. (Once a segment has been placed, it is not moved.) We try to fit the current segment after each conflicting segment. To maximize efficiency, we maintain a list of the conflicting segments, sorted in increasing order to the right endpoint of each segment. If the segment won't fit, we move on to the next conflicting segment and try again.

Using cliques

A clique in the conflict graph yields a lower bound on the length of the traffic light cycle. Specifically, since all members of a clique conflict with one another, the cycle cannot be any shorter than the sum of the weights of the elements of the clique.

This fact implies that one should employ a preprocessing step consisting of finding the clique of maximum weight and placing the elements of this clique one after another. We note that the clique of maximum weight and the clique containing the largest number of vertices may not be the same, since the vertices are weighted. However, finding the maximum-weight clique is almost certainly NP-hard, since we know that finding a maximal clique, which appears to be easier, is NP-hard.

Nevertheless, a limited version of this preprocessing step may be both useful and feasible. If we restrict ourselves to finding the clique of maximum weight of three or fewer nodes, for example, this can be accomplished in $O(n^3)$ time by checking all combinations of three nodes, which takes $n(n-1)(n-2)$ steps. Since the greedy algorithms, for example, are also in $O(n^3)$ on dense graphs, doing this is 'free', asymptotically speaking, and yields a solution that is at least as good, and in general better, than a solution which starts from an arbitrary vertex. In the current implementation, we use a trivial form of this preprocessing by choosing the clique of maximum weight of size 1.

Some result on local optimality

In this section we will argue that the solution produced by the greedy algorithm cannot be compacted any more (or at least not in a straightforward fashion; we do not claim that the greedy algorithm yields an optimal answer). We will assume that the reader is familiar with the basic results about interval graphs (for a nice introduction see *Graph Theory* by West). Consider the solution obtained by the greedy algorithm. If we construct a graph in which the nodes are the intervals and edges correspond to overlapping intervals, then the graph is an *Interval Graph*. Interval graphs have *perfect elimination orderings*. This means that we can compute the shortest interval needed to represent the interval graph on the number line in polynomial time.

Without loss of generality we can assume that the shortest interval on which the interval graph can be laid out starts at 0. The greedy algorithm tries to fit the next interval at the earliest possible location. This constraint tells us that it is not possible to move the interval any more to the left, and thus reduce the total width. In fact it can be shown that the greedy algorithm computes the perfect elimination ordering. This implies that the solution obtained by the greedy algorithm cannot be compacted, and is therefore locally optimal.

Compressing the solution onto a circle

Given a solution on the interval, the following procedure places the solution on the circle. In the process, it tries to decrease the circumference of the circle from its initial value of the length of the interval. This is accomplished by allowing segments to "wrap around" the endpoints of the interval, which are now identified as a single point on the circle.

Case 1: Only 1 interval touches the boundary, say x.

(a) We save $S - \max(\text{intervalends} \neq x)$.

(b) Look at all intervals adjacent to x in the graph, and calculate $\min(start(y) - end(x)) \neq 0 \pmod S$[1]. For all y adjacent to x such that $start(y) = end(x)$, we put y into $\mathcal{A}$.

(c) For all $x \in \mathcal{A}$ repeat (b).

(d) Terminate when either the original x appears in $\mathcal{A}$ (in which case we have a tight circle, and we cannot compress), or $\mathcal{A} = \emptyset$, in which case we compress by the minimum of all the numbers calculated.

Case 2: If there is more than one interval then we calculate this value for each of them and compress by the minimum of all of them.

We repeat this process until no more compressions are possible.

[1] Note if we wrap around the circle n times, to get there we use $(start(y) - end(x))/n$.

2-2.3 Optimal Solution Using Ilog Solver

The problem of finding the *weighted star-chromatic* number of a graph can be formulated as the following constraint satisfaction problem:

In what follows we assume that x_i denotes the mid-point of interval i (which corresponds to the node i in the incompatibility graph). w_i denotes the weight of the node i (or the length of the arc associated with node i). The incompatibility graph is $G = (V, E)$.

Minimize y subject to

$$\frac{w_i + w_j}{2} \leq |x_i - x_j| \leq y - \frac{w_i + w_j}{2} \text{ for all } (x_i, x_j) \in E(G)$$

and

$$0 \leq x_i \leq y \text{ for any } i$$

We used the *Ilog Solver*[2] to find optimal solutions to the above mentioned set of constraints. We gratefully acknowledge Junas Adhikary for the help provided on the Ilog package.

2-3 Testing

There were two approaches considered for testing: creating a series of random graphs, which would allow us to characterize the overall performance of the various approaches and creating a series of graphs which corresponded to what we considered realistic intersections. Since the motivation for this project was to derive useful approaches for calculation efficient traffic signal cycles, rather than approaches for calculating the weighted star-chromatic number of a graph, we decided on the latter scheme.

The tests were constructed as follows: we created three classes of graphs, corresponding to 3-way, 4-way and 5-way intersections. Since we lacked experimental data, we assigned weights to the traffic flows (vertices) that seemed to accord with our empirical knowledge of traffic patterns. Each member of a given class of graphs differed from the other members of its class by an amount that depended on a single variable, which corresponded roughly to time of day.

We hoped to use these tests to determine two things. We wanted to measure the relative performance of the heuristics on graphs whose structure was reminiscent of actual traffic patterns observed in nature (as well as their performance relative to the optimal solution). We also wished to do some sensitivity analysis on the weights of the nodes, to determine, among other things, how these algorithms might be used in an adaptive traffic signal network.

Unfortunately, we did not have the time to collect sufficient data to come to any definite conclusions, or to engage in a great deal of analysis. Based on our limited results, however, it appears that the LP method and the two greedy algorithms have comparable results on the classes of graphs that we investigated. No algorithm had significantly better performance overall, although there were some differences in the quality of the solution in a few cases.

[2] A C++ library for Constraint Programming.

2-4 Conclusion

In this document, we modeled the problem of finding a signal plan for an isolated intersection as a graph theoretic problem. In particular, we have computed the weighted star-chromatic number of an undirected graph.

We tested the three approaches on almost real world data (constructed over a period of a day). We modeled intersections with 3, 4 and 5 roads converging. We generated 11 instances for each intersection size by varying the traffic flows in a systematic fashion. Optimal solutions computed using Ilog were used to rate the performance of the greedy and the local search approaches. Local search and greedy almost always found the optimal solution. However, given the size of the data set and the assumptions, we cannot comment on the robustness of the approach. It remains to be seen how the techniques would perform on real world data.

2-5 Future Work

In this document we have studied the case for an isolated intersection. The real world application is closer to the online version of the problem of finding the weighted star-chromatic number. In the *online version* the traffic flows change continuously. Ideally, what we would like to do is obtain a family of related solutions, which have the property that while the sizes of the intervals might change, the relative orderings will remain relatively stable. This would allow the traffic signal to adapt to traffic patterns without unduly confusing the drivers. We believe that our algorithms can be adapted to the online version of the problem.

It should be noted that the problem is NP-hard (Garey and Johnson). Furthermore, even finding approximations within a factor or $n^{1+\epsilon}$ is NP-hard (Arora et. al.). Not much is know about the special classes of graph for which the problem can be solved in polynomial time. It would be interesting to investigate the existence of a dichotomy theorem (similar to Schaefer's for satisfiability, Hell et al. for graph homomorphism). Given the abundance of the negative results for the problem, we advocate the use of greedy approaches tailored to the input data as a cost effective solution.

The local search method can be made complete (so that it would find solution if one exists) by using the *reserve search* technique (Avis and Fukuda) for enumerating the vertices of a convex hull. It remains to be seen if a complete algorithm based on reverse search would perform any better or worse in practice when compared to the current implementation based on tabu search. It should be noted that such an algorithm would take exponential time in the worst case.

Chapter 3

Optimal Policies for Queueing Systems

Participants: R. Kuske (Mentor), B. Bart, M. Titcombe, Y. Lucet, S. Kavousian, S. Jensen, A. Sheshnev, K. El-Yassini, M. Neagu

PROBLEM STATEMENT: There are several conceptually simple problems which could be studied. These include make-to-stock production systems in which costs are incurred in production and held inventory, with competition between processes for a server, and queueing networks in which customers require more than one type of service, with again competition for service. These problems can be studied using either stochastic processes or deterministic fluid flow models. One can start by studying small systems to gain some intuition and works his/her way up to larger systems. One can use simulations as well as analytical tools.

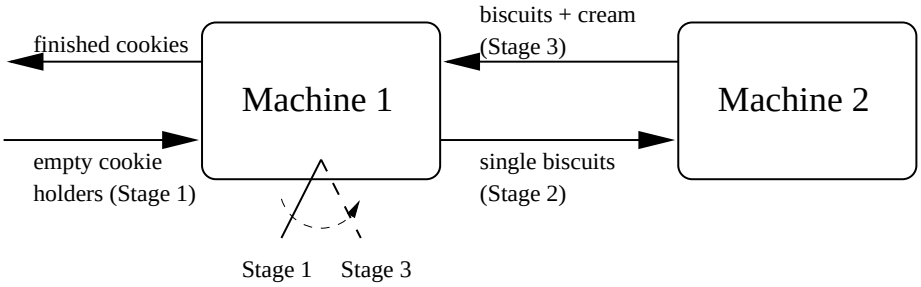


Figure 3-1: *The Oreo Cookie Queuing Model.*

3-1 Modeling a Queueing System - Simulation Approach

3-1.1 Introduction

The production of Oreo cookies can be modelled by a 3-step process. The Oreo cookie factory has two machines. The first machine adds a biscuit to a cookie in a progress and the second adds the cream filling. The process is analogous to the queuing system shown in Figure 3-1. Because machine 1 can service only one of stage 1 or stage 3 at any given time, it is equipped with a switch to define which service it is providing. Each stage is equipped with a queue which will begin to fill if the machine's service rate is slower than the arrival rate of the queue. We shall now define some quantities.

Definition 1 *For the Oreo cookie system, let Q_i represent the queue at stage i and let $Q_i(t)$ represent the number of partially completed cookies in Q_i at time t. Furthermore, let μ_i be the service rate of Q_i and let λ be the arrival rate to Q_1. The system is said to be* neutrally stable *if the all queue sizes are bounded above for all time.*

Stability is highly dependent on the algorithm we use to change the switch. It is known that if we use the optimal switching strategy, then the system is neutrally stable if

$$\frac{1}{\lambda} > \frac{1}{\mu_1} + \frac{1}{\mu_3} \qquad \text{and} \qquad \frac{1}{\lambda} > \frac{1}{\mu_2} \tag{3-1}$$

Unfortunately, to employ the optimal switching strategy, we must assume that we can constantly monitor the system and its queues. This would be a realistic model if a computer was monitoring the system.

But suppose that for reasons of affordability, job unions, etc. that Nabisco does not have such a monitor available. Instead, they will pay a human to change the switch. Hence, we introduce Mr. Christie, our factory worker hero who, of course, makes good cookies. Mr. Christie's switching algorithm is as follows.

loop
  drink coffee for t_1 minutes
  a: dummy statement
  change the switch
end loop

where the coffee break time t_1 is Mr. Christie's single parameter. There are two questions to be answered.

1. (*Nabisco's Problem*) If Nabisco wants to provide Mr. Christie with coffee breaks of t_1 minutes, then what size will Nabisco need to build the queues so that no cookies overflow a queue?
2. (*Mr. Christie's Problem*) If Mr. Christie runs a machine with specified queue lengths, then what is the longest time that Mr. Christie can have a coffee break without overflowing any of the queues?

It is clear that these questions are the converses of each other.

3-1.2 Analysis

Analysis of each possible step was found to be too difficult because there were too many cases to consider. In light of practicality, we have chosen the simulation approach. In our simulation, we let

$$\begin{aligned} \frac{1}{\mu_1} &= 3 \quad \text{minutes} \\ \frac{1}{\mu_2} &= 8 \quad \text{minutes} \\ \frac{1}{\mu_3} &= 6 \quad \text{minutes} \\ \frac{1}{\lambda} &= 10 \quad \text{minutes} \end{aligned}$$

One important simulation issue is how Mr. Christie handles partially serviced cookies when his coffee break ends. He could abort the service and leave it to be processed until later, but then (3-1) would not hold anymore. To ensure that (3-1) holds, we add dummy statement a to the algorithm.

a: wait until partially serviced cookie in machine 1 has finished

The simulation was carried out on a SPARC ULTRA 1. A graph of maximum queue size against time is shown in Figure 3-2.

3-1.3 Results

For all choices of t_1, the system reached an equilibrium state (i.e. the system was neutrally stable). The graphs of $Q_1(t)$ and $Q_2(t)$ are clearly linear with slopes 0.2000 and 0.1786, respectively. The graph of $Q_3(t)$ appears to be a region, rather than a line. Upon closer examination of this graph in Figure 3-3, $Q_3(t)$ is very jumpy. The slope of the peaks of $Q_3(t)$ is 0.2194 and the troughs drop by

Figure 3-2: *Queue size virsus coffee break time.*

Figure 3-3: *Queue size virsus coffee break time: a close-up.*

Figure 3-4: *Queue size virsus non-deterministic coffee break time.*

7% of the peaks from time to time. It has been suggested that dummy statement a is the cause of this occasional drop.

Given that Mr. Christie is only human, it might be fair to assume that his coffee break time is a random function. Figure 3-4 shows a graph of the maximum queue sizes if Mr. Christie's coffee break times are random subject to a binomial distribution. Notice that the sporadic nature of $Q_3(t)$ has ceased. The trendlines have slopes 0.2098, 0.1893 and 0.2255, for queues 1, 2 and 3, respectively- a 6% increase.

3-1.4 Conclusions

In a practical sense, running a simulation to determine the appropriate way to build a machine is an effective solution to the Oreo cookie problem. We have concluded that for regularly spaced coffee breaks, the maximum queue size is a linear function of the coffee break time, t_1. The maximum queue size almost remains linear for the random coffee break times. To justify the practicality of long coffee breaks, Nabisco may require Mr. Christie to monitor many cookie machines on the factory floor, thereby flipping many switches at interval t_1, rather than drinking so much coffee.

3-2 Queueing Problem for an Open System - Discrete Approach

3-2.1 Introduction

We consider a simple system on which we study different queueing strategies. In section 3-2.2 we consider the objectives of the company's management. Mathematical model of the above problem is defined in section 3-2.3. The transition diagram between different states of the system is described in section 3-2.4. In section 3-2.5 we state the necessary conditions for the existence of neutrally stable cycles. We also present results for various system parameters. The participants of this subgroup are Janez Ales, Shane Jensen and Yves Lucet.

3-2.2 Management Objectives

We consider a simple system devoted to the construction of an OREO cookie. Two servers are present, one for the wafer component of the cookie, and one for the creme filling. A total of three processes are needed, which are: (1) a bottom wafer being generated by Server I; (2) white filling being added by Server II; and (3) a top wafer being added by Server I. This is illustrated in Figure 3-5.

Since our first server must process two different jobs, a priority policy must be introduced. Three distinct priority policies are considered: *Job 1 first*, *Job 3 first* and *Job 1 and 3 weighted equally.* Based on intuition, we make two assumptions: (1) Job 1 and Job 3 are individually faster than Job 2 and (2) Job 1 and Job 3 together take more time than Job 2. The objective of the company's management is to minimize the processing time for a cookie.

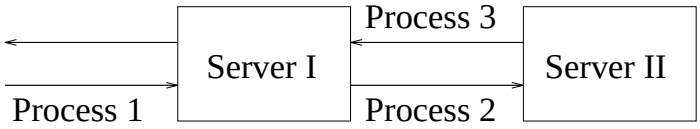


Figure 3-5: *Schematics of the model.*

3-2.3 Mathematical Model

In order to simulate the states of the above mentioned system we introduce some system parameters. Let λ be the *arrival rate* of the cookie holders to the Server I in a unit time (objects/time), let μ_i be the processing speed for Job i (objects/time), let u_i be the dedication of a server to Job i, and let x_i be the queue size of Job i (objects), for $i = 1, 2, 3$.

Assumptions made in section 3-2.2 lead to the following conditions imposed on system parameters:

$$\mu_2 \leq \min(\mu_1, \mu_3), \tag{3-2}$$

and

$$\frac{1}{\mu_2} \leq \frac{1}{u_1\mu_1} + \frac{1}{u_3\mu_3}. \tag{3-3}$$

To ensure stability of the system the time needed to process one object on Server I must be smaller than the interval time in which objects are arriving to the queue for Job 1. Also, the processing

Table 3-1: *State variables of the integral system.*

Queue	State 1	State 2	State 3	State 4	State 5	State 6	State 7
x_1	$\frac{\lambda}{\mu_1} - 1$	$\frac{\lambda}{\mu_2}$	$\frac{\lambda}{\mu_3}$	$\frac{\lambda}{\mu_1} - 1$	$\frac{\lambda}{\mu_3}$	$\frac{\lambda}{\mu_1 u_1} - 1$	$\frac{\lambda}{\mu_2} - \frac{\mu_1 u_1}{\mu_2}$
x_2	1	-1	0	$1 - \frac{\mu_2}{\mu_1}$	$-\frac{\mu_2}{\mu_3}$	1	$\frac{\mu_1 u_1}{\mu_2} - 1$
x_3	0	1	-1	$\frac{\mu_2}{\mu_1}$	$\frac{\mu_2}{\mu_3} - 1$	-1	$-\frac{\mu_3 u_3}{\mu_2} + 1$

time for Job 2 has to be smaller than the interval time of the arrivals. Hence, we get the following two conditions:

$$\frac{1}{u_1\mu_1} + \frac{1}{u_3\mu_3} \leq \frac{1}{\lambda}$$

for Server I and

$$\frac{1}{\mu_2} \leq \frac{1}{\lambda}$$

for Server II.

Three distinct priority policies are considered: *Job 1 first*, that is, $u_1 = 1$ and $u_3 = 0$, *Job 3 first*, $u_1 = 0$ and $u_3 = 1$, *Job 1 and 3 weighted equally*, $u_1 = \frac{\mu_1}{\mu_1+\mu_3}$ and $u_3 = \frac{\mu_3}{\mu_1+\mu_3}$,

3-2.4 States of the System

We describe all the possible states of the system in the state transition diagram in Figure 3-6. For example, when system is in idle state (State 0) we have all queues empty, that is, $x_1 = x_2 = x_3 = 0$, and State 7 corresponds to all queues being non-empty, $x_1, x_2, x_3 > 0$. As one will notice the number of possible states can grow exponentially in the number of queues considered.

3-2.5 Stability of Cycles

Under each of the three priority policies mentioned in section 3-2.3 three types of cycles are possible: stable, neutrally stable, and unstable, depending on the stability conditions (3-2) and (3-3).

Neutrally stable cycles appear when the system cycles between the states without queue growth or return to the idle state. In order for the system to reach idle state it has to have all queues empty for a non-zero time interval. That is, whenever the system gets a new arrival at the same time then all queues are emptied we consider a direct transition from State 3 to State 1, (see Figure 3-6). The necessary condition for the existence of a cycle is that the sum of the changes in the queue size has to add up to zero for every queue. All the state variables are listed in Table 3-1. In order to satisfy our necessary condition for a neutrally stable cycle, we must find all combinations of the columns which sum to zero, i.e., $A\vec{y} = \vec{0}$ with $y_i = 0$ or 1 where $\vec{y} = (y_1, y_2, ..., y_7)^T$ and the elements of matrix A are the state variables listed in the table. Diaz-Rivera, Armbruster, and Taylor [2] considered this integral system in a re-entrant queueing problem. Integer solutions to linear system are used to determine neutrally stable cycles for an entire range of arrival rates and processing speeds under the *equal weight* priority policy. Notice, that the above condition is only necessary, but not sufficient. Thus, not all integral solutions lead to cycles. Therefore, it is necessary to check if the given integral solution is indeed a cycle in the transition diagram.

Our objective was to characterise the periodicity between the different states this system can achieve for given arrival rates and processing speeds of our servers. Several examples of stable and unstable

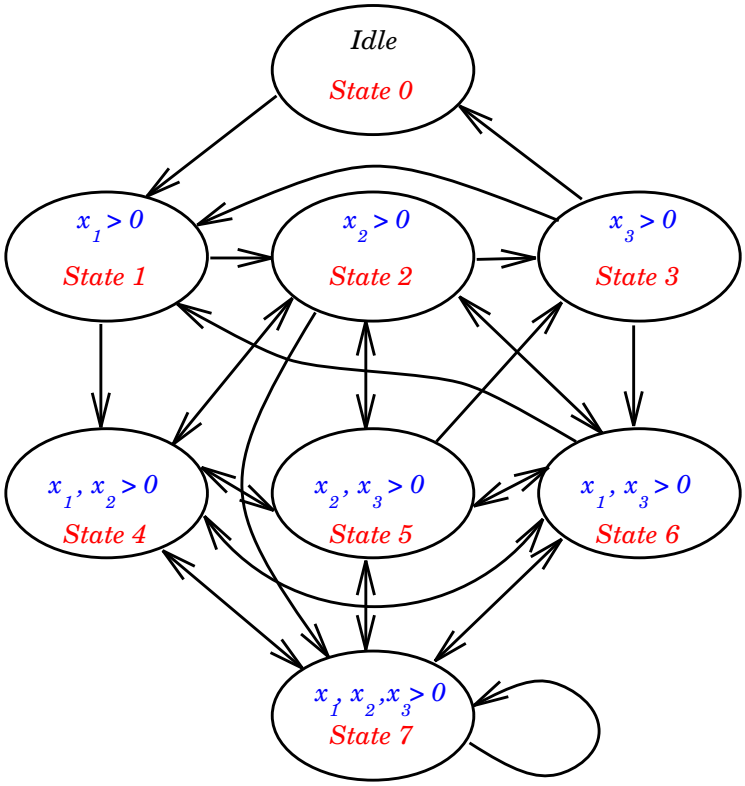


Figure 3-6: *Diagram of state transition with the arrows indicating the transitions between states. In addition to an idle state, seven non-idle states are possible for our system, each corresponding to a different combination of non-zero queue sizes.*

cycles were examined under each priority policy. This determines the behaviour of the system for given initial conditions, and some of the results are summarized in the following.

1. Neutrally stable system ($\lambda = 1$, $\mu_1 = 3$, $\mu_2 = 2$, $\mu_3 = 6$, and time is 1).
   For priority $u_1 = 0$ and $u_2 = u_3 = 1$, two solutions are
   $$\begin{aligned} \vec{y} &= (1,1,0,0,0,1,0)^T, \\ \vec{y} &= (1,1,1,0,0,0,0)^T. \end{aligned}$$
   For priority $u_1 = u_3 = 0.5$ and $u_2 = 1$, there is only one solution
   $$\vec{y} = (1,1,1,0,0,0,0)^T.$$
   For priority $u_1 = u_2 = 1$ and $u_3 = 0$, there are two solutions
   $$\begin{aligned} \vec{y} &= (0,1,1,0,0,1,0)^T, \\ \vec{y} &= (1,1,1,0,0,0,0)^T. \end{aligned}$$

2. Stable system ($\lambda = 1$, $\mu_1 = \mu_2 = \mu_3 = 4$ and time is $3/4$).

   For three different priorities tested: $u_1 = u_3 = 0.5$ and $u_2 = 1$; $u_1 = u_2 = 1$ and $u_3 = 0$; and $u_1 = 0$ and $u_2 = u_3 = 1$, there exists only one solution

$$\vec{y} = (1, 1, 1, 0, 0, 0, 0)^T.$$

3. Unstable system ($\lambda = 4$, $\mu_1 = \mu_2 = \mu_3 = 1$ and time is 2).

   For priority $u_1 = u_3 = 0.5$ and $u_2 = 1$, there are two solutions

$$\begin{aligned} \vec{y} &= (0, 0, 0, 1, 1, 1, 0)^T, \\ \vec{y} &= (1, 0, 1, 1, 1, 0, 0)^T. \end{aligned}$$

   For priority $u_1 = u_2 = 1$ and $u_3 = 0$, we have four solutions

$$\begin{aligned} \vec{y} &= (0, 0, 0, 0, 1, 1, 0)^T, \\ \vec{y} &= (0, 0, 1, 0, 1, 1, 1)^T, \\ \vec{y} &= (0, 0, 1, 1, 1, 1, 0)^T, \\ \vec{y} &= (1, 0, 1, 1, 1, 0, 0)^T. \end{aligned}$$

   For priority $u_1 = 0$ and $u_2 = u_3 = 1$, we also have four solutions

$$\begin{aligned} \vec{y} &= (0, 0, 0, 1, 0, 1, 0)^T, \\ \vec{y} &= (1, 0, 0, 1, 0, 1, 1)^T, \\ \vec{y} &= (1, 0, 0, 1, 1, 1, 0)^T, \\ \vec{y} &= (1, 0, 1, 1, 1, 0, 0)^T. \end{aligned}$$

These solutions correspond to cycles in the transition diagram from Figure 3-6. State i belongs to the cycle if the i^th component of the solution is 1, otherwise the state is not in the cycle.

3-2.6 Summary

We have considered a simple system devoted to the construction of an OREO cookie. Two servers are present in the system and a total of three processes are needed. The objective is to characterise the periodicity between the different states this system can achieve for given arrival rates and processing speeds of our servers. Since our first server must process two different jobs, a priority policy must be introduced. Three distinct priority policies have been considered: *Job 1 first*, *Job 3 first*, *Job 1 and 3 weighted equally*. Under each of these three priority policies, three types of cycles are possible: stable, neutrally stable, and unstable. Integer solutions to a linear system are used to determine neutrally stable cycles for an entire range of arrival rates and processing speeds under the *equal weight* priority policy. Several examples of stable and unstable cycles are examined under each priority policy.

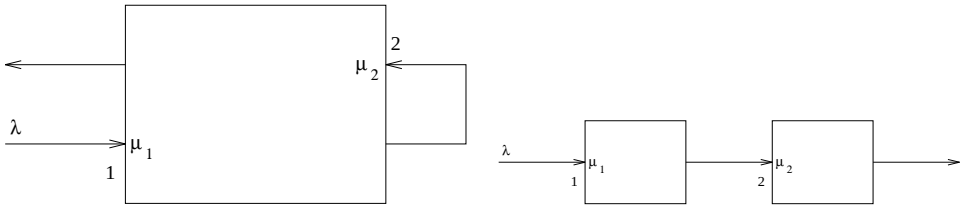
3-3 Single Server Re-entrant Systems - Continuous Approach

3-3.1 Introduction

Our subgroup of the Optimal Policies for Queueing Systems/Networks group examined the optimization based on a specific performance measure of a simple re-entrant manufacturing system using a continuous fluid model approach. The participants of this subgroup are Khalid El-Yassini, Mike Neagu, David Saunders, and Michèle Titcombe.

3-3.2 Statement of the Problem

We considered the system in Figure 3-7(a) consisting of a single server with two job classes, the second of which re-enters the server. The complication of this re-entrant system is the competition between job classes 1 and 2 at the server. We examined the optimization strategies based on the performance measure of minimizing the total holding cost of the system.



(a) *Single server re-entrant* (b) *Tandem queues*

Figure 3-7: *Flow diagrams of the two related systems: the single server re-entrant system (SSRS) and the tandem queues system (TQS).*

The parameters of the problem are

$$\begin{aligned} \lambda &= \text{rate of incoming jobs} \\ \mu_i &= \text{rate at which server processes job class } i \\ c_i &= \text{cost incurred by job class } i \text{ waiting to be served} \\ u_i &= \text{dedication of server to job class } i \\ x_i(t) &= \text{amount of job class } i \text{ waiting to be served,} \end{aligned}$$

where $i = 1, 2$. Feasibility of the system requires that

$$u_i \geq 0, \tag{3-4}$$

$$x_i \geq 0, \tag{3-5}$$

$$u_1 + u_2 \leq 1. \tag{3-6}$$

The stability of the system requires

$$\frac{\lambda}{\mu_1} + \frac{\lambda}{\mu_2} < 1. \tag{3-7}$$

Given the initial conditions $x_i(0)$, the arrival rate λ, the service rates μ_i and the costs c_i, for $i = 1, 2$, the problem is to determine the optimal controls $u_i(t)$ that minimize the total holding cost

$$\int_0^T (c_1 x_1(t) + c_2 x_2(t))\, dt \tag{3-8}$$

subject to (3-4)-(3-7). For this system, the job class amounts, x_i, satisfy the ordinary differential equations

$$\dot{x}_1 = \lambda - \mu_1 u_1 \tag{3-9}$$

$$\dot{x}_2 = \mu_1 u_1 - \mu_2 u_2. \tag{3-10}$$

In (3-8), T is the time at which both job classes have been depleted, i.e. $x_1 = x_2 = 0$. The fixed time T is possible in view of the stability condition (3-7).

Figure 3-7(b) shows a related system, consisting of two servers in tandem, that Avram et al. [1] considered in their study of fluid models of sequencing problems. In the case of the tandem queues problem, the stability condition is

$$\frac{\lambda}{\mu_1} < 1, \qquad \frac{\lambda}{\mu_2} < 1.$$

Since the simple tandem network has no competition of job classes in the server, they were able to develop the optimal policy for this network using Pontryagin's maximum principle. We have reproduced their tandem queues network results for Cases 3 and 4 in Figure 3-8, containing the switching curve of server dedication, and the corresponding optimal controls for the tandem queues problem in Table 3-2. In both the figure and the table, β is the slope of the switching curve, and is

$$\beta = \frac{c_1(\mu_2 - \mu_1)}{c_2(\mu_1 - \lambda)}. \tag{3-11}$$

Table 3-2: *Optimal controls for related tandem queue problem. (reproduced from Avram et al. [1]).*

Case	Conditions	$u(t)$
1	$c_1 \geq c_2$	$(1,1)$
2	$c_1 < c_2$, $\mu_1 \geq \mu_2$	$(0,1)$
3	$c_1 < c_2$, $\mu_1 < \mu_2$, $\frac{x_1(t)}{x_2(t)} < \frac{1}{\beta}$, $\beta = \frac{c_1(\mu_2-\mu_1)}{c_2(\mu_1-\lambda)}$	$(0,1)$
4	$c_1 < c_2$, $\mu_1 < \mu_2$, $\frac{x_1(t)}{x_2(t)} \geq \frac{1}{\beta}$	$(1,1)$

We claim that if the two systems, the single server re-entrant system (SSRS) and the tandem queues system (TQS), are in the same state (i.e., the same x_1 and x_2 at $t = 0$) and if the optimal controls $u_1(t)$ and $u_2(t)$ for the tandem queues problem satisfy (3-6) for some time $t_1 > 0$, then the optimal controls for the two problems coincide on the time interval $[0, t_1]$.

Now, we outline the argument behind this claim, first noting that if the dedications u_1 and u_2 minimize the cost integral (3-8) up to some time T, then these dedications also minimize the cost integral up to any time $t_1 < T$.

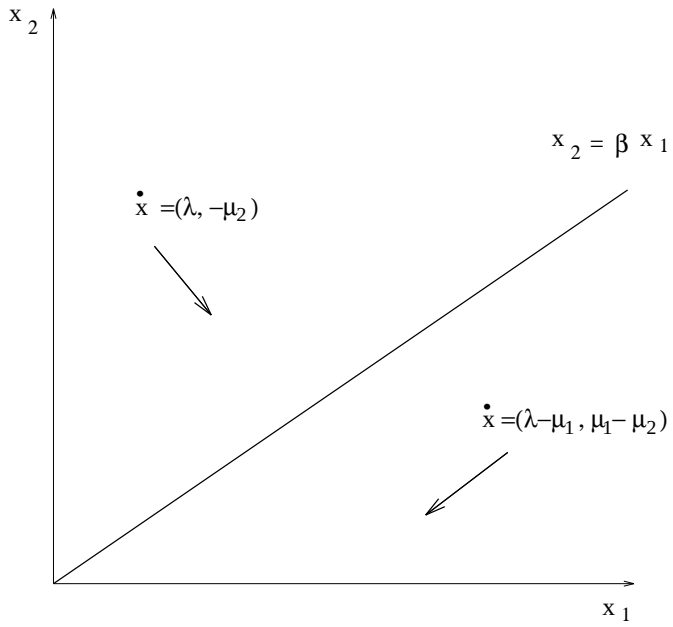


Figure 3-8: *Switching curve for cases 3 and 4 of the tandem queues network optimal policy (reproduced from Avram et al. [1]).*

Let u_1 and u_2 be optimal controls for the tandem queues system (TQS), satisfying (3-6) for all t in $[0, t_1]$. Also, let v_1 and v_2 be optimal controls for the single server re-entrant system (SSRS). We want to show that $u_1(t) = v_1(t)$ and $u_2(t) = v_2(t)$ for all t in $[0, t_1]$.

Since u_1 and u_2 minimize the cost integral for TQS up to time T, they also minimize the cost integral for TQS up to time t_1 from the note above. But $u_1 + u_2 \leq 1$ on the interval $[0, t_1]$, so u_1 and u_2 are acceptable controls for SSRS on $[0, t_1]$. Since the two systems are governed by the same set of differential equations in (3-9), and start with the same initial conditions, and any acceptable controls for SSRS are also acceptable for TQS, it follows that u_1 and u_2 minimize the cost integral for SSRS up to time t_1. But v_1 and v_2 also minimize the cost for SSRS up to time t_1. Assuming uniqueness of the optimal controls, it follows that $u_1 = v_1$ and $u_2 = v_2$ on the interval $[0, t_1]$.

Next, we describe a version of the claim which does not invoke uniqueness of the optimal controls. Suppose that u_1 and u_2 minimize the cost integral for TQS up to time T and satisfy (3-6) for all t in the interval $[0, t_1]$. Then, there exist optimal controls v_1 and v_2 for SSRS such that $u_1(t) = v_1(t)$ and $u_2(t) = v_2(t)$ for all t in $[0, t_1]$. The argument is similar to the one above.

To back up our claim, we have plotted the total holding cost from (3-8) versus $x_2(0)$, the initial queue size of job class 2, for various dedication strategies in the cases that satisfy (3-6). In all the plots, we fixed the arrive rate at $\lambda = 1$ and the initial queue size of job class 1 at $x_1(0) = 10$. The inset in each plot displays the dedication strategies in the x_1x_2 state space for given initial queue sizes $(x_1(0), x_2(0))$.

Figures 3-9 and 3-10 correspond to Case 1 of the tandem queue system, in which the cost at queue 1 is greater than or equal to that at queue 2. Our claim does not include Cases 1 and 4 in the tandem queue network, since those dedications violate (3-6) for the single server re-entrant network. The two dedication strategies that we used for Case 1 ($c_1 \geq c_2$) are: (i) serve queue 1 to depletion, then split the server so that queue 1 remains depleted while serving queue 2 to depletion, and (ii) serve queue 2 to depletion, then split the server so that queue 2 remains depleted while serving queue 1 to

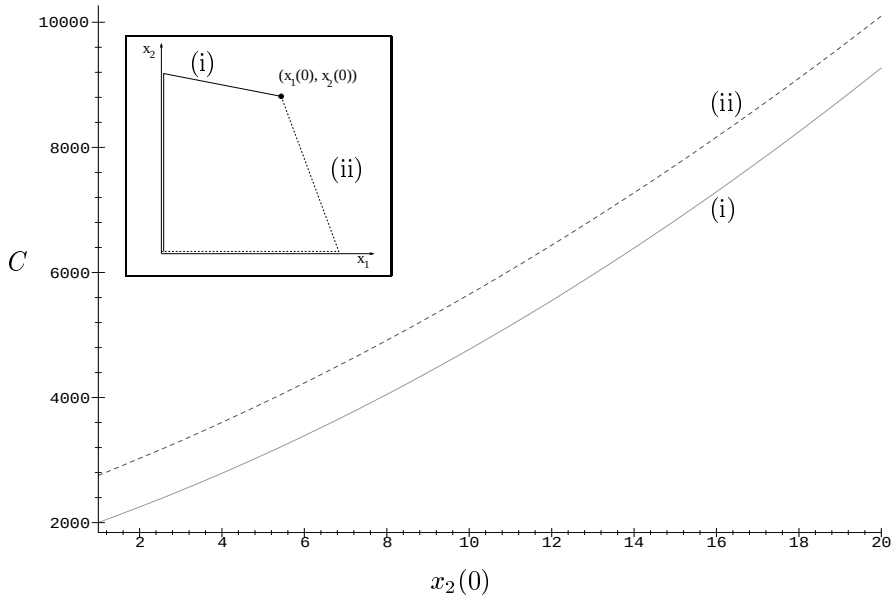


Figure 3-9: *Case 1(a): $c_1 > c_2$, $\mu_1 > \mu_2$. Total holding cost C versus initial queue size of job class 2, $x_2(0)$. Parameter set=$\{(c_1, c_2) = (10, 5)$, $(\mu_1, \mu_2) = (3, 2)$, $\lambda = 1$, $x_1(0) = 10\}$.*

depletion as well. Figure 3-9 indicates that the total holding cost is lower if the dedication strategy is to serve queue 1 to depletion, whereas Figure 3-9 displays a critical initial queue size of job class 2 of approximately $x_2(0) \approx 7$ below which it would be cheaper to serve queue 2 to depletion first.

Figure 3-11 corresponds to Case 2 of the tandem queue system. For Case 2, we used the same dedication strategies as in Case 1. The plot validates our claim for this case, since the total holding

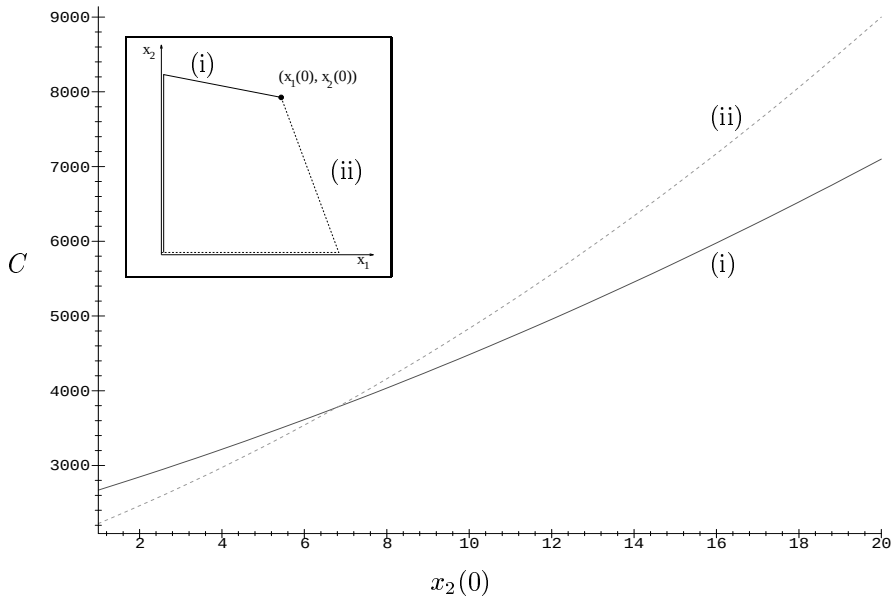


Figure 3-10: *Case 1(b): $c_1 > c_2$, $\mu_1 < \mu_2$. Total holding cost C versus initial queue size of job class 2, $x_2(0)$. Parameter set=$\{(c_1, c_2) = (10, 5)$, $(\mu_1, \mu_2) = (2, 3)$, $\lambda = 1$, $x_1(0) = 10\}$.*

cost for the single server re-entrant system is lowest using the dedication strategy of the tandem queues system for the same case, that is, to serve queue 2 to depletion first.

Figure 3-12 essentially combines Cases 3 and 4 of the tandem queue system, in which we have computed the total holding cost for three strategies: (i) serve queue 1 to depletion first, (ii) serve queue 2 to depletion first, and (iii) serve queue 2 until the switching curve $x_2 = \beta x_1$, then split the server dedication to stay on the line until both queues are depleted. Case 4 is included since the switching curve itself belongs to this case. For this plot, the initial point $(x_1(0), x_2(0))$ lies above the switching curve $x_2 = \beta x_1$, which corresponds to Case 3 of the tandem queue system. The figure validates our claim for this case, showing that the cheapest strategy is to serve queue 2 to depletion. Although we cannot use the strategy of the tandem queue Case 4 since the dedication violates (3-6), Figure 3-12 suggests that the dedication for Case 4 of our single server re-entrant system is to serve queue 2 to depletion first, as in our Case 3.

In Figures 3-13 to 3-15, we have plotted the results for simulations of the single server re-entrant system with exponentially distributed interarrival and service times. The total cost C is computed as an average over ten realizations for integer values of the initial queue length of job class 2, $x_2(0)$. We used the same parameter values as in Figures 3-9 to 3-11.

Comparing Figures 3-9 and 3-13 and Figures 3-10 and 3-14, we see that the results of the simulations do not agree with the predictions of the analysis in the case of $c_1 > c_2$. Assuming that we follow strategy (i) of our Case 1, then we empty queue 1 first and then divide the server so that it will attempt to keep queue 1 empty while also emptying queue 2. However, in the case of the simulation with discrete random arrivals, queue 1 will not remain empty. After one or more arrivals, there will be customers in the queue for a period of time during which costs are incurred. The simulation suggests that it may be better to dedicate more of the server to queue 1 when this occurs.

The simulation results in Figure 3-15 agree with the predictions of the analysis in the case with $c_1 < c_2$ and $\mu_1 > \mu_2$.

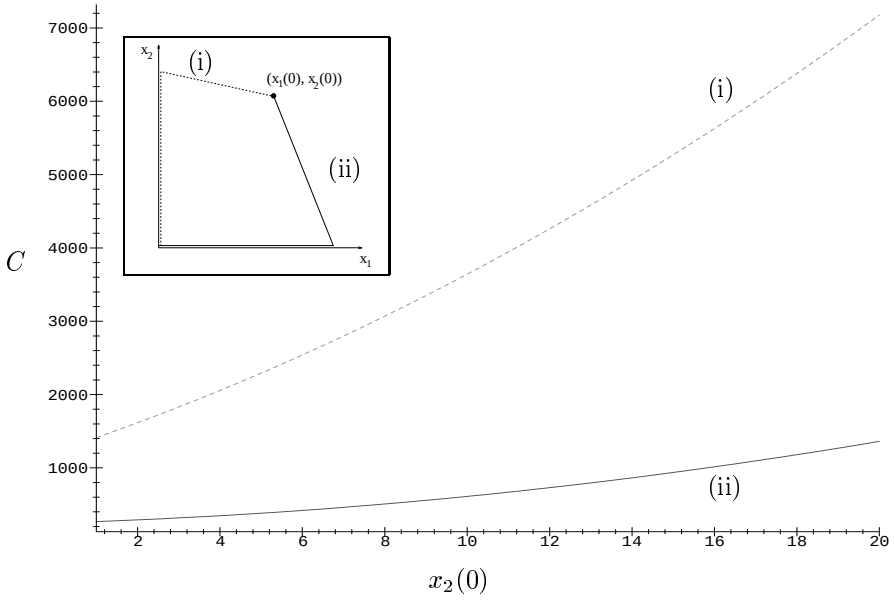


Figure 3-11: *Case 2:* $c_1 < c_2$, $\mu_1 > \mu_2$. *Total holding cost* C *versus initial queue size of job class 2,* $x_2(0)$. *Parameter set*=$\{(c_1, c_2) = (5, 10),\ (\mu_1, \mu_2) = (6, 3),\ \lambda = 1,\ x_1(0) = 10\}$.

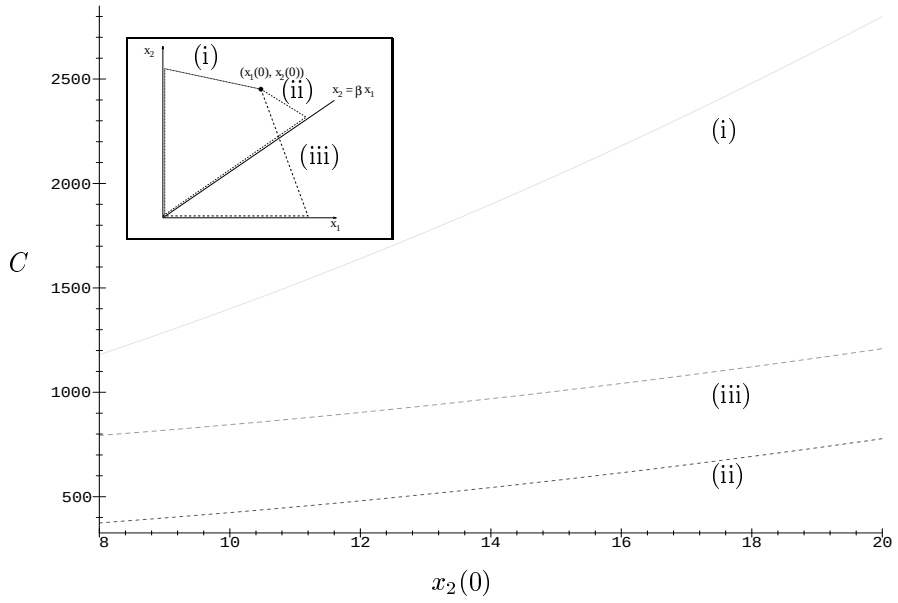


Figure 3-12: *Case 3:* $c_1 < c_2$, $\mu_1 < \mu_2$. *Total holding cost* C *versus initial queue size of job class 2,* $x_2(0)$. *Parameter set*=$\{(c_1, c_2) = (5, 10),\ (\mu_1, \mu_2) = (3, 6),\ \lambda = 1,\ x_1(0) = 10\}$.

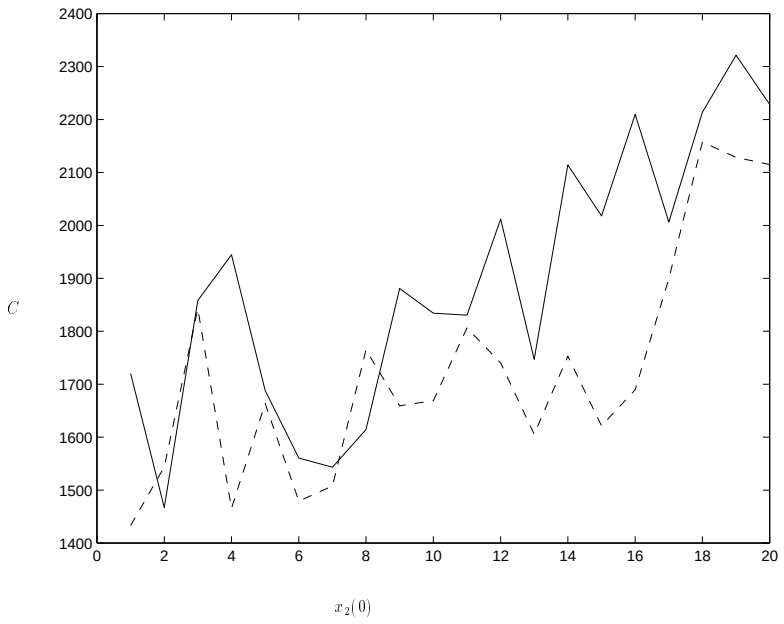


Figure 3-13: *Total holding cost* C *versus* $x_2(0)$ *for simulation with exponentially distributed inter-arrival and service times with the same parameters as Figure 3-9. Strategies: serve queue 1 to depletion first (solid), serve queue 2 to depletion first (dotted).*

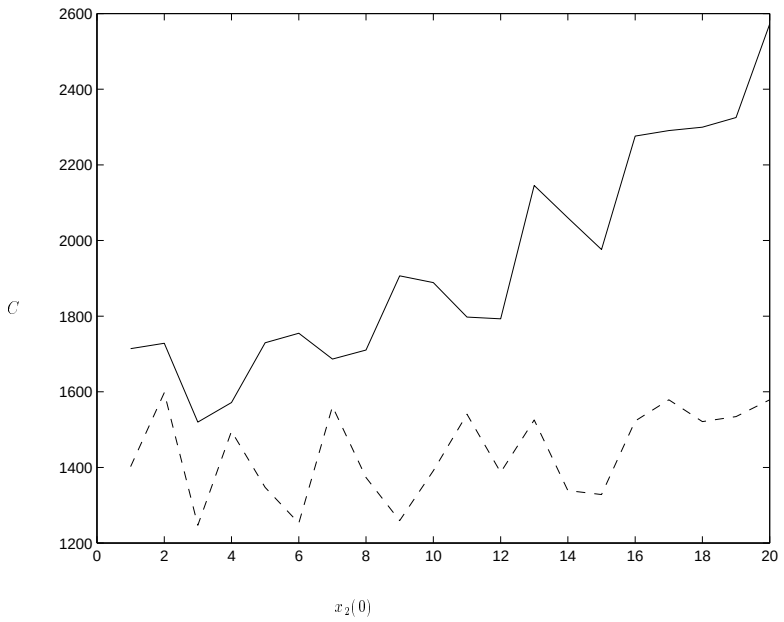


Figure 3-14: *Total holding cost C versus $x_2(0)$ for simulation with exponentially distributed inter-arrival and service times with the same parameters as Figure 3-10. Strategies: serve queue 1 to depletion first (solid), serve queue 2 to depletion first (dotted).*

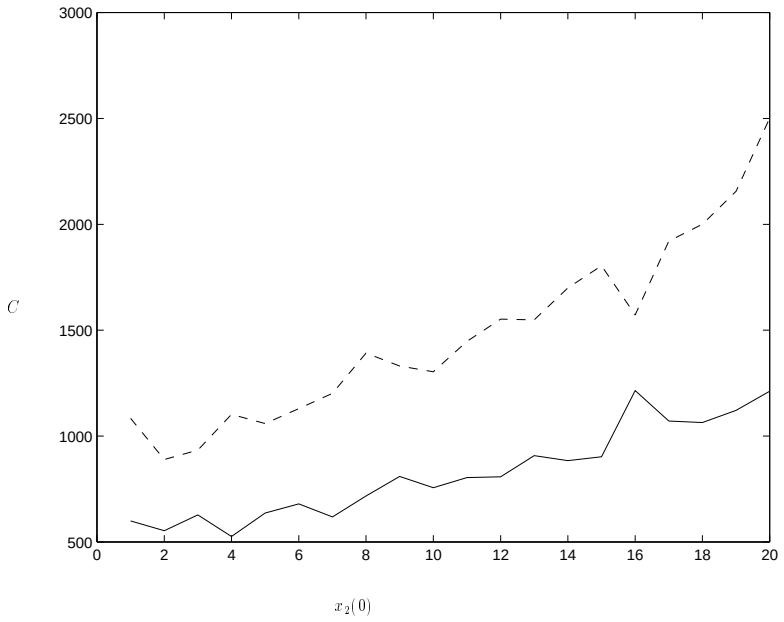


Figure 3-15: *Total holding cost C versus $x_2(0)$ for simulation with exponentially distributed inter-arrival and service times with the same parameters as Figure 3-11. Strategies: serve queue 2 to depletion first (solid), serve queue 1 to depletion first (dotted).*

Chapter 4

Synchrony of Pituitary Cells

Participants: Y.-X. Li (Mentor), N. Costanzino, A. Dawes, T. Ewen, S. Reinker, J. Williams

PROBLEM STATEMENT:

Calcium is the primary trigger for initiating a large variety of biological processes. Thus, hormone secretion by endocrine cells in the pituitary gland is triggered by an increase in intracellular calcium concentration. Such a calcium increase, induced by an extracellular messenger, is often oscillatory. This pulsatile calcium profile has been shown to correlate to the secretory activities in pituitary cells. However, individual cells only secrete a tiny amount of hormone. Therefore, coherent, synchronized secretory activity is of great physiological importance. Whether and how intracellular calcium oscillations in individual pituitary cells synchronize with each other remain obscure. Recent experiments suggest that plasma membrane electrical activities in neighboring cells can synchronize. The question remains if synchronized plasma membrane electrical activity causes synchrony in agonist-induced intracellular calcium oscillations between neighboring cells. The experimentalists cannot answer this question yet. We here answer the question by using mathematics.

4-1 Introduction and Statement of the Problem

The systems studied in mathematical biology are exceedingly complex. In modeling any complex phenomena, it is important to include only the mechanisms that play a fundamental role in the dynamics of the system. In this way, one hopefully obtains a simpler system that captures the 'essential features' of the more complex phenomena. Therefore, having some background information of the mechanisms involved is essential in order to properly model any system. With this in mind, we present some biological background in Section 4-1.1. Then in Section 4-1.2 we state the problem which we want to investigate. Finally, in Section 4-1.3 we propose a method by which we will approach the problem.

4-1.1 Background

Calcium is the primary trigger for initiating a large variety of biological processes. Thus, hormone secretion by endocrine cells in the pituitary gland is triggered by an increase in intracellular calcium concentration. Such a calcium increase, induced by an extracellular messenger, is often oscillatory. This pulsatile calcium profile has been shown to correlate to the secretory activities in pituitary cells. Individual cells however, only secrete a tiny amount of hormone. Therefore, coherent, synchronized secretory activity is of great physiological importance. It is not clear whether the intracellular calcium (IC) oscillations in individual pituitary cells synchronize with each other or how this process takes place. Recent experiments suggest that plasma membrane (PM) electrical activities in neighboring cells can synchronize. However, the question remains if synchronized plasma membrane electrical activity causes synchrony in agonist-induced intracellular calcium oscillations between neighboring cells. This question has not been answered by the experimentalists yet. In the next Section we pose the problem which we wish to study.

4-1.2 Statement of the Problem

The question of whether or not calcium oscillations of each individual pituitary cell can synchronize or phase-lock with neighboring pituitary cells can be answered by investigating the behavior of the inner workings of a single pituary cell. Each individual cell is a coupled system of a plasma membrane (PM) oscillator and an intracellular calcium (IC) oscillator as shown in Fig. 4-1. We thus pose the following problem: Can the intracellular calcium oscillations of each pituitary cell synchronize or phase-lock with those of neighboring cells? In answering this, we must first investigate the coupled IC-PM system and the coupled PM-PM system.

4-1.3 Method of Approach

Mathematically, the problem of two coupled pituitary cells can be simplified to the problem of a network of four coupled oscillators. This is because each cell is by itself a coupled system of a plasma membrane (PM) oscillator and an intracellular calcium (IC) oscillator. It is known that the PM oscillator activates the IC oscillator but the latter inhibits the former. The coupling between the two cells is diffusive, or nearest neighbor coupling, and is only between the two PM oscillators. Therefore, a sufficient condition is to show that since the IC oscillator in each cell is coupled to its own PM oscillator and since the two PM oscillators synchronize, the two IC oscillators should also be synchronized, although they do not talk directly to each other. Hence, we will concentrate our efforts on exploring whether the PM and IC oscillations are synchronized. Despite the outcome of this exploration, we will also briefly explore synchronization between cells.

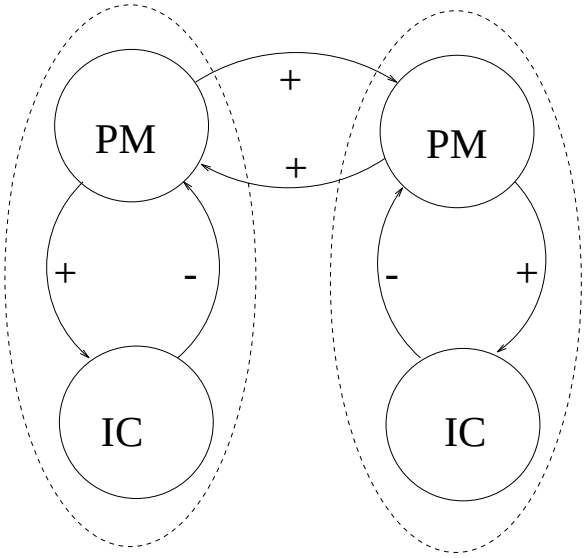


Figure 4-1: *Schematics of the coupled cells.*

4-2 The Two Models

To construct a model of the pituitary cell, we modified two existing models for mechanisms that occur within the cell. The first is the Morris-Lecar model, which has successfully been used in describing the potential difference across a membrane. To describe the intracellular calcium oscillations in the cytosol, we used the Li-Rinzel model. In the next two subsections, we briefly present these two models.

4-2.1 The Morris-Lecar Model

These equations describe the potential v and channel gating w across the plasma membrane.

$$C_m \frac{dv}{dt} = I_{app} - I_{Ca} - I_K - I_L \tag{4-1}$$

$$\frac{dw}{dt} = \lambda(v)(w_\infty(v) - w) \tag{4-2}$$

where I_{app} is the current externally applied to the membrane (for instance by an electrode), I_{Ca} and I_K are the calcium and potassium currents respectively, and I_L describes the cumulative effect of other small currents and is called the leak currents. The functional forms of the currents are given by

$$I_{Ca} = \bar{g}_{Ca} m_\infty(v)(v - v_{Ca}) \tag{4-3}$$

$$I_K = \bar{g}_K w(v)(v - v_K) \tag{4-4}$$

$$I_L = \bar{g}_L(v - v_L) \tag{4-5}$$

where $\bar{g}$ is the maximum conductance. The functions $m_\infty(v)$ and $w(v)$ are the activation probabilities which are typically sigmoidal curves, and determine the gating or channel opening of the cell.

Under certain parameter values the Morris-Lecar system exhibits periodic firing as can be seen in Fig. 4-2. Experiments have shown that the period is on the order of seconds.

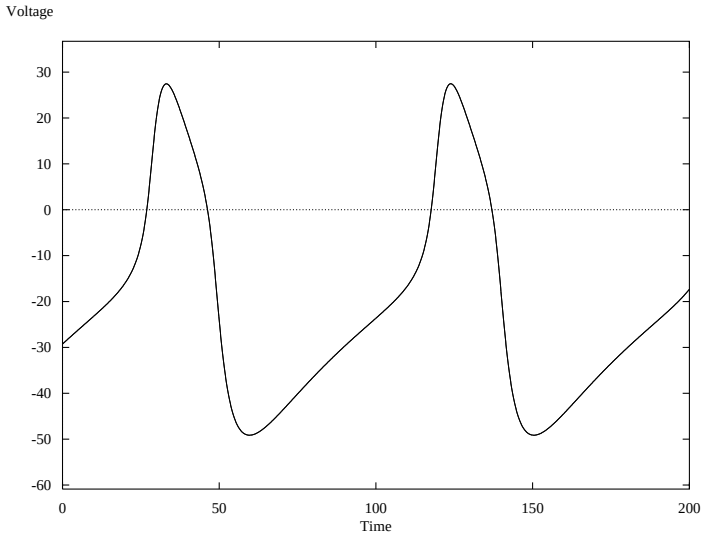


Figure 4-2: *Morris-Lecar plasma membrane voltage oscillator.*

4-2.2 The Li-Rinzel Model

This model describes the IC oscillations in the cytosol. The calcium is introduced through both the plasma membrane (PM) and the endoplasmic reticulum (ER).

$$\frac{dc}{dt} = J_{rel} - J_{fill} + J_{in} - J_{out} \tag{4-6}$$

$$\frac{dh}{dt} = \frac{h_\infty(c) - h}{\tau(c)} \tag{4-7}$$

$$\frac{da}{dt} = J_{in} - J_{out} \tag{4-8}$$

where $a = c + \sigma c_{ER}$.

The flux of calcium out of and into the cytosol is given by the following equations:

$$J_{rel} = p(c_{ER} - c) \tag{4-9}$$

$$J_{fill} = \frac{v_{ER} c^2}{k_{ER}^2 + c^2} \tag{4-10}$$

$$J_{out} = \frac{v_{PM} c^2}{k_{PM}^2 + c^2} \tag{4-11}$$

$$P = L + M \left(\frac{ch}{d_a + c} \right)^3 \tag{4-12}$$

For a fairly robust set of parameter values, this system for the calcium concentration in the cytosol also settles to a periodic oscillation (Fig. 4-3) which may also have a similar period to solutions of the Morris-Lecar system.

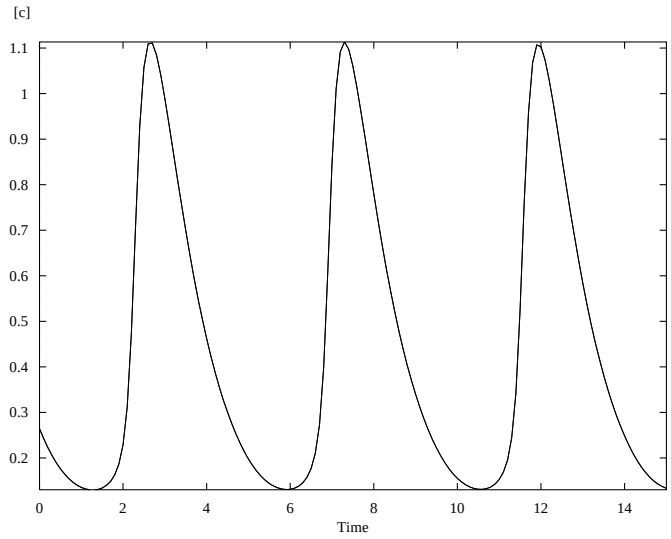


Figure 4-3: *Li-Rinzel intra-cellular Calcium oscillator.*

4-3 Coupling

Recall that our goal is to determine if the IC oscillations in two separate cells become synchronized when the two PM oscillators become synchronized. As previously described, we attempt to do this by taking two existing models, the Morris-Lecar mode- and the Li-Rinzel model, and model how they couple based on experimental evidence. This will give a model for the calcium oscillations within the cell. We will then take two identical cells modeled in this manner and couple them across the plasma membranes. We want to see, if the PM becomes synchronized, whether the IC oscillations also become synchronized even though they are not directly affecting each other.

4-3.1 Coupling Within the Pituitary Cell

Having previously explored the uncoupled Morris-Lecar and Li-Rinzel models with XPP, we decided it was time to model the coupling between the IC and the PM of each individual pituitary cell. The models were coupled by the addition of two extra terms. The first was through the equations for v and c, where these oscillations are related through the equation for J_{in};

$$J_{in} = J'_{in} - \alpha I_{Ca} \tag{4-13}$$

Second, we introduced a calcium activated potassium current $I_{K(Ca)}$, which affects the voltage across the plasma membrane due to the influx of potassium ions. Hence equation (4-2) becomes

$$C_m \frac{dv}{dt} = I_{app} - I_{Ca} - I_K - I_L - I_{K(Ca)} \tag{4-14}$$

where

$$I_{K(Ca)} = \bar{g}_{K(Ca)} \frac{(v - v_K)c^4}{c^4 + k_{Ca}^4} \tag{4-15}$$

See Fig 4-4 for an example of phase-locking. The resulting 5-dimensional system corresponds to a feedback loop between the PM and the IC regulation oscillator. Here PM activates the IC while IC inactivates the PM. Numerical explorations with the program XPP revealed that indeed the voltage oscillations of the PM and the calcium oscillations of the ER do phase-lock. In fact, they phase-lock over a large set of parameter values, and for each parameter value the basin of attraction for the phase-locked solution seemed to be the whole phase-space.

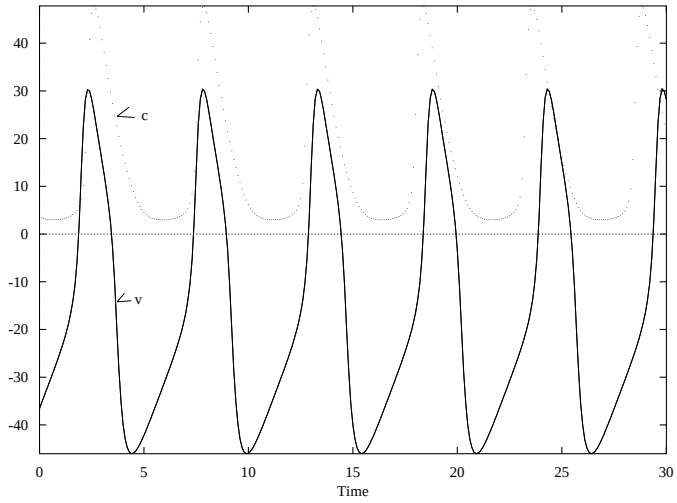


Figure 4-4: *Phase-lock between PM and IC.*

However, since we are only interested in solutions which are periodic, it is natural to ask ourselves if the behaviour of the cell can be captured with a much simpler set of equations. Indeed, it is often the case that the dynamics of oscillators, such as the two above, can be described solely though one variable - their phase difference. Descriptions of this sort are called phase models. In these types of models, the state of the oscillator during its cycle is parameterized by a phase variable. This greatly simplifies the analysis because the model reduces to a problem in one periodic variable. We investigated a popular method for coupling mutually attracting/repelling oscillators, based on the work of Strogatz [4] and Li [3]. In Section 4-3.1, we describe this method, and (unsuccessfully apply it to our system). We also give an argument why this approach does not work.

The Integrate and Fire Model of Coupling

This is a specific example of a phase model. Let ϕ describe the phase of an oscillator with period 1 and let f be a function such that $x = f(\phi)$ subject to the conditions:

$$\begin{aligned}
&i) && \frac{d\phi}{dt} = 1 \\
&ii) && f(0) = 0 \\
&iii) && f(1) = 1 \\
&iv) && f \text{ is concave down, i.e. } f' > 0,\ f'' < 0
\end{aligned}$$

One function that satisfies these conditions is:

$$f(\phi) = c(1 - e^{-\phi}) \tag{4-16}$$

$$c = \frac{1}{1 - e^{-1}} \tag{4-17}$$

The time evolution of this function is shown in Fig 4-5, x increases towards its threshold value at which point it fires and then returns to 0 and the cycle repeats. When dealing with two coupled oscillators, the firing of one causes a displacement of the other oscillator. It follows this time evolution:

$$x_i(t) = 1 \Rightarrow x_j(t^+) = min\{1, x_j(t) + \varepsilon\}, \qquad \begin{array}{l} i = 1, 2 \\ j \neq i, j = 1, 2, \varepsilon > 0 \end{array}$$

Hence, we use the following equations for our integrate-and-fire model.

$$\begin{aligned} f(\phi) &= c(1 - e^{-\phi}) \\ g(x) &= \ln \frac{c}{c - x} \end{aligned}$$

where g is the inverse function of f (ie. fg $\equiv 1$)
It is easy to see that to construct the return map $R(\phi) = h_2(h_1(\phi))$. We chose

$$\begin{aligned} h_1(\phi) &= g(f(1 - \phi) - \varepsilon_1), \\ h_2(\phi) &= g(f(1 - \phi) + \varepsilon_2). \end{aligned}$$

The size of $\varepsilon_i > 0$ determines the strength of the interaction between the oscillators. The sign change of ε describes an activating/inactivating process. For simplicity we considered the case $\varepsilon_1 = \varepsilon_2$. Then the return map in this case is given by

$$\begin{aligned} R(\phi) &= h_2(h_1(\phi)) \\ &= -ln(e^{-1}[e^{\phi - 1} - \frac{\varepsilon}{c}]^{-1} + \frac{\varepsilon}{c}) \end{aligned}$$

Fixed points of $R(\phi)$ correspond to synchronized or phase-locked solutions. Assuming the existence of a fixed point $R(\phi^*) = \phi^*$, we found that

$$\phi^* = \ln \frac{\frac{\varepsilon}{c} \pm \sqrt{e(\frac{\varepsilon^2}{c^2} - 4)}}{2} \tag{4-18}$$

This gives a lower bound on ε :

$$\varepsilon_c = \frac{2c}{\sqrt{\varepsilon}} \tag{4-19}$$

At this critical value of ε, the fixed point $\phi^* = \frac{1}{2}$ which corresponds to a solution with half a phase difference. Unfortunately the lower bound on ε is not valid in our region $[0, 1]$. It is easily verified that $R(\phi)$ does not have a fixed point.

There are several reasons why this model does not work. The most simple and compelling reason can be described by considering two runners on a circular track. Let us assume that runner A is ahead of runner B and A activates B while B inhibits A. When runner A crosses the mark, B gets pushed closer to A by an amount ϵ. Then when B crosses the mark, he pulls A closer to him by the same

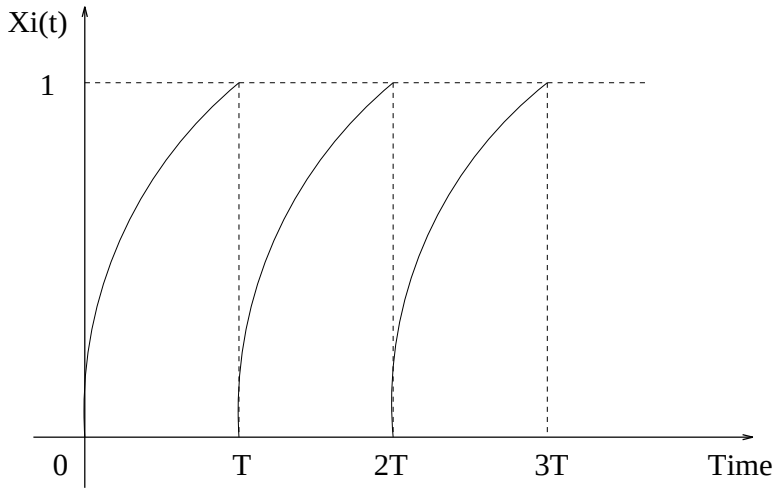


Figure 4-5: *Integrate-and-Fire Oscillator.*

amount. We thus see that A and B get closer and closer together. However, as this continues, let us say that A and B are now running side-by-side (ie. synchronized). Then as A and B simultaneously cross the finish line they will repell each other because B will push A backwards and A will push B forwards. In this way, we can see that A and B can never synchronize or phase-lock. Hence, this model is not appropriate in describing the coupling between the IC and the PM.

4-3.2 Coupling Between Cells

Here we take two pituitary cells modeled as above and couple them proportional-y to the voltage difference across the intercellular gap. The electrical gap junction coupling gives positive/positive feedback (Fig. 4-1). The coupling takes the form:

$$C_m \frac{dv}{dt} = I_{app} - I_{Ca} - I_K - I_L - I_{K(Ca)} + \varepsilon(\bar{v} - v) \quad (4\text{-}20)$$

$$C_m \frac{d\bar{v}}{dt} = I_{app} - I_{Ca} - I_K - I_L - I_{K(Ca)} - \varepsilon(\bar{v} - v) \quad (4\text{-}21)$$

$$(4\text{-}22)$$

The voltage oscillations of the two cells synchronize over time, as do those for the IC oscillations despite there being no direct coupling between them (Figs.4-6 and 4-7). We determined numerically the response and interaction functions (Figs. 4-8 and 4-9). The response function estimates the systems phase-shift due to perturbation. Roots of the interaction function correspond to fixed points in the phase difference between oscillators. Here we found roots at zero and half-phase, with zero being the stable fixed point.

4-4 Conclusion

The combination of the Morris-Lecar and and the Li-Rinzel model gave a similar behaviour to experimentally observations in pituitary gland cells. We were able to obtain coupling of the two

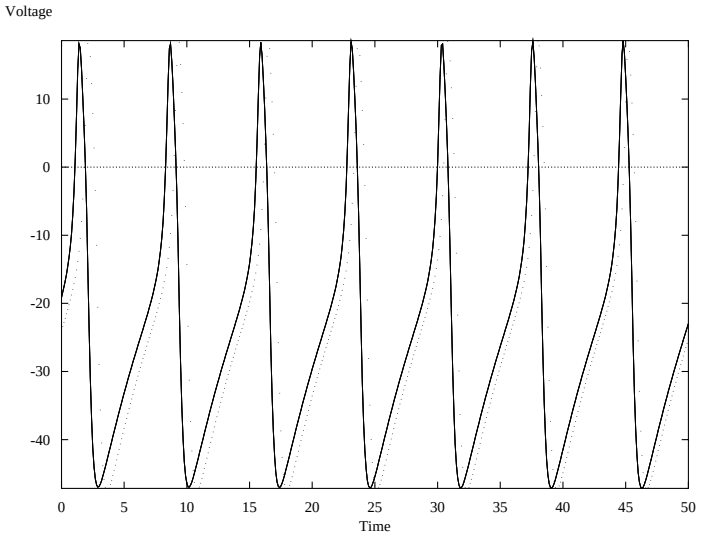


Figure 4-6: *Synchronization of two PM oscillators.*

oscillators in the PM and ER membrane. Furthermore two cells that were coupled only through the plasma membrane, corresponding to gap junction coupling, had their ICs synchronized. Therefore, we found that calcium oscillations of each individual pituary cell do synchronize with neighboring pituary cells. In answering this we also found that the PM and IC oscillations within each pituitary gland cell phase-lock, and that the PM oscillations of two neighbouring cells synchronize.

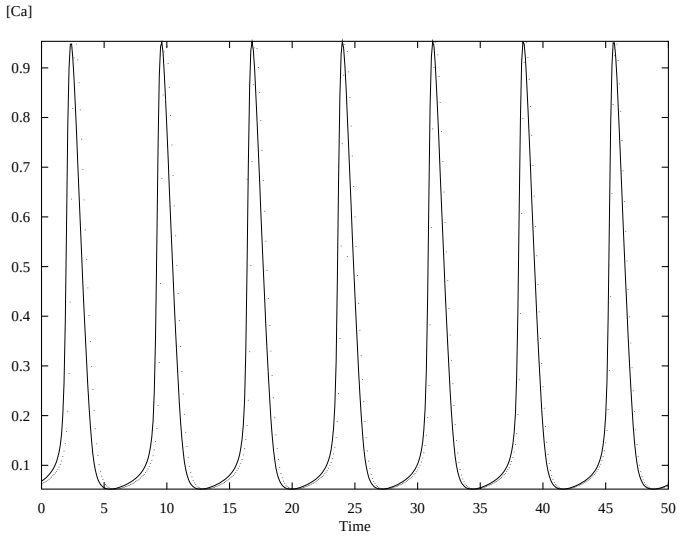


Figure 4-7: *Synchronization of two IC oscillators.*

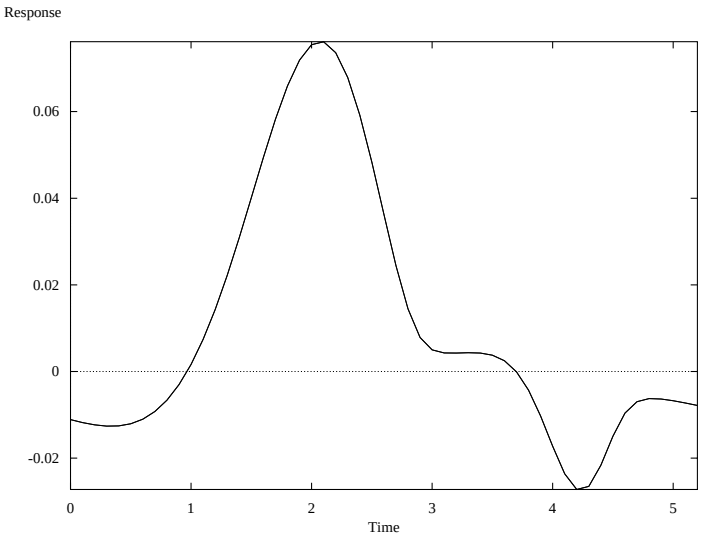


Figure 4-8: *Response Function.*

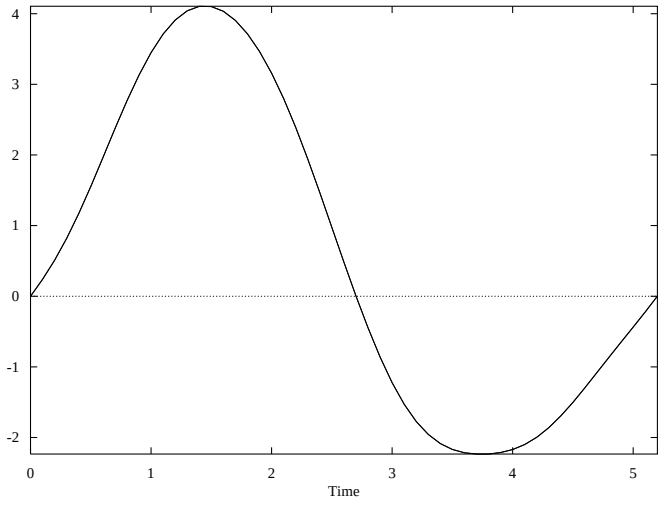


Figure 4-9: *Interaction Function.*

Chapter 5

Air Impact on Green Sand

Participants: C. Please (Mentor), S. Bohun, A. Bona, D. Chertok, J. Modayil, J. Samuel, C. Stoica, A. Wacher

PROBLEM STATEMENT: Toowoomba Foundry is a producer of cast-iron parts for a number of automobile and agricultural machinery manufacturers. To produce their cast-iron products, such as truck brake drums and water pump housings, they use a conventional method involving making a mould, for every item, out of sand. Once made, the mould is filled with molten metal which is then cooled and the mould then broken to reveal the cast object.

The technology of creating a sand mould consists of placing sand over a pattern and then compressing the sand to create compacted sand which has sufficient rigidity that the pattern can be removed easily, and the sand will remain in place when molten metal is poured onto it.

The sand itself has some very special properties and consists of relatively uniform grains of "Dune" sand mixed with water and very fine particles of bentonite (clay) which cover the sand. The purpose of this project will be to investigate how the sand gets compacted. The method of interest consists of using a sealed box with the pattern as the base in which the sand is loosely placed. The box is then very suddenly subjected to high pressure air (around 7 atmospheres of pressure). If the pressure is applied too slowly very little compaction occurs while when applied very quickly the sand becomes very well compacted with at particularly strongly compacted region adjacent to the pattern. However, the rapid compaction tends to create regions relatively void of sand and hence causes problems when the molten metal is poured in.

Can a model be made to demonstrate how the sand compacts so that methods of creating good compaction without voids be suggested?

5-1 Introduction

An Australian steel casting company, Toowoomba Foundry, employs a sand mould casting procedure as part of its operation. To produce a mould, a sand-based substance called *green sand* is poured into a box containing the moulding pattern. Subsequently, high pressure air at 0.7 MPa is quickly (on the time scale of $0.1s$) applied to the top of the sand to compress it. The profile of the pressure pulse is shown in Figure 5-1. Upon compression, the sand is packed into a solid structure which preserves the shape of a pattern that is later used for casting.

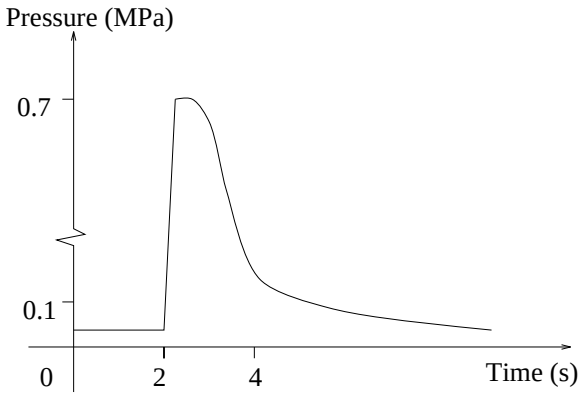


Figure 5-1: *Compaction pressure pulse.*

To optimize the compaction process, the manufacturer wishes to reduce inhomogeneities in thee compacted sand that lead to imperfections in the mould. An empirical solution to this problem involves the application of two consecutive compacting air pulses as shown in Figure 5-2.

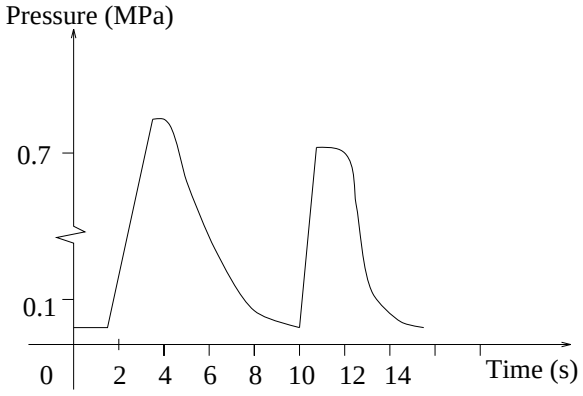


Figure 5-2: *Preceding pressure pulse with a pre-compaction pulse.*

It has been observed that the mould imperfections, *soft spots*, occur at specific positions in the mould and that the size and location of such imperfections depend on the geometry of the pattern. An example of such a dependency is given in Figure 5-3. Other tools for reducing the size of soft spots at the manufacturer's disposal include: 1) roughening of the pattern surface, 2) the introduction of air vents at the outer surface of the mould.

The object of this research was to develop a qualitative mathematical model of the problem and suggest improvements for the manufacturing process.

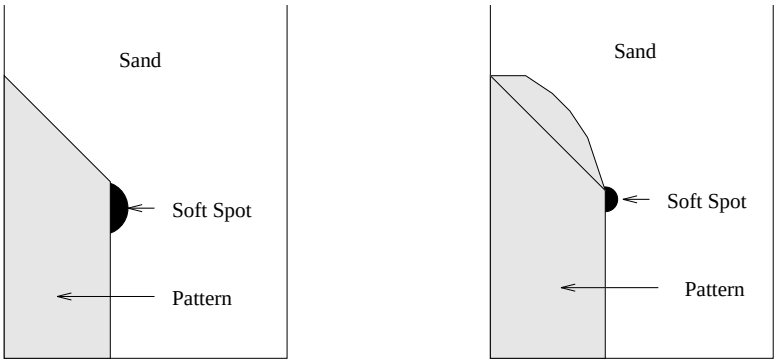


Figure 5-3: *Reduction of soft spots due to the change in mould shape: larger for sharp angles (left figure), smaller for smooth profiles (right figure).*

5-2 Physical properties and assumptions

1. The actual particles of solid in the sand mixture are assumed to be incompressible. However, the air fraction of the sand mixture is compressible. It is the air fraction that makes the sand a compressible material.
2. During the compaction process, the permeability of the sand changes insignificantly. This assumption was made on incomplete experimental data and should be verified experimentally.
3. Based on experimental data, the sand never experiences decompression.
4. Air is treated as an ideal isothermic Newtonian fluid with constant viscosity.
5. We consider a one-dimensional model in space. The equations are based on the classical conservation laws of mechanics (i.e. the conservation of mass and the conservation of momentum).

5-3 Formal Construction of the Model

Variables

- $y =$ vertical coordinate.
- $t =$ time.

Constants

- $h_0 = 0.4$m.
- $K/\mu = 1/\lambda = 10^{-6} m^3 s/kg$ (Darcy's 'constant' for green sand).
- $\alpha_0 = 0.6$ (Fraction of air in un-compacted sand mixture).

State Variables

- $\alpha(t, y)$: Volume percentage of air in green sand mixture.
- $\rho_A(t, y)$: Density of air.
- $\rho_S(t, y)$: Density of a sand particle.
- $v_A(t, y)$: Velocity of air.
- $v_S(t, y)$: Velocity of a sand particle.
- $h(t)$: Position of the top of sand mixture.
- $P_0(t)$: Experienced pressure of air at top of the sand box.
- $P_A(\rho_A) = \lambda\rho_A$, where $\lambda = (1atm/kg{\cdot}m^3)$: Air pressure as given by an ideal, isothermal gas.
- $P_S(\alpha)$: Pressure of sand on sand (Experimentally obtained).

5-3.1 Conservation of Mass and Momentum

Mass

$$\text{1. Air:} \quad \frac{\partial(\alpha\rho_A)}{\partial t} + \frac{\partial(\alpha\rho_A v_A)}{\partial y} = 0$$

$$\text{2. Sand:} \quad \frac{\partial((1-\alpha)\rho_S)}{\partial t} + \frac{\partial((1-\alpha)\rho_S v_S)}{\partial y} = 0$$

Momentum

$$\text{3. Air:} \quad \frac{\partial}{\partial t}(\alpha\rho_A v_A) + \frac{\partial}{\partial y}(\alpha\rho_A v_A^2) = -\frac{\partial P_A}{\partial y} - \gamma v_A$$

$$\text{4. Sand:} \quad \frac{\partial}{\partial t}((1-\alpha)\rho_S v_S) + \frac{\partial}{\partial y}((1-\alpha)\rho_S v_S^2) = -\frac{\partial P_S}{\partial x}\frac{\partial \alpha}{\partial y} - \frac{\partial P_A}{\partial \rho_A}\frac{\partial \rho_A}{\partial y}$$

5-3.2 Non-dimensionalization

We scale all variables with respect to a set of characteristic sizes $P = (1\text{MPa})\bar{P}$, $\rho_A = \rho_0\bar{\rho}_A$, $h = h_0\bar{h}$, $y = h_0\bar{y}$, $t = t_0\bar{t}$, $v = (h_0/t_0)\bar{v}$, $\rho_S = \rho_S$, $\alpha = \alpha$, $\gamma = \gamma$. Now examine the changes to the above equations.

$$1. \quad \frac{\partial(\alpha\rho_A)}{\partial t} + \frac{\partial(\alpha\rho_A v_A)}{\partial y} = 0$$

$$2. \quad \frac{\partial(1-\alpha)}{\partial t} + \frac{\partial((1-\alpha)v_S)}{\partial y} = 0$$

$$3. \quad \frac{\rho_0 h_0^2}{t_0^2(1\text{MPa})}\left(\frac{\partial}{\partial t}(\alpha\rho_A v_A) + \frac{\partial}{\partial y}(\alpha\rho_A v_A^2)\right) = -\frac{\partial P_A}{\partial y} - \frac{\gamma h_0^2}{t_0(1\text{MPa})}v_A$$

$$4. \quad \frac{\rho_S h_0^2}{t_0^2(1\text{MPa})}\left(\frac{\partial}{\partial t}((1-\alpha)v_S) + \frac{\partial}{\partial y}((1-\alpha)v_S^2)\right) = -\frac{\partial\left(P_A(\rho_A) + P_S(\alpha)\right)}{\partial y}$$

If the momentum of the air is significant, then the proper scaling factor would be:

$$\frac{\rho_0 h_0^2}{t_0^2(1\text{MPa})} = 1, \quad \text{or} \quad t_0 \simeq 0.1ms.$$

If the momentum of the sand was significant then:

$$\frac{\rho_S h_0^2}{t_0^2(1\text{MPa})} = 1, \quad \text{or} \quad t_0 \simeq 3ms.$$

If the friction of the air on the sand was significant, then $t_0 \simeq 0.16s$ which is comparable to the $0.2s$ supplied by the machine. Hence, the momentum of sand and/or air does not play any significant role. Therefore, the most significant process is governed by the Darcy's law. The initial conditions associated with the momenta, $v_A(0, y) = 0$ and $v_S(0, y) = 0$, are discarded.

5-3.3 Simplified Equations

$$\frac{\partial(\alpha\rho_A)}{\partial t} + \frac{\partial(\alpha\rho_A v_A)}{\partial y} = 0 \tag{5-1}$$

$$\frac{\partial(1-\alpha)}{\partial t} + \frac{\partial((1-\alpha)v_S)}{\partial y} = 0 \tag{5-2}$$

$$\frac{\partial}{\partial y}(P_S(\alpha) + P_A(\rho_A)) = 0 \tag{5-3}$$

$$v_A = -\frac{\partial P_A}{\partial y} \tag{5-4}$$

The initial value conditions and the boundary conditions become:

$\alpha(0, y) = \alpha_0$	$v_A(t, 0) = 0$ (no vents)
$\rho_A(0, y) = 1$	$v_S(t, 0) = 0$
$h(0) = 1$	$\rho_A(t, 0) = 1$ (vents)
$P_S(\alpha_0) = 0$	$dh/dt = v_S(t, h(t))$
$P_S(\alpha) =$ experiment data	$\rho_A(t, h(t)) = P_0(t)$

Considering the third equation and the last boundary condition, we find

$$P_S(\alpha) + \rho_A = P_0(t)$$

since $P_A = \rho_A$.

5-4 Analysis

The relation

$$P_S(\alpha) + \rho_A = P_0(t)$$

is critical. Through some mathematical consideration (omitted for brevity) we can investigate the compression of the sand mixture in some special cases:

1. Assume that the experienced pressure at the top of the sand box increases much faster than $t_0 \simeq 0.16$s. In this case, no air flows through the sand and the sand compacts uniformly. Vents at the bottom of the box do not play any role.
2. Assume that the experienced pressure a the top of the sand box increases much slower than t_0.
   (a) If there are no air vents at the bottom of the pattern, then no compression occurs.
   (b) If there are vents then compression increases linearly; the bottom will be the most compressed region.

5-5 Conclusions

- *Ignore inertia*: Inertia does not play any role in the compaction process.
- *No shock waves*: Since momentum does not contribute to the compaction process, no waves can be propagated.

5-6 Recommendations

- *No shock waves*: Shock waves should not be considered, even in higher dimensions.
- *Permeability*: Values of permeability are neither easily, nor accurately calculated by theoretical means. Moreover, the value of permeability can critically affect the compression of the sand, especially considering that the critical time t_0 is so close to the machine's capability of raising air pressure. The company is urged to calculate the permeability of the sand mixture for various values of α.
- *Partial fill*: Soft spots are believed to be formed by transverse motion of the sand particles. One method to reduce these defects is to fill the box with sand to the 'critical shoulder', followed by a fast compression. Now, fill the box and perform another fast compression.
- *Shoulder air vents*: Place air vents on the 'shoulder' above the defect. For this to be effective, the increase in air pressure must occur over a time scale of $\simeq 10t_0$
- *Plastic top*: If an airtight cover were placed on top of the sand, then the air pressure in the sand-air mixture would be much lower than without the cover. Hence, the sand must compact more. Moreover, as the air pressure in the sand-air mixture is lower, the time designated for pressure release can be shortened significantly.
- *Special air vents*: If air vents are placed at the bottom of the box and are opened shortly after the pressure reaches its maximum, we expect the efficiency of the machine to increase drastically. Installing such vents could lead to two advantages:
  - Denser compaction for the same air pressure pulse;
  - The relaxation time can be reduced.

Chapter 6

Surface Tension in a Flowing Fluid

Participants: David Ross (Mentor), Vladislav Agapov, Roger Coroas, Tony Cabal, Charles Cuell, Andrew Irwin, Tim Rogalsky, Bruce Rout

PROBLEM STATEMENT: Photographic film is made by coating a thin fluid layer of gelatin solution on a hard backing. The solution contains light-sensitive silver-halide crystals and dye-forming chemicals (and other stuff). The gelatin dries and hardens. Because of the importance of gelatin to our manufacturing processes, we are interested in the viscosity of gelatin solutions. As we all know from experience with Jell-O, (yes, it's the same stuff that we use to make film), gelatin solutions start out a fluids, they get thicker and thicker, and eventually turn into solids. The problem would be to find a mathematical model of the chemical process by which this happens.

6-1 Introduction

Eastman-Kodak Corporation presented a problem regarding the application of chemicals onto a polymer film using a long thin waterfall known as a coating curtain. The company wishes to keep a low surface tension on the curtain to prevent the curtain from breaking up during chemical application. One method the company has utilized is to add surfactant to the aqueous reservoir of the curtain.

The surfactant molecules consist of a hydrophobic 'tail' and neutral 'head' which congregate at the surface of the curtain, thereby lowering surface tension.

The company expected that by using surfactants of higher hydrophobicity the surface tension of the curtain would continually decrease in some well-behaved function. Although surfactants below a certain level of hydrophobicity behave as expected, above this level surfactants of higher hydrophobicity actually increase the surface tension.

It is believed micelles are responsible for this behaviour.

6-2 Mathematical Formulation

Surfactant accumulates on the surface of the curtain by fluid transport (the emulsion is falling) and by diffusion through the curtain. However, experiments show that air bubbles transported by the fluid are essentially in free fall at all points in the curtain. This allows us to rule out the fluid transport process. A good estimate for diffusion transport is given by $\frac{Dt}{x^2} = 1$. Using $t = 0.1s$ (the length of time the emulsion falls) we get $x \approx 3 \times 10^{-4}$cm, which allows for diffusion to be a possibility, since the typical cross section is 2×10^{-3}cm at the bottom of the curtain and the surfactant content in this outer sheath, if transported to the clean surface, is sufficient to reduce the surface tension.

In addition to diffusion, when the monomer concentration goes to some value typical for the surfactant, groups of sixty monomers will form non–hydrophobic micelles so that the concentration of monomers remains constant at this value. This concentration is called the CMC (critical micelle concentration). Given this, we can write the diffusion equations for the monomers and micelles as:

$$\frac{\partial C}{\partial t} = D_c \frac{\partial^2 C}{\partial x^2} - 60KH(C - CMC) + 60KH(CMC - C)M \tag{6-1}$$

$$\frac{\partial M}{\partial t} = D_m \frac{\partial^2 M}{\partial x^2} + KH(C - CMC) - KH(CMC - C)M, \tag{6-2}$$

where C is the monomer concentration, M is the micelle concentration, K is the rate at which monomers form micelles, D_c is the monomer diffusion rate and D_m is the micelle diffusion rate. H is the Heaviside function.

The initial conditions are

$$C(x, 0) = A \tag{6-3}$$

$$M(x, 0) = B, \tag{6-4}$$

where A and B are constants.

The boundary conditions are

$$\frac{d?}{dt} = D_c \left.\frac{\partial C}{\partial x}\right|_{x=0} \tag{6-5}$$

$$= k_f C(0,t)(?_{max} - ?) - k_b ? \quad (6\text{-}6)$$

$$\left.\frac{\partial M}{\partial x}\right|_{x=0} = 0. \quad (6\text{-}7)$$

In the first boundary condition, ? is the concentration of monomer on the surface of the curtain. The first equality comes from Fick's equation, $\phi = -D\frac{\partial C}{\partial x}$, where ϕ is the monomer flux. The second equality comes from modelling the interaction of the monomer with the surface of the curtain as a second order kinetic reaction, with forward (on to the surface) and backward rates given by k_f and k_b respectively.

The boundary condition for M is a zero flux condition at the water/fluid boundary.

We notice that if we add 60 times the equation for M to the equation for C, the Heaviside functions cancel and we end up with

$$\frac{\partial C}{\partial t} + \frac{\partial 60M}{\partial t} = D_c\frac{\partial^2 C}{\partial x^2} + D_m\frac{\partial^2 60M}{\partial x^2}. \quad (6\text{-}8)$$

We can define a new variable $\eta = C + 60M$, and recognize that as we move from the surface into the medium, η increases from zero, and reaches CMC at some point (provided the initial concentration is sufficient). Up to that point, $M = 0$, so $\frac{\partial^2 M}{\partial t^2} = 0$. After that point, $C = CMC$, so $\frac{\partial^2 C}{\partial t^2} = 0$. Given this, we can write the following with impunity

$$\frac{\partial \eta}{\partial t} = \frac{\partial}{\partial x}\left(D(\eta)\frac{\partial(\eta)}{\partial x}\right), \quad (6\text{-}9)$$

where

$$D(\eta) = D_c, \quad \text{if} \quad \eta < CMC \quad (6\text{-}10)$$

$$= D_m, \quad \text{if} \quad \eta > CMC \quad (6\text{-}11)$$

Boundary conditions become

$$\eta(x,0) = A + 60B \quad (6\text{-}12)$$

$$\frac{d?}{dt} = k_f \min(\eta(0,t), CMC)(?_{max} - ?) - k_b ? \quad (6\text{-}13)$$

$$= D_c \left.\frac{\partial \eta}{\partial x}\right|_{x=0} \quad (6\text{-}14)$$

6-3 Numerical Solution of the Partial Differential Equation

To verify the model, the diffusion equation was solved numerically. Three approaches to this were attempted. An IMSL routine appeared promising, but was unable to deal with the non-linear boundary condition. However, consistent results were obtained from two algorithms using the method of lines in Matlab.

The method of lines is a technique which reduces a partial differential equation to a system of ordinary differential equations. This system can then be solved using standard techniques. One team used the Matlab routine ODE45 to solve it. The other team wrote an Euler solver. Both resulted in the same solution when the initial concentration was lower than the CMC.

The method of lines discretizes the partial differential equation along the x-axis. That is, $\eta(x,t)$ is subdivided into $N+1$ functions $\eta_i(t)$, where

$$\eta_i(t) = \eta(i\Delta x, t), \quad i = 0, 1, \ldots, N, \tag{6-15}$$

for some given interval Δx. Each of these is then followed along a 'line' in the t-direction by the ordinary differential equation solver.

In the interior of the domain, the relationship between the functions $\eta_i(t)$ is determined by the diffusion equation (6-9) using the finite difference method. Thus,

$$\frac{d\eta_i}{dt} = \frac{D\left(\frac{\eta_{i+1}+\eta_i}{2}\right)\left(\frac{\eta_{i+1}-\eta_i}{\Delta x}\right) - D\left(\frac{\eta_i+\eta_{i-1}}{2}\right)\left(\frac{\eta_i-\eta_{i-1}}{\Delta x}\right)}{\Delta x}, \quad i = 1, \ldots, N-1. \tag{6-16}$$

Assuming symmetry, the boundary at $i = N$, corresponding to the inside of the waterfall, requires no flux of surfactant. That is, $\frac{\partial \eta_N}{\partial x} = 0$. This is enforced by setting

$$\frac{d\eta_N}{dt} = \frac{d\eta_{N-1}}{dt}, \tag{6-17}$$

ensuring a horizontal slope there. Initial conditions are given by the initial concentration,

$$\eta_i(0) = 10^{-6}, \quad i = 1, \ldots, N. \tag{6-18}$$

In the first numerical scheme, the equation in η_0 was removed and equation (6-13) in ? was added to the system. Then $\eta_0(t)$ was found algebraically, using the finite difference form of equation (6-14),

$$D_c \frac{\eta_1 - \eta_0}{\Delta x} = k_f \min(\eta_0, CMC)(?_{max} - ?) - k_b ?. \tag{6-19}$$

The results were encouraging, and are presented below for the surfactants having a CMC greater than the initial concentration of surfactants.

In the second numerical scheme, the boundary condition given in equation (6-13) is simplified. Analyzing the stability for the worst-case scenario (when $\eta_0(t) > CMC$), we require that

$$(k_f CMC + k_b)\Delta t < 1. \tag{6-20}$$

Since $k_b/k_f = 10^{-3}CMC$, and $k_b = 10^{-5}$, this ordinary differential equation requires $\Delta t < 10^{-8}$. The condition for stability of the partial differential equation, however, is $\frac{D_c \Delta t}{\Delta x^2} < 1$, where $D_c = 10^{-6}$. Thus the partial differential equation requires $\Delta t < 10^{-2}$.

This is a stiff system, with the time-scale of ? being much smaller than the time-scale of η. What this really means, however, is that the concentration at the surface equilibrates virtually instantaneously with the concentration at the subsurface. That is, on the micro scale, $\dfrac{d?}{dt} = 0$. Solving equation (6-13) gives

$$? = ?_{max} \frac{\min(\eta_0(t), CMC)}{\min(\eta_0(t), CMC) + 10^{-3}CMC}. \tag{6-21}$$

Differentiating this (on the macro scale!) and recalling that $\dfrac{d?}{dt} = D_c \dfrac{\partial \eta}{\partial x}\bigg|_{x=0}$, we have

$$\frac{d\eta_0}{dt} = \frac{D_c \left.\frac{\partial \eta}{\partial x}\right|_{x=0} \left(\eta_0(t) + 10^{-3}CMC\right)^2}{?_{max} 10^{-3} CMC}, \qquad \eta_0 < CMC. \tag{6-22}$$

Critical Micelle Concentration (mol cm^{-3})	Dynamic Surface Tension (dynes cm^{-2}) at 0.1 s
$1 \cdot 10^{-4}$	53.0432
$1 \cdot 10^{-5}$	41.98
$1 \cdot 10^{-6}$	30.54
$1 \cdot 10^{-7}$	30.76
$5 \cdot 10^{-8}$	40.72
$1 \cdot 10^{-8}$	60.02
$2 \cdot 10^{-8}$	55.0450

Table 6-1: *Computed dynamic surface tension as a function of initial surfactant concentration.*

When $\eta_0 > CMC$, the concentration at the subsurface is above critical. Therefore a no flux condition is required. In finite difference form, the partial differential equation at the boundary becomes

$$\frac{\partial \eta_0}{\partial t} = \frac{2D_m\,(\eta_1 - \eta_0)}{(\Delta x)^2}, \qquad \eta_0 > CMC. \tag{6-23}$$

Finally, since we are assuming that $\eta_0(0) \approx ?\,(0)$, initially

$$\eta_0(0) = 0. \tag{6-24}$$

The numerical scheme is now complete. Ordinary differential equations (6-16), (6-17), (6-22), and (6-23) together with initial conditions (6-18) and (6-24) form the system comprising the method of lines.

6-4 Results

We solved the diffusion equation (6-9) numerically for a variety of critical micelle concentrations of surfactant. The initial surfactant concentrations and computed surface tensions are reported in Table 6-1 and experimental data obtained from Eastman-Kodak are in Table 6-2. These data are plotted on the same set of axes in Figure 6-1.

Critical Micelle Concentration (mol cm^{-3})	Dynamic Surface Tension (dynes cm^{-1}) at 0.1 s
$1.0 \cdot 10^{-7}$	41
$1.5 \cdot 10^{-7}$	30
$1.566 \cdot 10^{-7}$	59
$1.95 \cdot 10^{-7}$	56.5
$3.5 \cdot 10^{-7}$	29
$2.0 \cdot 10^{-6}$	33
$2.0 \cdot 10^{-5}$	43

Table 6-2: *Experimentally measured dynamic surface tension as a function of critical micelle concentration (CMC). The initial surfactant concentration is* $\eta_0 = 10^{-6}$.

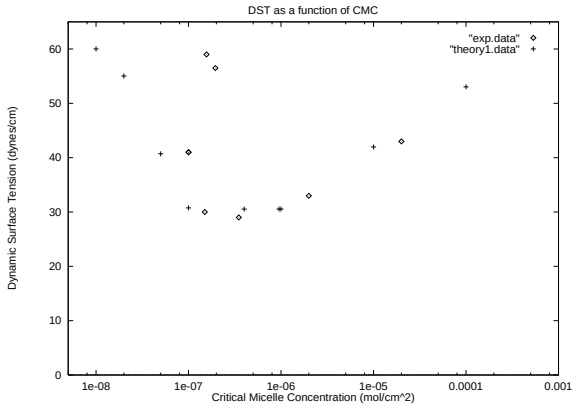


Figure 6-1: *Comparison of experimental and theoretical dynamic surface tensions as a function of critical micelle concentration. A small CMC corresponds to a strongly hydrophobic surfactant (a molecule with a longer tail). The initial total surfactant density was* $\eta(0) = 10^{-6}$ *mol/cm*^{3}.

Figures 6-2 and 6-3 show a three-dimensional plot of the total surfactant density as a function of space and time for two important cases: (a) $CMC > \eta(0)$ and (b) $CMC < \eta(0)$. In the first case, all the surfactant is in the highly mobile monomeric form, and the surfactant rapidly accumulates on the surface. There is an early minimum in the sub-surface surfactant concentration folowed by an equlibration phase which is nearly complete in 0.1s. In the second case, much of the surfactant is in micelle form and the diffusion process slows down the accumulation of surfactant on the surface.
Figures 6-4 and 6-5 shows the surfactant density on the surface of the curtain (?) as a function of time for two important cases: (a) $CMC > \eta(0)$ and (b) $CMC < \eta(0)$. The slower accumulation of surfactant on the surface can be seen clearly in the second figure.

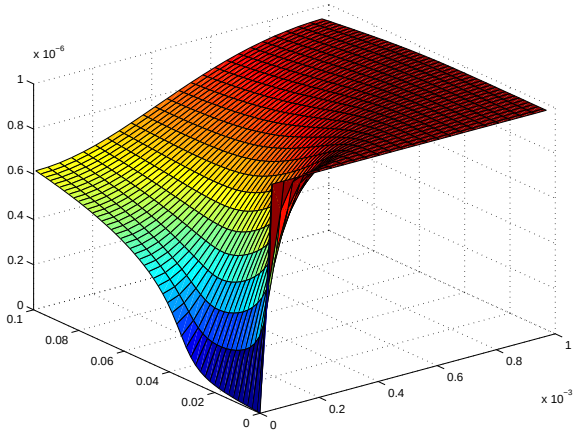


Figure 6-2: *Total surfactant density* $\eta(t)$ *(mol cm*$^{-3}$*, vertical axis) as a function of space (measured from the air-fluid boundary in cm, x-axis) evolving over time (for* $0 \leq t \leq 0.1s$*, y-axis) and CMC=* 10^{-5} *mol cm*$^{-3}$*. Computations were performed with a finer mesh than shown; the spatial mesh width was* $\Delta x = 10^{-5} cm$ *and approximately 5000 time steps were used.*

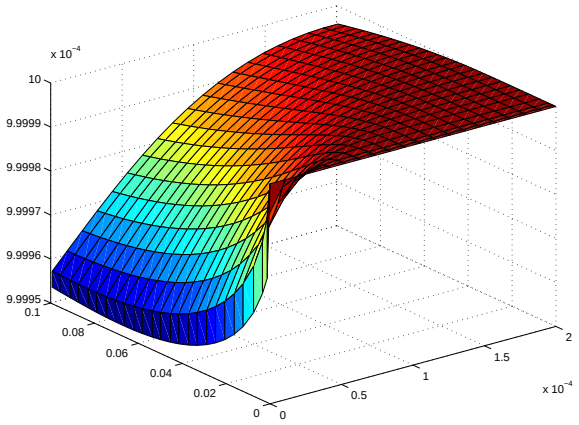


Figure 6-3: *Total surfactant density $\eta(t)$ (mol cm^{-3}, vertical axis) as a function of space (measured from the air-fluid boundary in cm, x-axis) evolving over time (for $0 \leq t \leq 0.1s$, y-axis) and CMC= 10^{-8} mol cm^{-3}. Computations were performed with a finer mesh than shown; the spatial mesh width was $\Delta x = 10^{-5} cm$ and approximately 5000 time steps were used.*

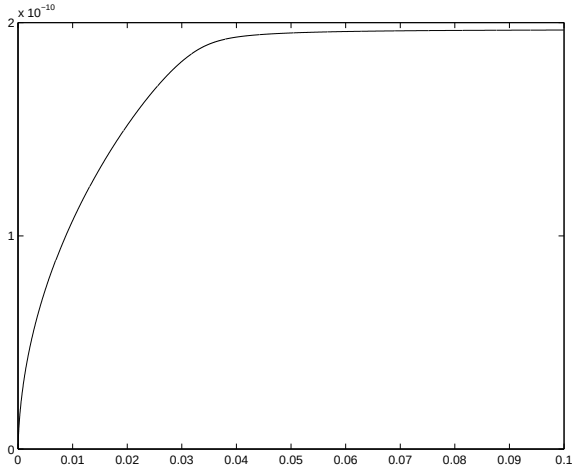


Figure 6-4: *Surfactant density on the surface of the curtain (? , mol cm^{-2}, vertical axis) evolving over time (for $0 \leq t \leq 0.1s$, horizontal axis) for CMC= 10^{-5} mol cm^{-3}.*

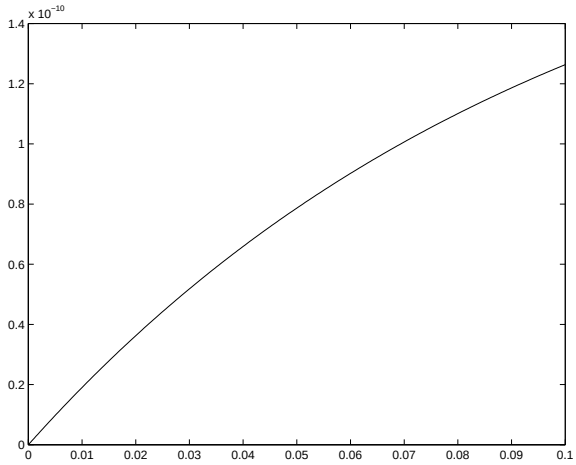


Figure 6-5: *Surfactant density on the surface of the curtain (? , mol cm^{-2}, vertical axis) evolving over time (for $0 \leq t \leq 0.1s$, horizontal axis) for CMC= 10^{-8} mol cm^{-3}.*

6-5 Conclusions

The excellent agreement between the experimental data and a simplified computational method suggest that the anomalous increase in dynamic surface tension (DST) for increasingly hydrophobic surfactants is in large part explained by our diffusion model. The initial decrease in DST with increasing hydrophobicity (shown by decreasing CMC on the right hand side of Figure 6-1) is easily understood. The longer surfactants are better at reducing the DST. Surfactants with a CMC smaller than the initial surfactant density of 10^{-6} mol/cm^3 form micelles in the holding tank and the slower mobility of the micelles in the diffusion process described above accounts for the larger DST after 0.1s.

Bibliography

[1] Avram, F., Bertsimas, D., and Ricard, M.: Fluid models of sequencing problems in open queueing networks; an optimal approach. In *Stochastic Networks*, volume 71 of *IMA Volumes in Mathematics and its applications*, Springer, New York: 199-234 (1995).

[2] I. Diaz-Rivera, D. Armbruster, and T. Taylor, Periodic orbits in re-entrant manufacturing systems. Preprint, 1998.

[3] Li, Y.X.: Modeling synchronized calcium oscillations in pituitary cells coupled through gap juctions. *Soci. Neurosci. Abstr.* 23: 422 (1997).

[4] Mirollo, R.E. and Strogatz, S.H.: Synchronization of Pulse-Coupled Biological Oscillators. *SIAM J. Appl. Math.* 50 (No. 6): 1645-62 (1990).

List of Participants

Organizing committee

Arvind Gupta	Simon Fraser University
Huaxiong Huang	PIMS and The University of British Columbia
Keith Promislow	Simon Fraser University

Mentors

Luis Goddyn	Simon Fraser University
Rachel Kuske	The University of Minnesota
Yue-xian Li	The University of British Columbia
Colin Please	The University of Southampton
David Ross	Eastman Kodak

List of Participants

Students

Janez Ales	Simon Fraser University
Brad Bart	Simon Fraser University
Daniel Chertok	Simon Fraser University
Roger Coroas	Simon Fraser University
Nick Costanzino	Simon Fraser University
Daya Guar	Simon Fraser University
Kevin Hare	Simon Fraser University
Shabnam Kavousian	Simon Fraser University
Yves Lucet	Simon Fraser University
Snezana Mitrovic	Simon Fraser University
Mike Neagu	Simon Fraser University
Bruce Rout	Simon Fraser University
Abagail Wacher	Simon Fraser University
Jeff Williams	Simon Fraser University
Vladislav Agapov	The University of Alberta
Andreea Amariei	The University of Alberta
Joseph Modayil	The University of Alberta
Adriana Dawes	The University of British Columbia
Kelly Kwok	The University of British Columbia
Joshua Madden	The University of British Columbia
Stefan Rienker	The University of British Columbia
Michele Titcombe	The University of British Columbia
Wai Tse	The University of British Columbia
Andrej Bona	The University of Calgary
Charles Cuell	The University of Calgary
Sean Bohun	The University of Victoria
Jonathan Samuel	The University of Victoria
Christina Stoica	The University of Victoria
Shane Jensen	McGill University
Andrew Irwin	Queen's University
Tracey Ewen	The University of Manitoba
Tim Rogalsky	The University of Manitoba
Khalid El-Yassini	The University of Sherbrook
David Saunders	The University of Toronto
Andrew Sheshnev	The University of Toronto
Antonio Cabal	The University of Western Ontario