

# Consulting in Applied Mathematics

CAAM 513, Spring 1998

## Preface

This report contains a description of four projects brought to the attention of the Consulting Course, CAAM 513 in the Department of Computational and Applied Mathematics at Rice University. The enclosed reports reflect the work done by students Genetha Gray, Nathan Hillson, Shannon Walsh and the instructor Liliana Borcea, in the Spring semester of 1998.

We thank the faculty in the CAAM department for their enthusiastic support of this new course. We also thank Professors Matthias Heinkenschloss, Yin Zhang and William Symes for very helpful lectures and suggestions.

**Final Report: Course CAAM 513**

**Boundary Condition Approximation and Composite  
Grid Development for Modeling the Gramicidin  
Channel**

**Student:** Nathan Hillson

**Instructor:** Liliana Borcea

May 4, 1998

**Client:** Thad Harroun  
Department of Physics  
Telephone: x4898  
Internet: thad@rice.edu

Now that we have a picture of the domain, we need to write down some notation for the sets we will be discussing shortly. Let us write

$$\Omega^h = \{(x, y) \in \mathbb{R}^2 \mid x/h, y/h \in \mathbb{N}\}, \Omega^H = \{(x, y) \in \mathbb{R}^2 \mid x/H, y/H \in \mathbb{N}\},$$

$$\Omega_c^h = \Omega_l \cap \Omega^h, \Omega_c^H = (\Omega \setminus \Omega_l) \cap \Omega^H, \Omega_c^{h,H} = \Omega_c^h \cup \Omega_c^H,$$

$$\Gamma_{11} = \{(x, y) \in \mathbb{R}^2 \mid x = \gamma_{11}, \gamma_{21} \leq y \leq \gamma_{22}\},$$

$$\Gamma_{12} = \{(x, y) \in \mathbb{R}^2 \mid x = \gamma_{12}, \gamma_{21} \leq y \leq \gamma_{22}\},$$

$$\Gamma_{21} = \{(x, y) \in \mathbb{R}^2 \mid y = \gamma_{21}, \gamma_{11} \leq x \leq \gamma_{12}\},$$

$$\Gamma_{22} = \{(x, y) \in \mathbb{R}^2 \mid y = \gamma_{22}, \gamma_{11} \leq x \leq \gamma_{12}\},$$

$$\Gamma = \Gamma_{11} \cup \Gamma_{12} \cup \Gamma_{21} \cup \Gamma_{22}, \Gamma^h = \Gamma \cap \Omega^h, \Gamma^H = \Gamma \cap \Omega^H, \text{ and}$$

$$\Gamma_h^* = \{(x, y) \in \Omega_c^h \mid \text{dist}((x, y), \Gamma) = h\}.$$

With this notation, we can describe how to solve for the values of all of the points on the composite grid. At the grid points  $\Omega_c^H \cup \Gamma^H$ , we use the standard stencil with coarse unit distance between points equal to  $H$ . At the grid points  $\Omega_c^h \setminus (\Gamma^h \cup \Gamma_h^*)$ , we use the standard stencil with fine unit distance between points equal to  $h$ .

For the remaining grid points, we do not use the standard stencil because our mesh does not include the necessary points outside of  $\Omega_l$ . Instead, we interpolate these classes of points. For the set of grid points  $\Gamma^h \setminus \Gamma^H$ , we have

$$u(x, y) = \text{average}\{u(x', y') \mid (x', y') \in \Gamma^h \cap \{(x', y') \mid \text{dist}((x, y), (x', y')) = h\}\}.$$

In other words, the value at a given grid point in this set is just an average of its immediate neighbors also in  $\Gamma^h$ . Similarly, for the set of grid points  $\Gamma_h^*$  we have

$$u(x, y) = \text{average}\{u(x', y') \mid (x', y') \in \Omega_c^h \cap \{(x', y') \mid \text{dist}((x, y), (x', y')) = h\sqrt{2}\}\}.$$

So, the value for one of these points is just the average value of its diagonal neighbors. With the two types of stencils, and the two types of interpolation schemes, we have a large system that when solved will produce a value at every one of the points in  $\Omega_c^{h,H}$ , which is the entire composite grid.

## Discussion

We examined two techniques that may aid in the theoretical modeling of the gramicidin channel. We approximated the boundary conditions near the channel using a perturbative approach, and developed a composite grid scheme which allows for placing more grid points in the interesting regions of the membrane. It is hoped that these techniques will aid the pursuits of current researchers.

and that

$$\left. \frac{\partial u}{\partial r} \right|_{r_o} = s,$$

we find that  $C_1$  and  $C_2$  are given by the solution of the system

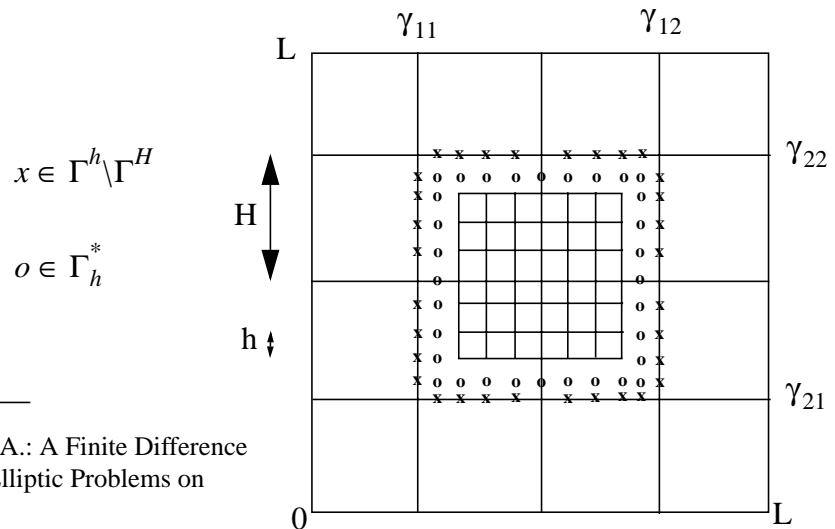
$$\begin{aligned} C_1 + C_2 &= u_o, \\ \alpha_1 C_1 + \alpha_2 C_2 &= s. \end{aligned}$$

Once we know the coefficients  $C_1$  and  $C_2$ , the perturbative solution above, and the radial distance to a grid point very close to the edge of the channel, we can calculate what the deformation should be that point. Note that our perturbative solution matches the boundary conditions and is an excellent approximation to the true solution so long as we are in a small enough neighborhood about the channel. This is much better, and not much more expensive to implement, than the first order approximation to the boundary conditions. The problem with first order is that although it satisfies the boundary conditions, it very quickly diverges from the true deformation.

## Composite Grid Development

It is now time to examine composite grids, the second technique we hope will make a contribution to modeling the gramicidin channel. The basic idea behind a composite grid is that it allows us to place a higher density of grid points in regions of interest, like those immediately between channels, and a lower density in regions where there are no channels nearby<sup>1</sup>. In the following description, we will make several simplifying decisions about the domain we are using. This will allow us to explain the basic ideas with a minimum of encumbering notation, and at the same time without losing generality. The principles are easily extended to more complicated domains and multiple mesh sizes.

Let our domain be given by  $\Omega = [0, L] \times [0, L]$ . Let the composite grid be composed of a global coarse grid covering  $\Omega$  and a local fine grid covering  $\Omega_l = [\gamma_{11}, \gamma_{12}] \times [\gamma_{21}, \gamma_{22}]$ . We only consider coarse grid sizes  $H$  such that  $L/H \in \mathbb{N}$ ,  $\gamma_{11}/H \in \mathbb{N}$ ,  $\gamma_{12}/H \in \mathbb{N}$ ,  $\gamma_{21}/H \in \mathbb{N}$ , and  $\gamma_{22}/H \in \mathbb{N}$ . We only consider fine grid sizes  $h$  such that  $h = H/\sigma$ ,  $\sigma \in \mathbb{N}$ . The figure below depicts the domain.



1. Ferret, P. J. J., Reusken, A. A.: A Finite Difference Discretization Method for Elliptic Problems on Composite Grids.

If we substitute this for the laplacian in the above expression we have

$$\frac{K_C}{2} \left[ \frac{d^4 u}{dr^4} + 2 \frac{d^3 u}{dr^3} - \frac{d^2 u}{dr^2} + \frac{du}{dr} \right] + \frac{B_o}{a} u = 0.$$

Now we can begin our perturbative approach. Let us write the radial distance  $r = r(\rho)$  as

$$r = r_o + \varepsilon \rho,$$

where again  $r_o$  is the radius of the channel,  $\varepsilon \ll 1$  is a constant parameter, and  $\rho$  is the independent variable. If we substitute this in for  $r$  above we have

$$\frac{K_C}{2} \left[ \frac{1}{\varepsilon^4} \frac{d^4 u}{d\rho^4} + \frac{2}{\varepsilon^3 (r_o + \varepsilon \rho)} \frac{d^3 u}{d\rho^3} - \frac{1}{\varepsilon^2 (r_o + \varepsilon \rho)^2} \frac{d^2 u}{d\rho^2} + \frac{1}{\varepsilon (r_o + \varepsilon \rho)^3} \frac{du}{d\rho} \right] + \frac{B_o}{a} u = 0.$$

Now if we have  $\varepsilon \rho \ll r_o$ , we can replace the terms  $r_o + \varepsilon \rho$  with  $r_o$  to get

$$\frac{K_C}{2} \left[ \frac{1}{\varepsilon^4} \frac{d^4 u}{d\rho^4} + \frac{2}{\varepsilon^3 r_o} \frac{d^3 u}{d\rho^3} - \frac{1}{\varepsilon^2 r_o^2} \frac{d^2 u}{d\rho^2} + \frac{1}{\varepsilon r_o^3} \frac{du}{d\rho} \right] + \frac{B_o}{a} u = 0.$$

We know that the solution to this is of the form

$$u = e^{\alpha \rho},$$

where  $\alpha$  solves the roots of the equation

$$\alpha^4 + 2 \left( \frac{\varepsilon}{r_o} \right) \alpha^3 - \left( \frac{\varepsilon}{r_o} \right)^2 \alpha^2 + \left( \frac{\varepsilon}{r_o} \right)^3 \alpha + \frac{2B_o \varepsilon^4}{K_C a} = 0.$$

If we set  $\varepsilon = h$ , where  $h$  is the smallest length unit of our stencil and solve for  $\alpha$  we will have

$$u = C_1 e^{\alpha_1 \rho} + C_2 e^{\alpha_2 \rho} + C_3 e^{\alpha_3 \rho} + C_4 e^{\alpha_4 \rho}.$$

Note that if

$$Real(\alpha_i) > 0 \Rightarrow C_i = 0,$$

since our solution can not blow up as  $\rho \rightarrow \infty$  by the boundary condition at infinity. This means that there will be only two non zero coefficients. So, we can write

$$u = C_1 e^{\alpha_1 \rho} + C_2 e^{\alpha_2 \rho}.$$

If we observe the boundary conditions at  $r = r_o$ , namely that

$$u(r_o) = u_o,$$

ary conditions about each channel

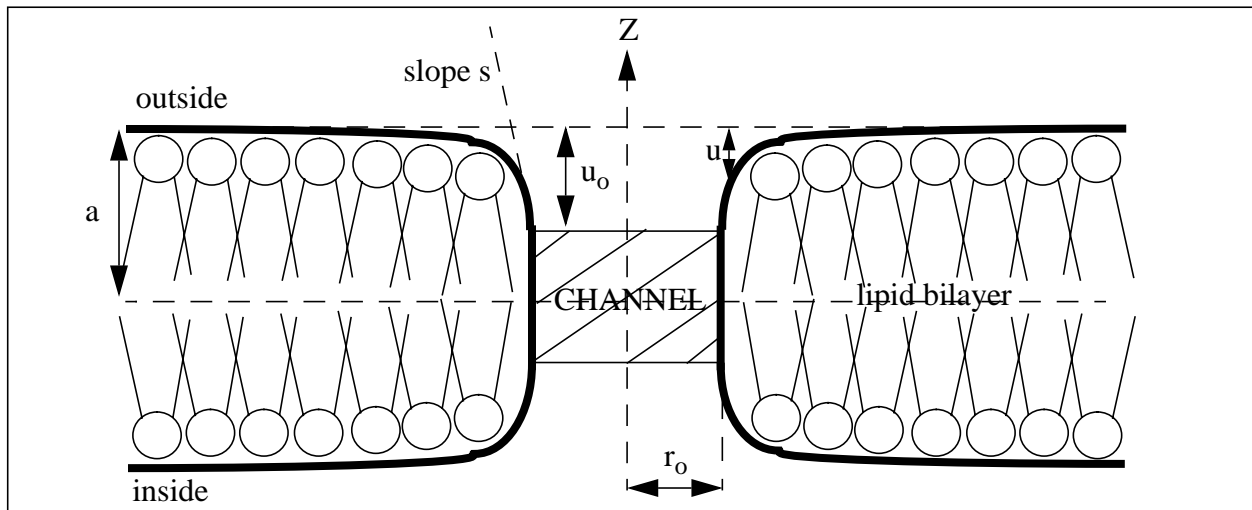
$$u(r_o) = u_o,$$

$$\left. \frac{\partial u}{\partial r} \right|_{r_o} = s,$$

and as we go outwards towards infinity,

$$u, \left. \frac{\partial u}{\partial r} \right|_{r \rightarrow \infty} = 0,$$

where  $r_o$  is the radius of the channel,  $2u_o$  is the change in the bilayer thickness at the edge of the channel, and  $s$  is the slope of the membrane along the edge of the channel. The following figure depicts graphically the situation. The view is a cross section of the channel looking down on the y-z plane.



## Boundary Condition Approximation

Now that we have outlined the basics, it is time to examine the two techniques we hope will make a contribution to modeling the gramicidin channel. The first involves approximating the boundary condition near the actual boundary. One might wonder why this is even necessary. The reason is that we are using finite differences in our calculations, and often times the grid points of the mesh do not exactly coincide with the edge of a given channel. There are grid points near the boundary, but not exactly superimposed on it. Because of this we need to come up with values for the deformation that will approximate the boundary conditions.

In a small enough neighborhood about the edge of a channel, we can reasonably assume that this single channel determines the deformation of the membrane. Taking advantage of this, we can assume a cylindrical symmetry near the channel, and write  $u = u(r)$  to give the radial lapacian

$$\Delta = \frac{1}{r} \frac{\partial}{\partial r} \left[ r \frac{\partial}{\partial r} \right].$$

# Boundary Condition Approximation and Composite Grid Development for modeling the gramicidin channel

Nathan Hillson  
Dr. Borcea  
5/4/98

## Abstract

We examine two techniques that may aid in the theoretical modeling of the lipid bilayer membrane gramicidin channel. First, we approximate the boundary conditions near the channel using a perturbative approach that is superior to first order while maintaining a low cost of implementation. Second, we develop a composite grid scheme which allows for more detail in regions of interest, like those located immediately between two nearby channels, while keeping the data points in other regions to a minimum.

## Introduction

Cells have stringent control over what is allowed to cross their membranes. Molecules are not allowed to come and go as they please. In order to ensure this, cells have a plethora of regulatory mechanisms that allow them to maintain chemical gradients and other conditions that are vital to their existence. When they lose control of their borders, cells either become significantly impaired or cease to function.

Gramicidin channels in the lipid bilayer have been proposed to be one such cause of loss of control. These channels are more or less cylindrical passages that extend from the inside to the outside of the membrane. The first thing to note about these channels is that they are not very large, only a few angstroms in diameter. While the smallest of molecules might be allowed to pass through them, it is very unlikely that large hazardous chemicals could penetrate them or that the cells would lyse as a consequence. The gramicidin channels have more of an effect because they distort the surface of the membrane nearby them. It is possible that when many of the channels aggregate, they can collectively cause such a disturbance as to disrupt gap junctions or other types of transmembrane channels, forcing them open or closed.

Research on the role that the gramicidin channels play in the disruption of the lipid bilayer is currently underway. The basic theoretical model consists of finding the minimization of the total deformation energy of the membrane induced by the gramicidin channels. The equation that governs the deformation such that the energy is minimized is given by

$$\frac{K_C}{2}\Delta^2 u + \frac{B_o}{a}u = 0,$$

where  $K_C$  is the elastic coefficient of splay,  $B_o$  is the elastic coefficient of compression,  $2a$  is the standard bilayer thickness,  $2u(x, y)$  is the change in the bilayer thickness and  $\Delta u = \nabla^2 u$  is the laplacian operator. In addition to minimizing the energy, the deformation must satisfy the bound-

**Final Report: Course CAAM 513**

**Particle Aggregation Problem**

**Student:** Genetha Gray

**Instructor:** Liliana Borcea

May 4, 1998

**Client:** Julia Aziz  
Environmental Engineering  
Telephone: x2450  
Internet: jaziz@ruf.rice.edu

# 1 Introduction

A graduate student in the environmental engineering department contacted the consulting class to get advice about a problem dealing with particle aggregation. The problem attempts to describe the change over time of the concentration of particles of a given size in an aggregating heterodisperse suspension. The results of such an equation are of particular interest to the water treatment industry. Many of the processes used in treating water depend on the aggregation of specific substances.

## 2 The Original Equation

The environmental engineers started with the following equation, which was proposed in the early 1900s:

$$\frac{dn_k}{dt} = \frac{1}{2}\alpha \sum_{i+j \rightarrow k} K(i,j)n_in_j - n_k \sum_i K(i,k)n_i. \quad (2.1)$$

The initial conditions are determined by the problem. For example, there may be no particles in the system initially. Then, particles could be added to the aggregating medium continuously over time or all at once at a given time. In cases where the initial concentrations are not zero, the engineers may take samples to determine the initial conditions. In the laboratory, the initial concentration of the particles can be specified to suit the system being modeled. The time span is also determined by what type of system is being studied.

To understand this equation, we first had to identify and describe its variables. Particles sizes are defined by their diameters,  $d_k$ . The concentration of particles of size  $d_k$  is  $n_k$ . Collision rates between particles of size  $d_i$  and  $d_j$  are given by  $K(i,j)$ . Knowing these variables, we could then examine each term. The first term represents particles of size  $dk$  that are created by collisions of other sized particles. The coefficient  $\frac{1}{2}$  accounts for the fact that  $K(i,j) = K(j,i)$ . In other words, it assures that particles created are not counted twice. Similarly, the second term accounts for particles of size  $d_k$  lost due to collisions with other sized particles. Hence, the rate of change in concentration of particles of diameter  $dk$  is equal to the concentration of particles gained minus the concentration of those lost.

## 3 The Modified Equation

The above equation was modified to include a way to monitor the location of the particles. The aggregating medium was divided into horizontal layers. Each layer was given a number. So,  $n_{k,l}$  is the concentration of particles with diameter  $d_k$  in layer  $l$ . This additional piece of information increases the size of the problem dramatically. At each time step, the engineers now want to find  $KL$  concentration numbers (where  $K$  is the total number of particle sizes and  $L$  is the total number of layers). The modified equation is:

$$\frac{dn_{k,l}}{dt} = \frac{1}{2}\alpha \sum_{i+j \rightarrow k} K(i,j)n_in_j - n_{k,l}\alpha \sum_i B(i,k)n_i + n_{k,l-1}\frac{v_{sk}}{l} - n_{k,l}\frac{v_{sk}}{l} + \text{break-up term}. \quad (3.2)$$

The initial conditions are still set by the engineers as described for the original equation. The additional terms describe particle settling. Each layer gains particles from the layer directly above it and loses particles to the layer directly below it. The variable  $v_{sk}$  is the velocity at which the particles descend. The specifics of the break up term are still under discussion. The engineers need it to correct the equation and account for activity between particles that are not in the same layer.

## 4 Client's Current Solver

The client approached our group with a program currently used by some of the other engineers to solve this particle aggregation problem. It is based on forward Euler method. The environmental engineers are dissatisfied with the speed of the algorithm and the quality of its results. After learning a little more about the process of particle aggregation, we were able to explain their discontent. When most aggregation procedures begin, there is a lot of activity that occurs very quickly. The time steps must be quite small in order to describe the system accurately. However, as the process progresses, the amount of activity decreases and slows down. Hence, the engineers are working with a stiff system of differential equations. Euler's method works well so long as the time steps are very small. It becomes quite unstable if the time steps are too large. In other words, small errors in the algorithm are magnified resulting in a large global error.

## 5 Advice of the Consulting Group

The main consideration in choosing an appropriate method for this problem was the stiffness of the system of differential equations. We needed to recommend a method that performed well for stiff systems. After some consideration, we decided on a predictor-corrector scheme. So, we introduced our client to the fourth order Adams-Bashforth predictor and Adams-Moulton corrector. The order of a method determines its truncation error. We choose fourth order based on the accuracy that the engineers were looking for. In addition, we also explained how to use fourth order Runge-Kutta to obtain the initial points needed to apply the predictor-corrector. Eventually, the client would like to write a code in the C programming language to solve the equation. However, we suggested that our client first practice and become familiar with the new method. To aid in this learning process, we furnished a Matlab code.

The fourth order Adams-Moulton scheme belongs to a group of differential equation solvers called multistep methods. It is a particularly useful method because it combines an explicit and an implicit method allowing the user the benefits of both kinds of methods. In particular, it inherits the desirable stability, accuracy, and convergence properties of implicit methods. The presence of this stability will help the engineers obtain more accurate results.

## 6 Additional Advice

Adaptive time step is another important improvement that can be made in the solving of this algorithm. By studying the history of the solution at each time step, the algorithm can be adjusted to amend the size of the time step if necessary. Thus, only a minimal number of time steps will be taken. Furthermore, a nonuniform mesh will allow the user to take very small steps at the beginning of aggregation and larger ones as the process settles down. In an effort not to overwhelm our client, we decided to start with a uniform time discretization. Once our client masters the algorithm using a uniform mesh, he can then adapt the mesh to fit the needs of the aggregation problem.

## 7 Conclusion and Discussion

There is much work to be done by our client and his colleagues in the environmental engineering department to prepare their code. It will be a while before they will be able to examine the quality of their new code and see how accurately it represents their data. The consulting group also feels

that there are a few other things to be considered about the equation itself. For example, where do the particles eventually end up? Is there a way that particles leave the system? If not, the layers will eventually be filled with particles and the numerical solution to the equation will blow up. Also, there is the question of how particles are added to the system. Are they added continuously or are they added all at one time? The equation should reflect this. We realize that modeling real world phenomena in the laboratory is very difficult. Describing results numerically is also a complex process.

This project taught us that despite the fact that there are many methods available to solve differential equations, not every method works well in every case. The properties of the equation being solved will help determine the benefits and drawbacks of each method. It also gave us the opportunity to learn a little about the applications of numerical methods in environmental engineering.

## References

- [1] Kincaid, David and Cheney, Ward, *Numerical Analysis*, Second Edition, Brooks/Cole Publishing, Pacific Grove, CA, 1996.
- [2] Miranker, Willard, *Numerical Methods for Stiff Equations*, D. Riedel Publishing, Dordrecht, Holland, 1981.
- [3] Wiesner, Mark, *Kinetics of Aggregate Formation in Rapid Mix*, Water Resources, volume 26, no. 3, pp. 379-387, 1992.

**Final Report: Course CAAM 513**

**Solving a Constrained Differential Equation for  
Modeling Air Flow through the Lungs**

**Student:** Shannon Walsh

**Instructor:** Liliana Borcea

May 4, 1998

**Client:** Dr. Niranjan, S. C.  
Elec. & Comp. Engineering MS366  
Telephone: 713-527-8101 x3559  
Internet: niranjan@rice.edu

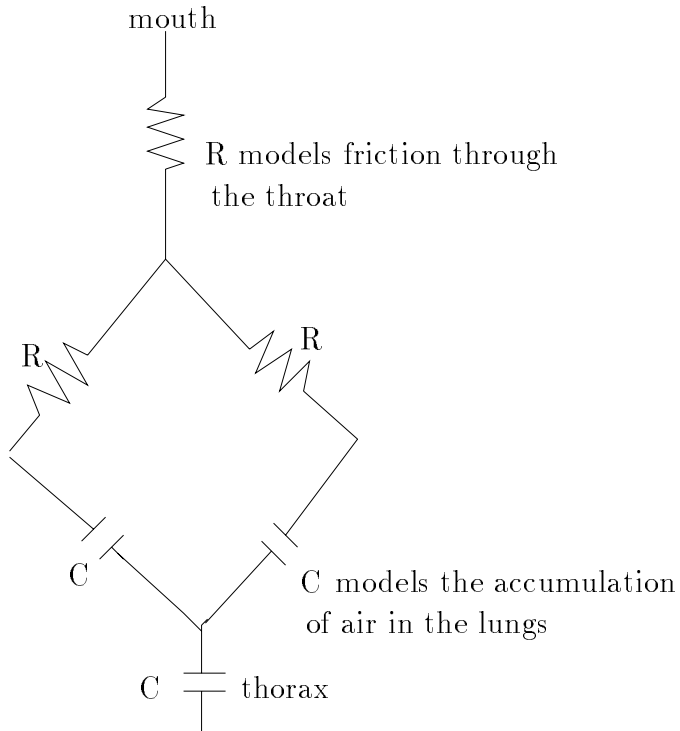


Figure 1: Representation of lumped, two-compartment mechanics

## 1 Introduction

The client, a member of the Electrical and Computer Engineering Department at Rice University, is conducting research on the air flow from the mouth to the lungs within the human body. He is utilizing a model which simulates this air flow as current within a network. In figure 1 we show a simple diagram of his model. After writing the flow equations for this resistor-capacitor network, the mathematical statement of the problem is:

Solve

$$x'(t) = f(x(t), t), \quad (1.1)$$

where  $x$  satisfies the constraint  $\phi(x) = 0$ . The variable  $x(t)$  represents the volume of air at a given point in time  $t$ , while  $x'(t)$  is the rate of the air flow. The constraint  $\phi(x) = 0$  refers to whether or not both lungs are functioning correctly. If both lungs are "healthy" then  $\phi(x) = 0$ , otherwise  $\phi(x) \neq 0$ . Suppose that we are modeling the flow of air through healthy lungs. Then we must solve (1), where the right hand side  $f$  is given by  $f_1$  if air is inspired and  $f_2$  if air is expired. Therefore, we must solve constrained initial value problems such as:

$$\begin{aligned} x'(t) &= f_i(x(t), t) \quad i = 1, 2, \quad t \in [t_0, t_f] \\ \phi(x) &= 0 \\ x(t_0) &= x_0 \text{ is given.} \end{aligned} \quad (1.2)$$

For air flow through diseased lungs, we have:

$$\begin{aligned}x'(t) &= g_i(x(t), t) \quad i = 1, 2, \quad t \in [t_0, t_f] \\ \psi(x) &= 0 \\ x(t_0) &= x_0 \text{ is given,}\end{aligned}\tag{1.3}$$

where  $g_1$  and  $g_2$  refer to inspiration and expiration, respectively. The new constraint  $\psi(x)$  comes from:  $\phi(x) = D = \text{constant}$  for the diseased lungs, or, equivalently,  $\psi(x) = \phi(x) - D = 0$ .

## 2 Method

### 2.1 The Method Which the Client Had Employed

Before he came to see us, the client was using a fourth-order Runge-Kutta method to solve (1.2). However, at each step during the integration process, he checked the constraint  $\phi(x) = 0$ . If the constraint was not satisfied, then he changed the right hand side of the equation from  $f$  to  $g$ . Clearly, this is unacceptable because a lung does not change from healthy to diseased in a small time step. Even worse, during a time step, while evaluating the slopes needed by Runge-Kutta, he changed from  $f$  to  $g$  if  $\phi(x) \neq 0$ . As expected, this led to incorrect results.

### 2.2 The Method We Proposed

General discussion: Suppose that we are given a first order system:

$$F(t, y(t), y'(t)) = 0,\tag{2.4}$$

where  $F$  and  $y$  are vector valued. Often, we have (2.4) in the explicit form:

$$y'(t) = f(t, y(t)).\tag{2.5}$$

There are times, though, when we would want to work directly with (2.4). For instance, rewriting (2.4) as (2.5) may be impossible. In this case, we defer to using differential-algebraic equations (DAE's) rather than ordinary-differential equations (ODE's). There are four basic types of DAE's:

1. Linear constant coefficient
2. Linear time varying (fully-implicit)
3. Linear time varying (semi-explicit)
4. Nonlinear time varying (semi-explicit)

A semi-explicit time varying DAE has the form:

$$\begin{aligned}x'(t) &= f(x(t), y(t), t) \\ 0 &= g(x(t), y(t), t).\end{aligned}\tag{2.6}$$

By differentiating the constraint equation in (2.6) with respect to  $t$  we arrive at:

$$\begin{aligned}x'(t) &= f(x(t), y(t), t) \\ g_x(x(t), y(t), t)x'(t) + g_y(x(t), y(t), t)y'(t) &= -g_t(x(t), y(t), t).\end{aligned}\tag{2.7}$$

If  $g_y$  is invertible, then (2.7) can be written as a system of ODE's and the semi-explicit DAE (2.6) has index one.

**Definition:** The minimum number of times that all or part of (2.4) must be differentiated with respect to  $t$  in order to determine  $y'(t)$  as a continuous function of  $y(t)$  and  $t$  is the index of the DAE.

**Specific Case:** We then apply the general principles to the problem at hand:

$$\begin{aligned}x'(t) &= f(x(t), t) \\ \phi(x) &= 0.\end{aligned}\tag{2.8}$$

To enforce the constraint we solve the semi-explicit DAE:

$$\begin{aligned}x'(t) &= f(x(t), t) + (\phi_x)^T \mu(t) = h(x(t), \mu(t), t) \\ 0 &= \phi(x(t), t),\end{aligned}\tag{2.9}$$

where  $\mu(t)$  is a Lagrange multiplier. Problem (2.9) has the solution:  $x(t)$  solves (2.8) and  $\mu(t) = 0$  (see [1]).

In order to determine the index of (2.9) we differentiate the constraint equation with respect to  $t$ :

$$\begin{aligned}x'(t) &= h(x(t), \mu(t), t) \\ \phi_x x'(t) + \phi_t + \phi_\mu \mu'(t) &= 0.\end{aligned}\tag{2.10}$$

Note:  $\phi_\mu = 0$  because there is no dependence on  $\mu(t)$  and, because zero is not invertible, this problem is not index one. We differentiate the constraint equation again and find that the DAE is of index two [i.e. this time the coefficient (a full low rank square matrix) of  $\mu'(t)$  is invertible].

**Solving the DAE:** The backward differentiation formulas (BDF's) are the most popular of the linear multistep methods (LMMS) for solving DAEs. Our semi-explicit nonlinear system (2.9) is of the form:

$$\begin{aligned}F(x(t), x'(t), \mu(t), t) &= 0 \\ \phi(x(t), t) &= 0,\end{aligned}\tag{2.11}$$

where we know that:

1. (2.11) has index two.
2. The inverse of  $(F_{x'})$  exists and it is bounded in a neighborhood of the solution.
3.  $F$  and  $\phi$  have as many partial derivatives as desired in a neighborhood of the solution.

To solve (2.11) we propose the one-step BDF: This method involves using the implicit Euler method and replacing  $x'(t)$  with a backward difference.

We construct a uniform time discretization  $t_0, t_1, t_2, \dots, t_f$  with time step  $h$ . For the time  $t_n = nh$  we have:

$$\begin{aligned}F\left[x(t_n), \frac{x(t_n) - x(t_{n-1})}{h}, \mu(t_n), t_n\right] &= 0 \\ \phi(x(t_n), t_n) &= 0\end{aligned}\tag{2.12}$$

which we solve with Newton's method for  $x(t_n)$  and  $\mu(t_n)$ .

### 3 Results

I met with the client on Monday, March 23, to go over the method described in this report. He quickly grasped the concept and even assured me that he himself had already been thinking about using *DAE's* to solve his problem. He is currently working on the implementation of this solution.

### References

- [1] Brenan, Campbell, and Petzold, Numerical Solutions of Initial-Value Problems in Differential-Algebraic Equations. Society for Industrial and Applied Mathematics; New York, 1996.

**Final Report: Course CAAM 513**

**An optimal control problem for the analysis of the cooperation between investors in a common project**

**Students:** Genetha Gray, Shannon Walsh, Nathan Hillson

**Instructor:** Liliana Borcea

May 4, 1998

**Client:** Professor Fouad El Ouardighi  
Groupe ESSEC, Logistics and Production Dept.,  
Av. B. Hirsch, B.P. 105  
95021 Cergy Pontoise Cédex, France  
internet: felouardighi@escna.fr

# 1 Introduction

We consider the following deterministic optimal control problem: Maximize over  $a_1(t)$  and  $a_2(t)$  the functional  $J(a_1(t), a_2(t))$ , where

$$J(a_1(t), a_2(t)) = \int_0^{\infty} e^{-rt} [f(A(t), B(t)) - g(a_1(t), a_2(t))] dt \quad (1.1)$$

and

$$A'(t) = h(a_1(t), a_2(t)) - k(B(t))A(t), \quad A(0) = A_0 \quad (1.2)$$

$$B'(t) = m(a_1(t), a_2(t)) - n(A(t))B(t), \quad B(0) = B_0.$$

Equations (1.1) and (1.2) model the problem of cooperation between two players that invest in a common project. The unknown functions  $a_1(t)$  and  $a_2(t)$  represent the investments of each player in time  $t$  and the functions  $A(t)$  and  $B(t)$  that obey (1.2) “describe” the project. In (1.1),  $r > 0$  is a discount rate. Furthermore, the functions  $f(\cdot)$ ,  $g(\cdot)$ ,  $h(\cdot)$ ,  $k(\cdot)$ ,  $m(\cdot)$  and  $n(\cdot)$  are positive and the unknown functions  $a_1$  and  $a_2$  are constrained to the intervals  $a_1 \in [0, a_{1Max}]$  and  $a_2 \in [0, a_{2Max}]$ . We are also given the following conditions:

$$\begin{aligned} f_A, f_B > 0, \quad f_{AA}, f_{BB} < 0, \quad f_{AB} = f_{BA} = 0 \\ g_{a_1}, g_{a_2}, g_{a_1a_1}, g_{a_2a_2} > 0, \quad g_{a_1a_2} = g_{a_2a_1} = 0 \\ h_{a_1}, h_{a_2} > 0, \quad h_{a_1a_1}, h_{a_2a_2} < 0, \quad h_{a_1a_2} = h_{a_2a_1} \geq 0 \\ k_B > 0, \quad k_{BB} < 0 \\ m_{a_1}, m_{a_2} < 0, \quad m_{a_1a_1}, m_{a_2a_2} \leq 0, \quad m_{a_1a_2} = m_{a_2a_1} \leq 0 \\ n_A > 0, \quad n_{AA} < 0. \end{aligned} \quad (1.3)$$

## 2 Proposed Methodology

We suggest the following steps towards the solution of the problem.

### 2.1 Checking the concavity of the functional J

The Hessian matrix

$$\nabla^2 F(a_1, a_2)_{i,j} = \frac{\partial^2 F(a_1, a_2)}{\partial a_i \partial a_j}, \quad i, j = 1, 2 \quad (2.4)$$

where

$$F(a_1, a_2) = f(A(t), B(t)) - g(a_1(t), a_2(t)) \quad (2.5)$$

is given by

$$\begin{aligned} \nabla^2 F_{11} &= f_{AA}(A_{a_1})^2 + f_{BB}(B_{a_1})^2 - g_{a_1a_1} + f_A A_{a_1a_1} + f_B B_{a_1a_1} \\ \nabla^2 F_{22} &= f_{AA}(A_{a_2})^2 + f_{BB}(B_{a_2})^2 - g_{a_2a_2} + f_A A_{a_2a_2} + f_B B_{a_2a_2} \\ \nabla^2 F_{12} &= \nabla^2 F_{21} = f_{AA} A_{a_1} A_{a_2} + f_{BB} B_{a_1} B_{a_2} + f_A A_{a_1a_2} + f_B B_{a_1a_2}. \end{aligned} \quad (2.6)$$

If the Hessian is negative definite at all times, then the problem is concave and the maximizers  $a_1$  and  $a_2$  are unique. If not, the functional  $J$  could have many maxima and finding the global solution is very difficult.

To study the sign of the Hessian, we must look at the sign of the expressions  $\nabla^2 F_{11}$  and  $(\nabla^2 F_{12})^2 - \nabla^2 F_{11} \nabla^2 F_{22}$ . If both expressions are  $\leq 0$ , the Hessian is negative definite and the functional  $J$  has a unique maximizer. To determine the sign, one must use the conditions (1.3) and the differential equations (1.2).

## 2.2 Study of the continuum problem

Although in the end we recommend to discretize the equations for the purpose of numerical computations, much insight can be gained from the study of the continuum problem. Thus, we look at the stationarity condition

$$\nabla J = 0 \tag{2.7}$$

that must be satisfied at the solution. The hope here is to obtain from (2.7) differential equations for  $a_1$  and  $a_2$  in time. For the purpose of the illustration of the ideas proposed, we consider the unconstrained scalar problem, where  $J = J(a(t))$  and  $a(t) \in R$ . To compute the Jacobian, let us assume that we take a small variation  $\delta a$  in  $a$ . Then,

$$\delta J_T = \int_0^T e^{-rt} [f_A \delta A + f_B \delta B - g' \delta a] dt, \tag{2.8}$$

where  $T$  is later taken to infinity. To calculate the Jacobian, it is clear that we must factor  $\delta a$  in (2.8). We take derivatives of both sides of equations (1.2) in the direction  $\delta a$  and obtain

$$(\delta A)' = h_a \delta a - k_{BA} \delta B - k \delta A \tag{2.9}$$

$$(\delta B)' = m_a \delta a - n_A B \delta A - n \delta B,$$

or, equivalently,

$$\frac{d}{dt} \begin{pmatrix} \delta A \\ \delta B \end{pmatrix} = M \begin{pmatrix} \delta A \\ \delta B \end{pmatrix} + \begin{pmatrix} h_a \\ m_a \end{pmatrix} \delta a, \quad M = \begin{pmatrix} -k & -k_{BA} \\ -n_A B & -n \end{pmatrix}. \tag{2.10}$$

Here we use the notation

$$\delta A = \lim_{\delta a \rightarrow 0} \frac{A(a + \delta a) - A(a)}{|\delta a|}$$

and we assume Frechet differentiability of  $h$  and  $m$ .

We define the adjoint equations

$$\frac{d}{dt} \begin{pmatrix} p \\ q \end{pmatrix} = -M^T \begin{pmatrix} p \\ q \end{pmatrix} + \begin{pmatrix} f_A e^{-rt} \\ f_B e^{-rt} \end{pmatrix}, \quad \begin{pmatrix} p(T) \\ q(T) \end{pmatrix} = \mathbf{0} \tag{2.11}$$

and, from (2.8) we have, after a simple integration by parts:

$$\delta J_T = \frac{dJ_T}{da} \delta a = - \int_0^T (e^{-rt} g' + p h_a + q m_a) \delta a dt. \tag{2.12}$$

Since (2.12) must hold for any  $\delta a$ , the stationarity condition implies the following implicit equation satisfied by  $a(t)$ :

$$e^{-rt} g' + p h_a + q m_a = 0, \quad \text{for } t \in [0, T]. \tag{2.13}$$

A similar calculation can be done in the vector case, where  $a(t) = (a_1(t), a_2(t))$ . The negativity of the Hessian at the solution should be explored as well. This line of thought can prove very useful in some situations and it can give insight into the behavior of the solution.

### 2.3 Study of the discretized problem

To solve numerically the time control problem (1.1)-(1.2) we limit ourselves to the interval  $t \in [0, T]$ , where  $T$  will be assigned a large value. We discretize the time as

$$t_k = k\Delta t, \quad k = 0, 1, \dots, N \quad N\Delta t = T$$

and, by using some quadrature rule (for example trapezoidal) for the integral in (1.1), we obtain

$$\max \sum_{k=0}^N \phi_k(A_k, B_k, a_{1k}, a_{2k}), \quad (2.14)$$

where  $A_k = A(t_k)$  and  $B(t_k)$  are recursively defined by

$$\begin{cases} A_{k+1} = \psi_k(A_k, B_k, a_{1k}, a_{2k}) \\ B_{k+1} = \chi_k(A_k, B_k, a_{1k}, a_{2k}) \end{cases}, \quad k = 0, 1, \dots, N-1. \quad (2.15)$$

Equations (2.15) are obtained by solving (1.2) with Euler's method, for example.

Then, we solve the discrete problem

$$\max_{a_{1k}, a_{2k}, \lambda_k, \eta_k} L(A_0, \dots, A_N, B_0, \dots, B_N, a_{10}, \dots, a_{1N}, a_{20}, \dots, a_{2N}, \lambda_0, \dots, \lambda_{N-1}, \eta_0, \dots, \eta_{N-1}), \quad (2.16)$$

where the Lagrangian is

$$\begin{aligned} L(A_0, \dots, A_N, B_0, \dots, B_N, a_{10}, \dots, a_{1N}, a_{20}, \dots, a_{2N}, \lambda_0, \dots, \lambda_{N-1}, \eta_0, \dots, \eta_{N-1}) = \\ \sum_{k=0}^N \phi_k(A_k, B_k, a_{1k}, a_{2k}) - \sum_{k=0}^{N-1} \{ \lambda_k [A_{k+1} - \psi_k(A_k, B_k, a_{1k}, a_{2k})] + \eta_k [B_{k+1} - \chi_k(A_k, B_k, a_{1k}, a_{2k})] \}. \end{aligned} \quad (2.17)$$

In (2.17) we do not account for the constraints

$$a_{ik} \in [0, a_{iMax}], \quad i = 1, 2, \quad k = 0, 1, \dots, N. \quad (2.18)$$

However, these conditions can be included in the Lagrangian (2.17) as follows: We define  $q_{ik} = a_{ik} \geq 0$  and  $p_{ik} = a_{iMax} - a_{ik} \geq 0$  for  $i = 1, 2, \quad k = 0, 1, \dots, N$ . Then, we modify the Lagrangian by adding to (2.17) the term

$$\sum_{i=1}^2 \sum_{k=0}^{N-1} (\nu_{ik} p_{ik} + \mu_{ik} q_{ik}).$$

For simplicity, let us consider the unconstrained problem for which we have the Lagrangian (2.17).

At a maximizer, the Karush-Kuhn-Tucker (KKT) first order necessary conditions are: *The state equations*

$$\begin{aligned} \partial_{\lambda_k} L = 0 &\rightarrow A_{k+1} = \psi_k \\ \partial_{\eta_k} L = 0 &\rightarrow B_{k+1} = \chi_k \end{aligned} \quad (2.19)$$

where  $k = 0, 1, \dots, N-1$  and  $A_0, B_0$  are given. *The adjoint equations*

$$\partial_{A_k} L = 0, \quad \partial_{B_k} L = 0, \quad k = 1, 2, \dots, N, \quad (2.20)$$

or, equivalently,

$$\begin{aligned}
\lambda_{k-1} &= \partial_{A_k} \phi_k + \lambda_k \partial_{A_k} \psi_k + \eta_k \partial_{A_k} \chi_k \\
\eta_{k-1} &= \partial_{B_k} \phi_k + \lambda_k \partial_{B_k} \psi_k + \eta_k \partial_{B_k} \chi_k \\
\lambda_{N-1} &= \partial_{A_N} \phi_N \\
\eta_{N-1} &= \partial_{B_N} \phi_N.
\end{aligned} \tag{2.21}$$

The gradient equations

$$\partial_{a_{ik}} L = 0, \quad i = 1, 2, \quad k = 0, 1 \dots N \tag{2.22}$$

or

$$\begin{aligned}
\partial_{a_{ik}} \phi_k + \lambda_k \partial_{a_{ik}} \psi_k + \eta_k \partial_{a_{ik}} \chi_k &= 0, \quad k = 0, 1, \dots, N-1 \\
\partial_{a_{iN}} \phi_N &= 0.
\end{aligned} \tag{2.23}$$

The maximization (2.16) will be done with Newton's method. This means that we must compute numerically  $\nabla L = (\partial_{a_{10}} L, \dots, \partial_{a_{1N}} L, \partial_{a_{20}} L, \dots, \partial_{a_{2N}} L)^T$  and the Hessian, which, for simplicity of notation, is called  $H$ . Then, the update is done as

$$\mathbf{a} = \mathbf{a} + \Delta \mathbf{a}, \quad H \Delta \mathbf{a} = \nabla L, \tag{2.24}$$

where  $\mathbf{a} = (a_{10}, \dots, a_{1N}, a_{20}, \dots, a_{2N})^T$ .

The numerical computation of  $\nabla L$  is done as follows: Given a current  $\mathbf{a}$ , compute  $A_k$  and  $B_k$ ,  $k = 1, \dots, N$  from (2.19). Next, compute  $\lambda_k$  and  $\eta_k$  from the adjoint equations (2.21) and, finally, calculate  $\nabla L$  as

$$\begin{aligned}
\partial_{a_{ik}} L &= \partial_{a_{ik}} \phi_k + \lambda_k \partial_{a_{ik}} \psi_k + \eta_k \partial_{a_{ik}} \chi_k, \quad i = 1, 2 \quad k = 0, 1, \dots, N-1 \\
\partial_{a_{iN}} L &= \partial_{a_{iN}} \phi_N
\end{aligned} \tag{2.25}$$

The calculation of the Hessian requires, as expected, additional steps. However, the first two steps are identical to the above, ie. we use  $\lambda_k$ ,  $\eta_k$ ,  $A_k$  and  $B_k$  from the gradient calculation above. Suppose that we want to calculate  $\begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{pmatrix} = H \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix}$ , where  $\mathbf{v}_i \in R^{N+1}$ . Then, we calculate the vectors  $\mathbf{w}_i = (w_{i0}, \dots, w_{iN})$ ,  $i = 1, 2$  from

$$\begin{aligned}
w_{1k+1} &= \partial_{A_k} \psi_k w_{1k} + \partial_{a_{1k}} \psi_k v_{1k} \\
w_{2k+1} &= \partial_{B_k} \psi_k w_{2k} + \partial_{a_{2k}} \psi_k v_{2k} \\
w_{i0} &= 0.
\end{aligned} \tag{2.26}$$

Next, we calculate the vectors  $\mathbf{x}_i = (x_{i0}, \dots, x_{iN})$ ,  $i = 1, 2$ . We define the functions  $G_k = \lambda_k \psi_k - \phi_k$  and  $E_k = \eta_k \chi_k - \phi_k$ . Then,

$$\begin{aligned}
x_{1k} &= \partial_{A_k} \psi_k x_{1k+1} + \partial_{A_k A_k} G_k w_{1k} - \partial_{A_k a_{1k}} G_k v_{1k}, \quad k = N-1, \dots, 1 \\
x_{2k} &= \partial_{B_k} \chi_k x_{2k+1} + \partial_{B_k B_k} E_k w_{2k} - \partial_{B_k a_{2k}} E_k v_{2k}, \quad k = N-1, \dots, 1 \\
x_{1N} &= \partial_{A_N A_N} \phi_N w_{1N} - \partial_{A_N a_{1N}} \phi_N v_{1N} \\
x_{2N} &= \partial_{B_N B_N} \phi_N w_{2N} - \partial_{B_N a_{2N}} \phi_N v_{2N}.
\end{aligned} \tag{2.27}$$

Finally,  $\mathbf{z}_1$  and  $\mathbf{z}_2$  are given by

$$\begin{aligned}
z_{1k} &= -\partial_{a_{1k}} \psi_k x_{1k+1} - \partial_{a_{1k} A_k} G_k w_{1k} + \partial_{a_{1k} a_{1k}} G_k v_{1k} \quad k = 0, 1, \dots, N-1 \\
z_{2k} &= -\partial_{a_{2k}} \chi_k x_{2k+1} - \partial_{a_{2k} B_k} E_k w_{2k} + \partial_{a_{2k} a_{2k}} E_k v_{2k} \quad k = 0, 1, \dots, N-1 \\
z_{1N} &= -\partial_{a_{1N} A_k} \phi_N w_{1N} + \partial_{a_{1N} a_{1k}} \phi_N v_{1N} \\
z_{2N} &= -\partial_{a_{2N} B_k} \phi_N w_{2N} + \partial_{a_{2N} a_{2k}} \phi_N v_{2N}.
\end{aligned} \tag{2.28}$$

### 3 Discussion

The methods presented in this report should be explored towards the solution of the control problem (1.1)-(1.2). Until now, the information given to the consulting service is not sufficient to allow us to give more precise solutions.

### Acknowledgements

We thank Professors M. Heinkenschloss and Y. Zhang for a wonderful and very helpful lecture on this topic.

### References

- [1] Heinkenschloss, M., *CAAM 454 Lecture Notes*, 1998.
- [2] Dennis, J. E., Schnabel, R. B., *Numerical methods for unconstrained optimization and nonlinear equations*, SIAM, PA 1983.
- [3] Fletcher, R., *Practical methods of optimization*, John Wiley & Sons, Chichester, 1987.
- [4] Ortega, J. M., Rheinboldt, W. C., *Iterative solution of nonlinear equations in several variables*, Academic Press, Inc., Boston, 1970.