

**Pacific
Institute**
for the Mathematical Sciences

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

Proceedings of the fifth
**PIMS Graduate Industrial
Math Modelling Camp**

May 18–23, 2002, Simon Fraser University

Co-sponsored by:

**The Natural Science and
Engineering Research Council
of Canada**

and

**The British Columbia Information,
Science and Technology Agency**

Editor: Manfred Trummer, Simon Fraser University

FOREWORD BY THE PIMS DIRECTOR

The annual PIMS **Graduate Industrial Math Modelling Camp (GIMMC)** is held in one of the PIMS universities as part of the PIMS Industrial Forum. It is part of PIMS commitment to providing training for young mathematical scientists who are either pursuing careers in academia or in industry.

The goal of the GIMMC is to provide experience in the use of mathematical modelling as a problem solving tool for graduate students in mathematics, applied mathematics, statistics and computer science. In addition to this it helps prepare them for the **Industrial Problem Solving Workshop** which is the other component of the PIMS Industrial Forum.

At the workshop students work together in teams, under the supervision of invited mentors. Each mentor poses a problem arising from an industrial or engineering application and guides his or her team of graduate students through a modelling phase to a resolution.

The Fifth GIMMC was held May 18–23, 2002, at Simon Fraser University. There were seven problems posed, with a total of 56 students in attendance. The students mainly came from all across North America with 14 from the United States.

My sincere appreciation and gratitude goes to everyone involved in this workshop, in particular I wish to thank the organisers (Jack Macki, Chris Bose, Randy LeVeque, Huaxiong Huang, Marc Paulhus, Manfred Trummer, Ian Frigaard) and mentors (Tim Myers, Chris Budd, Brett Stevens, Petra Berenbrick, Brian Wetton, Alexander Melnikov, Yongji Tan).

I look forward to the 2003 GIMMC which will be held at the Banff International Research Station.

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

EDITOR'S PREFACE

The PIMS site office at Simon Fraser University was pleased to host the 5th PIMS Graduate Industrial Mathematics Modelling Camp from May 18–23, 2002.

Fifty-five highly motivated graduate students from North America participated in the event. The students' curiosity and enthusiasm is an important ingredient for the success of the workshop. I hope the project reports in these proceedings convey the level of dedication of all participants.

GIMMC relies very much on the selfless efforts of the mentors. We were very fortunate to have found seven dedicated researchers who made this event a big success: Petra Berenbrink from Simon Fraser University, Chris Budd from the University of Bath in England, Alexander Melnikov from the University of Alberta, Tim Myers from the University of Cape Town, Brett Stevens from Carleton University, Yongji Tan from Fudan University in China, and last but not least, Brian Wetton from the University of British Columbia.

Marc Paulhus (University of Calgary) made the organisation of this event seem effortless; Ian Frigaard from the University of British Columbia, one of the organisers of the PIMS Industrial Problem Solving Workshop held in the week after the camp, helped tremendously in the selection of students—a difficult task given the large number of excellent student applicants.

The Modelling Camp and the Industrial Problem Solving Workshop benefited greatly from the scheduling around the MITACS (Mathematics of Information Technology and Complex Systems) Annual General Meeting. Students were able to present their modelling workshop project as a poster during the MITACS AGM.

The PIMS facilities at SFU provided the mentors and students with nice meeting spaces and excellent computing facilities. The PIMS Administrative Assistant at SFU, Ms. Fuyuko Kitazawa, made sure that all local arrangements were well taken care of. The first few days of the workshop fell on the long Victoria Day weekend; virtually all food outlets on campus were closed. We were extremely fortunate to have Arthur Giovinazzo, former proprietor of the world renowned No Pop Sandwich Shop in Whitehorse, Yukon, pamper the group with excellent food. Serious science requires high quality inputs.

Manfred Trummer
SFU PIMS Site Director and Local Organiser

Contents

1	The Flow of an Evaporating Thin Film Liquid	1
1.1	Derivation of Governing Equations	2
1.2	Energy Balance	5
1.2.1	Energy Fluxes	5
1.2.2	Total Energy Balance	7
1.2.3	Heat Equation	9
1.3	No Surface Tension Reduction to Burgers' Equation	10
1.4	Estimating a When the Rain Catch and Shear are Position Dependent	11
1.4.1	Key Idea: Integrate the Free Surface Equation over the Whole Layer	11
1.4.2	Simplifying Assumptions	11
1.4.3	The Usual Steady Estimate for a Holds Even When Accounting for Surface Tension σ and Wind Shear τ	11
1.4.4	More Simplifying Assumptions	12
1.4.5	Consequence: The Evaporation is Constant for $x > a_{\text{in}}$	12
1.4.6	Formula for a with E Constant in the Outer Region	12
1.5	List of Parameters	13
1.6	Conclusions	13
2	“One Fish, Two Fish, Red Fish, Blue Fish”	
	Mathematically Modelling a Fresh Fish Detector	15
2.1	Introduction	16
2.2	Data Filtering and Estimation of Acceleration	16
2.2.1	Wavelet Shrinkage De-noising	16
2.2.2	Numerical Computation of Acceleration	17
2.3	Analysis of the Signal	20
2.3.1	Phase A	21
2.3.2	Phase B	21
2.3.3	Phase C	23
2.4	Conclusions	27
	Source Code	28
3	Software Testing Using t-Covering	33
3.1	An Integer Programming Model	34
3.2	A Graph Model or Illustrating the Integer Programming Model	35
3.2.1	Graph Decomposition	35
3.2.2	Vertex Cover	35
3.3	Using the LP-Relaxation to Obtain a 2-Cover	37

3.4	On the Geometric-Algebraic/Array Model and Some Results	40
3.4.1	Background	40
3.4.2	Identifying Covers from Hyperplanes—Algebraic Approach	40
3.4.3	Developing Strategies for Zoom-in Test	41
3.5	Array Model: Results and Extensions	42
3.5.1	Basic Definitions	42
3.5.2	2-Covers: Algebraic Construction	43
3.5.3	2-Covers: Recursive Construction	44
3.5.4	3-Covers: Recursive Construction	46
3.5.5	Generalizations and Conclusions	48
	Acknowledgements	49
4	Routing in Ad Hoc Networks/Dynamic Networks	51
4.1	The Model	52
4.2	Balancing Algorithm	52
4.3	Previous Work	52
4.4	Our Approach	54
4.5	Results	54
4.5.1	Step One—Simple Network	54
4.5.2	Step Two—Networks with Branches	55
4.5.3	Step Three—Networks with Loops	56
4.6	Extension to Two Destinations:	57
4.7	Search for a Counter Example	57
4.8	Conclusions	59
5	Modelling Polymer Electrolyte Membrane Fuel Cells	61
5.1	Physical Problem	62
5.2	Mathematical Problem	62
5.2.1	Coupled System of ODEs	62
5.2.2	Concentrations in Terms of Fluxes	63
5.2.3	Relative Humidities in Terms of Fluxes	64
5.2.4	Initial Conditions	65
5.2.5	Pressure Changes Along Channel	66
5.2.6	Counterflow Case	67
5.3	Simulation and Computational Results	67
5.3.1	Determining Appropriate Voltage to Yield Desired Current	67
5.3.2	Results for Constant Pressure	68
5.3.3	Dependence on the Humidity Levels	70
5.3.4	Variable Pressure Case	71
5.3.5	Counterflow Simulation	71
5.4	Further Work	72
5.5	Conclusion	72
6	Hedging in the Presence of Market Constraints	73
6.1	Introduction	74
6.2	Continuous-Time Model	74
6.2.1	The Black-Scholes Model	74
6.2.2	The Bellman Equation	75
6.2.3	An Analytic Approximation to the Bellman Equation	76
6.2.4	Small Parameter Approach with the Probability Density Function (PDF)	77
6.2.5	Numerical Solution	77
6.2.6	Graphs Interpretation	78
6.3	Discrete-Time Model	79

Appendix: Advanced Pricing Methodologies in Contemporary Financial Mathematics	82
7 Resistivity Well Logging	91
7.1 Introduction	92
7.2 Mathematical Model	93
7.2.1 Problem Formulation	93
7.2.2 Simplification	93
7.2.3 Physical Boundary Conditions	93
7.2.4 Interface Conditions	93
7.2.5 The Inverse Problem	94
7.3 Variational Principle and Finite Element Method	95
7.3.1 Formulation	95
7.3.2 Implementation	95
7.3.3 Inverse Problem	95
7.3.4 Stability	95
7.4 Classical Method	96
7.4.1 Formulation	96
7.4.2 Implementation	96
7.4.3 Inverse Problem	97
7.5 Conclusions	99
List of Participants	101
Organising Committee	101
Mentors	101
Students	101

Chapter 1

The Flow of an Evaporating Thin Film Liquid

Participants: Tim Myers (Mentor), Lyudmyla Barannyk, Neville Dubash, Leslie Fairbairn, Jeffrey Gilmore, Maria Landry, Nicolas Robidoux, Bidhan Roy, Tetyana Segin, Carlos Trenado.

PROBLEM STATEMENT: Thin liquid films are ubiquitous. In geology, they appear as gravity currents under water or as lava flows. In biophysics, they appear as membranes, as linings in mammalian lungs or as tear films in the eye. They occur in Langmuir films and in foam dynamics. In engineering, thin films serve in heat and mass transfer processes to limit fluxes and to protect surfaces, and applications arise in paints, adhesives and membranes.

Thin liquid films display a variety of interesting dynamics. Since the interface between the liquid and the surrounding gas is a deformable boundary, these films display wave motion. The waves can travel and steepen under certain conditions for high flow rates, and also can make transitions into quasiperiodic or chaotic structures. Thin film can rupture, leading to holes in the liquid that expose the substrate to the ambient gas.

Of interest in this workshop is the development of thin films on aircraft structures. Thin film develops on aircraft structures because of deposition of supercooled liquid droplets. These droplets reduce the lift of the aircraft. A popular approach in eliminating such thin films is by heating the surface. However, heating an aircraft is expensive.

We develop a model that describes the development of thin film on a surface under constant shear. The model is then updated to account for evaporation when the bottom surface is heated. Parameters controlling the model are determined using real value data.

1.1 Derivation of Governing Equations

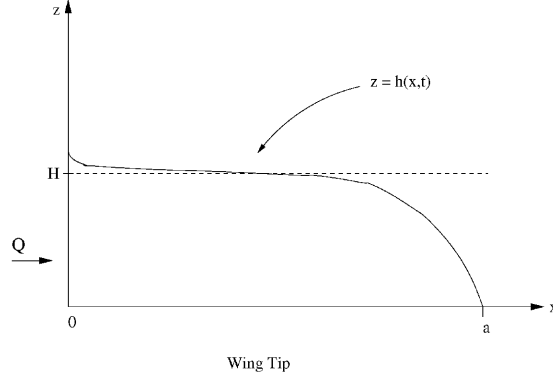


Figure 1.1: Half (due to the symmetry) of the wing tip.

Fluid is assumed to be flowing along a horizontal plane under the influence of interfacial continuous shear $\tau = \tau(x)$ and surface tension σ . The film has thickness h , the coordinates are such that x is directed down the plane, and z is perpendicular to the plane. The Navier-Stokes equations are assumed to govern the fluid flow. Using lubrication theory, these equations may be reduced to the following ones:

$$\frac{\partial^2 u}{\partial z^2} = \frac{1}{\mu} \frac{\partial p}{\partial x} \quad (1.1)$$

$$\frac{\partial p}{\partial z} = 0 \quad (1.2)$$

For an incompressible fluid the continuity equation is unaffected by lubrication theory:

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0 \quad (1.3)$$

Here p is the fluid pressure, (u, w) is the velocity vector and μ is the fluid viscosity.

The boundary conditions for solving this problem are

$$u(x, z, t) = 0|_{z=0} \quad \text{no-slip condition} \quad (1.4)$$

$$w(x, z, t) = 0|_{z=0} \quad \text{no penetration through the bottom surface} \quad (1.5)$$

Pressure difference across the free surface is taken to balance surface tension forces:

$$p - p_a = -\sigma \frac{\partial^2 h}{\partial x^2}$$

where p_a is the ambient pressure (which may vary over the fluid surface), and σ is the surface tension.

Another boundary condition at the interface is

$$\frac{\partial u}{\partial z} \Big|_{z=h} = \frac{\tau}{\mu} \quad (1.6)$$

where τ is the shear stress caused by the air flow.

At the interface $z = h(x, t)$ the kinematic boundary condition is

$$\frac{D(z - h(x, t))}{Dt} = E$$

where E is the evaporation rate and

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + u \frac{\partial}{\partial x} + w \frac{\partial}{\partial z}$$

Then

$$\frac{D(z - h(x, t))}{Dt} = -h_t + u(-h_x) + w \cdot 1,$$

so the kinematic condition can be written as

$$w = E + h_t + uh_x \quad (1.7)$$

Integrate the continuity equation (1.3) over the film

$$\begin{aligned} 0 &= \int_0^{z=h(x,t)} \left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} \right) dz = \int_0^{z=h(x,t)} \frac{\partial u}{\partial x} dz + w|_{z=0} \\ &= \int_0^{z=h(x,t)} \frac{\partial u}{\partial x} dz + w(x, z, t)|_{z=h} - w(x, z, t)|_{z=0}. \end{aligned}$$

Using the kinematic condition on the interface and zero normal velocity at $z = 0$, we have

$$0 = \int_0^{z=h(x,t)} \frac{\partial u}{\partial x} dz + E + \frac{\partial h}{\partial t} + \left(u(x, z, t) \frac{\partial h}{\partial x} \right) \Big|_{z=h(x,t)}.$$

Hence

$$\int_0^{z=h(x,t)} \frac{\partial u}{\partial x} dz = -E - \frac{\partial h}{\partial t} - \left(u(x, z, t) \frac{\partial h}{\partial x} \right) \Big|_{z=h(x,t)}. \quad (1.8)$$

Consider rate of the change of the mass flux with respect to x .

$$\begin{aligned} \frac{\partial}{\partial x} Q(x, t) &= \frac{\partial}{\partial x} \int_0^{z=h(x,t)} u(x, z, t) dz \\ &= \int_0^{z=h(x,t)} \frac{\partial u}{\partial x} dz + \left(u(x, z, t) \frac{\partial h}{\partial x} \right) \Big|_{z=h(x,t)} \\ &= -E - \frac{\partial h}{\partial t} - \left(u(x, z, t) \frac{\partial h}{\partial x} \right) \Big|_{z=h(x,t)} + \left(u(x, z, t) \frac{\partial h}{\partial x} \right) \Big|_{z=h(x,t)} = -E - \frac{\partial h}{\partial t}. \end{aligned}$$

Therefore,

$$\frac{\partial}{\partial x} Q(x, t) = \frac{\partial}{\partial x} \int_0^{z=h(x,t)} u(x, z, t) dz = -E - \frac{\partial h}{\partial t}. \quad (1.9)$$

Consider equations (1.1) and (1.2). It follows from equation (1.2) that pressure is the function of x alone: $p = p(x)$.

Integrating equation (1.1) once we get

$$\frac{\partial u}{\partial z} = z \frac{\partial p}{\partial x} + C_1$$

where the constant of integration C_1 can be found by using the boundary condition from the shear stress (1.6).

$$C_1 = \frac{\tau}{\mu} - h \frac{\partial p}{\partial x}$$

Another integration gives $u(x, z, t) = \frac{z^2}{2} \frac{\partial p}{\partial x} + C_1 z + C_2$. Using the no-slip condition (1.4), we get $C_2 = 0$. Hence, the velocity profile is

$$u(x, z, t) = \frac{z^2}{2} \frac{\partial p}{\partial x} + \left(\frac{\tau}{\mu} - h \frac{\partial p}{\partial x} \right) z. \quad (1.10)$$

Now, substitute $u(x, z, t)$ from (1.10) into (1.9) and integrate.

$$\begin{aligned} \frac{\partial}{\partial x} \int_0^{z=h(x,t)} u(x, z, t) dz &= \frac{\partial}{\partial x} \int_0^{z=h(x,t)} \left\{ \frac{z^2}{2} \frac{\partial p}{\partial x} + \left(\frac{\tau}{\mu} - h \frac{\partial p}{\partial x} \right) z \right\} dz = \\ &= \frac{\partial}{\partial x} \left\{ \frac{z^3}{6} \frac{\partial p}{\partial x} + \left(\frac{\tau}{\mu} - h \frac{\partial p}{\partial x} \right) \frac{z^2}{2} \right\} \Big|_0^{z=h(x,t)} = \\ &= \frac{\partial}{\partial x} \left\{ \frac{h^3}{6} \frac{\partial p}{\partial x} + \left(\frac{\tau}{\mu} - h \frac{\partial p}{\partial x} \right) \frac{h^2}{2} \right\} = \frac{\partial}{\partial x} \left\{ -\frac{h^3}{3} \frac{\partial p}{\partial x} + \frac{\tau h^2}{\mu} \right\}. \end{aligned}$$

Therefore,

$$E + \frac{\partial}{\partial x} \left\{ -\frac{h^3}{3} \frac{\partial p}{\partial x} + \frac{\tau h^2}{\mu} \right\} = -\frac{\partial h}{\partial t}, \quad (1.11)$$

which represents the mass conservation of the fluid.

Pressure can be found from $p = p_a - \sigma \frac{\partial^2 h}{\partial x^2}$. Substitute pressure into the mass balance equation (1.11) taking into account that

$$\frac{\partial p}{\partial x} = \frac{\partial p_a}{\partial x} - \sigma \frac{\partial^3 h}{\partial x^3} \sim -\sigma \frac{\partial^3 h}{\partial x^3}$$

since variation of ambient pressure p_a is smaller than variation of pressure p . Hence we have

$$E + \frac{\partial}{\partial x} \left\{ \frac{\sigma h^3}{3} \frac{\partial^3 h}{\partial x^3} + \frac{\tau h^2}{\mu} \right\} = -\frac{\partial h}{\partial t}. \quad (1.12)$$

Also we are mostly concerned with the film length at steady state. Thus all time derivatives can be set to zero $\frac{\partial h}{\partial t} = 0$ and equation (1.12) becomes

$$E + \frac{\partial}{\partial x} \left\{ \frac{\sigma h^3}{3} \frac{\partial^3 h}{\partial x^3} + \frac{\tau h^2}{\mu} \right\} = 0. \quad (1.13)$$

This is simply a first order differential equation in h . While there are a number of possible boundary conditions that we could apply, the best one to apply is a fixed rate of inflow, Q , at $x = 0$. We have accounted for evaporation (a loss of mass) in the equation (1.13) but not for mass addition (due to the accumulation of droplets). Thus we must account for this in the boundary condition if we hope to obtain a reasonable solution. The other boundary conditions are less significant.

Integrate equation (1.13) with respect of x taking into consideration the boundary condition for the flux at $x = 0$

$$Q(x)|_{x=0} = Q_0 = \int_0^h u(z) dz = \int_0^h \frac{\tau}{\mu} z dz = \frac{\tau}{2\mu} h^2$$

to get

$$Ex + \left(\frac{\tau h^2}{\mu} + \frac{h^3}{3\mu} \sigma h_{xxx} \right) = Q_0.$$

We can neglect the term with the high derivative of h since the surface tension is small to get an estimate for the height h

$$h \sim \sqrt{\frac{2\mu}{\tau} (Q_0 - Ex)}$$

and estimate the distance a at which the height $h(x = a, t) = 0$:

$$a = \frac{Q_0}{E}. \quad (1.14)$$

This gives the length of the fluid film. It is interesting to note that length is independent of the shear stress applied at the surface of the film.

1.2 Energy Balance

1.2.1 Energy Fluxes

There are a variety of mechanisms that add or remove energy from the system. The following are the main mechanisms considered.

1. **Convective Heat Transfer:** The water film loses energy to the surrounding atmosphere due to the temperature difference between the surface of the film and the surrounding air (see Fig.1.2).

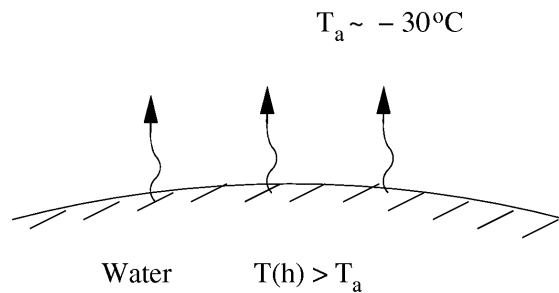


Figure 1.2: Convective heat transfer.

A simple model for this convection is Newton's Law of Cooling. The energy flux is given by

$$-k \frac{\partial T}{\partial z} = H_{tc}(T(z = h(t)) - T_a) \quad (1.15)$$

where k is the thermal conductivity of water; H_{tc} is the heat transfer coefficient for water and air, $T(z)$ is the temperature in the film, and T_a is the air temperature surrounding the film.

2. **Evaporative Heat Transfer:** The water film loses energy through the evaporation of water from the surface of the film (see Fig. 1.3).

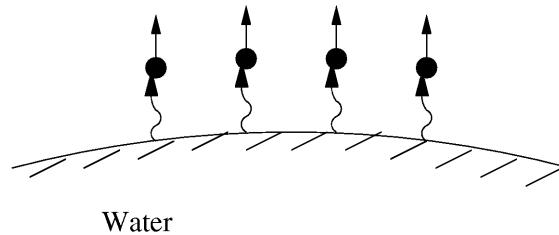


Figure 1.3: Evaporative heat transfer.

While there are complicated non-linear relations for evaporation that are derived empirically, to first order, they can be approximated by a linear relation where the evaporation rate is proportional to the temperature difference between the film surface and the surrounding air. In this case the energy flux is given by

$$-k \frac{\partial T}{\partial z} = e_0 (T(z = h(t)) - T_a) \quad (1.16)$$

where e_0 is an empirical evaporation coefficient.

3. **Thermal Energy of Droplets:** The droplets which collect on the surface of the film are at a temperature equal to that of the surrounding air (see Fig. 1.4).

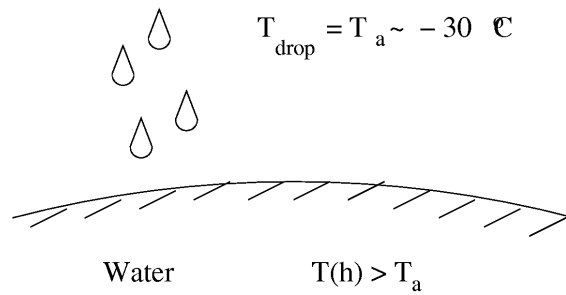


Figure 1.4: Thermal energy of droplets.

This effectively results in a loss of energy in the film proportional to the temperature difference of the surface of the film and the surrounding air.

$$-k \frac{\partial T}{\partial z} = C_{p,water} (cv \rho_{drop}) (T(z = h(t)) - T_a) \quad (1.17)$$

where $C_{p,water}$ is the specific heat of water (at constant pressure), c is the catch parameter, v is the freestream velocity, and ρ_{drop} is the density of water droplets in the air.

4. **Kinetic Energy of Droplets:** The droplets have a velocity equal to the freestream velocity (see Fig.1.5).

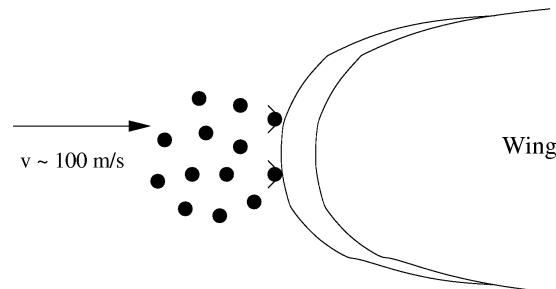


Figure 1.5: Kinetic energy of droplets.

For droplets which accumulate on the wing, some of their kinetic energy is converted to thermal energy which heats the water film. Assuming that all the kinetic energy of the droplets gets converted to heat the energy flux is

$$-k \frac{\partial T}{\partial z} = -\frac{1}{2}(c\nu\rho_{drop})v^2. \quad (1.18)$$

5. **Aerodynamic Heating:** The air near the surface of the wing is forced to decelerate losing energy to the water film (see Fig. 1.6).

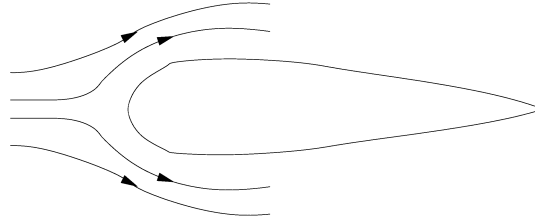


Figure 1.6: Aerodynamic heating.

This can essentially be thought of as heating due to friction. The energy flux due to aerodynamic heating is given by

$$-k \frac{\partial T}{\partial z} = \frac{rH_{tc}v^2}{2C_{p,air}} \quad (1.19)$$

where r is the recovery factor, and $C_{p,air}$ is the specific heat of air (at constant pressure).

6. **Wing Heating:** The surface of the wing is being heated through an external source (see Fig. 1.7).

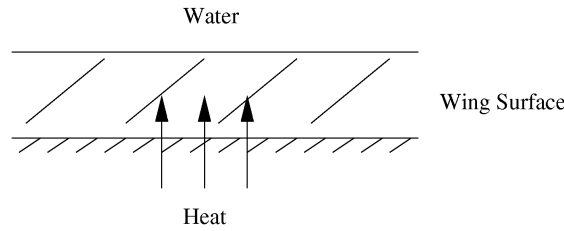


Figure 1.7: Energy applied at the bottom surface.

This results in a fixed energy flux (or fixed temperature) at the surface of the wing,

$$-k \frac{\partial T}{\partial z} = Q_t. \quad (1.20)$$

1.2.2 Total Energy Balance

The total energy balance equation for the whole film has the following form

$$\begin{aligned} & \left[\begin{array}{c} \text{Heat} \\ \text{from} \\ \text{Bottom} \end{array} \right] + \left[\begin{array}{c} \text{Kinetic} \\ \text{Energy} \\ \text{of Droplets} \end{array} \right] + \left[\begin{array}{c} \text{Aerodynamic} \\ \text{Heating} \end{array} \right] \\ = & \left[\begin{array}{c} \text{Thermal} \\ \text{Energy of} \\ \text{Droplets} \end{array} \right] + \left[\begin{array}{c} \text{Evaporative} \\ \text{Heat} \\ \text{Transfer} \end{array} \right] + \left[\begin{array}{c} \text{Convective} \\ \text{Heat} \\ \text{Transfer} \end{array} \right] \end{aligned} \quad (1.21)$$

or

$$Q_T + \frac{1}{2}(cv\rho_a)v^2 + \frac{rH_{tc}v^2}{2C_{p,air}} = C_{p,water}(T(h) - T_a)(cv\rho_a) + e_0(T(h) - T_a) + H_{tc}(T(h) - T_a), \quad (1.22)$$

where c is the catch parameter that describes distribution of droplets of water that stay of the surface of the wing; e_0 evaporation parameter. The graph for c is given on Fig. 1.8 (a).

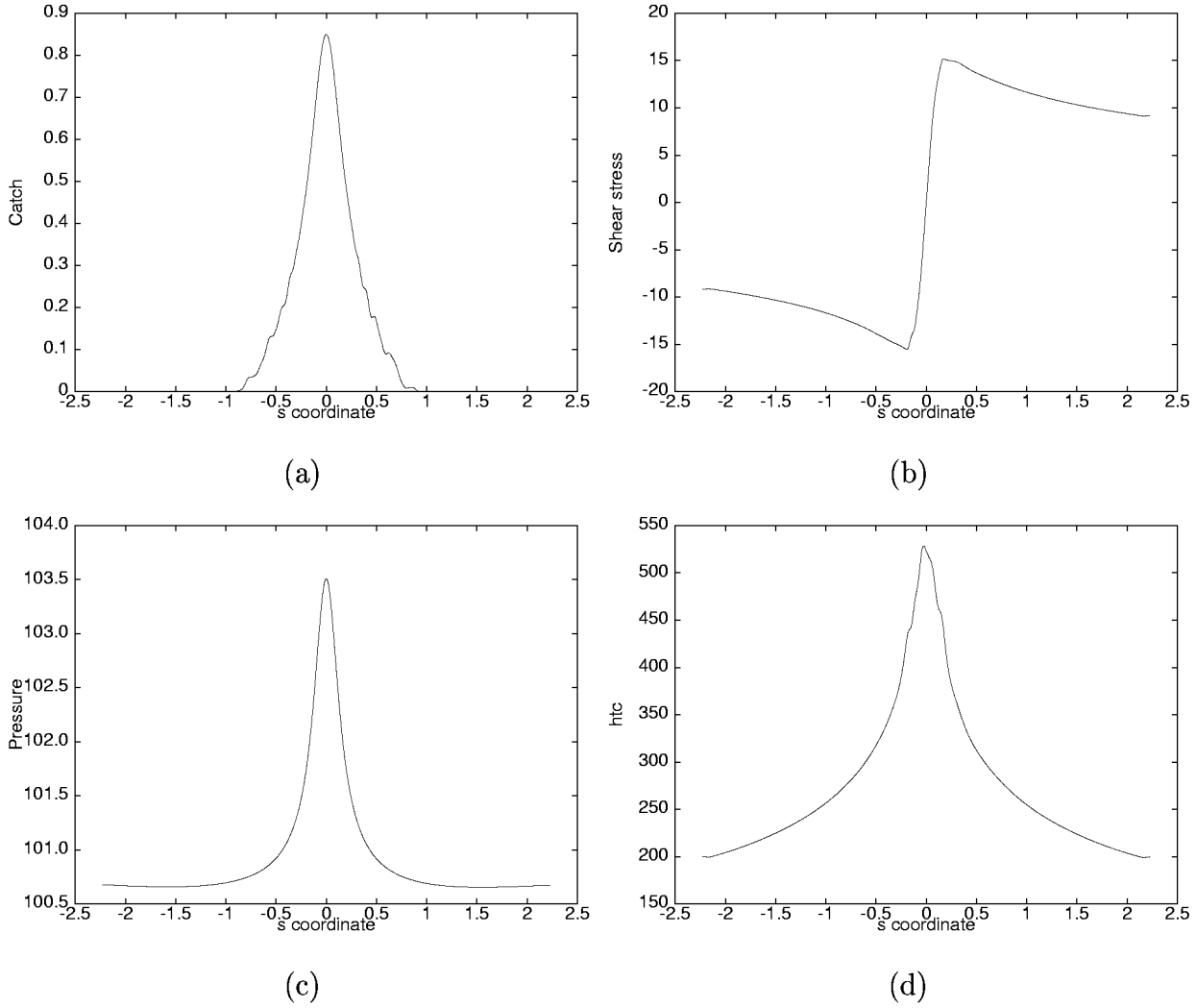


Figure 1.8: (a) Catch; (b) Shear stress; (c) Pressure; H_{tc} .

Solving equation (1.22) for T , we get the averaged temperature T

$$T_{aver} = \frac{Q_T + \frac{1}{2}(cv\rho_a)v^2 + \frac{rH_{tc}v^2}{2c_{p,a}}}{c_{p,w}(cv\rho_a) + e_0 + H_{tc}} \quad (1.23)$$

1.2.3 Heat Equation

To obtain the temperature profile $T(z)$ in the fluid film we must solve the heat equation

$$\rho C_{p,water} \frac{\partial T}{\partial t} = k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial z^2} \right) \quad (1.24)$$

Using the following scaling, where the primes ($'$) denote a dimensionless quantity we can dimensionalize equation (1.24):

$$T = T_0 T', \quad t = \frac{L}{U} t', \quad x = L x', \quad z = H z'$$

The non-dimensional heat equation is

$$\frac{\epsilon U}{L} \frac{DT'}{Dt'} = \frac{k}{\rho C_{p,water}} \left(\epsilon^2 \frac{\partial^2 T'}{\partial x'^2} + \frac{\partial^2 T'}{\partial z'^2} \right)$$

To first order this can be approximated by

$$\frac{\partial^2 T'}{\partial z'^2} = 0,$$

or in dimensional form

$$\frac{\partial^2 T}{\partial z^2} = 0. \quad (1.25)$$

The boundary conditions are given by the flux at the surface of the wing, equation (1.20),

$$-k \frac{\partial T}{\partial z} \Big|_{z=0} = Q_t, \quad (1.26)$$

and the flux at the surface of the film, equations (1.15) - (1.19),

$$-k \frac{\partial T}{\partial z} \Big|_{z=0} = -\frac{1}{2}(cv\rho_{drop})v^2 - \frac{rH_{tc}v^2}{2C_{p,air}} + C_{p,water}(cv\rho_{drop})(T(h) - T_a) + e_0(T(h) - T_a) + H_{tc}(T(h) - T_a) \quad (1.27)$$

$$= -\frac{1}{2}(cv\rho_{drop})v^2 - \frac{rH_{tc}v^2}{2C_{p,air}} + (C_{p,water}(cv\rho_{drop}) + e_0 + H_{tc})(T(h) - T_a) \quad (1.28)$$

$$= A + B(T(h) - T_a) \quad (1.29)$$

where, for simplicity, we write

$$A = -\frac{1}{2}(cv\rho_{drop})v^2 - \frac{rH_{tc}v^2}{2C_{p,air}} \quad \text{and} \quad B = C_{p,water}(cv\rho_{drop}) + e_0 + H_{tc}.$$

Note that A and B are just functions of the physical constants.

Solving equation (1.25) with the boundary conditions (1.26) and (1.29) we get (since $\frac{\partial T}{\partial z}$ is constant which we find from the boundary condition (1.26), we can use it to find $T(z)|_{z=h}$ by solving equation (1.29) for $T(z)|_{z=h}$)

$$T(z, t) = -\frac{Q_t}{k}(z - h(t)) + \frac{Q_t - A}{B} + T_a. \quad (1.30)$$

Now we can find the temperature difference

$$T(h) - T_a = \frac{Q_t - A}{B}.$$

Using equation for the energy balance due to evaporation

$$\rho_w L_e E = e_0(T - T_a)$$

we find the evaporation rate

$$E = \frac{e_0(T - T_a)}{\rho_w L_e} = \frac{e_0(Q_t - A)}{\rho_w L_e B} \quad (1.31)$$

Hence, using equation (1.14) for the estimate of the length a of the film, we can write

$$a = \frac{Q_0}{E} = \frac{Q_0 \rho_w L_e}{e_0(T - T_a)} = \frac{Q_0 \rho_w L_e B}{e_0(Q_t - A)}$$

Substituting all physical parameters we obtain the following estimate for a :

$$a = \frac{Q_0 \rho_w L_e C_{p,water}(cv\rho_{drop}) + e_o + H_{tc}}{e_o \left(Q_t + \frac{1}{2}(c\rho_{drop}v^2) + \frac{rH_{tc}v^2}{2Ca} \right)} \sim 8.8m \quad (1.32)$$

At the interface $z = h(x, t)$ we have another boundary condition that relates temperature T and height of the film h . This is so-called Stefan condition and it has the following form

$$-k \frac{\partial T}{\partial z} = \rho L_e \frac{\partial h}{\partial t} \quad (1.33)$$

It can be used to estimate the time needed for the full evaporation of the film of thickness h :

$$t^* = \frac{\rho L_e H^2}{k(T(h) - T_a)}. \quad (1.34)$$

1.3 No Surface Tension Reduction to Burgers' Equation

With the assumption that surface tension plays a negligible role in the profile of the fluid surface, the higher order derivative term can be removed to simplify the equation. The resulting equation is similar to Burgers' equation:

$$\frac{\partial h}{\partial t} + \frac{\tau(x)}{\mu} h \frac{\partial h}{\partial x} = R(x) - E(x)$$

Here, incoming rain and evaporation are generalized as functions of x . A closed form solution of this equation can be obtained by the method of characteristics. By introducing a new variable s , where

$$\frac{dt}{ds} = 1, \quad \text{and} \quad \frac{dx}{ds} = \frac{\tau(x)}{\mu} h.$$

Burgers' equation can now be expressed as a total derivative of h :

$$\frac{dh}{ds} = R(x) - E(x).$$

These ODEs can be solved when coupled with the initial condition that at $t = 0$, $x = x_0$ and $h = h_0$:

$$h = R(x)t - E(x)t + h_0$$

$$x = \frac{\tau(x)}{\mu} t \left(h_0 + [R(x) - E(x)] \frac{t}{2} \right) + x_0.$$

This closed form solution is only capable of tracking a single particle. It is possible to generate solution profiles numerically by tracking multiple points. Our solution fails to accurately model the physical situation when a shock forms (this will not happen with the surface tension included). Since we are only interested in damping the surface, this is not a problem. It is debatable, however, to what degree this solution is valuable since it does little to elaborate on the steady state solution.

1.4 Estimating a When the Rain Catch and Shear are Position Dependent

1.4.1 Key Idea: Integrate the Free Surface Equation over the Whole Layer

Consider the free surface equation (1.12) modified so that the rain's mass contribution comes through the top of the layer instead of through the left boundary (the derivation in Sections 1.1 and 1.2 are somewhat mismatched physically: in Section 1.2, the rain's heat contribution comes through the top, while in Section 1.1, the rain's mass contribution comes through the left boundary):

$$\frac{\partial h}{\partial t} = \text{rain} - E - \frac{\partial}{\partial x} \left\{ \frac{\sigma}{2\mu} h^3 \frac{\partial^3 h}{\partial x^3} + \frac{\tau}{2\mu} h^2 \right\}. \quad (1.35)$$

The key idea of this section is that integrating (1.35) over the entire thin water layer yields an integral equation parameterized by the layer length a ; this integral equation only involves a as an unknown if steady state is assumed.

An advantage of being able to compute a without solving the nonlinear fourth order PDE (1.35) is that this allows a more straightforward treatment of the boundary conditions: we know where the layer “ends” before solving the PDE; thus, the boundary is not quite “free”.

1.4.2 Simplifying Assumptions

For the sake of exposition, we assume that the only “parameters” which depend on x are the rain catch (effective cross section) c and the shear τ ; we also assume symmetry about $x = 0$, which implies that the shear must be odd, and in particular that the shear vanishes at the origin (this is quite reasonable, since the origin is a stagnation point of the air flow). Nonetheless, the approach carries over to situations when other parameters or terms other than the rain catch and shear (and, indirectly, the evaporation flux per unit length and time) depend on x . The approach also applies to situations in which c and τ are not constant away from the origin, situations with no symmetry about the origin, and the non-steady case (in which case the approach relates $\partial h/\partial t$ to a and its time derivative).

1.4.3 The Usual Steady Estimate for a Holds Even When Accounting for Surface Tension σ and Wind Shear τ

Assuming steady state, the time derivative vanishes. Integrating, one gets:

$$\int_0^a \text{rain} \, dx - \int_0^a E \, dx = \left\{ \frac{\sigma}{2\mu} h^3 \frac{\partial^3 h}{\partial x^3} + \frac{\tau}{2\mu} h^2 \right\} \Big|_{x=0}^{x=a} \quad (1.36)$$

The right hand side of (1.36) vanishes because

- h^3 and h^2 vanish at a (by definition, h vanishes at a);
- $\frac{\partial^3 h}{\partial x^3}$ vanishes at 0 by symmetry (being an odd derivative of an even function).
- τ vanishes at 0 (because it is odd).

Now:

- The total rain collected by the wing (per unit time) is $\int_0^a \text{rain} \, dx = Q$, and consequently can be computed from c (and some other constants).
- The total water loss through evaporation is $\int_0^a E \, dx$, which is equal to $\bar{E}a$ —the average evaporation per unit length—times a .

This yields a first relationship between a , (half of) the length of the water layer, Q , (half of) the total rain flux, and \bar{E} , the average evaporation flux per unit length:

$$a = \frac{Q}{\bar{E}}. \quad (1.37)$$

(Note: this is an equality, not an approximation.) That this should hold is actually obvious: the only source of water is rain, the only sink is evaporation, and we assume steady state, so the total rain flux must balance the total evaporation. Thus, it is reassuring to have the equation appear even when providing for longitudinal water flow (driven by shear) and surface tension. Again, this is not surprising: the term between brackets in (1.35), which arises from approximating the Navier-Stokes equation, accounts for water flow *within the layer*; consequently, it does not affect the *total* water balance in any way, only its distribution. Had it been otherwise, this would have been cause for worry.

Equation (1.35), unfortunately, does not readily lead to a very precise computation of a , for \bar{E} , being an *average* evaporative flux, it involves the length scale a . It is, however, possible to get an equation with a as the only unknown; in fact, very reasonable simplifying assumptions allow one to derive a “closed form” formula for a in terms of integrals of the catch c .

1.4.4 More Simplifying Assumptions

The Catch c Typically Vanishes Outside of a Small Neighbourhood of $x = 0$

The catch is not well approximated by a constant; as a function of x , the distance along the wing to the leading edge, c typically looks like a “narrow Gaussian with cut-off tails” centred at $x = 0$ (see Figure 1.8 (a)). In any case, outside a small neighbourhood of the origin, c can be safely assumed to be zero.

c enters the free surface equation (1.12) (nonlinearly) through the evaporation term (1.31): the heat and kinetic energy conversion flux terms are proportional to the value of c .

The Evaporation Flux E is Essentially Constant Outside of a Neighbourhood of $x = 0$

The formula for the evaporation flux E was obtained by assuming that heat flows solely in the z direction. Looking at the closed form expression for this flux, one sees that the evaporation depends on the catch c . Since c is constant away from the origin, so is E . (Of course, we may want to allow other parameters to be position dependent, in which case similar assumptions on the varying parameters are necessary to get a “closed form” for a .)

1.4.5 Consequence: The Evaporation is Constant for $x > a_{\text{in}}$

The above simplifying assumptions imply that there exists an a_{in} such that the catch—and consequently the evaporation—is constant for $x > a_{\text{in}}$.

Suppose that the steady state a is greater than a_{in} . Break the interval $[0, a]$ into an inner region $[0, a_{\text{in}}]$ and an outer region $[a_{\text{in}}, a]$. In the inner region, the evaporation flux E depends on position through c . In the outer region however, the evaporation is constant, this constant being obtained by setting c to zero in the formula for E ; call this constant E_{out} .

1.4.6 Formula for a with E Constant in the Outer Region

The free surface equation (1.12) now reads

$$\begin{aligned} \int_0^{a_{\text{in}}} \text{rain } dx - \int_0^{a_{\text{in}}} E dx &= \int_{a_{\text{in}}}^a E dx - \int_{a_{\text{in}}}^a \text{rain } dx \\ &= E_{\text{out}} \{a - a_{\text{in}}\}, \end{aligned}$$

in which it has been assumed that no rain falls in the outer region. The left hand side is completely determined by the rain distribution, the heat flux through the bottom, and various physical parameters. So:

$$a = a_{\text{in}} + \frac{\left\{ Q - \int_0^{a_{\text{in}}} E dx \right\}}{E_{\text{out}}},$$

where Q is the total flux of rain, E is the evaporation flux per unit length (which depends on the catch c , and consequently on position), and E_{out} is the constant value of the evaporation flux per unit length which holds for $x > a_{\text{in}}$.

1.5 List of Parameters

r_{drop} - radius of water droplet, equal to 50 mm ;
 H_{tc} - heat transfer coefficient, equal to 400 $W/m^2 K$;
 ρ_a - humidity (water density in the air), equal to $10^{-3} kg/m^3$;
 ρ_w - water density, equal to $10^3 kg/m^3$;
 v - free stream velocity, equal to 100 m/s ;
 L - length scale in the x direction, equal to 0.5 m ;
 k - thermal conductivity, equal to 0.6;
 e_o - evaporation coefficient, equal to 270 $Pa m / K s$;
 L_e - latent heat of evaporation, equal to $2.257 \cdot 10^6 J/kg$;
 Q - water flux, equal to $10^{-4} m^2/s$;
 Q_t - heat flux specified at the leading edge of the wing, $J/m^2 s$;
 μ - viscosity of water, $10^{-3} kg/ms$;
 p - pressure, Pa (profile presented by Figure F1.2 (c));
 τ - shear stress, Pa (profile presented by Figure F1.2 (b)), in case of constant shear stress equal to $10Pa$;
 $C_{p,air}$ - specific heat of air, 1000 $J/kg K$;
 $C_{p,water}$ - specific heat of water, 4200 $J/kg K$;
 c - number of droplets catching the surface;
 r - recovery factor of aerodynamical heating, equal to 0.8;
 σ - surface tension of water, equal to 0.0728 N/m ;
 T_a - temperature of the air, equal to $\sim -30^\circ$

1.6 Conclusions

- We model the dynamics of thin water films arising from the deposition of supercooled water droplets on the front of an heated airplane wing.
- The thin film approximation is used to select leading order effects.
- We derive several consistent estimates of the length of the steady state water layer, as well as the time to evaporate an initial layer.
- Energy balance is used to determine the evaporation rate from the heat convection (radiation), the heat content and kinetic energy of the incoming droplets, aerodynamic heating, and the imposed wing heating. This involves solving a collection of 1D steady heat equations.
- The shape of the thin water layer is modelled by a nonlinear fourth order parabolic partial differential equation water balance equation which includes the effects of surface tension and aerodynamic shear stress through a Stefan problem, as well as the leading order terms of the Navier-Stokes equations.
- We derive several consistent estimates for the length of steady film layers, as well as the time to evaporation of an initial layer.
- An exact formula for the steady film length is obtained by integrating the water balance equation over the whole layer. The resulting “closed form” formula for the length of the layer involves integrals of the position dependent effective cross section of the wing as well as the evaporation function.

Bibliography

- [1] Myers T.G. and Thompson C.P. Modelling the flow of water on aircraft in icing conditions, *AIAA J.*, 1998, **36** (6), pp.1010–1013.
- [2] Myers T.G. Thin films with high surface tension, *SIAM Review*, 1998, **40** (3), pp.441–462.
- [3] Oron A., Davis S.H. and Bankoff S.G. Long-scale evolution of thin liquid films, *Rev. Mod. Phys.*, 1997, **69** (3), pp.931–980.
- [4] Ockendon H. and Ockendon J.R. *Viscous Flow*, Cambridge University Press, 1995, 113p.

Chapter 2

“One Fish, Two Fish, Red Fish, Blue Fish”

Mathematically Modelling a Fresh Fish Detector

Participants: Chris Budd (Mentor), Ibrahim Agyemang, Erik Andries, Dhavide Aruliah, Mélanie Beck, Qingguo Li, Colin Macdonald, Matthias Müick, Robin Swain.

PROBLEM STATEMENT: The PIMS Mathematical Modelling spring workshop presented six different environments to be considered for modeling during the program. For this group, Chris Budd proposed that we study data obtained through experiments using a device designed to determine the freshness of fish. Through an electric current applied to a coil, a needle-shaped probe is projected by a force directly on the surface of a test sample. The depth to which the probe pushes the surface is recorded by the coil as a function of time. The goal of this project is to use the data to indicate what mechanisms govern the dynamics of the probe over time, namely models of ordinary differential equations from which parameters can be extracted to determine fresh fish from those which are not.

2.1 Introduction

Have you ever been to the supermarket to buy fresh fish and wonder: Just how fresh is this fish? Is the specified freshness of the fish accurate? If not, how could you tell?

In this report we describe the mathematical analysis of a system which is proposed to measure the freshness of fish. The device used to test freshness is a thin needle-like probe with a coil which provides an electromagnetic force to the probe and also measures its motion. We attempt to capture the dynamics of the motion of the probe in a mathematical model which can fit the given data. In doing so we hope to extract information on fish freshness from some fish-dependent parameters in the model(s).

The first section deals with cleaning up the data. Our group has been provided with some raw data collected by the fish-probe apparatus for several different materials:

1. A plaice fish
2. A foam or sponge
3. A clear cling wrap medium
4. A human hand

The data contains noise which was filtered for subsequent analysis. Fast Fourier Transforms (FFT) and wavelet de-noising techniques were applied to the data to remove the noise. The re-constructed data was useful in obtaining estimates for the velocity and acceleration of the probe during the experiments. The scope of our analysis was only concerned with fitting the data for a plaice fish and foam sponge, and does not include analysis of the cling wrap and human hand. Mathematical modelling techniques including nonlinear optimization and the physics of oscillatory systems were used to describe the dynamics of the probe.

2.2 Data Filtering and Estimation of Acceleration

In this part of the project, wavelet de-noising techniques are applied to the empirical fish probe data for filtering. Based on the filtered data, the acceleration was computed numerically.

2.2.1 Wavelet Shrinkage De-noising

Our objective was to suppress the noise and recover the signal. Both FFT and wavelet approaches were implemented to fulfil the task. Because of the multi-resolution property of the wavelet transform, wavelet-based de-noising produced better results than the FFT approach.

To recover a signal the noise must be removed before proceeding with further data analysis. The wavelet de-noising procedure consists of three steps:

1. A linear forward wavelet transform
2. A nonlinear shrinkage de-noising
3. A linear inverse wavelet transform

Let $X(t)$ represent a set of observed data, and assume

$$X(t) = S(t) + N(t) \tag{2.1}$$

contains the true signal $S(t)$ with additive noise $N(t)$ as functions in time t to be sampled. Let $\mathcal{W}(\cdot)$ and $\mathcal{W}^{-1}(\cdot)$ denote the forward and inverse wavelet transform operators. Let $\mathcal{D}(\cdot, \lambda)$ denote the de-noising operator with threshold λ . We intend to de-noise $X(t)$ to recover $\hat{S}(t)$ as an estimate of $S(t)$. The basic version of the procedure

consists of three steps, including decomposition, obtaining thresholding detail coefficients, and reconstruction. It is summarized as

$$\begin{aligned} Y &= \mathcal{W}(X) \\ Z &= \mathcal{D}(Y, \lambda) \\ \hat{S} &= \mathcal{W}^{-1}(Z) \end{aligned} \quad (2.2)$$

For the de-noising operator \mathcal{D} , given threshold λ for the data \mathcal{U} ,

$$\mathcal{D}(\mathcal{U}, \lambda) = \text{sgn}(\mathcal{U}) \max(0, |\mathcal{U}| - \lambda) \quad (2.3)$$

defines nonlinear soft thresholding. Many different schemes have been developed on selection of de-noising operator [1, 2].

In this application, the data was first processed by using 5-level Haar wavelet transform, then we removed the finer wavelet coefficients b_5 and b_4 . After that, we performed the inverse transform and recovered the clean data. The processes were carried out by using the wavelet toolbox in Matlab [3]. The results for the fish data are shown in Figures 2.1 to 2.2. Figure 2.2 contains the wavelet de-noising results and the original fish data (offset for comparison). The clean signals for the foam and cling film are included in Figures 2.3 and 2.4.

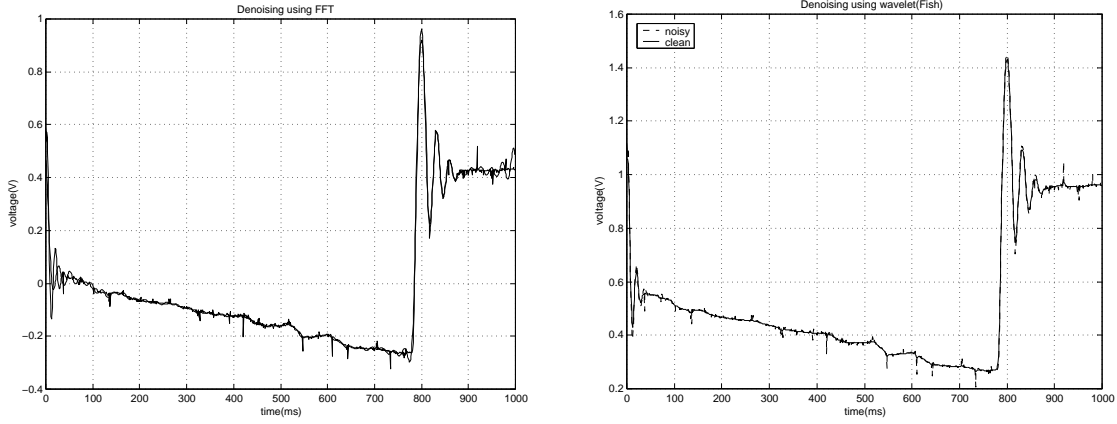


Figure 2.1: FFT (left) and wavelet (right) de-noising results for the original fish data.

2.2.2 Numerical Computation of Acceleration

After we obtained the clean data by using wavelet de-noising, the accelerations were computed using finite difference approximation. The position data $x(t)$ can be calculated from the de-noised voltage data $u(t)$ as follows:

$$x(t) = 10^{-3} \times 0.8u(t) \text{mm/V} \quad (2.4)$$

First the velocity $v(t)$ can be computed as

$$v(t) = \frac{x(t + \Delta t) - x(t)}{\Delta t} \quad (2.5)$$

where $\Delta t = 10^{-3}$ is the sample rate of the sensor.

Because of the differentiation process, the noise in the position signal was amplified, and in order to increase the signal to noise ratio (SNR), the de-noising routine was also applied to the velocity data $v(t)$ to generate the clean signal $v_1(t)$. The acceleration $a(t)$ was computed as

$$a(t) \approx \frac{v_1(t + \Delta t) - v_1(t)}{\Delta t} \quad (2.6)$$

With the de-noised data we obtained continuous acceleration data for further analysis.

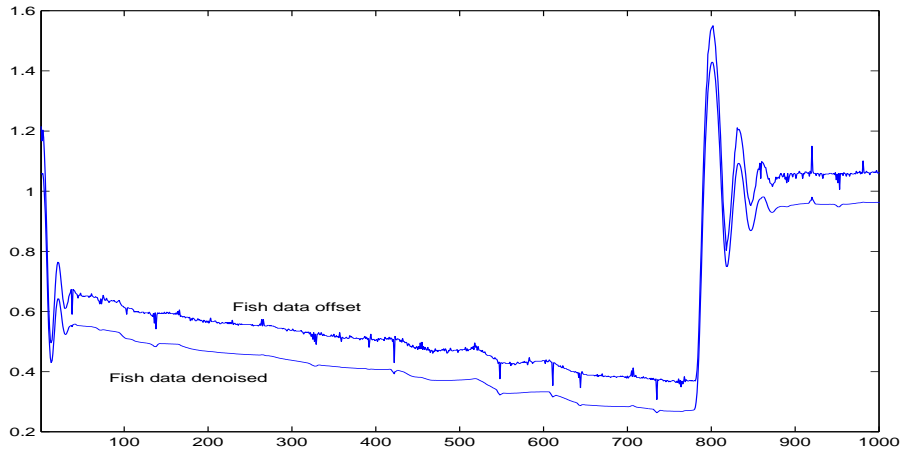


Figure 2.2: Comparison of the wavelet de-noised data and the original data (offset) for a plaice fish.

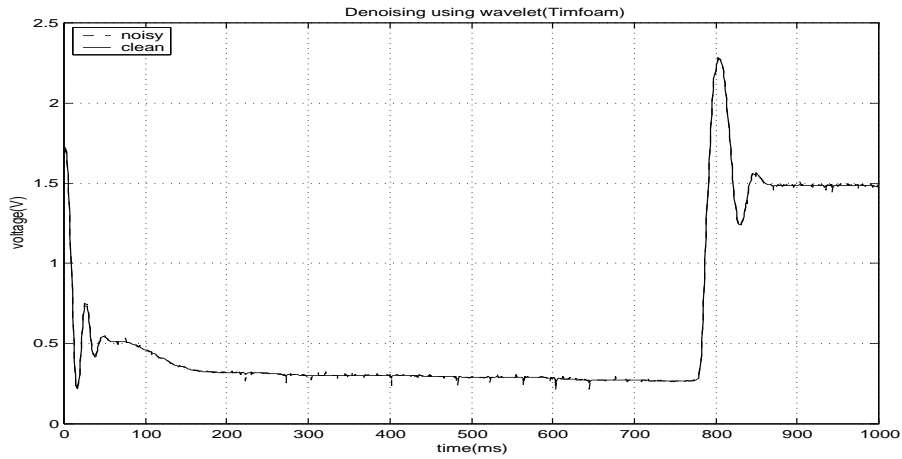


Figure 2.3: De-noising results of timfoam data.

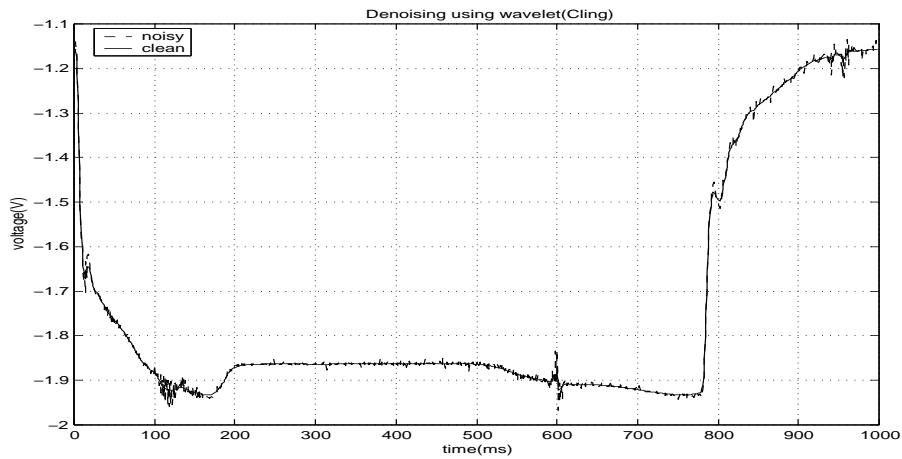


Figure 2.4: De-noising results of cling film data.

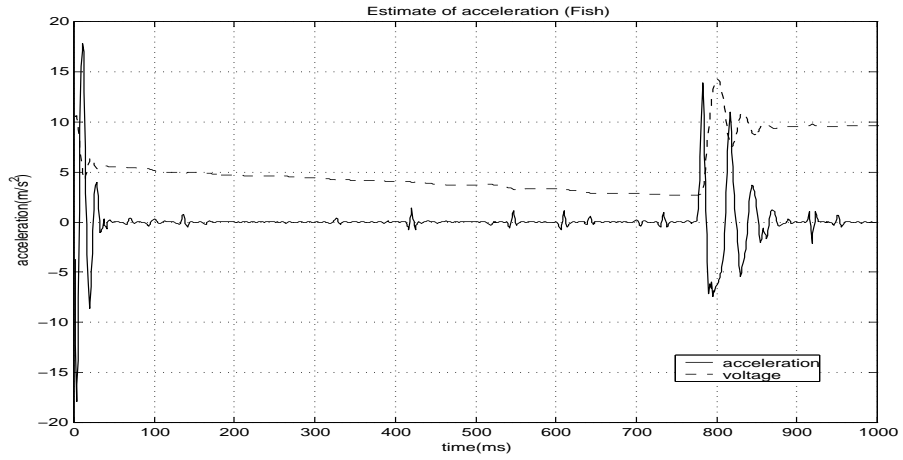


Figure 2.5: Estimation of acceleration from clean data (fish).

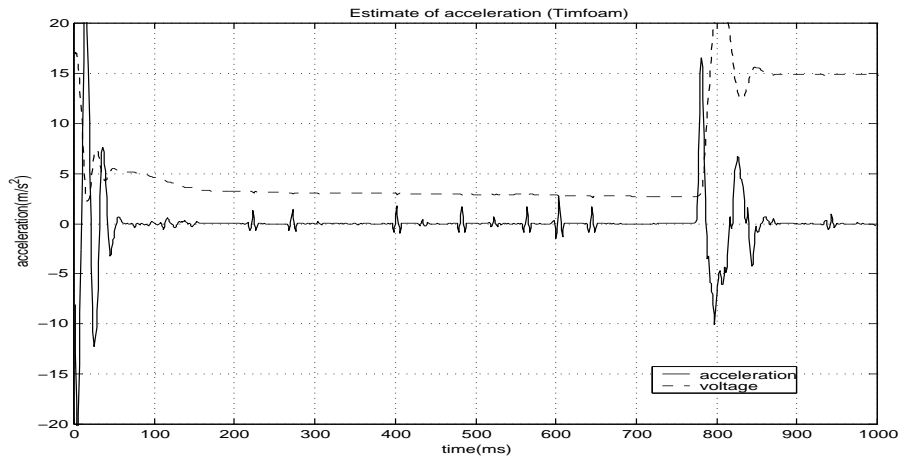


Figure 2.6: Estimation of acceleration from clean data (foam).

2.3 Analysis of the Signal

The graphs describing the response from the plaice fish and foam material are displayed in Figures 2.7 and 2.8. It is obvious from the Figures that the motion of the probe exhibits three distinct behaviours during an experiment (for both the fish and foam cases). Therefore we model these phases as independent processes:

1. **Phase A:** The initial phase resembles an oscillating decaying exponential function which might be modelled as a mass-spring system with damping. This phase coincides with the initial force applied by the probe on the fish or foam.
2. **Phase B:** The probe continues to exert a constant force on the fish (foam), and a linearly (exponentially) decaying function is observed which indicates motion with constant or slowly changing velocity.
3. **Phase C:** In the final phase damped oscillations are again observed as the force is removed.

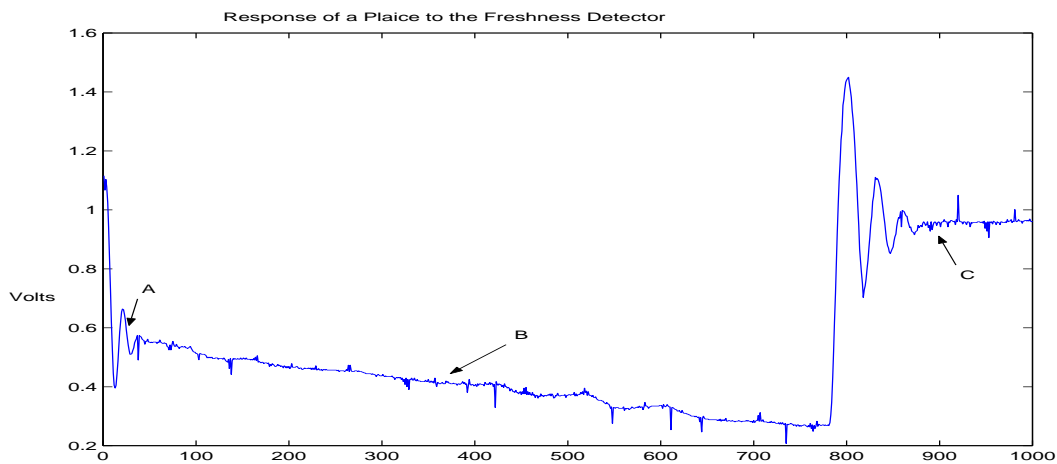


Figure 2.7: The response of the probe to application of a plaice fish.

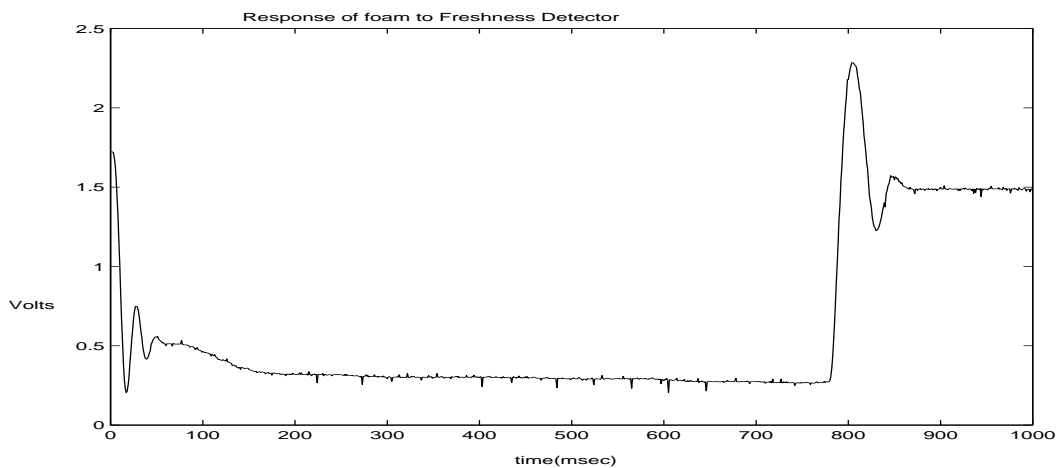


Figure 2.8: The response of the probe to application to foam block.

The following sections describe the motion of the fish-probe system in the the three phases as isolated processes.

2.3.1 Phase A

Due to the oscillatory nature of the voltage (position) over time, we first attempted to model the phenomena by a damped linear oscillator

$$M\ddot{x} + \beta\dot{x} + \alpha x + Mg = 0$$

where $M = 10\text{g}$ is the mass of the probe, g is the gravity constant and $x = x(t)$ is the position of the probe.

With constant coefficients, the solution to the above equation is

$$x(t) = e^{-\delta t} [A \cos(\omega t) + B \sin(\omega t)] + D,$$

where δ is the decay rate, D is the vertical shift and ω is the frequency. These intermediate parameters ω and δ can be expressed by the ODE parameters:

$$\omega = \sqrt{\frac{\alpha}{M} - \delta^2} \quad \text{and} \quad \delta = \frac{\beta}{2M}.$$

We now have to estimate the unknown parameters of the linear ordinary differential equation from the time series data and then inspect the compatibility of the reconstructed model with the data. The MATLAB nonlinear least squares fitting algorithm, NLINFIT, was used to perform the parameter estimation. The results in Figure 2.9 indicate that Phase A can indeed be modelled by the damped linear oscillator.

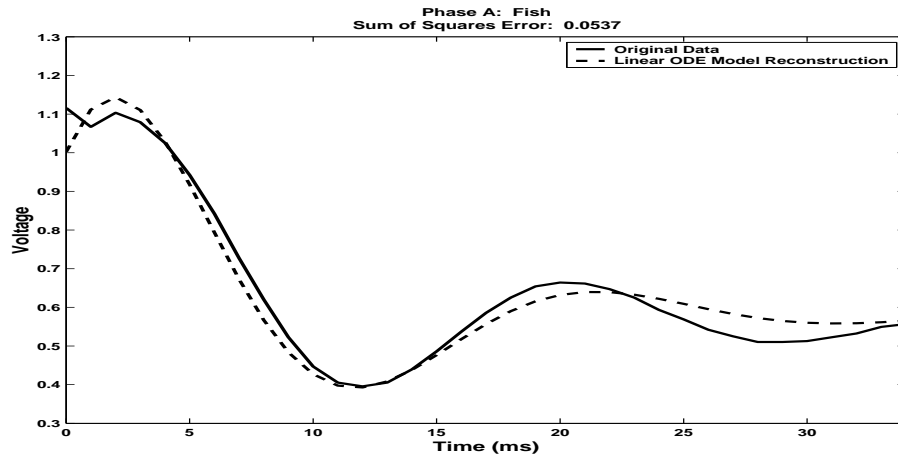


Figure 2.9: The fit obtained for Phase A by modelling the original fish data with a damped linear oscillator.

The initial and final phases of the motion of the probe were similar in nature - both were oscillating decaying exponential functions.

2.3.2 Phase B

The second phase of the time series is observed from 50 to 800ms after the initial contact. It is characterised by a steady decay in the magnitude of displacement following the transient oscillatory behaviour observed in Phase A. We hypothesized that phase B consists of an inelastic deformation because the final equilibrium state at $t = 1000\text{ms}$ is somewhat depressed below the initial condition at $t = 0$.

To model the decrease in the magnitude of the displacement, we attempt to fit the data first with a decreasing exponential function of the form

$$x(t) = Ae^{-\gamma t} + B, \quad (2.7)$$

with parameters A , B and γ . As an alternative, we used a linear model,

$$x(t) = Ct + D, \quad (2.8)$$

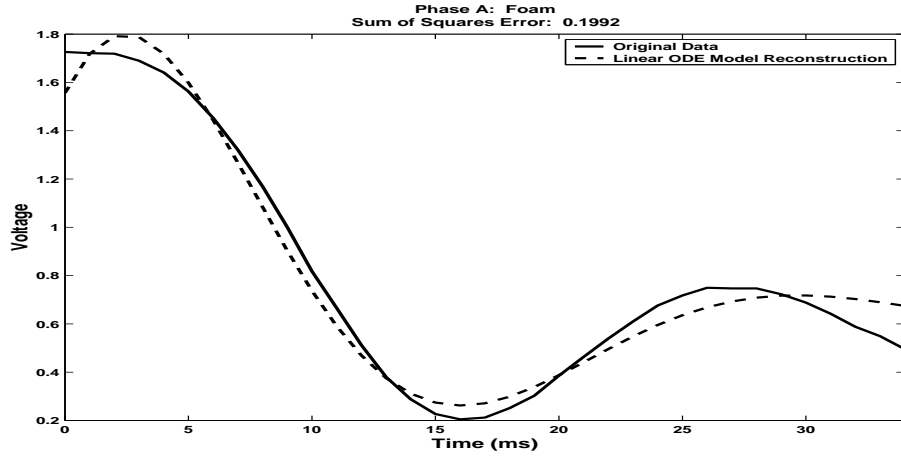


Figure 2.10: The fit obtained for Phase A by modelling the original foam data with a damped linear oscillator.

with parameters C and D .

The residuals are minimized in the least squares sense for the above data sets giving the functions

$$x(t) = 28.45e^{-0.01368t} - 27.91 \text{ or} \quad (2.9)$$

$$x(t) = -0.3874t + 0.5357 \quad (2.10)$$

that fit the fish data and

$$x(t) = 0.2306e^{-19.56t} + 0.2994 \text{ or} \quad (2.11)$$

$$x(t) = -0.3104t + 0.3928 \quad (2.12)$$

for the foam data. These results are plotted in Figure 2.11 and we note the linear and exponential functions are essentially identical for the fish data, both having a least squares residual of about 0.05. For the foam data, the exponential function provides a much better fit with a least squares residual of 0.016 as opposed to 0.47 for the linear fit; this is to be expected because, as seen in Figure 2.11, the linear function cannot model the curvature of the data effectively.

As a possible model of this behaviour, consider a particle subjected to a constant force F moving in a visco-elastic medium with elastic coefficient α and viscosity β . With this model, the kinematics are determined by the equation

$$\alpha x(t) + \beta \frac{dx}{dt} = F. \quad (2.13)$$

Integrating this equation gives the exact solution

$$x(t) = \left(X_0 + \frac{F}{\alpha} \right) e^{-\frac{\alpha}{\beta}t} + \frac{F}{\alpha}, \quad (2.14)$$

where $x(0) = X_0$ is the initial condition for the ODE. In the absence of an elastic term, the kinematics are determined by

$$\beta \frac{dx}{dt} = F \quad (2.15)$$

with corresponding exact solution

$$x(t) = X_0 + \frac{F}{\beta}t. \quad (2.16)$$

It also seems plausible that we could derive Equation (2.16) by considering Darcy's Law for the flow of fluid through a porous medium (the fish flesh). However, we unfortunately did not have sufficient time to develop this idea.

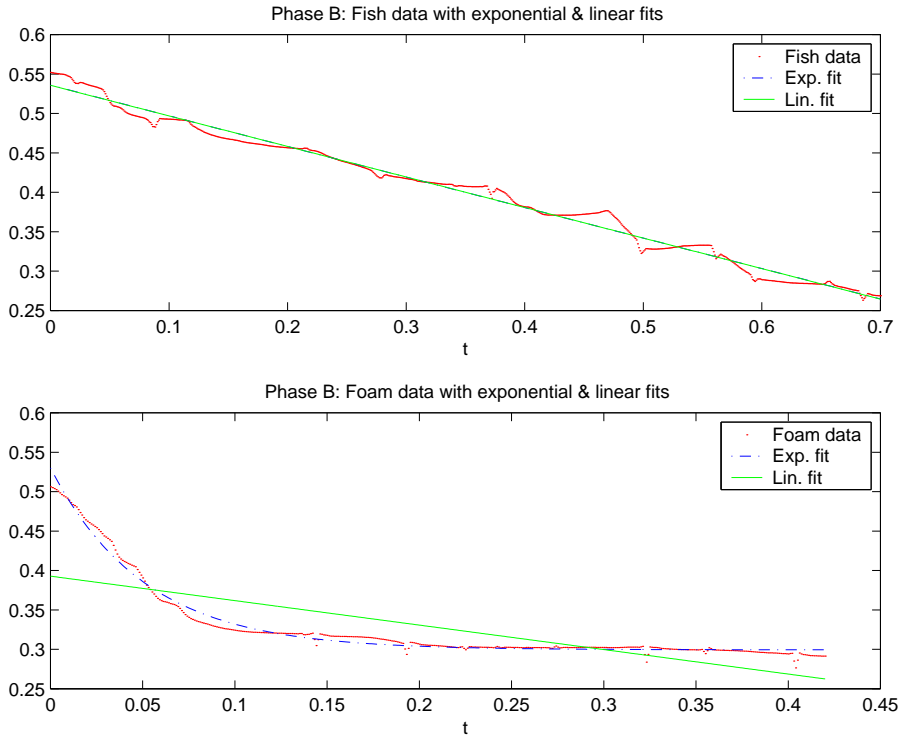


Figure 2.11: Exponential and linear functions fit to the phase B portion of the data from the fish and the foam experiments.

The parameters in the solutions Equation (2.14) and Equation (2.16) can be determined by matching with Equation (2.7) and Equation (2.8).

Further, in the limit of vanishing elasticity α , the solution Equation (2.14) reduces to Equation (2.16) since

$$\begin{aligned}
 x(t) &= \left(X_0 - \frac{F}{\alpha} \right) e^{-\frac{\alpha}{\beta}t} + \frac{F}{\alpha} \\
 &= \left(X_0 - \frac{F}{\alpha} \right) \left(1 - \frac{t}{\beta}\alpha + \frac{1}{2} \left(\frac{t}{\beta} \right)^2 \alpha^2 + \dots \right) + \frac{F}{\alpha} \\
 &= X_0 + (F - \alpha X_0) \left(\frac{t}{\beta} \right) - \frac{1}{2} (F - \alpha X_0) \left(\frac{t}{\beta} \right)^2 \alpha + \dots \\
 &\rightarrow X_0 + \frac{F}{\beta}t \quad \text{as } \alpha \rightarrow 0.
 \end{aligned}$$

Thus, this model suggests that the foam acts as a visco-elastic medium during phase B while the fish acts as a viscous medium with vanishing elasticity. Indeed, this is supported by the small time constant γ in the exponential fit for the fish data.

2.3.3 Phase C

We define phase C as beginning when the force on the probe is released at around 780ms. We consider both the fish and foam data during this phase.

The Fish Data

We first consider the fish data. After the force is released, the probe is pushed upwards by the skin of the fish and enters an apparent damped oscillation.

Our initial model was to describe the motion by the ODE

$$\mu\ddot{x} + \tilde{\beta}\dot{x} + \tilde{\alpha}x + \mu g = 0, \quad (2.17)$$

where $\mu = 1 + \frac{M}{m}$ is the effective mass of the oscillator (M is the mass of the probe, m is a mass unit of the skin) and $\tilde{\alpha}$ and $\tilde{\beta}$ are respectively the restoring and friction parameters. The μg term describes the force due to gravity. We fitted the general solution

$$x(t) = Ae^{-\delta t} \cos(\omega t - \phi) + D, \quad (2.18)$$

to the data by matching the parameters A , δ , ω , ϕ and D where δ and ω are related to ODE coefficients by

$$\delta = \frac{\tilde{\beta}}{2\mu}, \quad (2.19)$$

$$\omega^2 = \frac{\tilde{\alpha}}{\mu} - \delta^2. \quad (2.20)$$

Results of a least squares fit are shown in Figure 2.12.

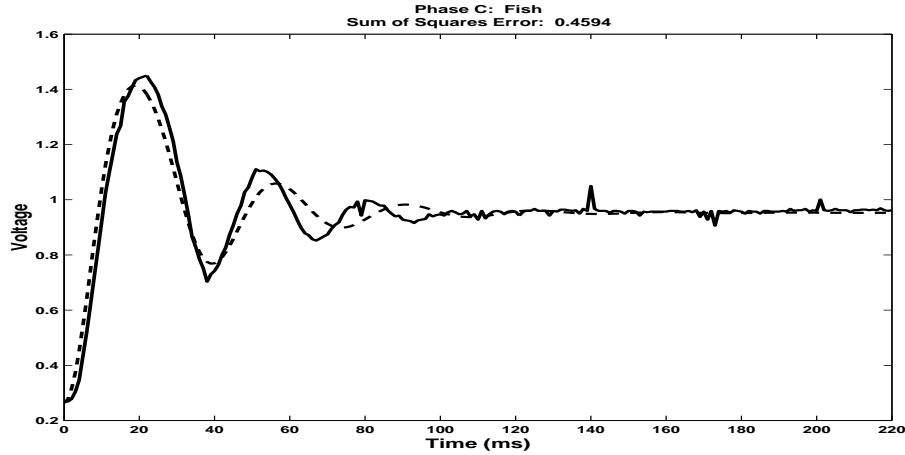


Figure 2.12: Phase C—The fit obtained by modelling the original fish data with a damped nonlinear oscillator.

We note that the fit apparently matches the first period of the oscillations quite well but the solution quickly drifts out of phase. Therefore, the model described by Equation (2.18) should be modified. One strategy is to replace the constant coefficients $\tilde{\alpha}$ and $\tilde{\beta}$ with functions of x and \dot{x} respectively. We considered these corrections up to second order, i.e.,

$$\tilde{\alpha}(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 \quad (2.21)$$

$$\tilde{\beta}(\dot{x}) = \beta_0 + \beta_1 \dot{x} + \beta_2 \dot{x}^2 \quad (2.22)$$

$$(2.23)$$

However, no significant change was noticed between this non-linear model and the linear approach shown in Figure 2.12.

After the failure of the non-linear model, we attempted to gain additional insight into the physics of the problem. Therefore, we considered the velocity and acceleration of the probe as computed from the de-noised data (see Figure 2.13).

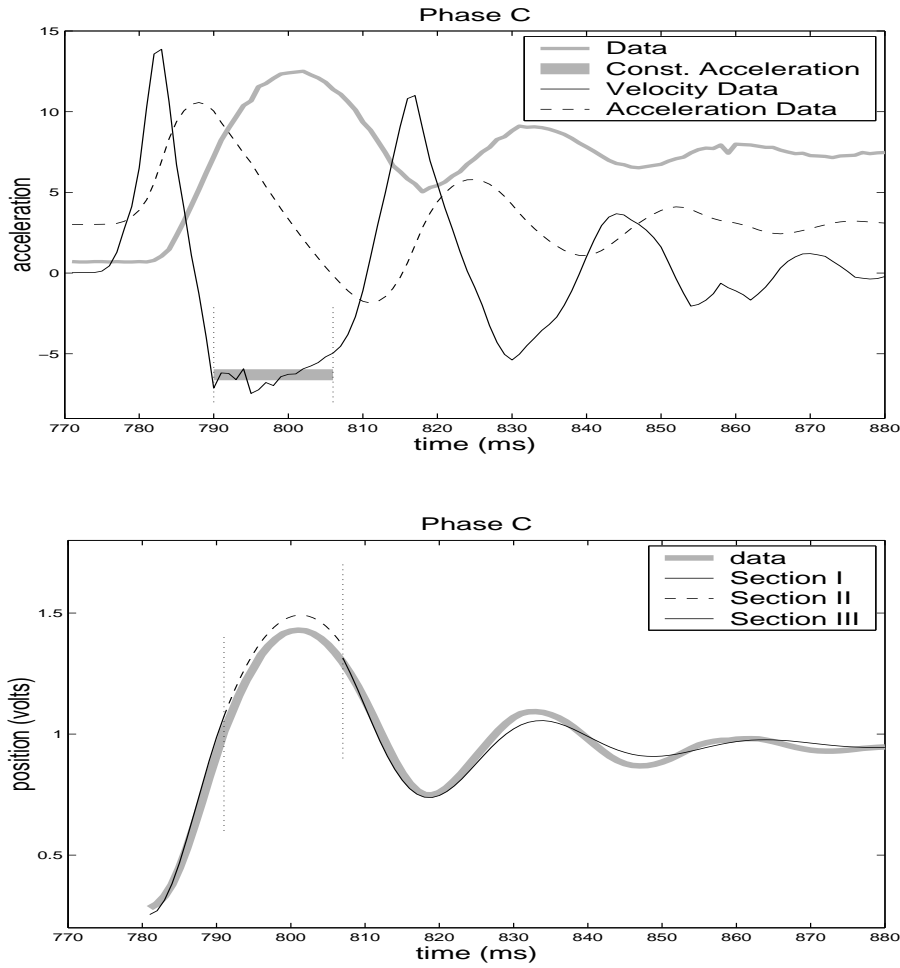


Figure 2.13: (Above) Original Phase C data with acceleration overlay. The region where the probe is in free fall is highlighted.

(Below) The fit obtained by modelling the original fish data as an impact oscillator (loss of contact model).

We noticed that between times $t_1 = 790\text{ms}$ and $t_2 = 808\text{ms}$, the acceleration is almost constant, indicating that the probe is a constant acceleration over 18ms which does not coincide with motion due to a damped oscillator. Note the agreement with the velocity which appears to have a linear behaviour during the same time. However, before and after this acceleration plateau, the motion seems plausibly described by damped oscillations. For convenience we call the time interval $[790\text{ms}, 808\text{ms}]$ Section II and refer to the intervals $[780\text{ms}, 789\text{ms}]$ and $[809\text{ms}, 1000\text{ms}]$ as Section I and Section III, respectively.

The different types of motion of the probe in these three sections suggest that the probe is starting to perform a harmonic oscillation (Section I) and loses contact near $t_1 = 790\text{ms}$. During Section II, the probe (apparently in “free-fall”) is dealt with below. Near time $t_2 = 808\text{ms}$ the skin and the probe collide and again behave as a single oscillator. The motion in Section I and III must be driven by the ODE given in Equation (2.18). Therefore the frequency ω , the decay rate δ , and the baseline D of the signal must be in agreement. However, we allowed that the intermediate Section II introduces an amplitude and a phase correction to the solution of the ODE in the last section. We then fitted the (de-noised) data to the functions

$$x(t) = Ae^{-\delta t} \cos(\omega t + \phi) + D, \quad t \in \text{Section I}, \quad (2.24)$$

$$x(t) = Be^{-\delta t} \cos(\omega t + \theta) + D, \quad t \in \text{Section III}. \quad (2.25)$$

The results of the least squares fit is displayed in the relevant parts of Figure 2.13 (i.e., the two solid curves). We emphasize that the fit is reasonably close to the data. This confirms our conjecture that the probe performs a damped oscillation as long as it is in contact with the skin.

We focus our attention on Section II to understand the motion of the probe over the entire time range of Phase C. After losing contact with the skin, the probe is constantly accelerated over the time range of Section II and therefore its position function $x(t)$ obeys the ODE

$$\ddot{x} - \hat{g} = 0. \quad (2.26)$$

The constant \hat{g} can be determined from the acceleration data in Figure 2.13. We expected that \hat{g} match the gravitational constant $g = -9.81\text{m/s}^2$. However, by taking the average of the acceleration data over the plateau in Section II, \hat{g} turned out to be $\hat{g} = -6.3\text{m/s}^2$. A plausible explanation for this reduced gravitational acceleration could be the effects of friction from inside the apparatus.

To solve Equation (2.26) we chose initial conditions such that the position function $x(t)$ and the velocity $\dot{x}(t)$ became continuous at the point, $t_1 = 790\text{ms}$. The solution is a parabolic path and is plotted in Figure 2.13 as a dashed curve. Note that the “free fall” not only fits the motion in Section II very well but also connects to the damped oscillatory motion in Section III.

Unfortunately, it is difficult to determine $\tilde{\alpha}$ and $\tilde{\beta}$ because of their dependence on μ . This is problematic because we have very little information about μ ; mostly because we did not succeed in finding a way to extract significant information about the motion of the fish skin during Section II. A possible approach is that some information could be extracted by assuming that momentum is conserved by the collision at t_2 . However, we had insufficient time to properly pursue this option.

The Foam Data

We began modelling the foam data in much the same way as we initially modelled the fish data, i.e., as a damped linear oscillator. The resulting model was not a good fit.

The next model attempted was the generalized damped oscillator with non-linear coefficients (again, with up to quadratic corrections) for viscosity, $\beta = \beta_0 + \beta_1 \dot{x} + \beta_2 \dot{x}^2$, and the restoring force, $\alpha = \alpha_0 + \alpha_1 x + \alpha_2 x^2$. This new model demonstrated considerable improvement over the linear case, i.e., the reconstructed ODE model was an excellent fit (see Figure 2.14).

From here, we can conclude that the probe probably did not leave the surface during Phase C. Overall, the probe response for foam can accurately be modelled as a visco-elastic system, as opposed to the probe response for the fish—indicating that the fish is not a perfect visco-elastic material.

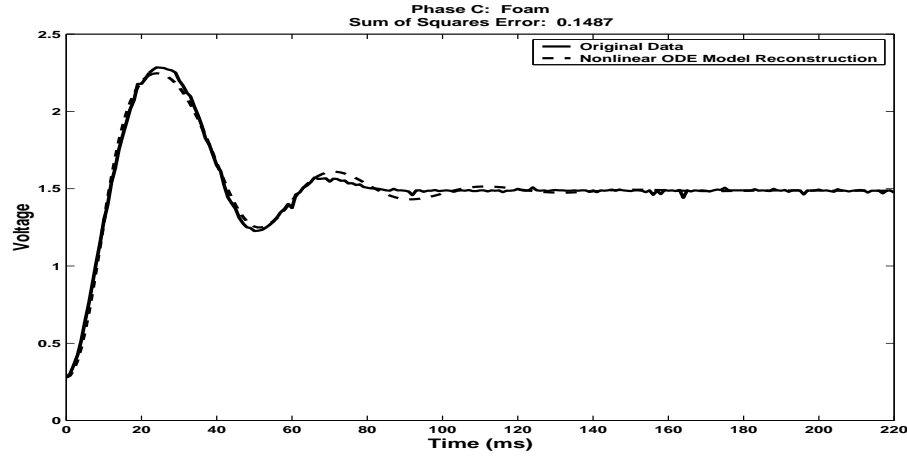


Figure 2.14: Phase C—The fit obtained by modelling the original foam data with a damped non-linear oscillator.

2.4 Conclusions

We gained a fundamental understanding of the dynamics of the given problem for both the fish and the foam data. In this sense, the modelling was successful. In particular, the loss-of-contact model as an impact oscillator during Phase C was surprisingly successful. However, although we have identified many parameters of the model, correlating these parameters with freshness requires several (preferably many) samples of fish of various freshness. Thus, when it comes to predicting freshness of the fish, we simply do not have enough data to make any claims.

It seems plausible that the restoring and damping parameters would be indicators of freshness. Recall that, in Phase C, we were unable to extract the values of these parameters from the frequency, ω , and decay rate, δ , since we could not determine μ . However, it is possible that ω and δ are themselves indicators using μ as a fish-dependent constant.

Acknowledgements

Many thank goes to Chris Budd for providing the group with an interesting project to model and for his guidance as a mentor. Also, many thanks to Jim Verner for his support, laptop, and car.

Source Code

Phase A Codes

Phase_A_Fit.m

```

=====
function [P, V, F]= Phase_A_Fit( time, voltage, t1, t2, baseline )
% Estimate the parameters (coefficients) of the nonlinear ODE for Phase A
% time:      time vector of the signal (SCALE IS 'MS', i.e. 0ms, 1ms, 2ms, ...
% voltage:   position data of the probe/needle
% -- time and voltage must be column vectors
% t1:       time of the first crest in the data
% t2:       time of the second crest in the data
% -- from t1 and t2, a rough estimate of the decay rate and period can be computed,
% whereby all subsequent initial parameters can be computed
% baseline:  guess for the steady-state voltage for Phase A
=====

% Reconstructed Voltage
global V;

% Estimate parameters
Mass = 10;
T = t2-t1; % Period
Omega = 2*pi/T; % Frequency
Delta = -1/T * log(voltage(t2)/voltage(t1)); % Decay rate
Beta = 2*Mass*Delta; % Damping coefficient
Alpha = Mass*(Omega^2 + Delta^2); % Restoring force coefficient
Shift = baseline; % Steady-state voltage
A = voltage(1)-baseline; % Initial amplitude for cos(omega t)
B = 0.1; % Initial amplitude for sin(omega t)
P0 = [Delta Omega Shift A B]; % Consolidate initial parameters into one vector

%Find Optimal Parameters
P = nlinfit(time,voltage,@Phase_A_NLObj,P0);

% Unroll the parameters
Delta = P(1);
Omega = P(2);
Shift = P(3);
A = P(4);
B = P(5);

% Reconstruct the voltage from the estimated parameters
V = exp(-Delta*time).*( A *cos(Omega*time) + B*sin(Omega*time) ) + Shift;

% Compute the sum of squares error term (not normalized to the length of the vector
F = sum((V-voltage).^2);

return

```

Phase_A_NLObj.m

```

=====
function x = Phase_A_NLObj(P,t)
% Objective function to be minimized by NLINFIT
=====

% Unroll parameters
Delta = P(1);
Omega = P(2);
Shift = P(3);
A = P(4);
B = P(5);

% Equation to fit
x = exp(-Delta * t) .* (A*cos(Omega*t) + B*sin(Omega*t)) + Shift;
return

```

Phase_C_Fit.m

```

=====
function [P, V, F] = Phase_C_Fit(time,voltage,t1,t2,baseline)

```

```

% Estimate the parameters (coefficients) of the nonlinear ODE for Phase C
% time:      time vector of the signal (SCALE IS 'MS', i.e. 0ms, 1ms, 2ms, ...
% voltage:   position data of the probe/needle
% -- time and voltage must be column vectors
% t1:       time of the first crest in the data
% t2:       time of the second crest in the data
% -- from t1 and t2, a rough estimate of the decay rate and period can be computed,
% whereby all subsequent initial parameters can be computed
% baseline:  guess for the steady-state voltage for Phase C
%=====

% Reconstructed Voltage
global V;

% Estimate parameters
Mass    = 10;
T       = t2-t1;
Omega   = 2*pi/T;
Delta   = -1/T * log(voltage(t2)/voltage(t1));
Beta    = 2*Mass*Delta;
Alpha   = Mass*(Omega^2 + Delta^2);
Shift   = baseline;
P0      = [[Beta 0 0] [Alpha 0 0] Shift];
IC      = [voltage(1)-baseline; 0];

% Period
% Frequency
% Decay Rate
% Damping coefficient
% Restoring Force Coefficient
% Steady-state voltage
% Consolidate initial parameters into one vector
% Initial conditions for ODE

%Find Optimal Parameters
[P,F] = fminsearch(@Phase_C_Obj, P0, [], time, voltage, IC);
return

```

Phase_C_Obj.m

```

%=====
function f = Phase_C_Obj(P,t,x,IC)
% Compute Objective Function where the minimizer, x, for the objective function is
% derived from the numerical solution of the ODE
%=====

% Plug in current ODE parameters and compute current minimizer, x, based on the
% the numerical solution to the second-order of ODEs
[T,X] = ode15s(@Phase_C_Ode, t, IC, [], P);

% Extract reconstructed voltage
global V;
shift = P(end);
V = X(:,1) + shift;
f = sum( (x-V).^2 );
disp(sprintf('f = %9.7f',f));
return

```

Phase_C_Ode.m

```

%=====
function f = Phase_C_Ode(t,x,P)
% M-function which define the system 2nd-order ODE as system of 1st-order
% ODEs which hypothetically describe the behavior of Phase C
%=====

% Define the coefficients of the ODE
M = 10;
[Beta,Alpha] = tweak_parameters(x,P);

% Compute the RHS of system of 1st-order ODEs
f = zeros(2,1);
f(1) = x(2);
f(2) = (-Beta/M)*x(2) + (-Alpha/M)*x(1);
return

%=====
function [Beta,Alpha] = tweak_parameters(x,P)
% Define the coefficients of the ODE
%=====

% Unroll the parameter vector into meaningful components
beta = P(1:3);
alpha = P(4:6);

```

```

% Define the nonlinear damping and restoring force
Beta = beta(1) + beta(2)*x(2) + beta(3)*x(2)^2;
Alpha = alpha(1) + alpha(2)*x(1) + alpha(3)*x(1)^2;
return

```

Phase C Codes

The code `phasesc_minimizer.m` matches the parameters for Section I and III of Phase C as described above. It uses `phasesc_cost.m` as the cost function to minimize.

`phasesc_minimizer.m`

```

%PHASEC_MINIMIZER This script minimizes the parameters in the
% general solution for Section I and Section III of Phase C
% simultaneously

close all;
clear all;

% uncomment for the fish data
%rawdata = load('bl.mat');

% uncomment for the clean fish data:
load('bl_clean');
rawdata(:, 1) = [1:1024]';
rawdata(:, 2) = ul';

% the time boundaries
T10 = 780+1; % these are off by one!
T1F = 790+1;
T20 = 806+1;
T2F = 879+1;

time1offset = rawdata(T10,1);
time1 = rawdata(T10:T1F, 1) - time1offset;
depth1 = rawdata(T10:T1F, 2);

time2offset = rawdata(T20,1);
time2 = rawdata(T20:T2F, 1) - time2offset;
depth2 = rawdata(T20:T2F, 2);

% this is just used for plotting
alldepth = rawdata(T10:T2F, 2);
alltime = rawdata(T10:T2F, 1);

% number of times to restart optimization. basically we want to
% avoid local minima
N = 100;

% stores parameters values and the associated cost value
Xs = zeros(N,6);
fvals = zeros(N,1);

figure(1);
clf;
plot(alltime, alldepth, 'b-');
hold on;

for n = 1:N
    % these are initial values for the parameters
    X0 = [.04*rand+.02 .4*rand .1*rand+.9 -1*rand -.4*rand pi*rand-pi/2];

    [X,fval] = fminsearch(@pc_secl_cost_5, X0, [], time1, depth1, time2, depth2);
    Xs(n,:) = X;
    fvals(n) = fval;

    delta = X(1);
    omega = X(2);
    D = X(3);
    A = X(4);
    B = X(5);
    phi2 = X(6);

    phi = atan(-delta/omega);

```

```

fval
y1 = A*exp(-delta*time1) .* cos(omega*time1 + phi) + D;
y2 = B*exp(-delta*time2) .* cos(omega*time2 + phi2) + D;
plot(time1+time1offset, y1, 'r-');
plot(time2+time2offset, y2, 'm-');
pause(0);
if(mod(n, 10) == 0)
    disp(sprintf('%10d/%d done', n, N));
end
end

% find the index of one of the global minima
[minfval, js] = min(fvals);
j = js(1);

minfval
delta = Xs(j,1)
omega = Xs(j,2)
D = Xs(j,3)
A = Xs(j,4)
B = Xs(j,5)
phi2 = Xs(j,6)
phi = atan(-delta/omega);

y1 = A*exp(-delta*time1) .* cos(omega*time1+phi) + D;
y2 = B*exp(-delta*time2) .* cos(omega*time2+phi2) + D;

figure(2);
clf;
plot(alltime, alldepth, 'b-');
hold on;
plot(time1+time1offset, y1, 'r-');
plot(time2+time2offset, y2, 'r-');
pause(0);

figure(3);
clf;
plot(sort(fvals));

save pc_plot HACK_y1 HACK_y1p HACK_y2 HACK_y2p T10 T1F T20 T2F delta ...
    omega D A B phi phi2 minfval time1 time2 time1offset time2offset ...
    alltime alldepth

```

phasesec_cost.m

```

function cost = phasesec_cost(X, time1, depth1, time2, depth2)
%PHASEC_COST Cost function to minimized for phase B
% The X is the input to the function and the times and depths
% never change (they are the parameters to this function)

% see below...
global HACK_y1 HACK_y1p HACK_y2 HACK_y2p;

delta = X(1);
omega = X(2);
D = X(3);
A = X(4);
B = X(5);
phi2 = X(6);

phi = atan(-delta/omega);

y1 = A*exp(-delta*time1) .* cos(omega*time1 + phi) + D;
y2 = B*exp(-delta*time2) .* cos(omega*time2 + phi2) + D;

cost = sum((y1 - depth1).^2) + sum((y2 - depth2).^2);

% don't ask don't tell...
HACK_y1 = y1;
HACK_y1p = -A*delta*exp(-delta*time1) .* cos(omega*time1+phi) - ...
    A*exp(-delta*time1) .* sin(omega*time1+phi)*omega;
HACK_y2 = y2;
HACK_y2p = -A*delta*exp(-delta*time2) .* cos(omega*time2+phi2) - ...
    A*exp(-delta*time2) .* sin(omega*time2+phi2)*omega;

```

Bibliography

- [1] D.L., Donoho, De-noising by soft-thresholding, IEEE trans. on Information theory, 41(3), 613–627, 1995
- [2] XP. Zhang, M.D., Desai, Adaptive de-noising based on SURE risk, IEEE signal processing letters, 5(10),265–267, 1998
- [3] Wavelet toolbox for user with Matlab manual, Ver.2, 2000
- [4] W.E., Boyce, R.C., DiPrima, Elementary Differential Equations and Boundary Value Problems, Sixth Edition., John Wiley & Sons, Inc., 1997

Chapter 3

Software Testing Using t -Covering

Participants: Brett Stevens (Mentor), Lorraine Dame, Charles Fortin, Jingxiang Luo, Huamei Yin, Tzvetalin Vasilev, Qingze Zou.

PROBLEM STATEMENT: One of the most expensive and time consuming aspects of industrial software testing is testing with all possible combinations of input parameter values. There are several mathematical models existing whose goal is to minimize the number of combinations of test parameters used to run the software while meeting specified criteria.

One possible criteria is to require a t -covering. A t -covering is a set of test runs that covers all possible t -tuples of parameter values. For instance, a 2-covering is a set of test runs that covers all possible pairs of parameter values. We will refer to each such test run as a test configuration or a test combination.

One important issue that occurs during industrial software testing is that of 'forbidden combinations'. It may be known before testing that a certain combination of parameter values always produces a failure. Performing any test runs that contain these combinations does not provide any information about other interactions of parameters in the test run. Test runs with these combinations should be forbidden.

Another important issue that arises is non-interaction of parameters. It may be known before testing that some parameters do not interact with each other in the software. In this case it might not be necessary to perform extraneous tests of all possible combinations of these parameters. A third issue is non-uniform coverage. It may be known before testing that certain combinations of parameters are very frequently used, and these combinations should be tested more than other combinations. A fourth issue that occurs is zooming in on the failure. When a test combination produces a failed result, it is important to determine which single, pair, triple, etc. of parameter values caused the failure. It is also critical to examine the flexibility of the model to adapt to higher order coverings.

During the workshop, we examined the models in existence to see if they addressed the given issues. We proposed a new graph model, and extended existing models to address some of these issues. The existing models we studied include graph model, algebraic model, array model [2, 3], and integer linear programming model.

Additionally, we examined some situations where we can have some efficient direct constructions. For these constructions, there are more limitations, for example, some models may only handle parameters that all have the same number of possible values (alphabet size). We also studied the case that the alphabet size of the parameters is a prime power, for which some efficient construction methods have been developed.

The remainder of this report describes the various models, and our work on how these issues are addressed in each model.

3.1 An Integer Programming Model

Because of the combinatorial structure of our 2-cov problem, it is natural to expect a formulation in terms of an integer program. We here present such a model (see [6]) and to simplify the notation we shall work our way through an example. It will not be a difficulty to see how the model can be formulated in the general case.

For our example, we consider the case where we have three parameters, each one can take on two values, 0 and 1, and where we wish to find a 2-cover of the parameters. It is clear that in this case there are 8 possible configurations of the parameters. We start by assigning a variable x_i , where $i = 1 \dots 8$, to each of the possible configurations. Precisely we have

$$\begin{aligned} x_1 &: (0, 0, 0) \\ x_2 &: (0, 0, 1) \\ x_3 &: (0, 1, 0) \\ x_4 &: (0, 1, 1) \\ x_5 &: (1, 0, 0) \\ x_6 &: (1, 0, 1) \\ x_7 &: (1, 1, 0) \\ x_8 &: (1, 1, 1). \end{aligned}$$

Each variable x_i is binary and $x_i = 1$ means we select the i -th configuration to appear in our 2-cov.

To have the pair of parameters $(0, 0)$ covered in the first two parameters, we introduce the notation $\{0_1, 0_2\}$, meaning 0 in the first coordinate and 0 in the second coordinate must appear in one of the configurations of our 2-cov. To force this pair of parameter to be 2-covered in our model, we have the constraint

$$x_1 + x_2 \geq 1.$$

Our objective is clear: we minimize

$$\sum_{i=1}^n x_i.$$

We can now formulate our integer programming model where we indicate for each constraint which pair of parameters is being covered.

$$\begin{array}{rcccccccc} & \min & x_1 + & x_2 + & x_3 + & x_4 + & x_5 + & x_6 + & x_7 + & x_8 \\ \{0_1, 0_2\} : & & x_1 & +x_2 & & & & & & & \geq 1 \\ \{0_1, 1_2\} : & & & & x_3 & +x_4 & & & & & \geq 1 \\ \{1_1, 0_2\} : & & & & & & x_5 & +x_6 & & & \geq 1 \\ \{1_1, 1_2\} : & & & & & & & & x_7 & +x_8 & \geq 1 \\ \{0_1, 0_3\} : & & x_1 & & +x_3 & & & & & & \geq 1 \\ \{0_1, 1_3\} : & & & x_2 & & +x_4 & & & & & \geq 1 \\ \{1_1, 0_3\} : & & & & & & x_5 & & +x_7 & & \geq 1 \\ \{1_1, 1_3\} : & & & & & & & x_6 & & +x_8 & \geq 1 \\ \{0_2, 0_3\} : & & x_1 & & & & +x_5 & & & & \geq 1 \\ \{0_2, 1_3\} : & & & x_2 & & & & +x_6 & & & \geq 1 \\ \{1_2, 0_3\} : & & & & x_3 & & & & +x_7 & & \geq 1 \\ \{1_2, 1_3\} : & & & & & x_4 & & & & +x_8 & \geq 1 \end{array}$$

$$x_i \in \{0, 1\}, \quad i = 1 \dots 8.$$

It is easy to see how this model extends in the general case, that is when we have multiple parameters that can take on different number of values (maybe not the same values for each parameter). The idea is to create a binary variable for each possible configuration and add the corresponding constraint for each pair of parameters we wish to cover (for a 2-cover we wish to cover all pairs, but we may decide as well that it is not important to cover some pairs of parameters). In the case of a t -cover, we then have a constraint for each corresponding arrangement of parameters values among all possible combinations of t parameters..

Returning back to our example, we consider now the effect of forbidding a pair of parameters to appear in our selection of testing configurations. Say we do not want to have configurations including the pair of parameters

values $\{0_1, 0_2\}$. We then modify our model by deleting the first constraint and removing the x_1 and x_2 variables (note that in our example, this change implies $x_3 = x_4 = x_5 = x_6 = 1$ and only the constraint $x_7 + x_8 \geq 1$ remains, but this phenomena does not occur in general).

We may also model non-uniform converge with this model. Precisely, if we wish for example to have the first and second parameters covered twice as much as the second and the third parameter, we change the right hand side column of 1's, to 2's for the constraints involving the first and second parameters (in our case, the first four constraints).

The model also allows a lot of freedom in the sense that we may choose the configurations we wish to consider for our tests and which pairs of parameters we wish to cover. This follows from the construction of the variables and the constraints. For example, if we do not want to test the interaction between the second and the third parameter, we would eliminate the four last constraints.

We see that the integer programming model is very flexible as it adapts to the general case where we have multiple parameters that can take on different values and where we consider a t -cover of these parameters. On the other hand, integer programs are hard to solve in general. Although for our particular example the constraint matrix is totally unimodular and therefore we will obtain an integer solution for the LP-relaxation (dropping the integrability constraint), this property does not follow in general and the LP-relaxation will most likely give fractional solutions. In Section 3.3 we consider how we can use the fractional solutions of relaxed linear programs to obtain an approximate solution.

3.2 A Graph Model or Illustrating the Integer Programming Model

We keep on working with the example of the previous section and see how the integer programming model translates to some problems in graph theory. We will see that in general it translates to the problem of decomposing a graph into particular subgraph. In a more particular case, we will also see the problem is equivalent to the vertex cover of a graph. This last model was a fortunate discovery we made during the workshop, while the first one was presented to us.

3.2.1 Graph Decomposition

Recall that for our example we have 3 binary parameters. In the graph in Figure 3.1, each vertical pair of vertices represents the 0-1 values that each parameter can take.

If we consider the configuration $\{0_1, 0_2, 1_3\}$ it covers the 3 pairs $\{0_1, 0_2\}$, $\{0_1, 1_3\}$ and $\{0_2, 1_3\}$. These pairs correspond to edges on the graph and the configuration corresponds to the triangle with vertices $0_1, 0_2$ and 1_3 . Finding a 2-cover is then equivalent to finding triangles (graph of the type K_3) with vertices in each vertical pair of vertex whose union covers all the edges of the graph. In general, if we have n parameters and are looking for a 2-cover, we then wish to find graphs of the type K_n (with vertices corresponding to a parameter value for each of the n parameters) whose union covers all of the edges of the graph. In the case of a t -cover, we do not want to cover edges anymore, but all the possible complete subgraphs with t -vertices.

3.2.2 Vertex Cover

We now show how we can formulate our problem in terms of vertex cover. In Figure 3.2, we assign to each vertex a configuration. Precisely,

$$\begin{aligned} a &: (0, 0, 0) \\ b &: (0, 0, 1) \\ c &: (0, 1, 1) \\ d &: (1, 0, 1) \\ e &: (1, 1, 1) \\ f &: (1, 1, 0) \\ g &: (1, 0, 0) \\ h &: (0, 1, 0) \end{aligned}$$

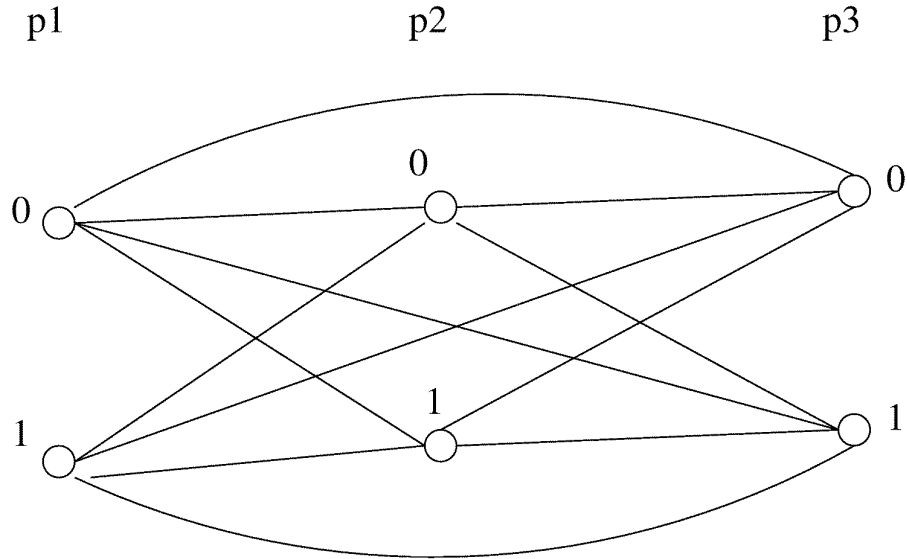


Figure 3.1: 2-cover for 3 binary parameters.

We then join with an edge the configurations that share a pair of parameter values. For example, we join a to b since they share the pair $\{0_1, 0_2\}$. Because only a and b share this pair in common, either configuration a or b must be in our 2-cover. Equivalently, either vertex a or b must cover (in the sense of vertex cover) the edge $\{a, b\}$. Hence, in this example, finding the minimal vertex cover is equivalent to finding a minimum 2-cover. In this case, $\{a, c, d, f\}$ form a vertex cover and $\{(0, 0, 0), (0, 1, 1), (1, 0, 1), (1, 1, 0)\}$ is a minimum 2-cover.

Now consider the example where we have 4 binary parameters and we still look for a 2-cover. We now label the 16 vertices in the following way,

a :	$(0, 0, 0, 0)$	i :	$(1, 0, 0, 0)$
b :	$(0, 0, 0, 1)$	j :	$(1, 0, 0, 1)$
c :	$(0, 0, 1, 0)$	k :	$(1, 0, 1, 0)$
d :	$(0, 0, 1, 1)$	l :	$(1, 0, 1, 1)$
e :	$(0, 1, 0, 0)$	m :	$(1, 1, 0, 0)$
f :	$(0, 1, 0, 1)$	n :	$(1, 1, 0, 1)$
g :	$(0, 1, 1, 0)$	o :	$(1, 1, 1, 0)$
h :	$(0, 1, 1, 1)$	p :	$(1, 1, 1, 1)$

and again join by an edge the vertices which share a common pair of parameter values. The graph we obtain is shown on Figure 3.3. A minimum vertex cover is $\{a, b, c, e, f, g, h, i, j, k, l, m, o, p\}$! Only two vertices do not appear in the cover. This is not a surprise as the graph is almost complete. Yet the corresponding configurations do not form a minimum 2-cover. The reason for this is that there is now more than two configurations sharing a same pair of parameter values. Thus the vertex cover here only give an upper bound on the size of a minimum 2-cover (a pretty bad one in this particular case). However if we would now consider 3-cover for the same problem, i.e. that our configuration must include all triples of possible parameter values, then there is no more than two configuration sharing a same triple. For example, $\{0_1, 0_2, 0_3\}$ is shared by the configurations a and b only. On the graph of Figure 3.4 we join each vertices that share a common triple of parameter values and we obtain the minimum vertex cover $\{a, d, f, g, j, k, m, p\}$. The set of corresponding configurations is also a minimum 3-cover. Thus what we have shown is that for n binary parameters the problem of finding a minimum $n - 1$ -cover is equivalent to finding a minimum vertex cover of the associated graph.

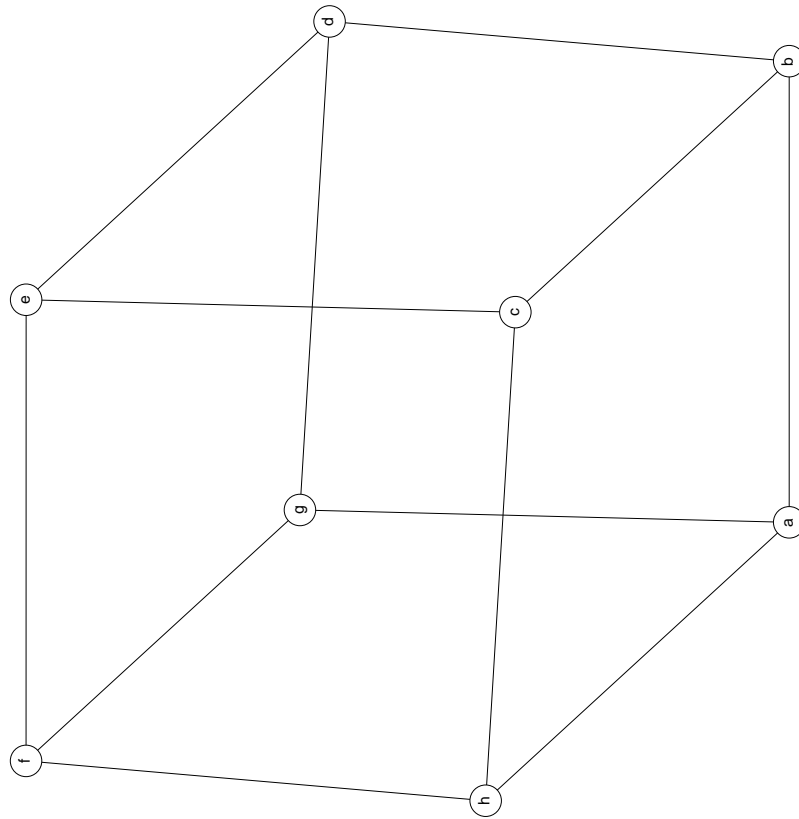


Figure 3.2: 2-cover for 3 binary parameters

3.3 Using the LP-Relaxation to Obtain a 2-Cover

We have mentioned that the optimum solution from the LP-relaxation of our integer programming model will most likely be fractional. Yet by rounding all non-zero variables of the optimum solution to 1, we obtain a feasible solution for the integer program, which may however not be optimal.

Another possibility is to relax the polytope defined by the linear constraints. We may do so by changing some of the right hand side scalars in the constraints, which are all 1's in the model, to 2. For example, we may change the first constraint in the example of Section 3.1 to

$$x_1 + x_2 \geq 2.$$

Obviously, if this constraint is satisfied, the original one with 1 on the right hand side is also satisfied. Therefore changing some of the 1's to 2's in the constraints enlarges the feasible polytope.

As an example, we could solve the following relaxed problem:

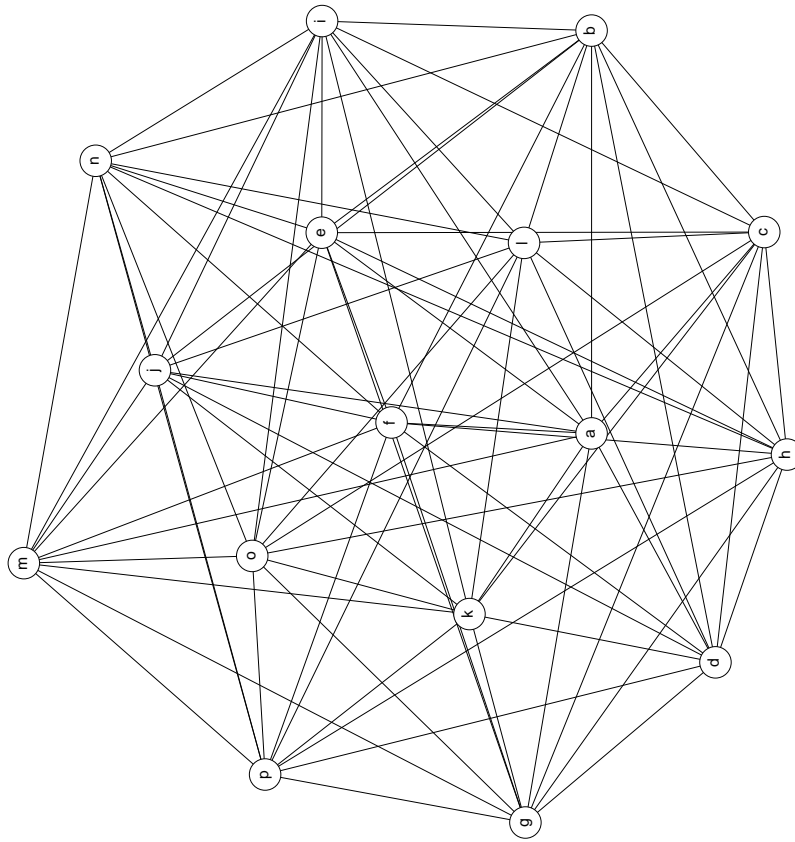


Figure 3.3: 2-cover for 4 binary parameters

$$\begin{array}{r}
 \min x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 \\
 \{0_1, 0_2\} : \quad x_1 \quad +x_2 \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \geq 1 \\
 \{0_1, 1_2\} : \quad \quad \quad \quad \quad x_3 \quad +x_4 \quad \quad \quad \quad \quad \quad \quad \quad \geq 2 \\
 \{1_1, 0_2\} : \quad \quad \quad \quad \quad \quad \quad \quad \quad x_5 \quad +x_6 \quad \quad \quad \quad \quad \geq 1 \\
 \{1_1, 1_2\} : \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad x_7 \quad +x_8 \geq 1 \\
 \{0_1, 0_3\} : \quad x_1 \quad \quad \quad +x_3 \quad \quad \quad \quad \quad \quad \quad \quad \quad \geq 1 \\
 \{0_1, 1_3\} : \quad \quad \quad x_2 \quad \quad \quad +x_4 \quad \quad \quad \quad \quad \quad \quad \geq 1 \\
 \{1_1, 0_3\} : \quad \quad \quad \quad \quad \quad \quad \quad \quad x_5 \quad \quad \quad +x_7 \quad \quad \quad \geq 2 \\
 \{1_1, 1_3\} : \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad x_6 \quad \quad \quad +x_8 \geq 1 \\
 \{0_2, 0_3\} : \quad x_1 \quad \quad \quad \quad \quad \quad +x_5 \quad \quad \quad \quad \quad \quad \geq 1 \\
 \{0_2, 1_3\} : \quad \quad \quad x_2 \quad \quad \quad \quad \quad \quad \quad \quad +x_6 \quad \quad \quad \geq 1 \\
 \{1_2, 0_3\} : \quad \quad \quad \quad \quad x_3 \quad \quad \quad \quad \quad \quad \quad \quad +x_7 \quad \quad \geq 2 \\
 \{1_2, 1_3\} : \quad \quad \quad \quad \quad \quad \quad \quad x_4 \quad \quad \quad \quad \quad \quad +x_8 \geq 1 \\
 \\
 x_i \geq 0, \quad i = 1 \dots 8
 \end{array}$$

From an optimal solution to this relaxed problem we may obtain a feasible (integer) solution to our integer programming model by setting all non-zero variables to 1. We can then repeat this procedure by generating the 1's and 2's randomly on the right hand side. Rounding each optimal solution as we described give us a feasible solution to the original integer programming problem and choosing the one which has the least non-zero variables

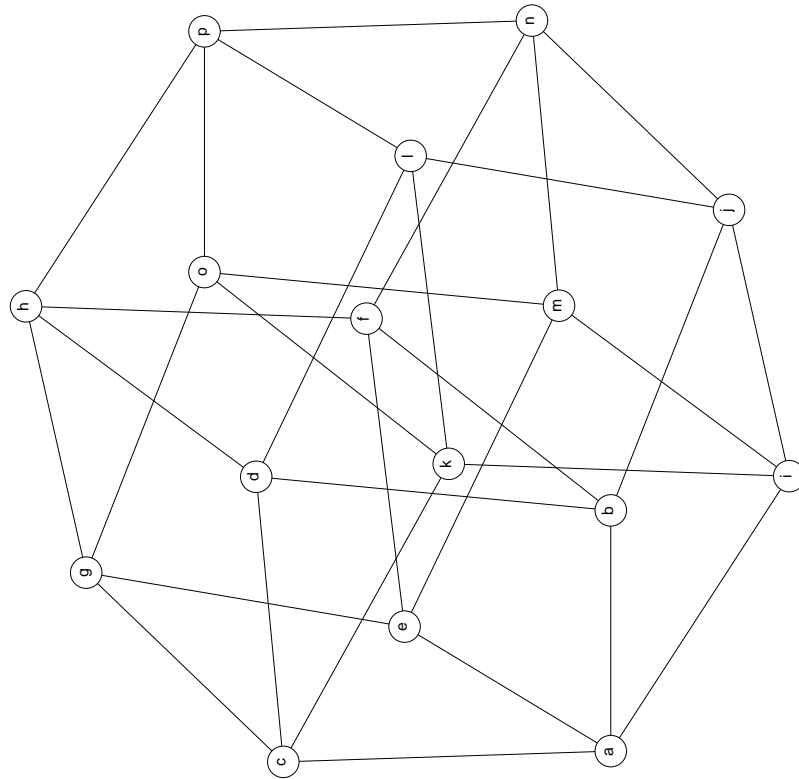


Figure 3.4: 3-cover for 4 binary parameters.

gives us an approximation for the optimal 2-cover (in the case where our integer model was modelling a 2-cover). We have tested this method on two different problems (unfortunately of small size since Maple was too slow on larger size problem and we did not have access to Matlab’s fast LP solver). Tables 3.1 and 3.2 below give the results of the experiments. For each problem we solved different relaxed LP problems. The first column of our tables gives the number of 1’s that were randomly replaced by 2’s in the relaxed problems. The second column gives the number of non-zero variables for the optimal solution (recall that if k variables are non-zero for an optimal solution, we can recover a 2-cover with k configurations). Each row in this column give the results for 4 different problems we randomly generated.

number of 2’s on the RHS	number of non-zero variables at optimum
0	8, 11, 8, 8
1	5*, 5*, 5*, 8
2	8, 5*, 5*, 5*
3	9, 6*, 8, 6*

Table 3.1: 2-cover for 4 binary parameters.

The starred numbers correspond to solutions that were already integral, i.e. no rounding of the non-zero variables was needed. For the first problem, since 5 is the smallest number which appears in Table 3.1, the smallest size 2-cover we can construct from our approximations has 5 configurations. It turns out this number is the optimal size of the 2-cover. For the second problem, we obtain a similar result: we can construct a 2-cover of

number of 2's on the RHS	number of non-zero variables at optimum
0	8, 8, 8, 8
1	11, 11, 16, 12
2	10, 10, 9, 10
3	6*, 6*, 10, 10

Table 3.2: 2-cover for 5 binary parameters.

size 6 and this number turns out to be the smallest size 2-cover for this problem.

3.4 On the Geometric-Algebraic/Array Model and Some Results

3.4.1 Background

Suppose we have k variables, and each variable can be assigned a value from an alphabet (size g). Each such assignment is called a test case, and will produce a result of pass / fail. Usually $k \gg g$.

which includes as few as possible test cases.

t -cover in such a situation.

g values in the alphabet Σ .

A geometric model maps a test case to exactly a point in the vector space Σ^k . The mapping is 1-1. All the possible test cases constitute a hypercube Σ^k . Each point is represented by (v_1, v_2, \dots, v_k) in Σ^k . A t -cover is a set of points that is incident to any t -dimensional hyperplanes. Each $(k-1)$ -cover is a hyperplane in Σ^k , however, the reverse is not true! To note, all such planes in $(\mathcal{F}_g)^k$ can be represented by an algebraic linear equation where any variable and any coefficients belongs to $(\mathcal{F}_g)^k$.

In the following, when g is a prime power, we use $(\mathcal{F}_g)^k$ to denote a finite field of size g (using modular g arithmetic). In this situation, the construction of $(k-1)$ -cover is known to be particularly efficient.

In an array model, a set of test cases is represented by a 2d-array: each column corresponds to a variable; each test case corresponds to a row, which designates an assignment of the k variables. Array model is very convenient for: (a) the examination of test result, (b) the expansion of a cover.

i.e. to expand t -cov(k, g) to t -cov(k', g), where $k' > k$, through the

i.e. to expand t -cov(k, g) to t -cov(k', g), where $k' > k$, through the method of using multiple copies of the original cover in columns and making additional test cases.

3.4.2 Identifying Covers from Hyperplanes—Algebraic Approach

This subsection identifies $(k-1)$ -cover from hyperplanes of $(\mathcal{F}_g)^k$. Corollaries are derived on $(k-1)$ -cov(k, g) and the number of such covers in $(\mathcal{F}_g)^k$.

We claim that the $(k-1)$ -cover can be identified very simply, as shown below.

Now from the view of geometric-algebraic construction, each test case is a point in $(\mathcal{F}_g)^k$. Note that in a $S \subset (\mathcal{F}_g)^k$, a $(k-1)$ -dimensional hyper-plane can be always represented by following:

$$C_{\alpha_1 \dots \alpha_k} : \alpha_1 p_1 + \alpha_2 p_2 + \dots + \alpha_k p_k = 0 \quad (3.1)$$

where (p_1, p_2, \dots, p_k) are the k parameters, $p_i \in \mathcal{F}_g, i = 1, 2, \dots, k, \alpha_1, \alpha_2, \dots, \alpha_k$ are the k coefficients, $\alpha_i \in \mathcal{F}_g, i = 1, 2, \dots, k$.

Now we will show that

Theorem 1 In formula (3.1),

$$(\forall i \alpha_i \neq 0) \longrightarrow C_{\alpha_1 \dots \alpha_k} \text{ represents a } (k-1) \text{-cover} \quad (3.2)$$

Proof From $(\mathcal{F}_g)^k$, we choose any combination of $(k - 1)$ parameters (say p_1, p_2, \dots, p_{k-1} , without losing generality), there is only one parameter left (in our case, p_k), which can be solved as:

$$p_k = -\alpha_k^{-1}(\alpha_1 p_1 + \alpha_2 p_2 + \dots + \alpha_{k-1} p_{k-1})$$

Since we discuss in $(\mathcal{F}_g)^k$, where g is a prime power, so α_k^{-1} exists, which means that for any $(p_1, p_2, \dots, p_{k-1})$, we can find the solution p_k shch that $(p_1, p_2, \dots, p_{k-1}, p_k)$ satisfies equation (3.1). By the definition, equation (3.1) represents a $(k - 1) - cover$.

Corollary 1.1 *The size of $(k - 1) - cov(k, g)$ can be obtained as:*

$$(k - 1) - cov(k, g) = g^{k-1}$$

Proof Since equation(1) represents a $(k - 1)$ -dimension hyper-plane in $(\mathcal{F}_g)^k$, it includes g^{k-1} points. This immediately leads to the corollary 1.1. ■

Corollary 1.2 *There exists exactly $(g - 1)^{k-1}$ different covers in $(\mathcal{F}_g)^k$.*

Proof We only need to notice:

1. C is a cover $\longrightarrow \forall i, \alpha_i \neq 0 \longrightarrow$ each α_i can pick up values from $(1, 2, \dots, g - 1)$.
2. If $(\alpha_1, \alpha_2, \dots, \alpha_k)$ represents a hyper-plane, by multiplying both sides with $\lambda \in (\mathcal{F}_g)^k$, i.e.

$$\lambda \alpha_1 p_1 + \lambda \alpha_2 p_2 + \dots + \lambda \alpha_k p_k = 0$$

still represents the same plane.

From Eq. (3.1) and (3.2), we conclude that the number of $(k - 1) - cover$ is

$$\frac{(g - 1)^k}{g - 1} = (g - 1)^{k-1}$$

■

3.4.3 Developing Strategies for Zoom-in Test

This subsection concerns zoom-in test, as we mentioned earlier. Suppose we have observed some failure cases in the initial test set. $t - cover$ test can detect any faulty $t - tuple$. In a practical scenario, sometimes we are interested in which $t - tuple(s)$ actually causes the failure.

set of tests to locate the faulty $t - tuple$, in which we want to have as few as possible test cases.

There are several situations:

Situation	Failure Cases in the initial set of tests	Assumptions of faulty pairs
A1	Single	Exact one
A2	Single	Multiple
A3	Multiple	Single/Multiple

To start, we start from the simple case $g = 2, t = 2, k > 2$ in the following discussion.

For situations as labelled in (A1), (A2), we claim that we can design a new test set which include $k, \binom{k}{2}$ test cases, respectively.

We first consider that there is a single failure case in the initial test set. Suppose $g = 5$. Without losing generality, assume the single failure case captures is $(0, 0, 0, 0, 0)$. Then we design the following test set, which includes $k = 5$ test cases, each time we flip one of the variables in the initial failure case. The strategy for detecting the faulty fair is as follows. If the result of a new test case is pass, then the variable flipped in this test

case is one of the faulty pairs, otherwise it is not. This strategy is justified as follows. Assume any faulty pair $(p_i, p_j)(i, j \neq 5)$ is faulty, in this example, we will get that $(0, 0, 0, 0, 1)$ will fail.

Thus we have designed a new set of tests to detect the faulty pair for situation (A1). To extend this strategy, we can design a new set of tests in which each time two of the variables are flipped. From this design the number of test cases is $\binom{k}{2}$ as we have claimed.

To further the study of this problem, we note several directions to be explored:

- If $k \gg g$, there is usually multiple cases for a special pair $(p_i = v, p_j = v')$. According to the definition of a faulty pair, we can immediately exclude the pair $(p_i = v, p_j = v')$ from being a faulty pair if any of the test cases that satisfies $(p_i = v, p_j = v')$ succeeds. This reveals, just by a through analysis of the output, some pairs can be excluded from being a faulty pair.
- At this point, we are unable to show that our construction of a new test set is the best possible solution. Attempting for a (possible) better solution for this question may shed lights on “smarter” design of test cases.
- To tackle the question more generally, e.g. $g > 2$, etc.

3.5 Array Model: Results and Extensions

In this section we look at the problem of generation of testing sets, from the point of view of combinatorial design. We present some results about 2-coverings and 3-coverings, recursive constructions of covering arrays and give some ideas about the future work towards the possible extension of this model.

3.5.1 Basic Definitions

We start our discussion with the following definitions:

Definition 1 *We assume that we have k parameters, they can be viewed as the inputs and/or outputs of the actual software to be tested. Each of the parameters can take v different values. For simplicity we assume that these are values of the alphabet V , consisting of all the symbols form 0 to $(v - 1)$. $V := \{0, 1, 2, \dots, v - 1\}$. We also assume that it will be sufficient to test all t -tuples of parameters, where $1 \leq t \leq k$.*

It is clear that the situation with $t = 1$ corresponds to the testing of each individual variable, and the situation with $t = k$ corresponds to the full testing of all possible combinations of values of the parameters. Naturally, no test except the full test can reveal the faulty behaviour of the tested system. However, in practise it might be sufficient to only test the smaller subsets of the parameters and it will give the acceptable level of certainty about the behaviour of the tested system. Another reason to try not to test the set of all possible combinations of parameter values is that industrial systems that are subject to such a test are of size that does not allow exhaustive testing in reasonable time.

Definition 2 *An covering array A is the array of size $N \times k$, such that each row in the array corresponds to one test combination and each column in the array corresponds to one of the parameters in our model. Thus, the i^{th} -element in the j^{th} row of A , denoted by $A(i, j)$, represents the value of the j^{th} parameter in the i^{th} test combination. Thus, $A(i, j)$ is one of the symbols in our alphabet. Furthermore, the covering array has the property that it contains all possible t -tuples of values for all possible choices of t parameters among the k parameters.*

From Def. 2, it is clear that if we add an row of arbitrary values to a covering array, the resulting array, which is of size $(N + 1)$ will also be a covering array. If there are redundant (equivalent, identical) rows in the covering array, we may remove all the copies and leave just one, this will also preserve the coverage. Thus, we shall assume that all the rows in the covering array are different. For some covering arrays it will be possible to remove a row (or number of rows) and still have the remaining rows representing a covering array. However, this is not always possible. For each triple (t, k, v) , there exists a covering array (or set of covering arrays) such that no row can be removed without a loss of the covering property.

Definition 3 For the triple (t, k, v) , the minimum covering array is a covering array that has minimum number of rows. The size of the minimum covering array (the number N of rows) is denoted by $t - cov(k, v)$.

As it is clear, the minimum covering array always exists, but there might be many of them, i.e. the minimum covering array is not necessarily unique.

Property 1 The size of the minimum $t - covering array with v alphabet and k parameters is greater or equal to the t^{th} power of the alphabet v :$

$$t - cov(k, v) \geq v^t \tag{3.3}$$

Suppose we have chosen t out of the k parameters, for simplicity—the parameter $1, 2, \dots, t$. To cover all t -tuples of their possible values, we need at least v^t rows, because there are v^t combinations that we need to test. Thus, we need at least v^t rows to test only the possible t -tuples for any t parameters. It may be enough in some cases to cover all the combinations. Generally though, it will not be sufficient. From this observation, we have the following two important consequences:

1. We want to be able to identify the cases when the $t - cov(k, v)$ is equal to v^t and more importantly, to construct covers with an optimal length, i.e. the testing array with minimal size.
2. We want to obtain as tight as possible the upper bounds for the size of the covering array, when the optimal size is not achievable.

For Examples, $1 - cov(k, v)$ is the following array with v rows

$$\begin{matrix} 0 & 0 & 0 & \cdot & 0 \\ 1 & 1 & 1 & \cdot & 1 \\ 2 & 2 & 2 & \cdot & 2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (v - 1) & (v - 1) & (v - 1) & \cdots & (v - 1) \end{matrix}$$

$k - cov(k, v)$ is the array with v^k rows representing all different k symbol words over the alphabet V

$$\begin{matrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & (v - 1) \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 1 \\ \vdots & \vdots & \vdots & \cdots & \ddots & \vdots \\ (v - 1) & (v - 1) & (v - 1) & \cdots & (v - 1) & (v - 1) \end{matrix}$$

3.5.2 2-Covers: Algebraic Construction

Lemma 1 For any number $q = g^r$ with g a prime number and $r \in \mathbf{Z}$ (The set of positive integer), the size of a $2 - cov(q + 1, q)$ is q^2 .

Proof Consider the finite field \mathcal{F}_q defined by the elements of $V = \{0, 1, 2, \dots, q - 1\}$, and the operations of addition and multiplication modulo q . Furthermore, construct the 3-dimensional vector space $V_3 := \mathcal{F}_q \times \mathcal{F}_q \times \mathcal{F}_q$. The affine space of V_3 has q^3 points, q^3 choices of 2 lines, etc. The projective geometry is defined as following: in the affine geometry, points represent the lines through the origin, and lines represent the planes through the origin, subspaces of order j (i.e., j -dimensional subspaces) represent the subspaces of order $(j + 1)$ through the origin, etc. We can always choose one of the points in the projective geometry as the infinity point. Then, all the lines through this point will have the same number of points from the point set of the projective geometry. If we label

the points on each of these lines uniformly with labels 0 to $q - 1$, and $q + 1$ such lines can be obtained. This can be seen from the fact that the cardinality of the set of the points in the projective geometry is $q^2 + q + 1 = q(q + 1) + 1$. The points on each line will then represent the corresponding values of one of the parameters, and we have $q + 1$ parameters. Now each test row will correspond to a hyperplane that has one point from each line. It can be shown that q^2 such hyperplanes cover the set of all possible pairs of values for each parameter. Another way to see this is to build the covering array explicitly. The $2 - cov(q + 1, q)$ is an array of size $q^2(q + 1)$. The first two columns of this array, labelled respectively a and b represent all possible pairs of symbols, and there are q^2 such pairs. The rest of the array with $q - 1$ columns are obtained by the linear combinations of a and b of the form $a + \alpha_i b$, where $\alpha_i \in \{1, \dots, q - 1\}$, represents the $i + 1^{th}$ column:

$$\begin{array}{cccccccc}
 a & b & a + b & a + 2b & a + 3b & \dots & a + (q - 1)b \\
 0 & 0 & 0 & 0 & 0 & \dots & 0 \\
 0 & 1 & 1 & 2 & 3 & \dots & (q - 1) \\
 0 & 2 & 2 & 4 & 6 & \dots & (q - 2) \\
 \vdots & \vdots & \vdots & \dots & \dots & \ddots & \vdots \\
 (q - 1) & (q - 1) & (q - 2) & (q - 3) & (q - 4) & \dots & 0
 \end{array}$$

The addition and multiplication are modulo q . Now we can show that each pair of values for each parameter is covered, i.e. that the array so constructed is in fact a $2 - cov(q + 1, q)$. Consider the pair (x, y) in the the columns a and $a + \alpha_i b$. We can determine that $a = x$, $a + \alpha_i b = y$, thus $\alpha_i b = y - x$, and therefore $b = (y - x)/\alpha_i$, which determines the location of the pair (x, y) . Analogously, if we are interested in finding the pair (x, y) in the pair of columns b and $a + \alpha_i b$, then we have: $b = x$, $a + \alpha_i b = y$, thus, $a = y - \alpha_i x$, which determines the location of the pair (x, y) . Last, if we want to find where the pair (x, y) appear in the pair of columns $a + \alpha_i b$ and $a + \alpha_j b$, we have: $a + \alpha_i b = x$, $a + \alpha_j b = y$, subtracting the first equation from the second we obtain $(\alpha_j - \alpha_i)b = y - x$, thus $b = (y - x)/(\alpha_j - \alpha_i)$, and $a = x - \alpha_j(y - x)/(\alpha_j - \alpha_i)$. Thus, we have proven that for each prime power, the existence of the above mentioned properties is guaranteed, the optimal 2-covering is possible, and $2 - cov(q + 1, q) = q^2$. ■

This construction gives us the optimal 2-coverings for infinitely many pairs of values (k, v) . It remains to see whether there are other values that have the same property and how this can be used to derive upper bounds for $2 - cov(k, v)$ for all possible pairs (k, v) . Such constructions are presented in the next section.

3.5.3 2-Covers: Recursive Construction

There are two logical questions that we can ask about the two-coverings, based on the fact that we know $2 - cov(k, v)$ for certain values of (k, v) . First, how can we increase the number of parameters keeping the symbol set constant? What effect will this have on the size of the array? Second, how can we increase the number of values for each parameter, i.e. the symbol set size, while keeping the number of the parameters constant? Thirdly, how will this affect the size of the optimal 2-covering?

Lemma 2 *If $2 - cov(k, q_1) = q_1^2$ and $2 - cov(k, q_2) = q_2^2$, then $2 - cov(k, q_1 q_2) = (q_1 q_2)^2$.*

Proof Consider the product of the two arrays, with sizes $q_1^2 \times k$ and $q_2^2 \times k$, defined as following. We align vertically q_2^2 copies of the array representing $2 - cov(k, q_1)$, we also assume that the symbols in the two alphabets are distinct, and we redefine the alphabet as the product of the two alphabets. In the i^{th} copy of $2 - cov(k, q_1)$ we redefine the symbols in the j^{th} row to be the product of the vectors representing the j^{th} row from $2 - cov(k, q_1)$ and the i^{th} row of $2 - cov(k, q_2)$. Thus we have taken into account of any possible pair of symbols that may occur, and thus $2 - cov(k, q_1 q_2) = (q_1 q_2)^2$. The merit of Lemma 2 is that for any number k of parameters that equals to one plus a prime power, we can construct $2 - cov(k, v) = v^2$, for any number v . To obtain this result, we need to notice two important properties of the coverings. ■

Property 2 *Row collapsing: $t - cov(k, v - 1) \leq t - cov(k, v)$.*

We can exclude the v^{th} symbol from V , or substitute it by any other symbol not in V , the array remains a covering — for any k parameters, all the combinations are covered (plus some extra combinations, including the symbol that is already not in alphabet). Thus the optimal covering $t - cov(k, v - 1)$ has at most $N = t - cov(k, v)$ rows. The above two results give us an opportunity to reduce the size of an already constructed optimal covering $t - cov(k, v)$, to a covering $t - cov(k - i, v - j)$, $i, j \geq 0$. However, this does not guarantee that the covering obtained will be the optimal covering for that case. That is to say that when we reduce, we only can use the initial array size as an upper bound for the size of the optimal covering of the reduced pair $(k - i, v - j)$. These considerations clear the case that the symbol set increases. We can achieve $2 - cov(k, v) = v^2$ for all pairs (k, v) .

Now we shall explore the way of changing the number of parameters. Suppose we have $2 - cov(k, v) = d$, we want to obtain $2 - cov(2k, v)$, i.e., to double the number of the parameters. Let A be the array representing the $2 - cov(k, v)$. If we duplicate A in the horizontal direction, we have:

$$A \quad A$$

But this array will not generally be a covering array for the new set of $2k$ parameters. To see this, compare the i^{th} column with the j^{th} column in the new array. If $i \not\equiv j \pmod{k}$, then these two parameters are already covered, since the corresponding columns $(i \pmod{k})$ and $(j \pmod{k})$ are present within A , thus $2 - covers$ of this two column is achieved. However, for the pairs of columns, for which $i = (j \pmod{k})$, only the pairs:

$$\begin{array}{cc} 0 & 0 \\ 1 & 1 \\ 2 & 2 \\ \vdots & \vdots \\ (v-1) & (v-1) \end{array}$$

are covered. This is v pairs out of the total of v^2 pairs. So, we just need to add the remaining $v^2 - v$ pairs. These are:

$$\begin{array}{cc} b & c \\ 0 & 1 \\ 0 & 2 \\ \vdots & \vdots \\ 0 & (v-1) \\ 1 & 0 \\ 1 & 2 \\ \vdots & \vdots \\ 1 & (v-1) \\ \vdots & \vdots \\ (v-1) & (v-2) \end{array}$$

If b and c are the columns as denoted above, the array B is the array obtained by k duplications of the column b in the horizontal direction. Analogously, the C is the array obtained by placing k columns c together in the horizontal direction. Then, by our construction, the array:

$$\begin{array}{cc} A & A \\ B & C \end{array}$$

covers all the possible pairs between any two columns (i.e., any two parameters). The array has $N = d + v^2 - v$ rows. Thus $2 - cov(2k, v) \leq d + v^2 - v$. Given that, we will derive the following more general property.

Property 3 *If $2 - cov(k, v) = d_1$ and $2 - cov(n, v) = d_2$, then we have $2 - cov(nk, v) \leq d_1 + d_2$. In the special*

case that one of these two coverings has the special property that it contains the rows

$$\begin{array}{cccccc} 0 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (v-1) & (v-1) & (v-1) & \cdots & (v-1) \end{array}$$

the upper bound can be reduced to $2 - \text{cov}(nk, v) \leq d_1 + d_2 - v$.

Proof The property follows from the construction above. We just need to make n replicas of the array A representing $2 - \text{cov}(k, v)$, instead of only 2. And in some cases we will be able to reduce the number of rows in the $2 - \text{cov}(nk, v)$ by v as it is noted above. ■

This proof completes the considerations on 2-coverings. We have shown how to increase the number of the parameters, and the number of symbols of the alphabet, i.e. the parameter values, and what effect this will have on the upper bound. It remains to show that the growth of $2 - \text{cov}(k, v)$ is logarithmic in v .

Lemma 3 $2 - \text{cov}(k, v) \leq (v \exp 2) \cdot \log k$.

Proof To show this, consider $2 - \text{cov}(k, v) = d$. Apply the procedure described in Property 3, i.e. the multiplication of two 2-coverings, using $2 - \text{cov}(k, v) = d$, we have $2 - \text{cov}(k^2, v) \leq 2d$. If we continue with this procedure, we have $2 - \text{cov}(k^m, v) \leq md$. Thus, if we denote $n = k^m$, $2 - \text{cov}(n, v) \leq \log 10d$. And it is clear that d can be considered as a constant. Basically we can pick $d = v^2$ for some v prime power and then reduce as appropriate. Stronger result is presented in [1].

Theorem 2

$$\lim_{k \rightarrow \infty} \frac{2 - \text{cov}(k, v)}{\log k} = \frac{v}{2}, \quad (3.4)$$

$$\lim_{k \rightarrow \infty} \frac{3 - \text{cov}(k, v)}{\log k} = \left(\frac{v}{2} \right). \quad (3.5)$$

We will return to this result later, next we discuss the natural extension of this schemes to 3-coverings. The Theorem 2 is given in [5].

3.5.4 3-Covers: Recursive Construction

One of the natural extension of the 2-covers are the 3-covers. For the 3-covers the same algebraic construction can be applied. In the case of q prime power, we will have $3 - \text{cov}(q+1, q) = q^3$. If q is even, we have $3 - \text{cov}(q+2, q) = q^3$. This result can also be generalized to the following:

Theorem 3 For any prime power $q \geq t - 1$, we have $t - \text{cov}(q+1, q) = q^t$.

This result is proven in [1], and an alternative construction using *orthogonal arrays* is given in [4]. We try to extend the constructions we used in previous section for 2-coverings to 3-coverings. It is more convenient, however, in constructing the 3-covering of larger set of parameters (say, multiples of k) to use not only the $3 - \text{cov}(k, v)$, but also $2 - \text{cov}(k, v)$. As we may expect $2 - \text{cov}(k, v) \ll 3 - \text{cov}(k, v)$, and as it is seen from Theorem 11, this difference is one order of magnitude as k approaches ∞ . Here we will present a recursive construction for the case $v = 3$, i.e. we have ternary variables. The construction triples the number of parameters while keeping the size of the symbol set constant.

Lemma 4 If $3 - \text{cov}(k, v) = a$, $2 - \text{cov}(k, v) = b$, then $3 - \text{cov}(3k, v) \leq a + 2b + 18$.

Proof: Let A denote the $a \times k$ array that represents the $3 - cov(k, 3)$, similarly B is the $b \times k$ array representing $2 - cov(k, 3)$. The arrays denoted by $B + 1$ and $B + 2$ represent the array B with 1 and 2 added to each of its element respectively, where the addition is modulo 3. Then the array

$$\begin{array}{ccc} A & A & A \\ B + 1 & B & B + 2 \\ B + 2 & B & B + 1 \\ C1 & C2 & C3 \end{array}$$

is a 3-covering of the $3k$ ternary variables. Assume we have three columns, not all of which are in the same part of the array. If we take their indices, for example, i, j and k , modulo 3, we have the following 3 possibilities:

1. All the indices are different modulo 3. Then these three parameters (columns) are covered according to the property of A.
2. Exactly two of the indices are equal modulo 3. Then all the possible pairs between the two columns with equal indices and the third column are covered in the part A, moreover, the construction of A assures that for each possible combination between the entries in the two "equal" columns there is an entry for each symbol (0, 1, 2) in the alphabet. Thus it is enough to 2-cover all possible combinations between the columns with equal indices modulo 3. This is achieved by the addition of the 2 blocks of the array B and its shifts $B + 1$ and $B + 2$.
3. All three indices are equal modulo 3. We will examine this case in detail in the following. more detail.

The A blocks cover the following triples:

$$\begin{array}{ccc} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 2 & 2 \end{array}$$

Further, the $B + 1, B,$ and $B + 2$ blocks cover the triples:

$$\begin{array}{ccc} 1 & 0 & 2 \\ 2 & 1 & 0 \\ 0 & 2 & 1 \end{array}$$

Similarly, the $B + 2, B, B + 1$ blocks cover the triples:

$$\begin{array}{ccc} 2 & 0 & 1 \\ 0 & 1 & 2 \\ 1 & 2 & 0 \end{array}$$

Thus, we have covered 9 of all $3^3 = 27$ triples between these parameters. We just add the other 18 triples as follows:

$c1$	$c2$	$c3$
0	0	1
0	0	2
0	1	0
0	0	2
0	1	0
0	1	1
0	2	0
0	2	2
1	0	0
1	0	1
1	1	0
1	1	2
1	2	1
1	2	2
2	0	0
2	0	2
2	1	1
2	1	2
2	2	0
2	2	1

and the blocks $c1$, $c2$ and $c3$ consist of k copies of the columns $c1$, $c2$ and $c3$, respectively. Thus, we have covered all possible triples between any 3 of the $3k$ parameters, and the claim is verified. It will be desirable to have a recursive construction that relates $3 - cov(nk, 3)$ to $3 - cov(k, 3)$ and eventually $3 - cov(n, 3)$, thus eliminating the need of a lower order coverings (in the above case — the $2 - cov(k, 3)$). The question about the existence of such a construction remains open.

3.5.5 Generalizations and Conclusions

Based on the considerations given in previous sections, the asymptotic behaviour of the t -coverings might be as follows:

$$\lim_{k \rightarrow \infty} \frac{t - cov(k, v)}{\log k} = \mathbf{O}(v^{t-1}) \quad (3.6)$$

where $\mathbf{O}(v)$ denotes the number of order of variable v . Another interesting thing to look at is the non-optimality of the t -coverings of (k, v) for fixed (k, v) . In other words, how much the optimal $t - cov(k, v)$ differs from the theoretical lower bound of v^t ?

Definition 4 *The non-optimality function f is defined as follows. For each pair of integers (k, v) , and each $1 \leq i \leq k$, $f_i(k, v) = i - cov(k, v) - (v^i)$.*

By the definition of the function, it is evident that $f_1(k, v) = 0$ and $f_k(k, v) = 0$. The question is what the general behaviour of the function f is? We have shown in Lemma 1 some of the cases when f will have internal zeros. It remains to study whether these are all the case when f will have internal zeros. If we are able to efficiently compute this function, there is one practical application of this property. In testing we will usually know the pair (k, v) , i.e. the parameters of the tested system. We will also know the capabilities and requirements of the testing equipment - i.e. the computing power and the time constraints. Given all these and combining them with the upper bounds for $t - cov(k, v)$, we may determine what maximum level of coverage we can test within our limits. In other words, we can adjust our testing to our capabilities and some additional requirements. The non-optimality function, as defined is a discrete function. It is interesting to see what the asymptotic behaviour of f is when $k \rightarrow \infty$, assuming that $k \rightarrow \infty$ without assuming any value that will allow f to have internal zeros. To our knowledge these questions have not been answered yet.

Acknowledgements

This part of the report reflects some of the models and constructions that were discussed during the days of the 5th Graduate Mathematic Modelling Camp, held at Simon Fraser University, Burnaby, British Columbia, Canada. The authors — Lorraine Dame, Charles Fortin, Jingxiang Luo, Huamei Luo, Tzvetalin Vassilev and Qingze Zou — would like to acknowledge the brilliant leadership provided by the mentor of the group — Prof. Brett Stevens. We appreciate the friendly and stimulating team-work atmosphere that made the experience wonderful and memorable. Thanks also go to the organizers of the event.

Bibliography

- [1] M. Chateauneuf and D. L. Kreher. On the state of strength-three covering arrays. *Journal of Combinatorial Designs*, 2000. Submitted.
- [2] D. M. Cohen, S. R. Dalal, M. L. Fredman, and G. C. Patton. The aetg system: an approach to testing based on combinatorial design. *IEEE Transactions on Software Engineering*, 23(7):437–444, July 1997.
- [3] D. M. Cohen, S. R. Dalal, J. Parelius, and G. C. Patton. The combinatorial design approach to automatic test generation. *IEEE Transactions on Software Engineering*, 13:83–88, 1996.
- [4] A. S. Hedayat, N. J. A. Sloane, and Stufken J. *Orthogonal Arrays*. Springer, 1999.
- [5] N. J. A. Sloane. Covering arrays and intersecting codes. *Journal of Combinatorial Designs*, 8(3):189–200, 2000.
- [6] Alan W. Williams and Probert Robert L. Formulation of the interaction test coverage problems as an integer program. *http: www.sit.uottawa.ca/~wiliam*, 2002.

Chapter 4

Routing in Ad Hoc Networks/Dynamic Networks

Participants: Petra Berenbrick (Mentor), Isabelle Déchène, Tereza Neocleous, Anita Parkinson, Bo Peng, Adam Webber.

PROBLEM STATEMENT: Dynamic networks provide new challenges for packet routing. In a dynamic network there is the possibility that any node or edge can move or disappear at any time. As a result paths have short lifetimes and establishment of fixed routes is not possible. Examples of dynamic networks include faulty networks where links often fail, mobile ad hoc networks where nodes move and the internet.

Previous work has been done proving the stability of single destination networks¹. These proofs, however, did not utilize actual structures within the networks to reach their conclusions. In this paper we develop proofs based on the structures of single destination networks with the hopes that they can be extended to multiple destination networks in the future. We will also include a discussion of insights gained into networks of both single and multiple destinations.

¹Baruch Awerbuch, Petra Berenbrink, André Brinkmann, Christian Scheideler, **Simple Routing Strategies for Adversarial Systems**, Proceedings of the 42th Symposium on Foundations of Computer Science (FOCS), 2001, pp 158–167.

4.1 The Model

In the model of the network, the following assumptions are used:

Time:	time is discrete and divided into synchronous steps
Messages:	Messages are represented by fixed-size packets with a fixed source and destination. A single destination network contains one type of packet. A multiple destination model contains multiple types of packets each with their own destination.
Communication Link:	Messages or packets are transported down links or edges that can transport one packet of any type in one time step. The edges are considered undirected.
Nodes:	Each node can store an unlimited number of all packet types. The node knows how many of each type of packet it is storing.

When discussing networks several parameters should be defined. One is the way the packets are injected. There are two main methods - packet and flow injection. In packet injection, unit packets are injected into the system. This represents discrete injection as the packet can not be broken down into smaller units and must flow down only one edge. Flow injection represents a continuous injection and can be split up and distributed between multiple edges. In this work we have focused on the packet injection model.

Once an injection model has been decided upon, the pattern of injection can be specified as either static injection or adversarial injection. In these network problems the concept of an adversary represents the worst case scenario. We assume that we are working against an adversary who has control of the network. Since he controls the configuration of the network, he determines a route for each packet through the system before they are injected. He is also aware of our dependence on the balancing algorithm and can remove edges that he did not plan to use, but that we would use.

Finally, the network can be considered static or adversarial. In a static network, all edges present in the first time step are present for all future time steps. This is in contrast to an adversarial network in which the adversary can choose to remove or add edges.

4.2 Balancing Algorithm

Packets are transferred from one node to another according to the balancing algorithm. For the single destination network, a packet is transferred from node A to node B if the number of packets stored in A is at least T more than the number of packets stored in B. T is at least as large as the maximum degree, d , of all the nodes to ensure that if packets are transferred out of node A then node A will not be left with such a low number of packets that some packets will flow back in the next step. The degree of a node is the number of edges connected to that node.

The algorithm for the multiple destination case is essentially the same as for one destination. A packet of a given type will move along an edge if the difference in packets of that type stored in the nodes at the ends of the edge is at least T and is greater than the difference in number of packets of the other types stored in the nodes at the ends of the edge. Each edge can only transport one packet in each time step. If the maximal difference is the same for many types of packets, then the one to be moved is randomly chosen.

The balancing algorithm creates a distribution of packets that results in a potential slope towards the destination. Once the slope has been built, new packets entering the network flow down the gradient and exit at the destination. The gradient indicates the path to the destination.

One method of implementing this algorithm would begin with feeding dummy packets into the system to create the potential profile. Once enough dummy packets have been injected, real packets can be injected and they will all flow to their destination. The problem is to determine if this number of dummy packets that must be injected can be bounded, or if eventually real packets will become part of the potential and will never reach the destination.

4.3 Previous Work

For the case of one type of packets with one destination, it has been proven that there is an upper bound on the number of packets that get stuck in the network. That is, the network is stable using the balancing algorithm.

Progression of Algorithm
One packet injected per time step

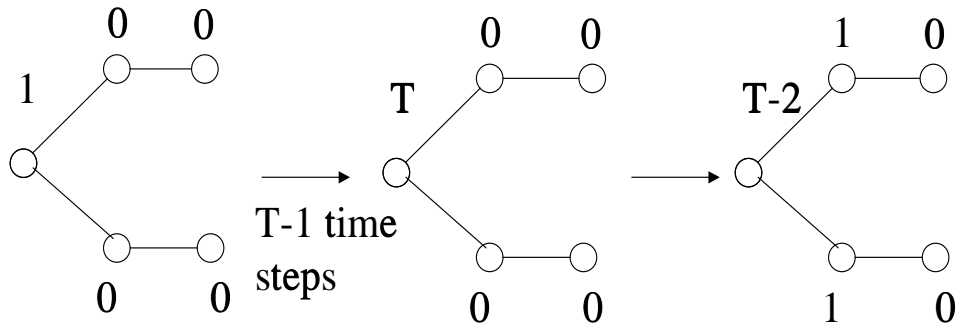
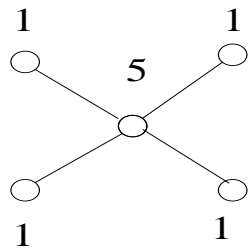
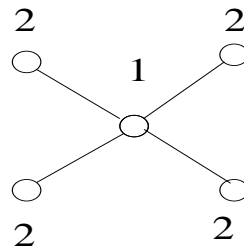


Figure 4.1:

Motivation for Balancing Threshold T
 $T = 5$



$5 \quad 1 < T$
No balancing



If balancing did occur
node would be left with fewer
packets than neighbour.

Figure 4.2:

In this project, we attempt to gain insight into the open question “Is the network stable under this algorithm with two types of packets each with a different destination?”

We will focus on the static network with packet injection model with static injection.

4.4 Our Approach

Previous work has proven that single destination networks are stable under the balancing networks. The proofs for the one destination case do not utilize the specific structure of each network, but are based on the number of nodes in the network. These proofs can not be extended to the multiple destination case as they use assumptions that all packets are the same and are exchangeable. This is obviously not possible with more than one packet type.

Our strategy to solving the problem was to approach it from two directions. The first direction was to find a formal proof of stability. First, a proof using the specific structure of the network for the single destination case would be developed. This would then be extended to multiple destination cases. In all cases we assumed static network and static injection of one packet of each type. It is believed that two destination cases and more than two destination cases may not act the same; two destination cases were therefore focused upon.

The second method was to search for a counterexample that showed an unstable network. In order to test networks and gain some insight as to how the algorithm functions a simulator was developed. This program was written in S-Plus and can handle single and multiple destination networks. Many ideas were tested on this simulator and it proved to be a great step forward in our progress.

4.5 Results

After spending three days learning about this problem, it is believed that a static network with static injection of two types of packets each with their own destination is stable under the algorithm.

4.5.1 Step One—Simple Network

We began with a simple network consisting of a chain of nodes connecting the source and the destination. Each node is in the path between the source and destination. In our approach, in order to prove that the network is stable under the balancing algorithm it needed to be proven there was an upper bound on the number of packets in each node. The following proof shows that there will be a profile down the chain in increments of T . The destination will have 0 packets, the next node will contain T packets and counting away from the destination, node k will contain kT packets.

Lemma 1 *The network is stable for the line case.*

Proof. The nodes are marked v_0, v_1, \dots, v_n from the destination to the source. The potential of node i at time t is marked by S_i^t . We claim that at time step t , the maximum potential for v_i is iT , which is independent of time. In this case, the total potential in the network is less than or equal to $Tn(n+1)/2$ and the network is therefore stable.

At time step 1, it is obvious that $S_i^1 \leq iT$. Induction step: Suppose that for time $t = m$, $S_i^m \leq iT$. We have to show this holds for time $m+1$. For $i = 0$, $S_i^{m+1} = 0$ by definition. Now if $i > 0$, two cases need to be considered. If $S_i^m < iT$, then $S_i^{m+1} \leq iT$ since the network is a straight line. If $S_i^m = iT$, since $S_{i-1}^m \leq (i-1)T$ by the induction hypothesis, then $S_i^m - S_{i-1}^m \geq T$ and S_i^m and S_{i-1}^m will be balanced according to the algorithm. Therefore $S_i^{m+1} = S_i^m - 1 + \text{possible injection from } v_{i+1} \leq iT$. ■

Since we know what the profile will be, an upper bound on the total number of packets in the system can be calculated and it can be concluded that this network under the balancing algorithm is stable.

Simple Network

$$T = 3$$

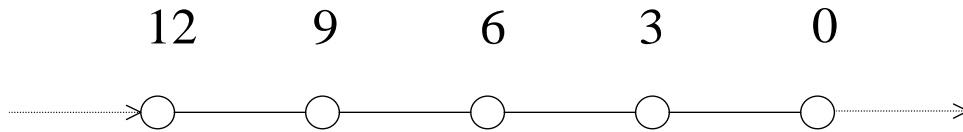


Figure 4.3:

4.5.2 Step Two—Networks with Branches

A slightly more complicated network consists of one path with side branches. The proof for the simple network can be extended to this case. With the ability to predict the potential profile, an upper bound on the number of packets in the system can be determined and hence the network under this algorithm is stable.

Branched Network

$$T = 4$$

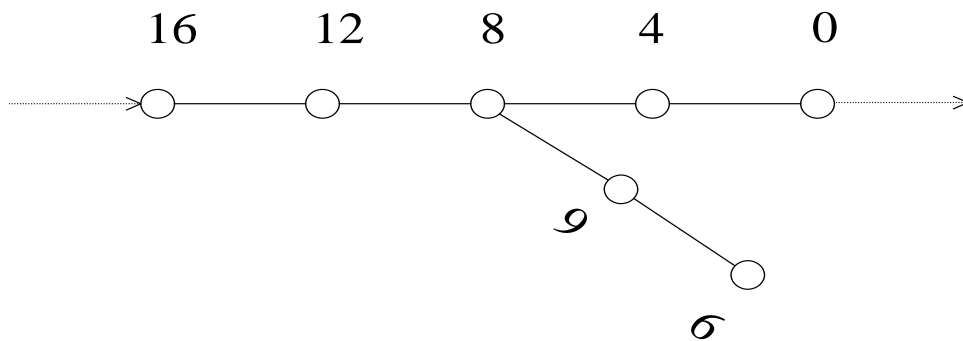


Figure 4.4:

Lemma 2 *If $T > \frac{d}{2}$ and the graph of the network is a tree, then it is stable.*

Proof. In this case, there is a unique path from source to destination since the graph has no circuit which we will later refer to as *main path*. The injection rate is 1 per time step and the network capacity is 1.

The nodes on the main path are marked v_0, v_1, \dots, v_n from the destination to the source. The nodes that are directly connected to the v_i (called *adjacent node of v_i*) are marked as $v_{ij}, j \geq 0$. Nodes that are not on the main path but are connected to v_i will be called *side nodes of v_i* and are marked as v_{i*} . We claim that $S_i^t \leq iT$, $S_{i*}^t \leq (i - 1)T + 1$ holds for any t . Therefore, the total potential in the network is less than N^2T where N is the number of nodes for the tree. The network is therefore stable.

The result is trivial for time step 1. Suppose that for time $t = m$, $S_i^m \leq iT$, $S_{i*}^m \leq (i - 1)T + 1$ for $0 \leq i \leq n$. The following facts can be observed:

1. For $i = 0$, we have $S_0^{m+1} = S_0^m = 0$ since v_0 is the destination. The following cases assume $i > 0$.
2. If $S_i^m = iT$, since $S_{i*}^m \leq S_i^m$ and $S_{i-1}^m \leq (i-1)T$, v_i will be balanced with v_{i-1} or some of its adjacent nodes. Since the injection for v_i is at most 1, $S_i^{m+1} = S_i^m - 1 + \text{possible injection from } v_{i+1} \leq iT$.
3. If $S_i^m < iT$ and $S_i^m < iT - d + 1$, then even if all adjacent nodes inject to v_i , we still have $S_i^{m+1} \leq S_i^m + 1 < iT - d + 1 + d \leq iT$.
4. If $S_i^m < iT$ and $S_i^m \geq iT - d + 1$, since $S_{i*}^m \leq (i-1)T + 1$, $S_{i*}^m - S_i^m \leq (i-1)T + 1 - iT + d - 1 = -T + d \leq T$ since $T > \frac{d}{2}$ by hypothesis. That is to say, v_{i*} will not be able to inject a packet to v_i . Since the only possible injection to v_i is from v_{i+1} , $S_i^{m+1} \leq S_i^m + 1 \leq iT$.

Therefore, $S_i^{m+1} \leq iT$ at any time step in all cases. For side nodes, since $S_{i*}^m \leq (i-1)T + 1$ by the induction hypothesis and all its adjacent nodes have potential less than or equal to iT , $S_{i*}^{m+1} \leq (i-1)T + 1$. ■

In the two cases above, only one packet would enter the path to the destination at a time and hence the predicted profile will always hold. Next we look at a case when more than one packet can enter the path to the destination at a time.

4.5.3 Step Three—Networks with Loops

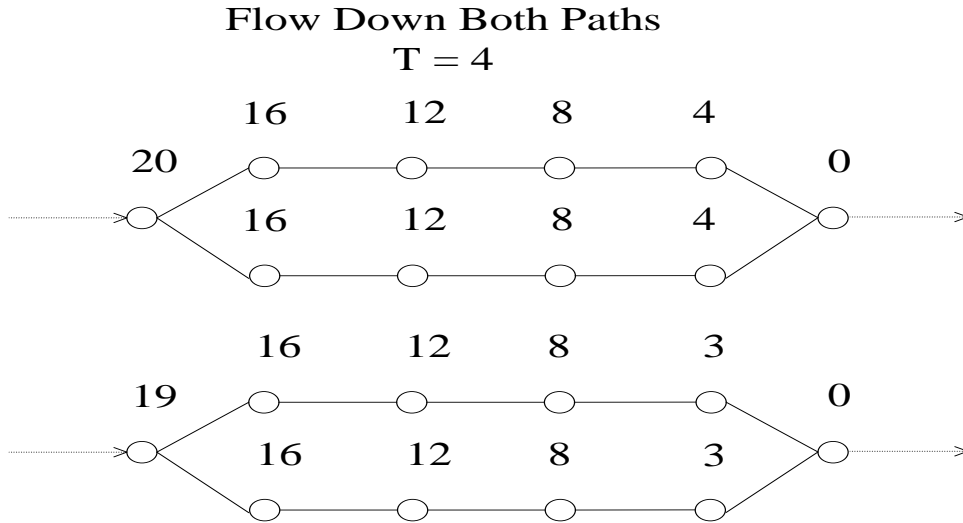


Figure 4.5:

In this type of network, with one type of packet we have found two types of behaviour can occur. Either all the packets use only one of the two paths, or both paths are used. In the second case, the network oscillates between a pair of states. In one time period two packets enter the loop, one down each edge of the loop. In the subsequent step, no packets enter the loop.

In investigating this situation, a formula has been determined that will predict if both paths will be used or if one path will be used.

Lemma 3 *If $nT < m(T-1)$, the long path is never used to deliver a packet to the destination.*

Proof. The nodes on the long path are labelled $v_1 \dots v_m$ with no label on the destination or the source node. Nodes on the short path are labelled u_1, \dots, u_n . m is the number of nodes in the long path and n is the number of nodes in the short path.

Flow Down One Path

$$T = 4$$

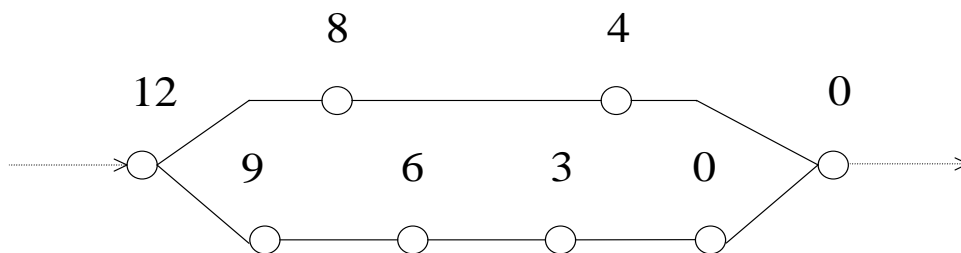


Figure 4.6:

Assume a packet is delivered down the long path.

$\exists t'$ s.t. $S_{v_1}^{t'} = T$. Since we are bounded by the algorithm, $\exists t'' < t'$ s.t. $S_{v_2}^{t''} = T - 1$ and $S_{v_2}^{t''} = 2T - 1$. S_{v_2} must reach $2T - 1$ to be able to pass to v_1 when $S_{v_1} = T - 1$.

By induction, the previous step can be applied to successive nodes. Therefore, there exists $t^m < t''$ such that $S_{v_m}^{t^m} = m(T - 1) + 1$. Therefore, there exists $t^w < t^m$ s.t. $S_{v_m}^{t^w} = m(T - 1)$.

$m(T - 1)$ is the number of packages v_m must have before it receives a package to deliver to the destination via the long path. By lemma 1, we know that S_{u_m} can reach a potential no higher than nT . By our algorithm, if a packet is not sent along the long path, then $S_{v_m} > S_{v_n}$. If S_{v_m} is greater than the maximum of S_{v_n} , it will always be greater than S_{v_n} and therefore the long path will never be used to deliver a packet. We can therefore conclude that if $nT < m(T - 1)$, the long path will never be used to deliver a packet. ■

This formula holds if the destination is immediately off the loop or if there are a number of nodes between the loop and the destination, although Lemma 3 holds only when the destination is at the end of the loop.

4.6 Extension to Two Destinations:

The algorithm for the multiple destination case is essentially the same as for one destination. A packet of a given type will move along an edge if the difference in packets of that type stored in the nodes at the ends of the edge is at least T AND is greater than the difference in number of packets of the other types stored in the nodes at the ends of the edge. Each edge can only transport one packet in each time step.

The simple case with two types of packets each with their own destination is two branches, each ending with a destination for one type of packet.

In this case we predict the number of each type of packets at each node. Along the path to the destination of a given type of packet, there will be a profile in increments of T . Along the other path, there will be a profile of increment $T-1$. When the two paths meet, the first node along the blocked path must have one more packet than the first node along the path to the destination. This will prevent flow down the blocked path.

With an understanding of this type of network, we move to the network with loops.

4.7 Search for a Counter Example

In looking at networks with two destinations, each with an independent path, and two possible paths for each packet type, it was learnt that each packet will use the shortest path to the destination. In the case when there is

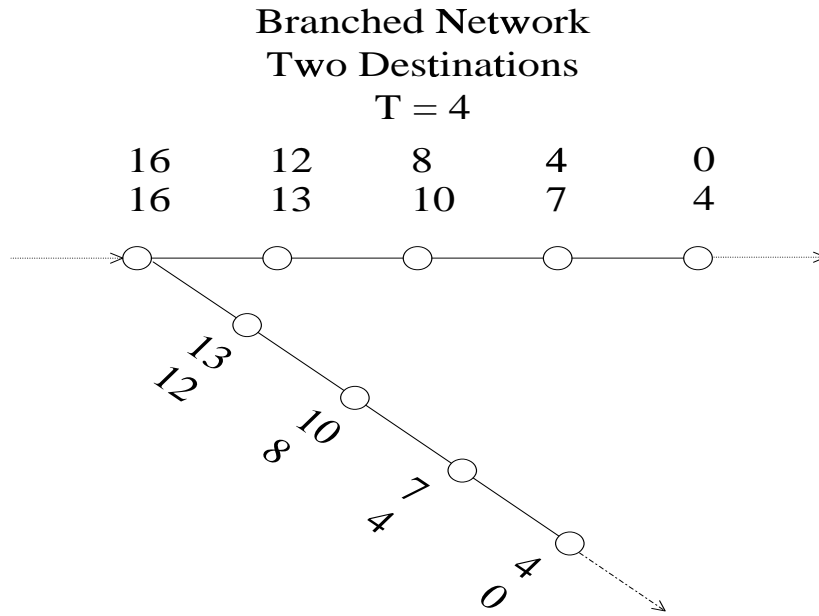


Figure 4.7:

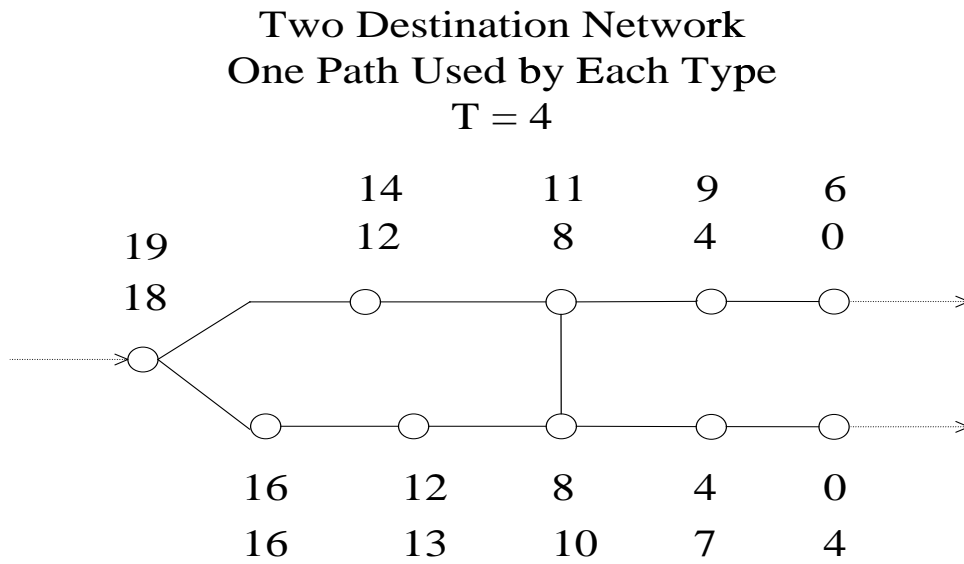


Figure 4.8:

one short path for one packet type, and two equivalent paths for another packet type, each packet type will take one path. This led to the question "What happens when both packets share the shortest path?". The speculation was that one packet type would settle on the shortest path and the other on the longest path. The choice of packet on the shortest path was arbitrary and might be decided when the first tie was settled.

The case was entered into the simulator and the following results obtained.

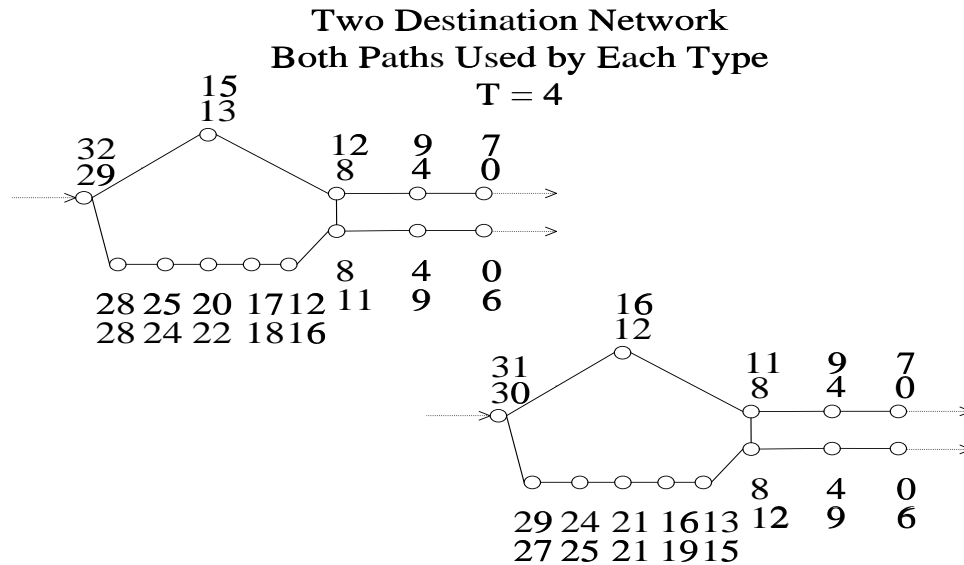


Figure 4.9:

In this case, both packets use both paths in an alternating fashion. Further thought resulted in the realization that if the adversary were to close the link between the two destinations, packets would continue to flow down both lines. This would happen until a profile was established that would result in each packet type taking the path open to its destination. By closing the link, additional packets are trapped in the network. Would repeated openings and closings result in repeated trappings and produce a counter example?

Unfortunately the answer, in this case is no. Eventually there are enough packets of each type stored in each node that if the edge is closed and a packet flows down the "wrong" path it will not result in a missed opportunity to send a packet to the destination.

This example lends support to the opinion that two destination networks are also stable.

4.8 Conclusions

From experimentations by hand and with the simulator, as well as the proofs that have been constructed, we believe that the two-destination static injection network is stable. Two-destination dynamic networks, however, under the control of an adversary, could behave in a different way. Networks with more than two destinations have not been examined, but there is potential that they could behave more problematically.

Future work should include the development of proofs for those conjectures we have made. Proofs that have been done should also be extended, if possible, to two (or more) destination networks. In order to accomplish this, a more general simulator, able to stop automatically upon stability and utilize symbolic representations of potentials, may need to be constructed so one can gain a general knowledge of what will happen in more complex networks.

Finally, it may interesting to observe the behaviour of the network under a modified balancing algorithm.

Chapter 5

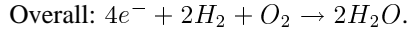
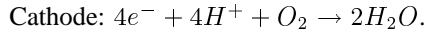
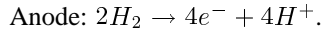
Modelling Polymer Electrolyte Membrane Fuel Cells

Participants: Brian Wetton (Mentor), Elaine Beltaos, Mikalai Birukou, Olivier Dubois, Katherine Hegewisch, Heather Lehr, Anuj Mubayi, Tingting Shu, Fabien Youbissi.

PROBLEM STATEMENT: In recent years, with increasing environmental concerns over vehicle generated pollution and limited range associated with battery powered electric vehicles, the proton exchange membrane (PEM) fuel cell system is gaining more attention as an alternative power generation source for electric vehicles. Attractive characteristics of the PEM fuel cell system include the simplicity of its design and operation. The PEM we will study here functioned with humidified hydrogen in the anode and humidified air ($O_2 + H_2O + N_2$) in the cathode. The mathematical formulation reduces to a system of nonlinear ODEs, subject to appropriate boundary conditions. Results are presented for various operating conditions and design parameters in order to identify the important factors in the performance of the fuel cell.

5.1 Physical Problem

The PEM fuel cell uses a simple chemical process to combine hydrogen and oxygen into water, producing electric current. At the anode, hydrogen molecules give up electrons and form hydrogen ions, a process which is made possible by the Platinum catalyst. These electrons travel to the cathode through an external circuit, producing electric current. The PEM allows protons to flow through, but stops electrons from passing through it. As a result, while the electrons flow through an external circuit, the hydrogen ions flow directly through the proton exchange membrane to the cathode, where they combine with oxygen molecules and electrons to form water. In this way, the hydrogen's natural tendency to oxidize and form water is used to produce electricity. The reactions we get are as follows:



In the anode, we have humidified hydrogen gas travelling in a channel. This gas enters the pores of the teflon layer and interacts with the catalyst layer, which speeds up the reaction $H_2 \rightarrow H^+ + 2e^-$. The electron transfers to the conductive teflon layer and conducts to the graphite layer. Since the graphite layer on the anode and on the cathode are connected in a circuit, there is a total current. The hydrogen proton carries a water molecule across the membrane to the cathode layer.

In the cathode, we have humidified air (composed of oxygen, nitrogen and water vapour) travelling in a channel. This gas also enters the pores of the teflon layer and interacts with the catalyst layer, which speeds up the reaction $4e^- + 4H^+ + O_2 \rightarrow 2H_2O$. The electrons are transferred to the medium from the graphite layer to the porous teflon layer.

5.2 Mathematical Problem

To derive a system of ODEs that describes our physical problem, we discretize the interval $x \in [0, 1]$ to $x = \Delta x * n$, where $n = 0, \dots, N$. Consider the change in flux, $Q_{n+1} - Q_n$ between the points $x = n\Delta x$ and $x = (n + 1)\Delta x$. We see that

$$Q_{n+1} = Q_n + \Delta x f(x)$$

where $f(x)$ describes the sources/sinks of the quantity. Letting $\Delta x \rightarrow 0$, we see that the equation becomes

$$\frac{d}{dx}Q = f(x)$$

Thus, for each of the gas fluxes, we need only consider the sources/sinks of the quantity.

5.2.1 Coupled System of ODEs

The rate of change of the flow of water in the anode depends only on the water transfer across the boundary between anode and cathode.

$$\frac{d}{dx}Q_{H_2O}^a = -\frac{i(x)}{F} - .252 (r_a(x) - r_c(x))$$

The rate of change of the flow of water in the cathode depends only on the water transfer across the membrane between the anode and cathode and the amount of water generated by the chemical reaction in the cathode.

$$\frac{d}{dx}Q_{H_2O}^c = \left(\frac{i(x)}{F} + .252 (r_a(x) - r_c(x)) \right) + \frac{i(x)}{2F}$$

Since the reaction $H_2 \rightarrow 2H^+ + 2e^-$ takes place in the anode, the amount of hydrogen needed is twice the amount of current generated, so

$$\frac{d}{dx}Q_{H_2} = -\frac{i(x)}{2F}$$

Since the reaction $4e^- + 4H^+ + O_2 \rightarrow 2H_2O$ takes place in the cathode, we see that the amount of oxygen is four times the amount of current, so

$$\frac{d}{dx}Q_{O_2} = -\frac{i(x)}{4F}$$

Since the flow of nitrogen in the channel does not change as a function of x , we have that the rate of change of the flow is zero, i.e.

$$\frac{d}{dx}Q_{N_2} = 0$$

Thus our system of differential equations in the fluxes becomes:

$$\begin{aligned}\frac{d}{dx}Q_{H_2O}^a &= -\frac{i(x)}{F} - .252(r_a(x) - r_c(x)) \\ \frac{d}{dx}Q_{H_2} &= -\frac{i(x)}{2F} \\ \frac{d}{dx}Q_{H_2O}^c &= \frac{3i(x)}{2F} + .252(r_a(x) - r_c(x)) \\ \frac{d}{dx}Q_{O_2} &= -\frac{i(x)}{4F} \\ \frac{d}{dx}Q_{N_2} &= 0\end{aligned}$$

where $r_a(x)$ and $r_c(x)$ are the relative humidities in the anode and cathode respectively. We can find $i(x)$ by inverting the following equation for $i(x)$, which also depends on the properties of the membrane

$$U = E_0 - i(x)\frac{0.0001}{24} \left(\frac{1}{r_a(x)} + \frac{1}{r_c(x)} \right) - \frac{RT}{F} \sinh^{-1} \left(\frac{8.6 i(x)}{0.0044 C_{O_2}(x)} \right)$$

where E_0 is the voltage across the fuel cell when there is no current, R is the ideal gas constant and T is the temperature. We take temperature to be constant throughout this paper.

In order to get a system of ODEs, we would like to have expressions for $r_a(x)$, $r_c(x)$, and C_{O_2} in terms of the fluxes Q .

5.2.2 Concentrations in Terms of Fluxes

To calculate the concentrations of the gases in terms of the fluxes, we consider the flux of a gas travelling at velocity v through a cross sectional area A . We assume that the velocity of the gases are all roughly the same.

For the gases in the cathode

$$\begin{aligned}Q_{N_2} &= C_{N_2}Av \\ Q_{O_2} &= C_{O_2}Av \\ Q_{H_2O}^c &= C_{H_2O}^cAv\end{aligned}$$

Then $Av = \frac{Q_{N_2}}{C_{N_2}} = \frac{Q_{O_2}}{C_{O_2}} = \frac{Q_{H_2O}^c}{C_{H_2O}^c}$

From Dalton's law, we have that $C_i = \frac{P_i}{RT}$ and the law of partial pressures gives $\sum_i P_i = P_{tot}^c$. Then

$$\sum_i C_i = \frac{\sum_i P_i}{RT} = \frac{P_{tot}^c}{RT}$$

and we have

$$C_{H_2O}^c = \frac{P_{tot}^c Q_{H_2O}^c}{RT(Q_{H_2O}^c + Q_{N_2} + Q_{O_2})}$$

This procedure can be used for all gases in either the anode or the cathode yielding

$$\begin{aligned} C_{O_2} &= \frac{P_{tot}^c Q_{O_2}}{RT(Q_{H_2O}^c + Q_{N_2} + Q_{O_2})} \\ C_{N_2} &= \frac{P_{tot}^c Q_{N_2}}{RT(Q_{H_2O}^c + Q_{N_2} + Q_{O_2})} \\ C_{H_2O}^a &= \frac{P_{tot}^a Q_{H_2O}^a}{RT(Q_{H_2O}^a + Q_{H_2})} \\ C_{H_2} &= \frac{P_{tot}^a Q_{H_2}}{RT(Q_{H_2O}^a + Q_{H_2})} \end{aligned}$$

If the cathode is saturated, we must make modifications to these formulae:

$$\begin{aligned} C_{H_2O}^c &= C_{sat} \\ C_{O_2} &= \left(\frac{P_{tot}^c}{RT} - C_{sat} \right) \frac{Q_{O_2}}{Q_{N_2} + Q_{O_2}} \\ C_{N_2} &= \left(\frac{P_{tot}^c}{RT} - C_{sat} \right) \frac{Q_{N_2}}{Q_{N_2} + Q_{O_2}} \end{aligned}$$

Similarly, if the anode is saturated, we use the following relations:

$$\begin{aligned} C_{H_2O}^a &= C_{sat} \\ C_{H_2} &= \frac{P_{tot}^a}{RT} - C_{sat} \end{aligned}$$

5.2.3 Relative Humidities in Terms of Fluxes

We can obtain expressions for the humidities, $r_a(x), r_c(x)$, of the anode and cathode respectively in terms of the fluxes, by using the above formulae:

$$\begin{aligned} r_a(x) &= \frac{P_{H_2O}^a}{P_{sat}} = \frac{C_{H_2O}^a RT}{P_{sat}} \\ r_c(x) &= \frac{P_{H_2O}^c}{P_{sat}} = \frac{C_{H_2O}^c RT}{P_{sat}} \\ \Rightarrow r_a(x) &= \frac{P_{tot}^a Q_{H_2O}^a}{P_{sat}(Q_{H_2O}^a + Q_{H_2})} \\ \Rightarrow r_c(x) &= \frac{P_{tot}^c Q_{H_2O}^c}{P_{sat}(Q_{H_2O}^c + Q_{N_2} + Q_{O_2})} \end{aligned}$$

We notice that we need to set $r_a(x) = 1$ if $P_{H_2O}^a > P_{sat}$ and $r_c(x) = 1$ if $P_{H_2O}^c > P_{sat}$

5.2.4 Initial Conditions

For our initial flux conditions, we choose some initial relative humidities, r_{a0} , r_{c0} , for the anode and cathode, respectively. Thus we take initial water fluxes of

$$Q_{H_2O}^a(0) = \frac{(Q_{O_2}(0) + Q_{N_2}(0))P_{sat}r_{a0}}{P^a - P_{sat}r_{a0}}$$

$$Q_{H_2O}^c(0) = \frac{Q_{H_2}(0)P_{sat}r_{c0}}{P^c - P_{sat}r_{c0}}$$

In choosing initial values of the oxygen and hydrogen, we consider that at the target current, I_T , the total current drawn from the cell is the integral over the current density, $i(x)$, through the membrane, so

$$I_T = \int_0^1 i(x)dx = -2F \int_0^1 \frac{d}{dx} Q_{H_2}(x)dx = -2F (Q_{H_2}(1) - Q_{H_2}(0))$$

$$I_T = \int_0^1 i(x)dx = -4F \int_0^1 \frac{d}{dx} Q_{O_2}(x)dx = -4F (Q_{O_2}(1) - Q_{O_2}(0))$$

Define the stoichiometric ratios S_{H_2} , S_{O_2} of hydrogen and oxygen such that

$$Q_{H_2}(0) = \frac{S_{H_2}I_T}{2F}$$

$$Q_{O_2}(0) = \frac{S_{O_2}I_T}{4F}$$

A lack of either hydrogen or oxygen will cause the current density to equal zero. To avoid this possibility, we enforce the final fluxes of hydrogen and oxygen be positive at $x = 1$. This leads to

$$Q_{H_2}(1) = Q_{H_2}(0) - \frac{I_T}{2F} = (S_{H_2} - 1) \frac{I_T}{2F} \geq 0$$

$$Q_{O_2}(1) = Q_{O_2}(0) - \frac{I_T}{4F} = (S_{O_2} - 1) \frac{I_T}{4F} \geq 0$$

To ensure that the fuel cell does not run out of either hydrogen or oxygen, we choose the stoichiometric ratios

$$S_{H_2} > 1$$

$$S_{O_2} > 1$$

Since dry air contains 79% N_2 and 21% O_2 , we take the initial condition of flow of nitrogen to be

$$Q_{N_2} = \frac{79}{21} Q_{O_2}$$

Thus we choose our initial fluxes by choosing only initial relative humidities for the anode and the cathode and initial stoichiometric ratios for the hydrogen and the oxygen.

Stoichiometric Ratios S_{O_2} , S_{H_2} , Initial Humidities r_{a0} , r_{c0}

$$Q_{H_2O}^a = \frac{(Q_{O_2} + Q_{N_2})P_{sat}r_{a0}}{P^a - r_{a0}P_{sat}}$$

$$Q_{H_2} = \frac{1}{4F} S_{H_2} I_{tar g}$$

$$Q_{H_2O}^c = \frac{Q_{H_2} P_{sat} r_{c0}}{P^c - r_{c0} P_{sat}}$$

$$Q_{O_2} = \frac{1}{2F} S_{O_2} I_{tar g}$$

$$Q_{N_2} = \frac{79}{21} Q_{O_2}$$

5.2.5 Pressure Changes Along Channel

Pressure Gradient when only Water Vapour is Considered

Assume gases are incompressible and have low Reynold's numbers (so low viscosity). We have the Navier-Stokes Equation:

$$\rho \frac{Du}{Dt} = -\nabla p + \nabla^2 u + \rho g, \text{ where } \frac{Du}{Dt} = u_t + (u \cdot \nabla)u \text{ (flow derivative)}$$

Assume the vector $u = u(y, 0, 0)$ and $p = p(x)$. Then the Navier-Stokes Equation becomes $\frac{\partial P}{\partial x} = \mu \frac{\partial^2 u}{\partial y^2}$. Since $\frac{\partial^2 u}{\partial y^2}$ is not a function of x , but $P = P(x)$ is a function of x , we have that the above equation must equal a constant, i.e. $\frac{\partial P}{\partial x} = \mu \frac{\partial^2 u}{\partial y^2} = \lambda$.

Consider $\frac{\partial^2 u}{\partial y^2} = \frac{\lambda}{\mu}$ with the boundary conditions $u(\frac{h}{2}) = u(-\frac{h}{2}) = 0$, since the velocity near the surfaces drops to zero. Then $u(y) = \frac{\lambda}{\mu} \frac{y^2}{2} - \frac{\lambda h^2}{8\mu}$. To get the pressure gradient in terms of the gas fluxes, we find the total gas flux:

$$Q(x) = C(x) \int_{-\frac{h}{2}}^{\frac{h}{2}} u(y) dy = \frac{-C(x)\lambda}{12\mu} h^3$$

Therefore, for the anode:

$$\frac{\partial P^a}{\partial x} = \frac{-12\mu(Q_{H_2O}^a + Q_{H_2})RT}{P^a h^3}$$

and for the cathode:

$$\frac{\partial P^c}{\partial x} = \frac{-12\mu(Q_{H_2O}^c + Q_{N_2} + Q_{O_2})RT}{P^c h^3}$$

In this case, the system of equations contains 7 variables (gas fluxes and pressures in the anode and cathode).

Pressure Gradient when the Height of the Liquid is Taken into Account

Assuming the depth of the liquid water in the cathode is \bar{h} then we have the boundary conditions for flows of liquid and gas:

$$\begin{aligned} u_g(h) &= u_l(0) = 0 \\ u_g(\bar{h}) &= u_l(\bar{h}) \\ \mu_g \frac{\partial u_g}{\partial y} \Big|_{\bar{h}} &= \mu_l \frac{\partial u_l}{\partial y} \Big|_{\bar{h}} \end{aligned}$$

where u_g is the velocity of the gas flow, and u_l is the velocity of the liquid water in the channel. Using the same arguments as above, we obtain the velocities:

$$\begin{aligned} u_g(y) &= \frac{\lambda}{\mu_g} y^2 + c_1 y + c_2 \\ u_l(y) &= \frac{\lambda}{\mu_l} y^2 + d_1 y + d_2 \end{aligned}$$

where c_1, c_2, d_1, d_2 are the constants of integration in terms of h, \bar{h} , and λ .

Explicitly, these constants are:

$$\begin{aligned} c_1 &= \lambda \frac{\bar{h}^2 \mu_g - \bar{h}^2 \mu_l + h^2 \mu_l}{\mu_g (\bar{h} \mu_l - \bar{h} \mu_g - h \mu_l)} \\ c_2 &= \frac{-\lambda h^2}{2\mu_g} - \frac{\lambda h}{\mu_g} \frac{\bar{h}^2 \mu_g - \bar{h}^2 \mu_l + h^2 \mu_l}{\mu_g (\bar{h} \mu_l - \bar{h} \mu_g - h \mu_l)} \\ d_1 &= \lambda \frac{\bar{h}^2 \mu_g - \bar{h}^2 \mu_l + h^2 \mu_l}{\mu_l (\bar{h} \mu_l - \bar{h} \mu_g - h \mu_l)} \\ d_2 &= 0 \end{aligned}$$

Integrating u_l and u_g gives us the respective fluxes for gases and liquid. Taking the ratio $\frac{Q_g}{Q_l}$ eliminates λ , and we get a 4th-degree polynomial equation in \bar{h} :

$$\alpha(\bar{h}) - \frac{Q_g}{Q_l} = 0$$

We use the root of this equation, \bar{h}^* , to find λ , thus getting the pressure gradient. This applies to both the anode and cathode.

5.2.6 Counterflow Case

In the coflow case, we have the air and hydrogen flow in the same direction. Changes in the fluxes due to the exchange of hydrogen and water through the membrane propagate in both channels in the same direction.

In the case of the counterflow, we reverse the direction of either hydrogen flow or airflow. The exchange of hydrogen and water on the membrane in this case propagates in opposite directions. Due to the counterflow, we have to reverse the sign of the derivatives of the fluxes for one of the channels, either anode or cathode.

5.3 Simulation and Computational Results

5.3.1 Determining Appropriate Voltage to Yield Desired Current

Our goal was to write a code which would compute the necessary voltage drop to obtain a given target current. To achieve this we first had to solve the problem of obtaining the current distribution down the channel given a voltage drop. Essentially, we needed to invert the following function:

$$U = E_0 - (4.167e - 6)i(x) \left(\frac{1}{r_a(x)} + \frac{1}{r_c(x)} \right) - \frac{RT}{F} \sinh^{-1} \left(1954.54 \frac{i(x)}{C_{O_2}(x)} \right)$$

Notice that in this expression, r_c , r_a , and C_{O_2} appear, so that in order to invert it, we first must obtain the values of the fluxes. To obtain the values of the fluxes, we solved our system which depended on the current distribution. In Matlab, we wrote a code which used the o.d.e. solver “ode15s” and took as its input function “Fun.m”. Within this input function, which defines the system of ordinary differential equations, we invert the above expression for a given U at each value of x given by the solver. To invert, we use bisection to find a zero of the function:

$$g = U - E_0 + (4.167e - 6)i(x) \left(\frac{1}{r_a(x)} + \frac{1}{r_c(x)} \right) + \frac{RT}{F} \ln \left(2 * 1954.54 \frac{i(x)}{C_{O_2}} \right)$$

Note that we are approximating $\sinh^{-1}(x)$ by $\ln(2x)$, since the argument is large. In order to do bisection, we need an interval on which g is positive and negative. By analyzing the intersection of the functions $i - c$ and $-\ln(di)$ (c and d are constants), we saw that if:

$$\frac{c}{d} = (1 - U) \frac{\frac{3909.09}{C_{O_2}}}{\left(\frac{1}{r_a} + \frac{1}{r_c} \right) (4.166e - 6)} < 1$$

Then $i \leq \frac{C_{O_2}}{3909.09}$. Otherwise, i is bounded above by $\frac{1-U}{(4.166e-6)\left(\frac{1}{r_a} + \frac{1}{r_c}\right)}$. The current is always bounded below by zero, so we take a small value (.001) as the lower bound for performing the bisection.

Once we could compute the current distribution given a voltage, we made a wrapper function (“VoltageToIntensity.m”, see Appendix) which, given a voltage drop, solves the system and returns the difference between the total simulated current ($\int_0^1 i(x) dx$) and a target intensity. Then, the bisection algorithm was applied to this wrapper function to find the voltage drop which gives a target intensity.

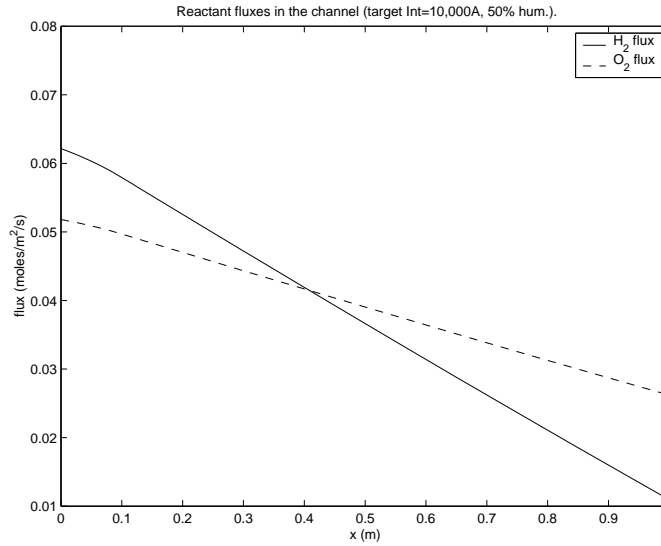
5.3.2 Results for Constant Pressure

In this subsection, we show results obtained for the constant pressure case. The following assumptions were made:

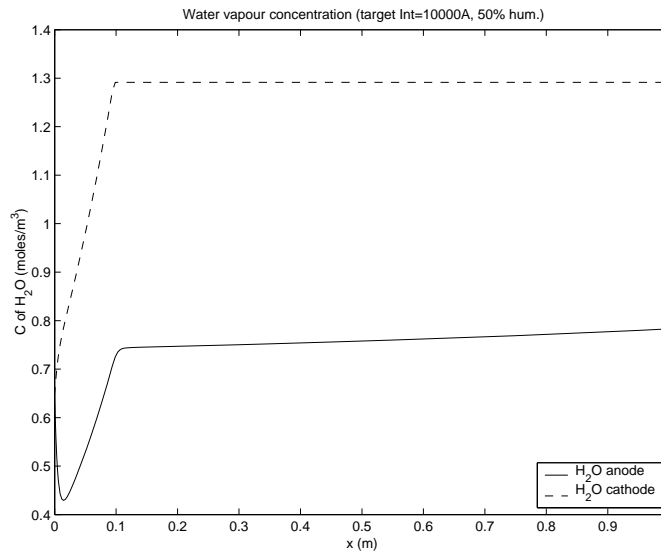
- Constant temperature (298K)
- Constant pressure (101.3kPa)
- Inlet humidities of 50% in the cathode and anode
- Inlet hydrogen flux of 1.2 x stoichiometric flow
- Inlet oxygen flux of 2 x stoichiometric flow
- Air is composed only of nitrogen (79%) and oxygen (21%)
- All flux values are averaged over channel height and gas components are assumed to be well-mixed
- The cell is 1 meter long ($x \in [0, 1]$)
- Target intensity of 10,000 Amps (used to compute initial values for the fluxes of hydrogen and oxygen in terms of their stoichiometric flows)

Simulation Results for the Forward Problem

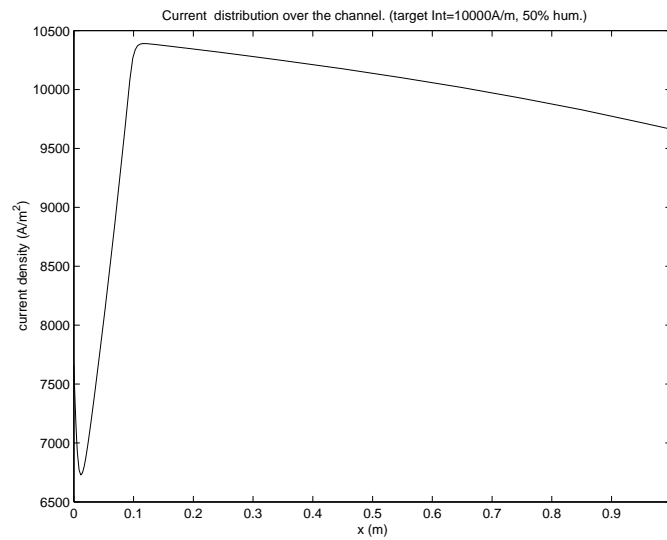
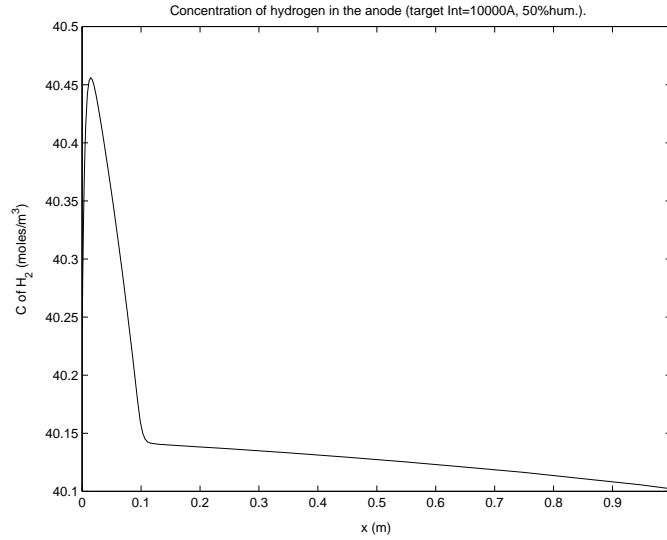
The main results of the simulation are summarized in the following graphs:



Note that the flux of hydrogen in the anode decreases faster than the flux of oxygen in the cathode since two molecules of H_2 are used for each molecule of O_2 in the chemical reaction.



Initially, the cathode and anode relative humidities are equal, so that by analyzing the equation for the anode water flux it is clear that the anode water flux (and so concentration) will decrease steadily. The cathode humidity grows quite fast and reaches saturation quickly, and so becomes larger than the anode humidity quickly, resulting in diffusion of water from the cathode to the anode. Once the cathode is saturated, the cathode humidity is fixed at 100%, and it appears that the diffusive flux to the anode nearly balances the flux drawn by the reaction.



Note that when reaching the water saturation in the cathode, the behaviour of the concentration of hydrogen and of the current density is modified. As the anode water flux drops sharply near the inlet, the concentration of hydrogen rises accordingly to fill the volume. Essentially the hydrogen concentration in the beginning mirrors the behaviour of the anode water concentration, and then dies off slowly after saturation, as it is consumed by the reaction.

Similarly, the current distribution near the inlet appears to be most strongly influenced by the humidities in the anode and cathode. Once the saturation state is reached in the anode, the depletion of the oxygen down the channel becomes the dominant factor, causing the slow decrease in current generation down the channel.

5.3.3 Dependence on the Humidity Levels

Using our inverse problem solver, we computed the voltage drop which would give an intensity of 10,000 amps for varying humidity level, in the cathode and anode.

Cell Voltage for Target Current at 10,000 Amps/meter

rc vs. ra	10%	25%	50%	100%
10%	.449185	.456813	.464998	.474523
25%	.468094	.471249	.475192	.480386
50%	.479853	.481143	.482819	.485153
100%	.486229	.486485	.486783	.487244

In the above table, rc (ra) is the relative humidity in the cathode (anode). From these numbers, we reached the following conclusions:

- Fuel cell performance improves with increased humidity
- Cathode humidity has slightly larger influence than anode

5.3.4 Variable Pressure Case

We modified our MATLAB code to consider variable pressure in the channel. Two ordinary differential equations were added to the original system, one each for the pressure in the cathode and in the anode, with boundary condition of 101.3 kPa at the inlet (see previous section). We tried three realistic channel heights (supplied by our mentor) and observed the resulting performance. In the second table below, the voltage drop was fixed at 0.4828087.

Dependence of Cell Performance on Channel Height with $I=10,000$ Amps

Channel Height	.01m	.003m	.001m
Voltage	.4828087	.4828049	.4826203

(In)Dependence of Current on Channel Height for fixed Voltage ($U = 0.4828087$)

Channel Height	.01m	.003m	.001m
Current	9.99 e3	1.00 e4	9.99 e3

Hence, for channel heights between 1cm and 1mm, the pressure change has little effect on the behaviour and performance of the fuel cell. This means that considering the presence of liquid water in the channels would also have little impact on the model, as the liquid water would only cause a small change in the channel height.

5.3.5 Counterflow Simulation

We began to extend our numerical model to simulate the counterflow case, i.e. when humidified air is put in the cathode from the left ($x = 0$) and humified hydrogen is put in the anode from the right ($x = 1$). The problem is more complex since we need to solve a first order system with some boundary values on the left and some on the right of the channel. Some kind of shooting method is required. We decided not to use a zero-finding routine for vector-valued functions because of robustness concerns. Our alternative idea involves nested scalar zero-finding:

1. Suppose we have a guess for the flux of water vapour at the anode outlet. Given the voltage drop corresponding to a target intensity, then we know the flux of hydrogen exactly at the anode outlet ($\frac{(stoich-flow-1)target-I}{2*F}$). We run bisection to find this voltage.
2. Run an outer bisection on the flux of water vapour at the anode outlet to find the value that corresponds to the specified flux of water vapour at the anode inlet.

We ran into difficulties, since the shooting on the water vapour flux is very sensitive, and our code does not handle properly the unphysical cases (for example, when a reactant or water runs out).

5.4 Further Work

The simulation of our fuel cell model with counterflow would probably reveal different behaviour and so it would be interesting to compare results with the coflow case. One could also look at the effect of liquid water inside the channel when the gas is saturated with water vapour, even though we have shown some evidence that it should have little effect for channel heights in the order of 1cm or 1mm. In our model, we assumed that the gases are well-mixed (in the direction perpendicular to the membrane). So, we could remove this assumption and add to our model the diffusion of the gases inside each channel. In that case, the gases would be moving towards the membrane as the reactants and water are used in the reaction. One can also consider a full 3D model. In this case, the derivative of the pressure would depend on $\frac{1}{h^4}$ instead of $\frac{1}{h^3}$ as in our model. So, the effect of pressure might become important.

5.5 Conclusion

As a summary, here are the main conclusions we obtained from our model:

1. The fuel cell performance is improved when increasing the humidity level, particularly on the cathode (air) side.
2. Incorporating variable pressures and the effect of liquid water have only small impact on the performance.

Chapter 6

Hedging in the Presence of Market Constraints

Participants: Alexander V. Melnikov (Mentor), Andrei Badescu, Hammouda Ben Mekki, Alicia Cantón, Raymond Ka Wai Cheng, Asrat Fikre Gashaw, Ana Granados, Yuanyuan Hua, Stephan Lawi, Marat Molyboga, Yuri Petratchenko, Victoria Skornyakova.

PROBLEM STATEMENT: We have studied mathematical models which describe financial (B, S) markets with both discrete and continuous time. The Black-Scholes model with stochastic volatility was investigated from point of view of controlled diffusion processes. To give an appropriate approximation of the Bellman equation solution we used (see Appendix and [20], [21], [14]) a small parameter approach and numerical methods. The Cox-Ross-Rubinstein model two different interest rates (borrowing and lending) was also investigated.

6.1 Introduction

Mathematics of Finance is one of the most applicable branches of the fundamental contemporary mathematics. The quantitative methods of financial mathematics based on stochastic analysis were introduced in papers by Black, Scholes and Merton and led to the modern option pricing theory and the contingent claims analysis. The pricing theory is based on the non-arbitrage principle and two fundamental theorems of contemporary financial mathematics:

1. There is no arbitrage on the financial market iff a risk neutral (martingale) measure exists.
2. The financial market is complete iff there is a unique martingale measure.

The classical model of financial markets such as Black-Scholes and Cox-Ross-Rubinstein are complete and, thus, the corresponding price of a contingent claim f_T can be calculated as $E^* \frac{f_T}{B_T}$, where B_T is the terminal non-risk asset price and E^* is the mathematical expectation with respect to the unique martingale measure P^* . In this situation we deal with the Black-Scholes and the Cox-Ross-Rubinstein formulae as well as the linear Black-Scholes equation (see Appendix).

In the case of incomplete markets (markets with constraints on the model structure and strategies) we deal with a range of “fair” prices:

$$\left[\inf_{P^*} E^* \frac{f_T}{B_T}, \sup_{P^*} E^* \frac{f_T}{B_T} \right].$$

The Black-Scholes model with stochastic volatility (see [20] and Appendix for details) is a representative example of such a market. This model is attractive from a theoretical as well as a practical perspective. We applied the general theory of controlled diffusion processes (see [14]) and the Bellman principle to arrive to a non-linear Bellman equation which appears to be an expansion of the linear Black-Scholes equation. The key to getting a good approximation of the Bellman equation solution was in applying a small perturbation approach. The first results in this area were obtained for the case of a generalized Bachelier model with stochastic volatility (see [21]). In the case of the generalized Black-Scholes model with stochastic volatility we have obtained a first approximation of the solution of the non-linear Bellman equation which is our key result.

We have also considered a generalized Cox-Ross-Rubinstein model with different risk-free rates of borrowing and lending. We showed completeness of this market by an explicit unique representation of a hedging strategy for any contingent claim. The result was obtained by applying backward induction to a binomial tree of stock prices.

We have used numerical methods to find a solution of the non-linear Bellman equation. The numerical and the analytical results coincide for the whole reasonable range of the model parameters. We have also developed an automated trading system for the generalized Cox-Ross-Rubinstein model with two different risk-free rates.

6.2 Continuous-Time Model

6.2.1 The Black-Scholes Model

Consider a complete (B, S) -market, composed of a non-risky bond B and a risky asset S . Let them satisfy the following stochastic differential equations:

$$\begin{cases} dB_t &= rB_t, dt \\ dS_t &= S_t(\mu dt + \sigma dW_t), \end{cases}$$

where r is a constant risk-free rate of return, μ is a risky average rate of return, σ is a constant volatility of the asset and W_t is the standard Brownian motion. In the framework of this model we can find the close form solutions for the price of various contingent claims, such as a European Call option.

$$C_0(T, r, \sigma) = S_0 \Phi(d_+(r)) - K e^{-rT} \Phi(d_-(r)) \quad (6.1)$$

where $\Phi(x)$ is the standard normal distribution and $d_{\pm}(r) = \frac{\ln \frac{S_0}{K} + T(r \pm \frac{\sigma^2}{2})}{\sigma \sqrt{T}}$.

6.2.2 The Bellman Equation

Consider an incomplete (B, S) -market. By the non-arbitrage principle for incomplete markets, some contingent claims cannot be replicated by any self-financing strategy. Therefore, one must consider a larger class of portfolios, namely the self-financing strategies with consumption C , where C is a non-decreasing process. The capital of the portfolio, $X_t^{\pi, C}$, satisfies the following stochastic equation:

$$dX_t^{\pi, C} = \gamma_t dS_t + \beta_t dB_t - dC_t. \quad (6.2)$$

From the optional decomposition theorem, adding the consumption process turns the discounted capital of the portfolio into a supermartingale. Furthermore, the minimal hedge theorem demonstrates the existence of a minimal hedge π^* of the form:

$$\frac{X_t^{\pi^*}}{B_t} = \text{ess sup}_{\tilde{P} \in \mathcal{M}(\frac{S}{B}, P)} \tilde{E} \left(\frac{f_T}{B_T} \middle| \mathcal{F}_t \right) \quad (6.3)$$

where $\mathcal{M}(\frac{S}{B}, P)$ is the set of all martingale measures for $\frac{S}{B}$ (see Appendix and [20] for details).

Consider a generalized Black-Scholes model with stochastic volatility:

$$\begin{cases} dS_t &= S_t(\mu dt + \Sigma_t dW_t), & \Sigma_t^2 &= \sigma^2 + (-1)^{\Pi_t} \Delta\sigma^2 \\ dB_t &= rB_t dt \end{cases}$$

where Π_t is a Poisson process with intensity λ and W_t is a standard Brownian motion. One recovers the Black-Scholes if $\Delta\sigma$ is set to 0. The stochastic process Σ_t defines a telegraph signal that oscillates between two values, i.e. $\sigma_{min} = \sqrt{\sigma^2 - \Delta\sigma^2}$ and $\sigma_{max} = \sqrt{\sigma^2 + \Delta\sigma^2}$, such that $\frac{\Delta\sigma^2}{\sigma^2} \ll 1$.

The key idea is now to find the upper price of a European Call option of pay-off $f(S_T) = (S_T - K)^+$. We emphasize the role of Σ_t as a control parameter in the following sense. Let $\alpha = (\alpha_t)_{t \leq T}$ be the process taking values in the set $\{\sigma_{min}, \sigma_{max}\}$. As a function of α , $S_t^{(\alpha)}$ is a controlled diffusion process. According to the minimal hedge structure, the minimal hedge is as follows:

$$v^*(x, t) = \sup_{\alpha} E[f_T e^{-r(T-t)}]. \quad (6.4)$$

By means of general theory of controlled stochastic processes and the Kolmogorov-Itô formula, we get the Bellman equation:

$$\begin{cases} \frac{\partial v}{\partial t} + rx \frac{\partial v}{\partial x} - rv + \frac{1}{2} \sigma^2 x^2 \frac{\partial^2 v}{\partial x^2} + \frac{1}{2} \Delta\sigma^2 x^2 \left| \frac{\partial^2 v}{\partial x^2} \right| = 0 \\ v(x, T) = (x - K)^+. \end{cases}$$

Now to simplify this equation we use the following idea. We know that the corresponding Black-Scholes equation ($\Delta\sigma = 0$) can be reduced to the classical heat equation through the following change of variables (see [30]):

$$\begin{aligned} \xi &= \ln x + (r - \sigma^2/2)(T - t) \\ \theta &= \sigma^2(T - t) \\ V(\theta, \xi) &= \exp(r(T - t))v(x, t). \end{aligned} \quad (6.5)$$

The expected result is the heat equation plus a perturbation term of the order $\Delta\sigma^2$.

$$\begin{cases} \frac{\partial V}{\partial \theta} = \frac{1}{2} \frac{\partial^2 V}{\partial \xi^2} + \frac{1}{2} \frac{\Delta\sigma^2}{\sigma^2} \left| \frac{\partial^2 V}{\partial \xi^2} - \frac{\partial V}{\partial \xi} \right| \\ V(0, \xi) = (e^\xi - K)^+ \end{cases} \quad (6.6)$$

An important remark can be added: this equation does not have a close form solution. Hence, the following sections deal with analytical and numerical techniques to approximate the solution.

6.2.3 An Analytic Approximation to the Bellman Equation

Since the corresponding parabolic operator is regular, from the general theory of partial differential equations one can use the well-known small perturbation approach:

$$V(\theta, \xi) = V_0(\theta, \xi) + \Delta\sigma^2 V_1(\theta, \xi) + o(\Delta\sigma^2) \quad (6.7)$$

After combining terms that don't have $\Delta\sigma^2$ in the equation (6.6) we get the following partial differential equation:

$$\begin{cases} \frac{\partial V_0}{\partial \theta} = \frac{1}{2} \frac{\partial^2 V_0}{\partial \xi^2} \\ V_0(0, \xi) = (e^\xi - K)^+, \end{cases}$$

and combining terms of the first order of $\Delta\sigma^2$ gives the following equation:

$$\begin{cases} \frac{\partial V_1}{\partial \theta} = \frac{1}{2} \frac{\partial^2 V_1}{\partial \xi^2} + \frac{1}{2\sigma^2} \left| \frac{\partial^2 V_0}{\partial \xi^2} - \frac{\partial V_0}{\partial \xi} \right| \\ V_1(0, \xi) = 0. \end{cases}$$

The equation (6.8) is the Black-Scholes equation and its solution has the following form:

$$V_0(\theta, \xi) = e^{\xi + \theta/2} \Phi\left(\frac{\xi + \theta - \ln K}{\sqrt{\theta}}\right) - K \Phi\left(\frac{\xi - \ln K}{\sqrt{\theta}}\right). \quad (6.8)$$

The absolute value term in equation (6.8) makes it difficult for one to solve the equation. Direct calculation shows that

$$\frac{\partial V_0^2}{\partial \xi^2} - \frac{\partial V_0}{\partial \xi} = \frac{K}{\sqrt{\theta}} \varphi\left(\frac{\xi - \ln K}{\sqrt{\theta}}\right) \geq 0,$$

where $\varphi(x)$ is the standard normal probability density function (PDF), which is obviously greater than 0.

It follows from the general theory of parabolic equations, the solution of equation (6.8) is given by the following formula:

$$V_1(\theta, \xi) = \int_0^\theta \int_{-\infty}^{\infty} \frac{f(x, \tau)}{\sqrt{2\pi(\theta - \tau)}} \exp\left(-\frac{(x - \xi)^2}{2(\theta - \tau)}\right) d\tau dx = \frac{K\sqrt{\theta}}{2\sigma^2} \varphi\left(\frac{\xi - \ln K}{\sqrt{\theta}}\right).$$

where $f(x, \tau) = \frac{K}{2\sigma^2\sqrt{\tau}} \varphi\left(\frac{x - \ln K}{\sqrt{\tau}}\right)$.

This calculation provides an approximation to the upper price (cf. equation (6.4)).

$$\begin{aligned} v^*(S_0, 0) &\approx S_0 \Phi\left(\frac{\ln S_0/K + (r + \sigma^2/2)T}{\sigma\sqrt{T}}\right) \\ &\quad - K e^{-rT} \Phi\left(\frac{\ln S_0/K + (r - \sigma^2/2)T}{\sigma\sqrt{T}}\right) \\ &\quad + \frac{K\Delta\sigma^2}{2\sigma^2} e^{-rT} \sigma\sqrt{T} \varphi\left(\frac{\ln S_0/K + (r - \sigma^2/2)T}{\sigma\sqrt{T}}\right) \end{aligned} \quad (6.9)$$

Similarly, an approximation for the lower price is given by:

$$\begin{aligned} v_*(S_0, 0) &\approx S_0 \Phi\left(\frac{\ln S_0/K + (r + \sigma^2/2)T}{\sigma\sqrt{T}}\right) \\ &\quad - K e^{-rT} \Phi\left(\frac{\ln S_0/K + (r - \sigma^2/2)T}{\sigma\sqrt{T}}\right) \\ &\quad - \frac{K\Delta\sigma^2}{2\sigma^2} e^{-rT} \sigma\sqrt{T} \varphi\left(\frac{\ln S_0/K + (r - \sigma^2/2)T}{\sigma\sqrt{T}}\right) \end{aligned} \quad (6.10)$$

6.2.4 Small Parameter Approach with the Probability Density Function (PDF)

Another way to solve the problem of pricing under a stochastic volatility model is to consider the PDF that satisfies the Bellman equation. The expectation is then computed by integrating the discounted pay-off multiplied by the density over all possible values of the underlying stock.

The expansion is this time applied directly to the PDF or pricing kernel, $U(\xi_t, \xi, \theta)$, following the idea that it should be a slight variation of the PDF, $U_0(\xi_t, \xi, \theta)$ satisfying the Black-Scholes equation.

$$U(\xi_t, \xi, \theta) = U_0(\xi_t, \xi, \theta) + \Delta\sigma^2 f_1(\xi_t, \xi, \theta) + o(\Delta\sigma^2) \quad (6.11)$$

where ξ_t is the value of the variable ξ at time t or $\theta = \sigma^2(T - t)$ and $f_1(\xi_t, \xi, \theta)$ is a function that must integrate to 0 in the ξ -space.

The problem is then reduced to solving equation (6.8) for f_1 , knowing that U_0 satisfies the heat equation (6.8). Unfortunately, the method is rather tedious and gives rise to complicated partial differential equations that discouraged the authors from further investigations.

The main advantage of this approach is that one considers the general solution of the initial PDE and sets aside the boundary conditions (in time) until the end of the computation. Once the PDF expansion is worked out, the result stays valid for any pay-off. This method has indeed a much wider range of applications since one density computation leads to the pricing of almost any contingent claim. (In the worst case, the results can always be integrated numerically.)

To conclude, here are a few comments on why the tedious calculations are not presented in the report and which paths should be followed for further insight. The first remark is that once the PDF is computed up to some order in $\Delta\sigma^2$, one must make sure that the result integrates to 1 over the whole ξ -space. It would indeed not make any sense to get a PDF with absorption for a given PDE (knowing that the model converges numerically). This unfortunately could not be completely proven and therefore prevented the discovery of promising results.

6.2.5 Numerical Solution

Since an exact solution of the Bellman equation (6.8) is unattainable, we solve it using numerical techniques, i.e. the second order approximation for the spatial first and second derivatives, and the explicit forward Euler method to progress the solution in time. Thus, our scheme is

$$\frac{V_i^{n+1} - V_i^n}{\Delta\theta} = \frac{1}{2} \frac{V_{i+1}^n - 2V_i^n + V_{i-1}^n}{\Delta\xi^2} + \frac{1}{2} \frac{\Delta\sigma^2}{\sigma^2} \left| \frac{V_{i+1}^n - 2V_i^n + V_{i-1}^n}{\Delta\xi^2} - \frac{V_{i+1}^n - V_{i-1}^n}{2\Delta\xi} \right|$$

where $\Delta\xi$ and $\Delta\theta$ are the space and time steps respectively.

To verify the stability of our scheme, we perform a Von Neumann stability analysis. First, assume a solution of the form $v_i^n = g^n e^{ikm\Delta\xi}$, and plug it into our scheme to obtain

$$g = 1 + \frac{\Delta\theta}{2\Delta\xi^2} (e^{ik\Delta\xi} - 2 + e^{-ik\Delta\xi}) + \frac{\delta\Delta\theta}{2} \left| \frac{e^{ik\Delta\xi} - 2 + e^{-ik\Delta\xi}}{\Delta\xi^2} - \frac{e^{ik\Delta\xi} - e^{-ik\Delta\xi}}{2\Delta\xi} \right| \quad (6.12)$$

where $\delta := \frac{\Delta\sigma^2}{\sigma^2}$.

We consider two cases:

- **Case 1:** $\frac{\partial^2 V}{\partial \xi^2} - \frac{\partial V}{\partial \xi} \geq 0$.

Here (6.12) becomes

$$g = 1 + \frac{\Delta\theta}{\Delta\xi^2} (\cos(k\Delta\xi) - 1) + \frac{\delta\Delta\theta}{\Delta\xi^2} (\cos(k\Delta\xi) - 1) - i \frac{\delta\Delta\theta}{2\Delta} \sin(k\Delta\xi)$$

We require $|g| \leq 1 + C\Delta\theta$ for stability, where C is a constant. For short, write A as

$$A := \left(1 + \frac{\Delta\theta}{\Delta\xi^2} (\cos(k\Delta\xi) - 1) + \frac{\delta\Delta\theta}{\Delta\xi^2} (\cos(k\Delta\xi) - 1) \right)^2,$$

then we have

$$|g|^2 = A + \frac{\delta^2\Delta\theta^2}{4\Delta\xi^2} \sin^2(k\Delta\xi) \quad (6.13)$$

Now, $A \leq 1$ iff

$$-2 \leq \frac{\Delta\theta^2}{\Delta\xi^2} (\cos(k\Delta\xi) - 1) + \frac{\delta\Delta\theta}{\Delta\xi^2} (\cos(k\Delta\xi) - 1) \leq 0,$$

and this holds if

$$\frac{\Delta\theta}{\Delta\xi^2} \leq \frac{1}{1+\delta} \quad (6.14)$$

because $\cos(k\Delta\xi) - 1 \geq -2$.

Now, we use (6.14) to bound the second part of (6.13):

$$\frac{\delta^2\Delta\theta^2}{4\Delta\xi^2} \sin^2(k\Delta\xi) \leq C\Delta\theta,$$

where the constant $C = \frac{\delta^2}{4(1+\delta)}$. Therefore, we obtain that $|g|^2 \leq 1 + C\Delta\theta$, i.e. $|g| \leq 1 + C\Delta\theta$, provided (6.14) holds.

- **Case 2:** $\frac{\partial^2 V}{\partial \xi^2} - \frac{\partial V}{\partial \xi} < 0$.

Similarly, we can show that the scheme is stable if $\frac{\Delta\theta}{\Delta\xi^2} \leq \frac{1}{1-\delta}$, using the given condition $\delta < 1$.

To guarantee stability, we impose the stronger condition (6.14). The scheme is consistent because the truncation error is $O(\Delta\theta + \Delta\xi^2)$, so by the Lax-Richtmyer equivalence theorem, we conclude that the numerical method converges.

6.2.6 Graphs Interpretation

We compute and plot the numerical solutions and analytical solutions of both the Black-Scholes equation and the Bellman equation, for a European call option. We do this for $\Delta\sigma = 0.05, 0.1$ and 0.2 . See attached plots. Here are some observations:

- Increasing the value of $\Delta\sigma$ increases the price of the call option.
- For the stochastic volatility model, the call option is more expensive.
- The approximate analytical solution and the numerical solution are indistinguishable for small $\Delta\sigma$.
- For $\Delta\sigma = 0.2 (= \sigma)$, the analytical solution computed using the small-parameter ($\Delta\sigma$) approach becomes less accurate.

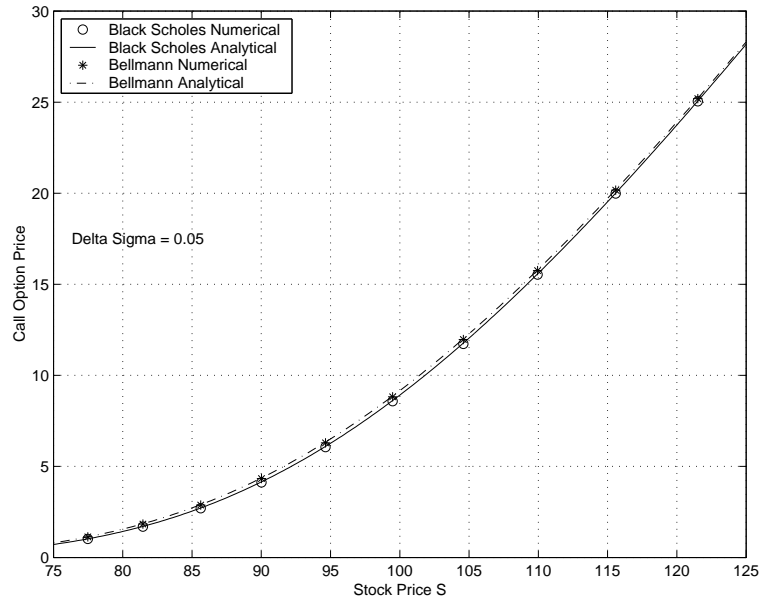


Figure 6.1:

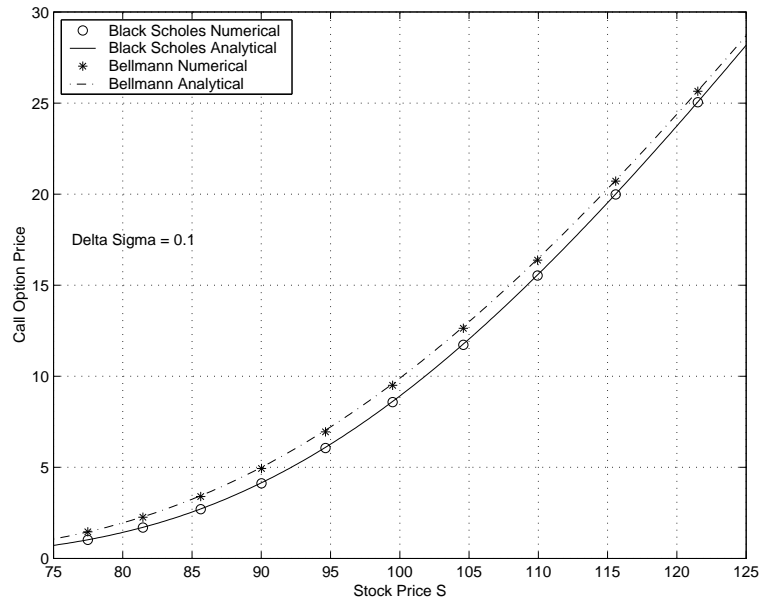


Figure 6.2:

6.3 Discrete-Time Model

The classical binomial model is described by the following evolution of the stock price:

$$\Delta S_n = \rho_n S_{n-1},$$

where ρ_n is a Bernoulli random variable which takes values b with probability p and a with probability $1 - p$; and by the evolution of the bond's price:

$$\Delta B_n = r B_{n-1},$$

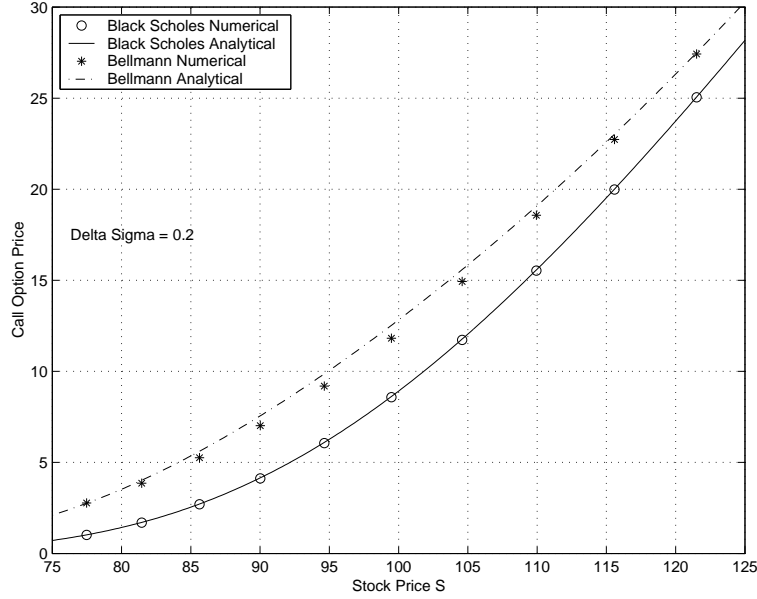


Figure 6.3:

where r is a constant interest rate.

For the model to make sense we need to impose $-1 < a$ to get positive stock, and $a < r < b$ otherwise, the strategy is trivial.

It is well-known that the market is complete, and there is even an explicit formula for pricing the standard European options. From the completeness of the market there is a unique “fair” strategy that replicates the given contingent claim. In other words, there is a unique strategy (β_n, γ_n) , where β_n is the number of bonds and γ_n is the number of stocks, that will be able to pay the claim, at the maturity. That is,

$$\beta_{n+1}B_{n+1} + \gamma_{n+1}S_{n+1} = \beta_n B_{n+1} + \gamma_n S_{n+1}.$$

The pair $\pi_n = (\beta_n, \gamma_n)$ is called a *portfolio*, and we will denote by X_n^π the capital of the portfolio at time n , i.e.:

$$X_n^\pi = \beta_n B_n + \gamma_n S_n.$$

We will say that the portfolio is *self-financing* if

$$\Delta X_{n+1}^\pi = X_{n+1}^\pi - X_n^\pi = \beta_{n+1} \Delta B_{n+1} + \gamma_{n+1} S_{n+1},$$

or equivalently,

$$\beta_{n+1}B_{n+1} + \gamma_{n+1}S_{n+1} = \beta_n B_{n+1} + \gamma_n S_{n+1}.$$

That is, at any given step we can build a new portfolio without taking (or putting) capital from (or to) other source than the portfolio.

In real market we have two different interest rates, a deposit rate r_1 , and a credit rate r_2 . To avoid the possibility of *free lunch* we need $r_2 > r_1$.

We will consider a (B^1, B^2, S) market described by:

$$\begin{aligned} \Delta S_n &= \rho_n S_{n-1} \\ \Delta B_n^i &= r_i B_{n-1}^i, \quad \text{for } i = 1, 2 \end{aligned}$$

where ρ_n is a Bernoulli random variable.

One would expect this market to be incomplete, but it turns out to be complete. It is possible to find the unique strategy that replicates any given claim. A strategy in this market has the form $(\beta_n^1, \beta_n^2, \gamma_n)$ with $\beta_n^1 \geq 0$ and $\beta_n^2 \leq 0$.

A key observation is that one of the β_n^i must be zero, hence the initial portfolio $(\beta_n^1, \beta_n^2, \gamma_n)$ can be interpreted as (β_n, γ_n) . Indeed, it makes no sense to borrow money at a high interest while investing at a lower rate.

The process of constructing the strategy is a backwards induction from time $n = N - 1$ down to $n = 0$. The capital at time N is $X_N^\Pi = f_N$. In what follows, to simplify the notation, set

$$i(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 2 & \text{if } x < 0 \end{cases}$$

The first step of the induction process will be given by the self-financing conditions:

$$\begin{cases} \beta_{N-1} B_N^{i(\beta_{N-1})} + \gamma_{N-1} S_{N-1} (1+b) = f_N^+ \\ \beta_{N-1} B_N^{i(\beta_{N-1})} + \gamma_{N-1} S_{N-1} (1+a) = f_N^- \end{cases}$$

where $f_N^+ = f(S_1, \dots, S_{N-1}(1+b))$ and $f_N^- = f(S_1, \dots, S_{N-1}(1+a))$.

Using the same consideration that proves this statement for the step $N - 1$ we can prove the other steps.

Assume that $\beta_{n+1}^\pm, \gamma_{n+1}^\pm$ are known, where the notation follows the same convention as in f_N^\pm . The self-financing condition of the portfolio is given by,

$$\begin{cases} \beta_n B_{n+1}^{i(\beta_n)} + \gamma_n S_n (1+b) = \beta_{n+1}^+ B_{n+1}^{i(\beta_{n+1}^+)} + \gamma_{n+1} S_{n+1} \\ \beta_n B_{n+1}^{i(\beta_n)} + \gamma_n S_n (1+a) = \beta_{n+1}^- B_{n+1}^{i(\beta_{n+1}^-)} + \gamma_{n+1} S_{n+1}. \end{cases}$$

Cramer's rule gives,

$$\begin{aligned} \beta_n &= \frac{S_n}{B_{n+1}^{i(\beta_n)} S_n (a-b)} (\beta_{n+1}^+ B_{n+1}^{i(\beta_{n+1}^+)} (1+a) - \beta_{n+1}^- B_{n+1}^{i(\beta_{n+1}^-)} (1+b) \\ &\quad + S_{n+1} (\gamma_{n+1}^+ (1+a) - \gamma_{n+1}^- (1+b))). \end{aligned}$$

Note that the expression for β_n is self-referencing. Nevertheless, the sign in the denominator is always negative regardless of β_n , and that the numerator does not depend on β_n . Therefore the sign β_n is determined by the numerator, and we can calculate uniquely β_n . Now find the corresponding γ_n . This proves the uniqueness of the strategy and therefore the completeness of the market.

To illustrate how this method works, we have implemented a numerical example for a put European option $f_N = (K - S_N)^+$ with $N = 40, K = 100, a = -0.01, b = 0.02, B_0^i = 1, r_1 = 0.001, r_2 = 0.002, S_0 = 100$.

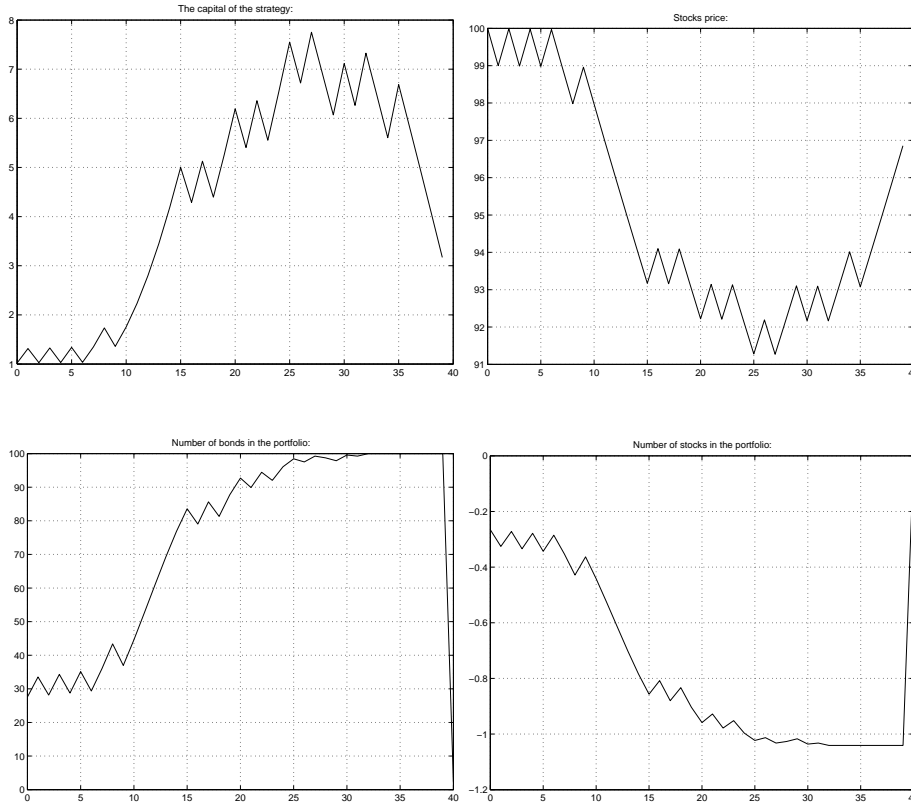


Figure 6.4:

Appendix: Advanced Pricing Methodologies in Contemporary Financial Mathematics

Consider a financial market model $((B, S)\text{-market})$ as a pair of non-risky (bank account) B and risky (stock) S assets represented by their prices B_t and S_t , where $t = 0, 1, 2, \dots$ (discrete-time model) or $t \geq 0$ (continuous-time model). Certainly, the risky component S can be a vector, i.e. consist of several risky assets. B and S are called the *primary assets*.

Let us define a *contingent claim* (for fixed T) as a function $f_T \equiv f_T(S_0, \dots, S_T)$. Taking a non-risky asset B_t and a risky asset S_t in amount β_t and γ_t respectively we form an *investment portfolio* $\pi_t = (\beta_t, \gamma_t)$. The *capital* of such a portfolio π is equal to

$$X_t^\pi(x) = \beta_t B_t + \gamma_t S_t, X_0^\pi(x) = x.$$

Denote by SF the following set of *self-financing* portfolios π :

$$\pi \in SF, \text{ if } X_t^\pi(x) - X_{t-1}^\pi(x) = \Delta X_t^\pi(x) = \beta_t \Delta B_t + \gamma_t \Delta S_t$$

in discrete-time models ($dX_t^\pi(x) = \beta_t dB_t + \gamma_t dS_t$ in continuous-time models if differentials dB_t and dS_t are correctly defined).

Arbitrage (at time T) means a possibility to make a positive capital at time T (with a positive probability) by some self-financing portfolio starting from zero initial capital.

Any *derivative security* on the given (B, S) -market can be identified with a corresponding contingent claim. For instance, *forward contract* with forward price F and expiration date T is equivalent to $f_T = f_T(S_T) \equiv S_T - F$ and *call option* with the strike price K is equivalent to $f_T = (S_T - K)^+$, etc. Thus, the set of derivative securities

induces a set of graphs (CCG). On the other hand, the set of self-financing portfolios forms a set of graphs of its terminal values $X_t^\pi(x)$ (TVG). The market is *complete* if TVG=CCG. Otherwise, it is *incomplete*. In other words, the market is called complete iff any contingent claim f_T can be *replicated*: there exist $x \geq 0$ and $\pi \in SF$ such that $X_T^\pi(x) = f_T$.

We face an important problem of defining mathematically a risky asset S . The natural answer is to model the prices S_t as a *random process* on a fixed probability space.

Denote by V_t the price of f_T at time $t \leq T$. One of the basic problems is to describe this stochastic process in terms of the (B, S) -market.

The *heuristic principle* of obtaining the price V_t consists of two steps: first, the value f_T should be discounted by the non-risky asset B_t and then its conditional expectation $E[f_T B_t B_T^{-1} | F_t]$ one can take as a candidate for V_t . The first step has no drawbacks, but the second one can be criticized because there is no proof that we should calculate the price V_t with respect to the physical measure P .

Any probability measure P^* on (Ω, F, P) defines its own “probability character” of the market prices B_t and S_t . It is clear that the “stable” character of the market should lead to a more “natural” price of the given contingent claim. So, this heuristic principle should be adjusted by choosing the most adequate probability measure P^* equivalent to P .

The above explanation leads to the following *non-arbitrage principle*. The (B, S) -market admits *no arbitrage possibility* if there exists a probability measure P^* equivalent to P such that the process of discounted risky asset prices $B_t^{-1} S_t$ is a martingale with respect to P^* . It turns out that often this statement can be reversed (see Harrison and Kreps, 1979, Stricker, 1990). Therefore, the measure P^* is called the *martingale* one and represents a level of the market stability. That is why P^* is called a “*risk-neutral*” measure.

According to Harrison and Pliska, 1981, the uniqueness of P^* leads to the statement:

On the non-arbitrage complete (B, S) -market the price of any contingent claim f can be defined uniquely iff the martingale measure is unique.

Here is the sketch of the proof.

(\Rightarrow) Let us assume there are two martingale measures $P_i^*, i = 1, 2$. Define two processes as price processes for given contingent claim f_T :

$$V_t^i = B_t E_i^* [B_T^{-1} f_T | F_t], i = 1, 2.$$

But $V_t^1 \equiv V_t^2$ and therefore $P_1^* \equiv P_2^*$.

(\Leftarrow) According to the previous statement there is a martingale measure P^* such that $\frac{S_t}{B_t}$ and $\frac{V_t}{B_t}$ are local martingales with respect to P^* . But P^* is unique and therefore the price process V_t can be defined uniquely as $V_t = B_t E^* [B_T^{-1} f_T | F_t]$.

Thus, we have the following methodology of pricing contingent claims in *complete markets*:

Let us assume that the (B, S) -market is complete and P^* is a unique martingale measure. If we define the price V_t of a given contingent claim f_T as

$$V_t = B_t E^* [B_T^{-1} f_T | F_t]$$

then (B, S, V) is a unique system of prices for (basic and derivative) securities iff the corresponding expanded market admits no arbitrage. Moreover, there exists such a *hedging* strategy $\pi \in SF$ that $X_t^\pi(V_0) = V_t, t \leq T$.

The above statement can be interpreted as *a possibility to reduce any risk related to any contingent claim to zero*. Below we give several classical examples of complete markets to illustrate that.

Binomial model or Cox-Ross-Rubinstein model, 1976.

Denote by $\rho_t = \frac{\Delta S_t}{S_{t-1}}$ for $t = 1, 2, \dots$ the rate of return of the risky asset S . Assume that $\rho = (\rho_t)_{t \geq 1}$ is a sequence of independent random variables with two values $b > a$ and corresponding probabilities of p and $1 - p$, where $p \in (0, 1)$. Denote, $\mu = bp + a(1 - p)$. Then we can represent $\rho_t = \mu + w_t$ as a random walk near the mean relative rate of return μ . Let us denote the rate of return of B as r . Then the unique martingale measure is defined by $p^* = \frac{r-a}{b-a}$ for $b > r > a > -1$.

Bachelier model, 1900.

This is a pure continuous (B, S) -market model where S is defined by $\frac{dS_t}{dt} \equiv \dot{S} = \mu + \sigma \dot{w}_t$, $S_0 > 0$. Here \dot{w}_t is the well-known Gaussian “white noise” and σ is a volatility parameter. The risk free rate of return is considered to be equal to zero ($dB \equiv 0$).

Define a martingale measure $dP_T^*/dP_T = \exp\left\{-\frac{\mu}{\sigma}w_T - \frac{1}{2}\left(\frac{\mu}{\sigma}\right)^2 T\right\}$. The non-arbitrage principle gives the following price C_T of the call option $f_T = (S_T - K)^+$:

$$C_T = V_0 = E^*(S_T - K)^+ = (S_0 - K)\Phi\left(\frac{S_0 - K}{\sigma\sqrt{T}}\right) + \sigma\sqrt{T}\phi\left(\frac{S_0 - K}{\sigma\sqrt{T}}\right),$$

where $\Phi(x) = \int_{-\infty}^x \phi(y)dy$ and $\phi(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}$ are the distribution and the density functions of a standard normal distribution respectively.

Black-Scholes-Merton model, 1973.

Consider the following relative rates of return for B and S:

$$\frac{\dot{B}_t}{B_t} = r, \quad \frac{\dot{S}_t}{S_t} = \mu + \sigma \dot{w}_t.$$

It is clear that the graphic representation of $\frac{\dot{S}_t}{S_t}$ will be the same as the one of \dot{S}_t in the Bachelier model. This model can also be presented either as a system of two stochastic differential equations:

$$dB_t = B_t r dt, \quad dS_t = S_t(\mu dt + \sigma dw_t),$$

where w_t is a standard Wiener process (Brownian motion), or in the following “exponential form”:

$$B_t = B_0 e^{rt}, \quad S_t = S_0 e^{\mu t - \frac{\sigma^2 t}{2} + \sigma w_t}.$$

The unique martingale measure can be determined as $dP_T^* = Z_T^* dP$, where the density $Z_T^* = \exp\left\{-\left(\frac{\mu-r}{\sigma}\right)w_T - \frac{1}{2}\left(\frac{\mu-r}{\sigma}\right)^2 T\right\}$. By applying the non-arbitrage methodology we obtain the celebrated Black-Scholes-Merton formula of the call option price:

$$C_T(T, r, \sigma) = E^* e^{-rT} (S_T - K)^+ = S_0 \Phi(d_+(r)) - K e^{-rT} \Phi(d_-(r)),$$

where $d_{\pm}(r) = \frac{\ln\left(\frac{S_0}{K}\right) + T(r \pm \frac{\sigma^2}{2})}{\sigma\sqrt{T}}$.

Note, the heuristic principle gives another price

$$C_{heuristic}(\mu) = e^{(\mu-r)T} S_0 \Phi(d_+(\mu)) - K e^{-rT} \Phi(d_-(\mu))$$

and, therefore,

$$C_{heuristic}(r) = C(T, r, \sigma).$$

The *third method* used by Black and Scholes is the method of differential equations (see Duffie, 1996) described below.

Let $f_T = g(S_T)$, $g \geq 0$ be a measurable function. Consider portfolios with the capital of the form $V_t = \nu(S_t, t)$, $t \leq T$, a smooth function, such that

$$\nu(S_t, T) = g(x), \quad x > 0$$

$$\nu(x, t) \geq 0, \quad x > 0, \quad t \leq T.$$

After applying the Kolmogorov-Ito formula to $\nu(S_t, t)/B_t$ we have

$$\frac{\nu(S_t, t)}{B_t} = \frac{\nu(S_0, 0)}{B_0} + \int_0^t \frac{\partial \nu}{\partial x} d\left(\frac{S}{B}\right)_u + \int_0^t \left(\frac{\partial \nu}{\partial t} + L^0 \nu - r\nu\right) B_u^{-1} du,$$

where $L^0 = rx \frac{\partial}{\partial x} + \frac{1}{2} \sigma^2 x^2 \frac{\partial^2}{\partial x^2}$ is the generator of the diffusion process S_t .

Evidently, $V_t = \nu(S_t, t)$ is the capital of a self-financing strategy iff $\frac{\partial \nu}{\partial t} + L^0 \nu - r\nu = 0$ (*Black-Scholes equation*). It means ν should be a harmonic function for the operator $L = \frac{\partial}{\partial t} + L^0 - r$. If $g(x)$ has a polynomial growth then such a solution does exist and $\nu(x, t)$ has the following representation:

$$\nu(x, t) = \int_0^\infty \beta(y, T-t, x) g(y) e^{-r(T-t)} dy,$$

where $\beta(y, t, x) = \frac{1}{y\sigma\sqrt{2\pi t}} \exp\left\{-\frac{\ln y - \ln x - (r - \frac{\sigma^2}{2})t}{2\sigma^2 t}\right\}$ is a log-normal distribution density function.

Let us consider the non-arbitrage principle for *incomplete markets*. There is a set $M(S/B)$ of martingale measures and a contingent claim f_T such that can not be *replicated* by any self-financing strategy. Consider a richer set of portfolios $\{(\pi, C) : \pi = (\beta, \gamma), C \text{ is an increasing process}\}$ called *portfolios* (π) with *consumption* (C) .

For such a portfolio (π, C) the corresponding capital has a form:

$$\frac{X_t^{\pi, C}}{B_t} = \frac{X_0^{\pi, C}}{B_0} + \int_0^t \gamma_u d\left(\frac{S}{B}\right)_u - \int_0^t B_u^{-1} dC_u.$$

It is a supermartingale with respect to any $P^* \in M(S/B)$. According to the *optional decomposition* (see Kramkov, 1996, Melnikov et al, 2001) any positive supermartingale can be represented in the above form.

Consider the following supermartingale $\frac{V_t^*}{B_t} = \operatorname{ess\,sup}_{P^* \in M(S/B)} E^* [B_T^{-1} f_T | F_t]$ (*Snell envelope*). Obviously, $\frac{V_T^*}{B_T} = \frac{f_T}{B_T}$ and, therefore, $V_T^* = f_T$ (a.s.). Let us denote by $V_t = X_t^{\pi, C}$ – the capital of a hedging strategy (π, C) . Since V_t is a supermartingale with respect to any $P^* \in M(S/B)$ we have

$$\frac{V_t}{B_t} \geq E^* \left(\frac{V_T}{B_T} | F_t \right) \geq E^* \left(\frac{f_T}{B_T} \right) \text{ (a.s.), for } t > 0.$$

Therefore, the process V_t^* is the minimal one in the above class of capital processes and according to the optional decomposition there exists a *minimal hedge* (π^*, C^*) such that V_t^* is its capital process. Thus, in the case of incomplete markets the “natural” price of f_T is equal to $\operatorname{ess\,sup}_{P^* \in M(S/B)} E^* [B_T^{-1} f_T | F_t] B_t$. This approach is called *superhedging*.

Below we show this methodology implemented for quite a representative incomplete market, so-called (B, S) -market with *stochastic volatility* (see Volkov and Kramkov, 1997, Melnikov et al, 2001):

$$dB_t = B_t r dt,$$

$$dS_t = S_t (\mu dt + \Sigma_t dw_t),$$

where r is a non-negative constant, $\Sigma_t^2 = \sigma^2 + (-1)^{N_t} \Delta\sigma^2$, $0 \leq \Delta\sigma^2 < \sigma^2$, N_t is a Poisson process with intensity $\lambda > 0$. According to the Kolmogorov-Ito formula:

$$d\left(\frac{S}{B}\right)_t = \left(\frac{S}{B}\right)_t \Sigma_t dw_t^*,$$

where $dw_t^* = dw_t + \frac{\mu - r}{\Sigma_t} dt$.

It is clear that a measure $\tilde{P} \in M(S/B)$ iff the process w^* is a (local) martingale with respect to \tilde{P} . The set $M(S/B) \neq \emptyset$ because by the virtue of the Girsanov theorem there exist at least two martingale measures defined as

$$\tilde{Z}_T^\pm = \exp \left\{ \int_0^T \left(\frac{r - \mu}{\sqrt{\sigma^2 \pm \Delta\sigma^2}} dw_u - \frac{(r - \mu)^2}{2(\sigma^2 \pm \Delta\sigma^2)} du \right) \right\}$$

and, thus, the market is incomplete.

Denote, $\sigma_{\min} = \sqrt{\sigma^2 - \Delta\sigma^2}$ and $\sigma_{\max} = \sqrt{\sigma^2 + \Delta\sigma^2}$. We can represent Σ_t in the following way:

$$d\Sigma_t = (\sigma_{\max} - \sigma_{\min})(I_{\{\Sigma_{t-} = \sigma_{\min}\}} - I_{\{\Sigma_{t-} = \sigma_{\max}\}}) dN_t$$

and $\Sigma_{t-} = \sigma_{\max}$. Let us represent the capital $\widehat{V}_t = \widehat{v}(S_t, t)$ of the minimal hedge $\widehat{\gamma}_t = \frac{\partial \widehat{v}}{\partial x}(S_t, t)$ as

$$\widehat{v}(x, t) = e^{-r(T-t)} \sup_{\alpha} E g(S_{T-t}^{(\alpha)}).$$

Here $S^{(\alpha)}$ is a controlled diffusion process which is a strong solution of the following stochastic differential equation

$$dS_u^{(\alpha)} = S_u^{(\alpha)}(r du + \alpha_u dw_u), S_0^{(\alpha)} = x,$$

where the ‘‘control’’ parameter $\alpha = (\alpha_t)_{t \leq T}$ takes values in the set $\{\sigma_{\min}, \sigma_{\max}\}$.

From the general theory of controlled diffusion processes (see Krylov, 1977) it follows that $\widehat{v} \in C^{2,1}$ and satisfies the *Bellmann equation*:

$$\begin{aligned} \frac{\partial \widehat{v}}{\partial t} + L^0 \widehat{v} - r \widehat{v} + \frac{1}{2} x^2 \left| \frac{\partial^2 \widehat{v}}{\partial x^2} \right| \Delta \sigma^2 &= 0, \\ \widehat{v}(x, T) &= g(x). \end{aligned}$$

To solve this equation let us apply a *small parameter approach* denoting by $V^*(\sigma^2, \Delta\sigma^2)$ the capital of the minimal hedge in this stochastic volatility model and $V^*(\sigma^2, 0)$ the capital in the corresponding Black-Scholes-Merton model. We have

$$V_t^*(\sigma^2, \Delta\sigma^2) \simeq V_t^*(\sigma^2, 0) + \left(\frac{\partial V_t^*}{\partial \Delta\sigma^2} \right) \Big|_{(\sigma^2, 0)} \Delta\sigma^2 = V^{(0)} + V^{(1)} \Delta\sigma^2.$$

Thus, \widehat{V} can be written in the form

$$\widehat{V} = \widehat{V}^{(0)} + \widehat{V}^{(1)} \Delta\sigma^2$$

and, therefore, the Bellmann equation is reduced to

$$L\widehat{V}^{(0)} + L\widehat{V}^{(1)} \Delta\sigma^2 + \frac{1}{2} \left| \frac{\partial^2 \widehat{V}^{(0)}}{\partial s^2} + \frac{\partial^2 \widehat{V}^{(1)}}{\partial s^2} \Delta\sigma^2 \right| \Delta\sigma^2 = 0$$

and

$$L\widehat{V}^{(0)} + L\widehat{V}^{(1)} \Delta\sigma^2 + \frac{1}{2} \left| \frac{\partial^2 \widehat{V}^{(0)}}{\partial s^2} \right| \Delta\sigma^2 + \frac{1}{2} \left| \frac{\partial^2 \widehat{V}^{(1)}}{\partial s^2} \right| (\Delta\sigma^2)^2 = 0,$$

where $L = \frac{\partial}{\partial t} + L^0 - r$.

Taking into account that $\widehat{V}^{(0)}$ is the capital of the minimal hedge in the Black-Scholes model we arrive to the following equation

$$L\widehat{V}^{(1)} \Delta\sigma^2 + \frac{1}{2} \left| \frac{\partial^2 \widehat{V}^{(0)}}{\partial s^2} \right| \Delta\sigma^2 = 0$$

(up to terms of the $\Delta\sigma^2$ order).

Therefore, the initial problem can be rewritten:

$$L\widehat{V}^{(1)}(x, u) = h(x, u)$$

$$\widehat{V}^{(1)}(x, T) = 0,$$

where $h(x, u) = -\frac{1}{2} \left| \frac{\partial^2 \widehat{V}^{(0)}}{\partial s^2} \right|$.

Applying the Kolmogorov-Ito formula we get the following *upper boundary* of the non-arbitrage price interval:

$$C^*(\sigma, \Delta\sigma^2) = C(\sigma) + \left\{ \sup_{Q \in \mathcal{M}(S/B)} E_Q \int_0^T \frac{1}{2} \left| \frac{\partial^2 \widehat{V}^{(0)}}{\partial s^2} du \right| \right\} \Delta\sigma^2,$$

where $C(\sigma)$ is the corresponding fair price in the Black-Scholes model with volatility σ .

There are pricing methods other than perfect hedging. We are going to mention two of them *mean-variance* and *quantile hedging approaches*. The first one suggests to use a strategy π^* with the terminal capital $X_T^{\pi^*}$ optimal in terms of mean variance (see Follmer, Sondermann, 1986 and Schweizer, 1996 for details):

$$E \left[(f_T - X_T^{\pi^*})^2 \right] \leq E \left[(f_T - X_T^\pi)^2 \right]$$

for any strategy π with its terminal capital X_T^π . The second one is based on hedging contingent claims with a probability $q < 1$. There are two problems of quantile hedging: minimizing the initial capital of the minimal hedge given the fixed probability level q and maximizing q given the initial capital value.

Now let us discuss a relationship between complete and incomplete markets (see Melnikov et al, 2000).

Let us consider the difference

$$\Delta = \operatorname{ess\,sup}_{\tilde{P} \in \mathcal{M}(S/B)} \tilde{E} B_T^{-1} f_T - \operatorname{ess\,inf}_{\tilde{P} \in \mathcal{M}(S/B)} \tilde{E} B_T^{-1} f_T.$$

We shall interpret Δ as a natural measure of incompleteness of the market ($\Delta = 0$ for any complete market). This is so-called *spread*. Let us consider a few more characteristics related to completeness or incompleteness: *leasing* and *transaction costs*. It's typical to consider such leasing $l_t S_t$ and transaction $\delta_t |\Delta \gamma_t| S_t$ costs, where S_t is a stock price, l_t and δ_t are *leasing* and *transaction costs coefficients*.

Involving new financial derivative securities makes the initial market "more complete" with smaller Δ, l and δ .

Above we have described the long term hedging and investment methods. Financial mathematics also suggests methods applicable to short term trading. Most short term trading systems are based on heuristic methods of technical analysis (see Granville, 1976). We are going to show how methods of stochastic financial mathematics can be applied for trading.

Let us consider a model of the financial market that starts trending at the moment $t = \theta$. It can be represented as the observed process of the following form (see Shiryaev, 2000):

$$X_t = r(t - \theta)^+ + \sigma B_t,$$

i.e.

$$dX_t = \begin{cases} \sigma dB_t, & t < \theta, \\ r dt + \sigma dB_t, & t \geq \theta, \end{cases}$$

Let us also assume that θ takes values in $[0, \infty)$ and has the following distribution:

$$P(\theta = 0) = \pi,$$

where $\pi \in [0, 1]$, and

$$P(\theta \geq t | \theta > 0) = e^{-\lambda t},$$

where $\lambda > 0$ is a known constant.

Denote,

$$\pi_t = P(\theta \leq t | F_t^X),$$

where $F_t^X = \sigma(X_s, s \leq t)$.

We are going to consider the problem of determining the stopping time τ^* optimal in the following sense: For any given $c > 0$, find

$$B(c, \pi) = \inf_{\tau} \{ P_\pi(\tau < \theta) + c E_\pi(\tau - \theta)^+ \} = P(\tau^* < \theta) + c E(\tau^* - \theta)^+.$$

According to the general theory developed for solving the well-known “quickest detection” problem the optimal stopping time can be calculated as follows:

Denote,

$$G(y) = \log \frac{y}{1-y} - \frac{1}{y},$$

$$\Lambda = \lambda / \frac{r^2}{2\sigma^2}, C = c / \frac{r^2}{2\sigma^2}.$$

Let us also define B_* as the root of the equation:

$$C \int_0^{B_*} e^{-\Lambda[G(B_*)-G(y)]} \frac{dy}{y(1-y)^2} = 1.$$

Then the time τ^* can be calculated as

$$\tau^* = \tau^*(B_*) = \inf\{t : \pi_t \geq B_*\}$$

is optimal for any $0 \leq \pi \leq 1$ and

$$B(c, \pi) = P_\pi(\tau^* < \theta) + c E_\pi(\tau^* - \theta)^+.$$

Here we have rigorously stated and solved the problem of detecting trends in financial markets which is probably the most challenging and important problem of technical analysis. Thus, rigorous quantitative methods of financial mathematics are replacing old qualitative heuristic methods of classical technical analysis.

Bibliography

- [1] Bachelier L. (1900), Theorie de la speculations, Ann. De l'Ecole Normale Superieure, Vol. 3, Gauthier-Vellars, Paris.
- [2] Black F., Scholes M. (1973), The pricing of options and corporate liabilities, J. Political Economy, 81, 637-659.
- [3] Cavanna H., Ed. 1992, Financial Innovations, Routledge, London & N.Y.
- [4] Cox J., Ross S., Rubinstein M. (1979), Option Pricing: A Simplified Approach, J. Fin. Economics, 7, 229-263.
- [5] Cummins J.D., Geman H. (1995), Pricing catastrophe futures and call spreads: an arbitrage approach, J. Fixed Income.
- [6] Delbaen F., Haezendock J. (1989), A martingale approach to premium calculation principles in an arbitrage-free market, Insurance: Math. and Economics, 8, 265-277.
- [7] Duffie D. (1996), Dynamic Asset Pricing Theory, Princeton Univ. Press, N.J.
- [8] Folmer H., Sondermann D. (1986) Hedging of non-redundant contingent claims, Contributions to Math. Economics, Ed. Hildebrandt and Mas-Colell, 205-223.
- [9] Granville J.E. (1976) New Strategy of daily stock market for maximum profit, Prentice-Hall, N.J.
- [10] Guillaume D.V., Dacorogna M.M., Dave R.R., Müller U.A., Olsen K.B., Pictet O.V. (1997), From the birds eye to the microscope: A survey of new stylized facts of the intra- daily foreign exchange markets, Finance and Stochastics, 1, 95-129.
- [11] Harrison M., Kreps D. (1979), Martingales and Arbitrage in Multiperiod securities Markets, J. Econom. Theory, 20,381-408.
- [12] Harrison M., Pliska S. (1981), Martingales and stochastic integrals in the theory of continuous trading, Stoch. Proc. Appl., 11, 215-260.
- [13] Kramkov D.O. (1996), Optional decomposition of supermartingales and hedging contingent claims in incomplete security markets, Probab. Theory and Related Fields, 105, 459-479.
- [14] Krylov N.V. (1977), Controlled diffusion type processes, Nauka, Moscow.
- [15] Lintner J. (1965), The valuation of risk assets and the selection of risky investment in stock portfolios and capital budgets, Rev. Econ. Stat, 47, 13-37.
- [16] Markowitz H. (1959), Portfolio selection: Efficient diversification of investment, Wiley, N.Y.
- [17] Melnikov A.V. (1995), On Stochastic Analysis in the Modern Mathematics of Finance and Insurance, Survey Appl. and Industrial Math., 2, 514-526 (in russian).

- [18] Melnikov A.V. (1999), On innovation and risk aspects of financial system evolution, *Questions of Risk Analysis*, 1, 1, p. 22-27 (in russian).
- [19] Melnikov A.V. (1999), *Financial System: Innovations and Pricing of Risks*, University Karlsruhe, Tech. Report 5, 21 p.
- [20] Melnikov A.V., Nechaev M.L., Volkov S.N. (2001), *Mathematics of Contingent Claims*, Higher School of Economics, Publishing House, Moscow, 260 p.
- [21] Melnikov A.V., Molyboga M.M. (2002), *Pricing of Equity-Linked Life Insurance Contracts*, *Russian Math Surveys*, 30 p. (accepted for publication).
- [22] Melnikov A.V., Volkov S.N., Nechaev M.L. (2000), *Mathematics of Contingent Claims*, Higher School of Economics Publishing House, Moscow.
- [23] Merton R.C. (1973), Theory of rational option pricing, *Bell J. Of Economics and Management Sci.*, 4, 141-183.
- [24] Merton R.C. (1990), *Continuous time Finance*, Basil Blackwell, Oxford.
- [25] Merton R.C. (1995), Influence of Mathematical Models in Finance on Practice: Past, Present and Future, *Math. Models in Finance*, Ed. Howison et al, Chapman & Hall, London, 1-12.
- [26] Schweizer M. (1996), Approximation Pricing and Variance-Optimal Martingale Measure, *Ann. Probab.*, 24, 1, 206-236.
- [27] Sharpe W.F. (1970), *Portfolio Theory and Capital Markets*, McGraw-Hill.
- [28] Shiryaev A.N. (1998), *Essentials in Stochastic Finance*, World Scientific.
- [29] Shiryaev A.N. (2000), Quickest Detection Problems in the Technical Analysis of the Financial Data, *Proceedings of Congress of Bachelier Finance Society*.
- [30] Shiryaev A.N. (2001), *Essentials of Stochastic Finance: Facts, Models, Theory*, World Scientific, 834 p.
- [31] Stricker C.(1990), Arbitrage et Lois de Martingale, *Ann. De l'Institute H. Poincare*, 26, 451-460.
- [32] Volkov S.N., Kramkov D.O. (1997), On hedging methodology for options, *Survey Applied and Industrial Math.*, 4, 1, 18-65 (in Russian).

Chapter 7

Resistivity Well Logging

Participants: Yongji Tan (Mentor), Maria Inez C. Goncalves, Jeff P. Grossman, Ying Han, Gerald H.W. Lim, Zheng Meng, Benjamin W.L. Ong.

PROBLEM STATEMENT: In petroleum industry resistivity well-logging is one important method to detect the resistivity of the formation. This method can be described as follows. After a well has been drilled, one puts a log-tool into the well. The log-tool is an insulation rod whose lateral surface is covered by metal as electrodes. While it works, some of the electrodes discharge currents of fixed intensity into the formation then the potentials in some other electrodes are measured. Raising the log-tool along the well-bore one gets the potential distribution on the electrodes of various positions. By this information one infers the resistivity of the geological formation. Combining this information with the porosity of the formation obtained by some other well-logging techniques, one can calculate the oil storage.

7.1 Introduction

In petroleum exploration, resistivity well-logging is performed in order to estimate oil saturation levels in porous media. The method employs a log tool consisting of 3 discharging and 20 measuring electrodes, which is lowered into a drilled well. Current is supplied by the discharging electrodes to generate a static electric field, and the resulting electric potential is sampled at the measuring electrodes. The goal is to infer the conductivity of the

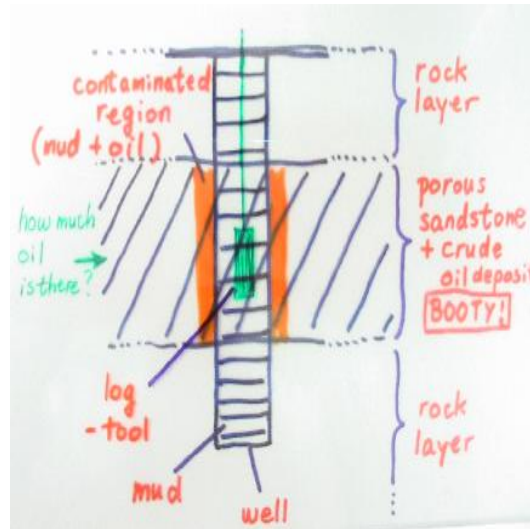


Figure 7.1: domain idealization

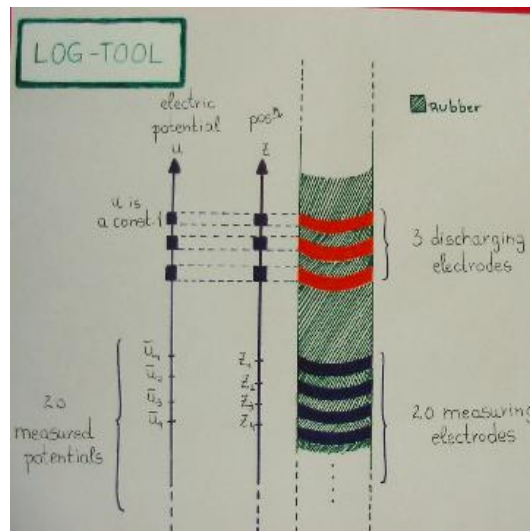


Figure 7.2: log tool

domain; this is achieved through an iterative scheme. There are two methods to solve the forward problem (i.e., find the potential given conductivities and boundary conditions): imposing the total boundary flux condition and re-expressing the problem in terms of classical PDEs.

7.2 Mathematical Model

7.2.1 Problem Formulation

Given

1. The governing Laplace equation of the electric potential u

$$\nabla \cdot (k(\vec{x})\nabla u(\vec{x})) = 0 \quad (7.1)$$

2. The current at the discharging electrodes,

$$\int \int_{A_i} k \frac{\partial u}{\partial \vec{n}} dS = I_i \text{ (known constant) ,} \quad (7.2)$$

where A_i stands for the surface of the i -th electrode, and I_i is the current discharged by it.

3. The electric potentials at the measuring electrodes,

$$\bar{u}_j, \quad j = 1, \dots, J \quad (7.3)$$

where J is the number of measuring electrodes.

We wish to find the conductivity $k(\vec{x})$. Unfortunately, the problem as posed is impossible to solve.

7.2.2 Simplification

We discretize our domain into several sub-regions, and assume that k is piece-wise constant in each sub-region. A natural discretization allows for a top rock layer, a bottom rock layer, an oil-saturated sandstone layer, a mud filled well bore and a contaminated region within the saturated layer (see figure 7.1). We note that although this discretization may appear somewhat simplistic, it nevertheless represents the essential geophysical features in a reasonable way. One can always refine the discretization to P subdomains provided $P \leq J$ (the number of measuring electrodes). In addition, we can infer the following physical boundary conditions.

7.2.3 Physical Boundary Conditions

- On the rubber insulating surface, $\frac{\partial u}{\partial \vec{n}} = 0$
- On the measuring electrodes

$$I = 0 \implies \int \int_{M_i} k_1 \frac{\partial u}{\partial \vec{n}} dS = I = 0$$

or equivalently, $\frac{\partial u}{\partial \vec{n}} = 0$.

- For far fields, $u \approx 0$, since $u \rightarrow 0$ as distance $\rightarrow \infty$
- By symmetry, $\frac{\partial u}{\partial \vec{n}} = 0$ on the cylindrical axis.

7.2.4 Interface Conditions

To obey the laws of physics, we require

- Continuity of electric potential $u|_- = u|_+$.
- Continuity of current, $k_- \frac{\partial u}{\partial \vec{n}} \Big|_- = k_+ \frac{\partial u}{\partial \vec{n}} \Big|_+$

across all interfaces.

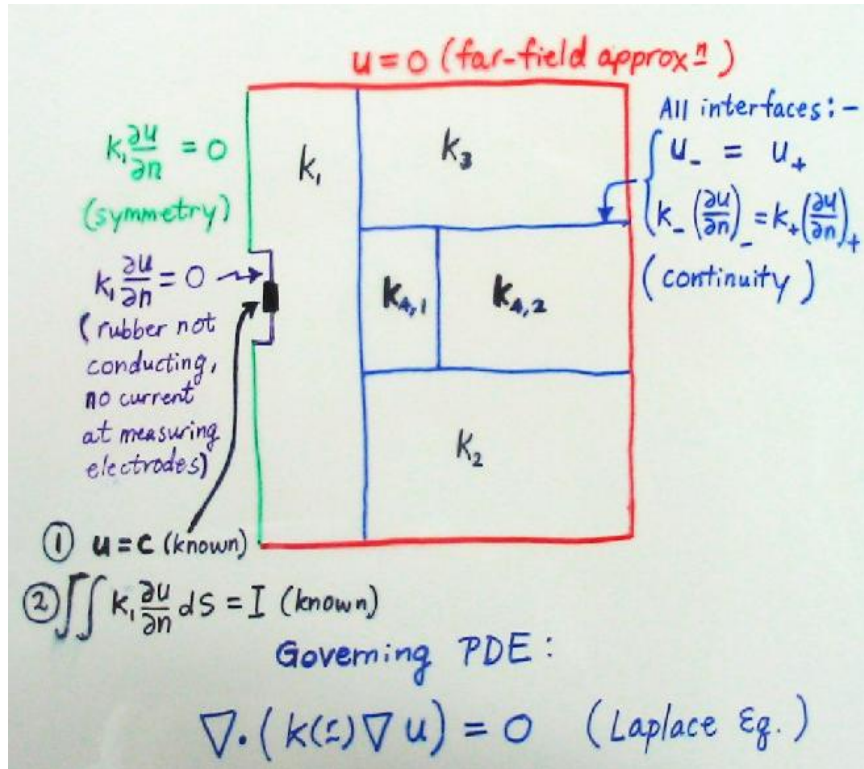


Figure 7.3: Numerical Domain

7.2.5 The Inverse Problem

This inverse problem cannot be solved in a straightforward manner. We adopt an iterative approach that enforces (7.2.3) by minimizing

$$\epsilon = \sum_{j=1}^J |u^i(z_j) - \bar{u}_j|, \quad (7.4)$$

so that the iterative solution at the measuring electrodes, $u^i(z_j)$, converge to \bar{u}_j . The outline of the iterative algorithm is as follows:

Algorithm:

1. Guess an initial set of conductivities, $\vec{k}^i = [k_1^i, k_2^i, k_3^i, k_{4,1}^i]$, and $k_{4,2}^i$ for $i = 0$.
2. Solve (7.1) for u^i excluding condition (7.3).
3. Calculate ϵ using (7.4)
4. If $\epsilon < \bar{\epsilon}$, where $\bar{\epsilon}$ is a user-defined tolerance, the \vec{k}^i is accepted; otherwise, it is modified to $\vec{k}^{i+1} = f(\vec{k}^i)$ by a simplex search algorithm in Matlab.
5. Return to step 2, using \vec{k}^{i+1} , and repeat steps 2 to 4 until the condition $\epsilon < \bar{\epsilon}$ is satisfied.

7.3 Variational Principle and Finite Element Method

7.3.1 Formulation

The resistivity well-logging problem can be re-expressed by incorporating (7.2) (the total flux boundary condition) in a variational formulation. This is equivalent to finding the minimum of the energy functional:

$$J(v) = \frac{1}{2} \int \int \int_{\Omega} k |\nabla v|^2 dx dy dz - \sum_{i=1}^L I_i v|_{\Gamma_3^i}, \quad (7.5)$$

where L is the number of discharging electrodes and the minimum is taken over $V_0 = \{v = v(\vec{x}) | v \in H^1(\Omega), v|_{\Gamma_1} = 0, v|_{\Gamma_3^i} = \text{const. } (i = 1, \dots, L)\}$, i.e., we wish to find $u \in V_0$ such that $J(u) = \min_{v \in V_0} J(v)$.

An equivalent formulation using the variational principle is to find $u \in V_0$ such that

$$\int \int \int_{\Omega} k \nabla u \cdot \nabla \Phi dx dy dz = \sum_{i=1}^L I_i \Phi|_{\Gamma_3^i}, \quad \forall \Phi \in V_0. \quad (7.6)$$

7.3.2 Implementation

Using the finite element method, the natural boundary conditions are automatically satisfied. Nevertheless, some special treatment of the total flux boundary condition is required. Without loss of generality, we simplify our problem as follows:

- We consider the $50\text{m} \times 50\text{m}$ cross section Ω_4 split into two sub-regions with piecewise constant conductivities k_1 and k_2 .
- Reduce problem to one discharging electrode and eight measuring electrodes.

$J(u)$ simplifies to

$$J(u) = \frac{1}{2} \sum_{i=1}^L \int \int_{\Omega_4^i} k_i |\nabla v|^2 dr dz - \frac{I}{2\pi} u|_{\Gamma_3}. \quad (7.7)$$

We can now form the stiffness matrix and treat the equi-valued boundary value condition explicitly. We use linear triangular elements based on the perpendicular network. The mesh is denser near the discharging electrode because the electric potential changes rapidly there..

In our numerical experiment, we set $I = 1$ on the discharging electrode, set $\vec{k} = (k_1, k_2) = (5, 0.05)$ and we computed the potential u over Ω_4 . The solution is plotted on the following page.

Observe that the potential u decreases rapidly from the tool to the far field boundaries. Also note the effect of current continuity at $r \approx 2$

7.3.3 Inverse Problem

Suppose we take eight potentials on the log tool (from the above computation) as measured data. Given another two resistivities k_1^0 and k_2^0 as initial values, can we recover $\vec{k} = (5, 0.05)$? Yes! Our final solution converges to our given data. The deviation ϵ is $o(10^{-12})$, and we recover $\vec{k} = [5.00003424142100, 0.04999434491546]$.

7.3.4 Stability

We tested the stability of our method by introducing 0.5% random error noise into our measured data. We find that the conductivities now converge to $\vec{k}' = [3.35348, 0.03354]$ with $\epsilon = o(10^{-10})$. This indicates that our method is not stable, and that some regularization of the parameters or some added weights is required in our future work, e.g., replacing (7.4) by

$$\epsilon = \min_{k_1, k_2} \sum_{i=1}^8 k_i (u - \bar{u}_i)^2 \quad (7.8)$$

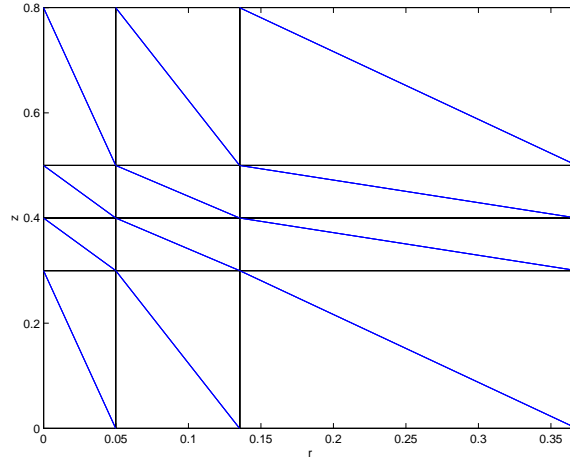


Figure 7.4: mesh

7.4 Classical Method

7.4.1 Formulation

We can also reformulate our boundary conditions to a more familiar form: Dirichlet and Neumann boundary conditions, namely

$$\text{replace } \int \int_{\Gamma_3^i} k_1 \frac{\partial u}{\partial \vec{n}} = I_i \text{ by } u|_{\Gamma_3^i} = 1 \quad (7.9)$$

In fact, through linearity of (7.1), if we just solve the PDE in section 7.2 with (7.2) replaced by

$$u|_{\Gamma_3^j} = 1, u|_{\Gamma_3^k} = 0, (j \neq k).$$

Our solution is then a linear combination

$$u = \sum_{j=1}^L \beta_j u_j$$

We can then solve for β_j by solving a linear system of equations. For example, the case with 3 discharging electrodes results in

$$\begin{aligned} \int \int_{\Gamma_3^1} \beta_1 k_1 \frac{\partial u_1}{\partial \vec{n}} + \int \int_{\Gamma_3^1} \beta_2 k_1 \frac{\partial u_2}{\partial \vec{n}} + \int \int_{\Gamma_3^1} \beta_3 k_1 \frac{\partial u_3}{\partial \vec{n}} &= I_1 \\ \int \int_{\Gamma_3^2} \beta_1 k_1 \frac{\partial u_1}{\partial \vec{n}} + \int \int_{\Gamma_3^2} \beta_2 k_1 \frac{\partial u_2}{\partial \vec{n}} + \int \int_{\Gamma_3^2} \beta_3 k_1 \frac{\partial u_3}{\partial \vec{n}} &= I_2 \\ \int \int_{\Gamma_3^3} \beta_1 k_1 \frac{\partial u_1}{\partial \vec{n}} + \int \int_{\Gamma_3^3} \beta_2 k_1 \frac{\partial u_2}{\partial \vec{n}} + \int \int_{\Gamma_3^3} \beta_3 k_1 \frac{\partial u_3}{\partial \vec{n}} &= I_3 \end{aligned} \quad (7.10)$$

7.4.2 Implementation

We used a finite element package (FEMLAB) to solve the problem posed in section 7.2.2. Only 4 regions are visible in the following plot because the computation was done to scale.

- the depth of the well is 100m

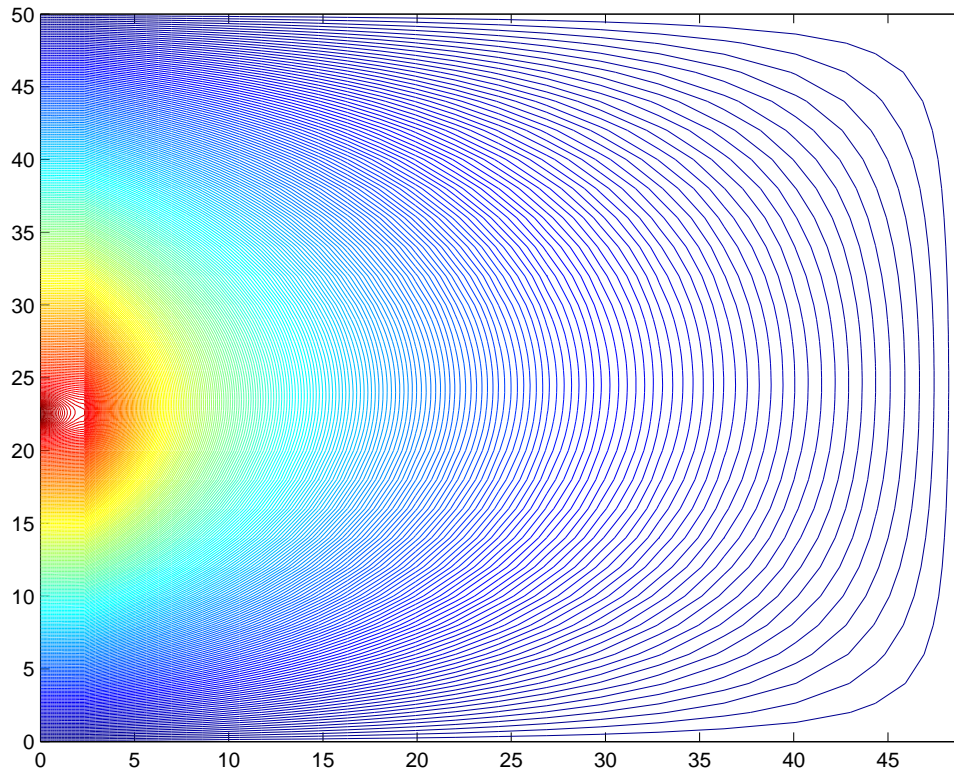


Figure 7.5: Electric Potential with $k_1 = 5$, $k_2 = 0.05$

- the electric field affects a radius of approximately 50 meters radially,
- the drill has a 10 cm radius resulting in a mud region with a 10cm radius
- the tool has a length of 8 meters
- the single discharging electrode has a height of 0.5 meters.
- the contaminated region is approximately 1 meter wide

When $I = 0.645$ and $\vec{k} = [k_1, k_2, k_3, k_{41}, k_{42}] = [10, 1, 1, 0.5, 0.01]$,

Notice that the same features are present in the more complex subdomain structure—the continuity of current provides a jump in $\frac{\partial u}{\partial \vec{n}}$.

A more detailed plot of the mesh is shown in figure 7.4.

7.4.3 Inverse Problem

As in section 7.3, the search direction for \vec{k} was computed using *fminsearch*, a routine found in the Matlab optimization toolbox. We had problems converging to the correct solution for the 5 parameter problem, $\vec{k} = [k_1, k_2, k_3, k_{41}, k_{42}]$. It appeared that there were local minimums in this problem. The problem did converge however when k_1, k_2 & k_3 was kept fixed, and $\vec{k} = [k_{41}, k_{42}]$ was optimized.

Finding a good optimization routine is of course the key to this problem. Due to time constraints, only *fminsearch* was attempted. *fminsearch* took some undesirable steps: the conductivities became negative, which is not physically possible in this problem. Given time, one could probably code a more suitable optimization routine which involves trust regions and appropriate constraints.

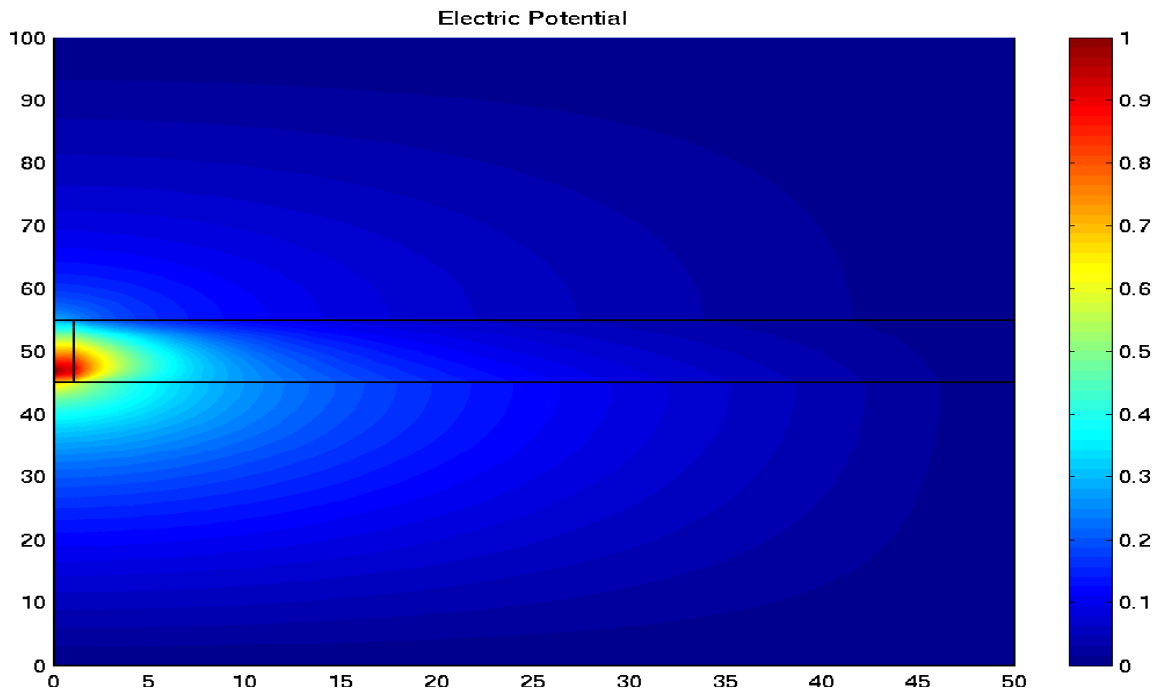


Figure 7.6:

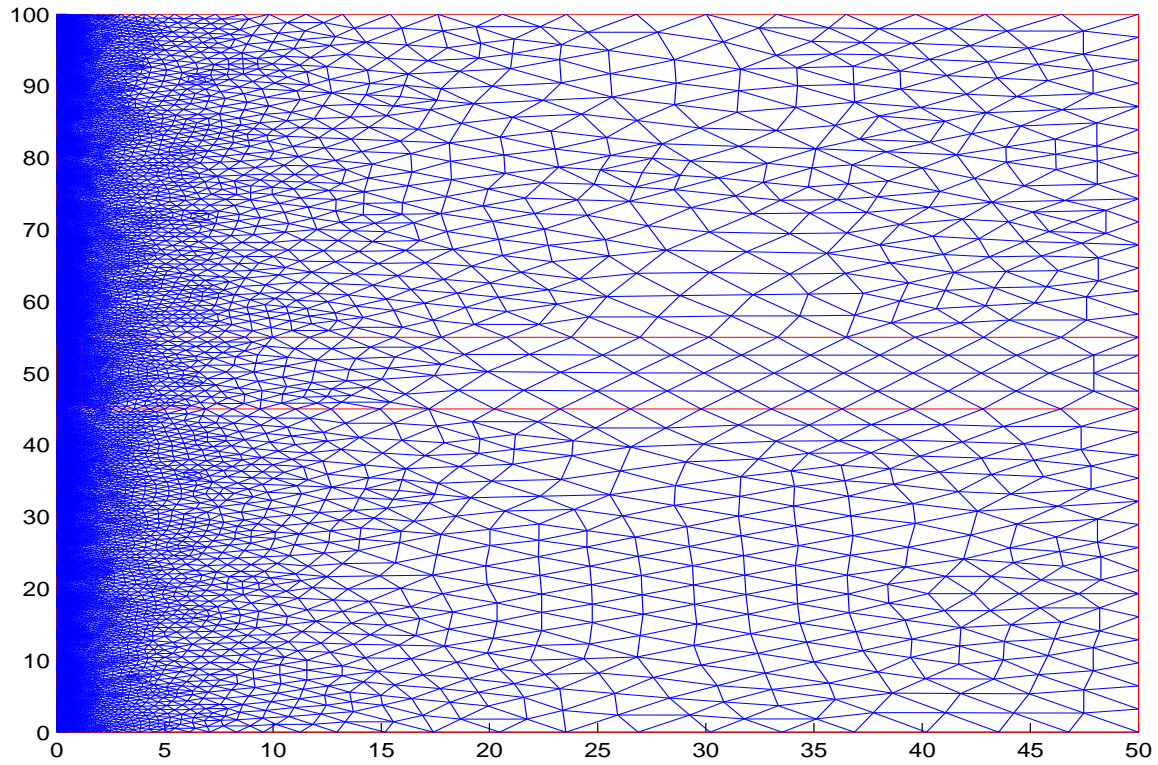


Figure 7.7: mesh

7.5 Conclusions

Given the time restrictions of a modelling camp, we were quite pleased with the results that we obtained. We think that our formulation of the problem correctly models the problem presented, and our finite element solvers correctly solved the Laplace equation with boundary conditions.

List of Participants

Organising Committee

Chris Bose	University of Victoria
Huaxiong Huang	York University
Randy LeVeque	University of Washington
Jack Macki	University of Alberta
Mark Paulhus	University of Calgary
Manfred Trummer	Simon Fraser University
Ian Frigaard	University of British Columbia

Mentors

Petra Berenbrick	Simon Fraser University
Chris Budd	Royal Institution of Great Britain
Alexander Melnikov	University of Alberta
Tim Myers	University of Capetown, S.A.
Brett Stevens	Carelton University
Yongji Tan	Fudan University
Brian Wetton	University of British Columbia

Students

Leslie Fairbairn	Simon Fraser University
Jeffrey Gilmore	Simon Fraser University
Qingguo Li	Simon Fraser University
Gerald H. W. Lim	Simon Fraser University
Colin Macdonald	Simon Fraser University
Benjamin W. L. Ong	Simon Fraser University
Nicolas Robidoux	Simon Fraser University & University of New Mexico
Ibrahim Agyemang	University of Alberta
Elaine Beltaos	University of Alberta
Tingting Shu	University of Alberta
Victoria Skornyakova	University of Alberta
Huamei Yin	University of Alberta
Neville Dubash	University of British Columbia
Maria Landry	University of British Columbia
Anita Parkinson	University of British Columbia
Jeff P. Grossman	University of Calgary
Hammouda Ben Mekki	University of Calgary

Lorraine Dame	University of Victoria
Maria Inez C. Goncalves	University of Victoria
Adam Webber	University of Victoria
Alicia Cantón	University of Washington
Ana Granados	University of Washington
Qingze Zou	University of Washington
Mélanie Beck	McGill University
Isabelle Déchène	McGill University
Olivier Dubois	McGill University
Charles Fortin	McGill University
Ying Han	McGill University
Yuanyuan Hua	McGill University
Robin Swain	Memorial University of Newfoundland
Marat Molyboga	Moscow State University
Yuri Petratchenko	Moscow State University
Lyudmyla Barannyk	New Jersey Institute of Technology
Tetyana Segin	New Jersey Institute of Technology
Carlos Trenado	New Mexico State University
Zheng Meng	Rice University
Bo Peng	Rice University
Dhavid Aruliah	University of Toronto
Raymond Ka Wai Cheng	University of Toronto
Stephan Lawi	University of Toronto
Matthias Mück	University of Toronto
Tereza Neocleous	University of Illinois
Bidhan Roy	University of Illinois
Fabien Youbissi	Universite Laval
Erik Andries	University of New Mexico
Jingxiang Luo	University of Saskatchewan
Tzvetalin Vassilev	University of Saskatchewan
Heather Lehr	University of Texas at Austin
Anuj Mubayi	University of Texas at Arlington
Andrei Badescu	University of Western Ontario
Mikalai Birukou	University of Winnipeg
Katherine Hegewisch	Washington State University, Pullman
Asrat Fikre Gashaw	York University

PIMS Contact Information

email: pims@pims.math.ca

<http://www.pims.math.ca>

- **Director: N. Ghoussoub**
Phone: 604-822-3922
Fax: 604-822-0883
email: director@pims.math.ca
- **SFU-Site Director: M. Trummer**
email: sfu@pims.math.ca
- **UAlberta-Site Director: J. Muldowney**
email: ua@pims.math.ca
- **UBC-Site Director: D. Rolfsen**
email: ubc@pims.math.ca
- **UCalgary-Site Director: G. Margrave**
email: uc@pims.math.ca
- **UVic-Site Director: F. Diacu**
email: uvic@pims.math.ca
- **UWashington-Site Director: S. P. Smith**
email: uw@pims.math.ca