Pacific Institute for the Mathematical Sciences

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

Proceedings of the fourth

PIMS Graduate Industrial Math Modelling Camp

June 11–15, 2001, University of Victoria

Cosponsored by:

The Natural Science and Engineering Research Council of Canada

and

The British Columbia Information, Science and Technology Agency

Editor: Chris Bose, University of Victoria

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 pursing 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 Fourth GIMMC was held at the University of Victoria, June 11–15, 2001. There were eight problems posed, a record, with a total of 56 students in attendance, another record. The students mainly came from all across North America with 16 from the United States. They were selected from over 130 applicants.

My sincere appreciation and gratitude goes to everyone involved in this workshop, in particular I wish to thank Chris Bose, the editor of these proceedings, the other organisers (Randy LeVeque, Huaxiong Huang, Mark Paulhus, Keith Promislow, Ian Frigaard) and mentors (Sergei Bespamyatnikh, John Chadam, Ian Frigaard, Lisa Korf, Hedley Morris, Tim Myers, Miro Powojowski, Moshe Rosenfeld). I am greatly looking foreward to the 2002 camp at Simon Fraser University.

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

EDITOR'S PREFACE

From June 11 through June 15, 2001 a record number of 56 graduate students gathered at the University of Victoria for the Fourth Annual PIMS Graduate Industrial Mathematical Modelling Camp – the GIMMC-4. This year marked a significant expansion of the Camp which in previous years had been limited to approximately 40 students chosen mostly from the five PIMS Universities. The expansion was suggested in recognition of the new role that PIMS is now playing on the broader Canadian mathematics scene, and also in view of the recent expansion of PIMS south of the border to include the University of Washington as a sixth founding institution. By February of last year, the organizers of the camp were struggling with the overwhelming response: more than 130 excellent applications from all over the continent, and even some from Europe! After the dust had settled on June 15, looking over our list of participating students we found that we had hosted students from 25 North American Universities. Approximately one-third of our participants were from US institutions and the participation from within Canada had expanded to include many students from central Canada and the maritimes. The Camp, and it's senior sibling, the Industrial Problem Solving Workshop have indeed arrived as premier events on the applied and industrial mathematics calendars throughout North America.

For those not yet familiar with the format of the camps or the industrial workshops let me say a few words about the organization, which was typical. Once the students got settled in, the week began by having each of the academic mentors give a short presentation describing their sample problem to the assembled group. It takes a considerable amount of judgement, skill and effort to come up with good problems for the camp, and this year we had eight excellent problems presented by our outstanding mentors. From Monday afternoon through Thursday evening, the individual workshop groups met under the guidance of the academic mentor. Typical activities during this period included short lectures on background material from the mentors, literature searches, pencil and paper calculations, work at the computers and so on. By Friday morning the groups were ready to present their findings (having elected one student to stay up all Thursday night preparing the group's presentation!) and shortly after I was given the formal writeups, which appear more or less as I received them in the rest of this document. While this may all sound fairly straightforward, in practice it is an extremely intense week for all concerned – students, mentors and organizers. The best way to appreciate this is to look at a few of the chapters which follow, keeping in mind that they represent the work of perhaps seven or eight graduate students having varied mathematical backgrounds, working in groups with a minimal amount of interference from the academic mentor over a period of three and one-half days.

An workshop of this size can only be successful through the effort and skill of numerous personalites both on stage and behind the scenes. First, as the backbone of the Camp, and the principal actors so to speak, let me thank the mentors. They were:

- Sergei Bespamyatnikh (UBC, Watchtower Placement)
- John Chadam (University of Pittsburgh, Portfolio Analysis)
- Ian Frigaard (UBC, Metal Spray Casting)
- Lisa Korf (University of Washington, Web Hosting Agreements)
- Hedley Morris (San Jose State University, Imaging Problem)
- Tim Myers (University of Capetown, S.A., Modelling Ice Accretion)
- Miro Powojowski (Algorithmics Corp., Risk Neutral Measures)
- Moshe Rosenfeld (University of Washington, Control of Streetlight Networks)

Some of these names will be familiar to those who have been following the evolution of the PIMS industrial program. Some were first-timers. All the mentors did outstanding work both leading up to and during the week, and I will never be able to thank them enough for their efforts. This is mitigated only slightly by my suspicion that, in truth, they enjoyed themselves throroughly during the week and they found the students to be a well-prepared, mathematically stimulating and energetic bunch.

As for the stage-hands behind the scenes, let me begin by thanking those in Victoria who anwered my call for help with this event. Pauline van den Driessche and Bill Reed came forward to read over all the student files during the selection process. Administrative and technical matters were, as usual, expertly and cheerfully handled by Kelly Choo, our systems administrator and the PIMS Web Manager, along with Timea Halmai, administrative assistant at the PIMS UVic site office. Ariana Clapton, one of our departmental secretaries, stepped in when the workload became too great for the rest of us combined.

At some point it became clear that we were not going to have enough borrowed computers to do the job. Eugeen Deen and his staff at the Human and Social Development Computer Laboratory bailed us out, providing expertly managed and timely access to all of the machinery and software that is so essential for this sort of event.

Finally, I must thank Marc Paulhus. Marc has been, in one way or another, instrumental in every GIMMC and IPSW I have been involved with and by extrapolation, I suspect with all of them. When things go wrong, and they always do, Marc's wit and unflappable nature make short work of the kind of problem us lesser mortals tend to get bogged down in. Although I was the local organizer for the GIMMC-4, in truth, it was Marc who once again pulled all the strings.

Christopher J. Bose, Editor
Department of Mathematics and Statistics
University of Victoria

Contents

List of Participants **69**

Chapter 1

Locating Watchtowers in Terrains

Participants: Sergei Bespamyatnikh (Mentor), Peter Anderson, Adrian Driga, Leslie Fairbairn, Jacky Li, Tatiana Marquez-Lago, Ling Zhao.

PROBLEM STATEMENT: A problem of current interest to investigators in Computational Geometry is to position a number of vertical watchtowers above a polyhedral surface such that every point on the surface can be seen from the top of some tower. With towers of zero height, the related problem of determining the minimum number of towers which collectively cover the surface by visibility has been shown to be NP-hard. The basic measurement of problem complexity is the number of faces (equivalently, the number of segments) needed to specify the surface.

Among all sets of k towers of finite height which permit every point of the surface to be covered, we seek ones whose tallest tower is as short as possible. It is of some importance in the sequel that the number of towers is fixed in advance. Algorithms are presented to solve several different versions of this problem.

1.1 Introduction

We consider how to minimize the common height of k towers, while still providing unobstructed lines of sight from the tops of the towers to each point on a given polyhedral 'terrain.'

Differently restricted cases of this problem are solved using new polynomial-time algorithms. The situations contemplated involve general numbers k of towers, but only two-dimensional terrains consisting of sequences of non-vertical line segments in the plane joined at the ends. We denote by n the number of segments in the terrain.

The first algorithm produces an approximate optimum tower height accurate to within an arbitrarily small additive constant. The running time estimate is polynomial in n and the reciprocal of this constant.

The restriction for the second algorithm is that the "local visibility" regions of the towers must collectively cover the terrain. Such a region consists of an interval containing the base of its tower, and extends in either direction as far as the first obstruction. Subject to this restriction, we find an exact solution with worst-case running time $O((n\log(n))^k)$.

The third algorithm seeks solutions in which any segment of the terrain is completely visible from some single tower. The algorithm produces an exact optimum relative to this second restriction, and operates in time $O(n^{4k+1}+n^6)$.

A full version of this paper is available at

`http://www.cs.ubc.ca/~besp/towers.ps.gz`

1.2 Approximation Algorithm for 2D k-watchtower problem

Problem (k-watchtower problem). Given a terrain P (polygonal line) and a positive integer k, find the position of k towers T_i that can visually cover the terrain P, and the height of the tallest tower, H^*, has the property

$$H^* = min\{\ max\{\text{height}(T_i) | 1 \leq i \leq k\} \mid T_i, 1 \leq i \leq k, \text{cover } P\}. \quad (1.1)$$

$H^* = H^*(P,k)$ is the height of the optimal solution for the polygonal line P and the integer k (*optimal height* for P).

Theorem 2.1 Let P be a 2D terrain without vertical lines and k be a positive integer. Let H^* be the optimal height for the corresponding k-watchtower problem. Then there is an algorithm such that:

1. $\forall c > 0$, the algorithm solves the k-watchtower problem and finds the approximately optimal height H with $H < H^* + c$;
2. the algorithm has polynomial time complexity in the number of segments of P, $\frac{1}{c}$, and X, the upper bound for the x axis.

Proof. Consider the algorithm in Figure 1.1. Let S be a division of the interval $[0, X]$, with $\delta, \alpha,$ and ϵ computed by the algorithm. Initially, the problem is solved for the terrain P and only one watchtower. Let $H1$ be the optimal height for the single-watchtower problem for P. Clearly, $H1$ is an upper bound for the optimal height of the k-watchtower problem for P.

Let $D = \{0 = h_1 < h_2 < ... < h_m = H1\}$ be a division of stride δ for the interval $[0, H1]$. The algorithm *Approx* finds the smallest point, H, of the division D such that there is a solution for P where the k towers have the height H, and their x-coordinates belong to the division S. The algorithm uses binary search to locate H.

For each point h of the division D considered by the binary search a verification algorithm called *Verify* (Figure 1.2) is used to check if a solution of height h can be found for the terrain P. The towers of this solution must be located at x-coordinates that form a subset of S. This verification algorithm considers all the possible combinations of k distinct x-coordinates from S and checks if the terrain P can be covered visually from the top of the k towers of height h built at the currently considered x-coordinates. The algorithm *Cover* decides if the k towers specified as input cover visually the terrain P. This is done by verifying that each segment of P is visible from the k towers.

```
Algorithm Approx
   Input  P: terrain, k: number of towers, c: positive error
   Output x = (x1, x2, ..., xk): the position of the k solution towers,
          H: the optimal height found.

   delta = c/2
   alpha = the measure of the smallest angle among the acute angles formed by
           the segments of the polygon line with the y axis
   epsilon = (c / 4) * tan( alpha )
   X = largest x co-ordinate of a polyline point (projX(P) = [0, X])
   S = { i * epsilon | i integer, i * epsilon <= X}

   H1 = OneTowerHeight( P ) //upper bound for the solution H
   left = 0
   right = int(H1 / delta)
   while (left <= right)
      mid = int( (left+right)/2 )
      h = mid * delta // current height
      if Verify(P, k, S, h, x) then
         right = mid-1
      else
         left = mid+1
   H = left * delta
```

Figure 1.1: Pseudo-code for the approximation algorithm

The following Lemma is essential for the proof of claim 1.

Lemma 2.1 Let P be a 2D terrain without vertical lines and k a positive integer. Let H^* be the optimal height for the corresponding k-watchtower problem and c a positive error. Then, the verification algorithm *Verify* returns "true" for all the heights h with $h \geq H^* + \frac{c}{2}$.

Proof of Lemma 2.1 Inside the algorithm *Approx* the following quantities are set:

- $\delta = \frac{c}{2}$,
- α = the measure of the smallest angle among the acute angles formed by the segments of the polygonal line with the y axis,
- $\epsilon = \frac{c}{4} \times tan(\alpha)$,
- X = largest x-coordinate of a polygonal line point ($proj_x(P) = [0, X]$),
- $S = \{i \times \epsilon \; | i \text{ integer}, i \times \epsilon \leq X\}$.

Let $h, h \geq H^* + \frac{c}{2}$, be the height that the algorithm *Verify* is verifying. Consider the situation depicted in Figure 1.3. This situation (or a symmetric one) is guaranteed to occur during the execution of *Verify*. The segment uu' is a sub-segment of the polygonal line, uv is an optimal tower for P, and $u'v'$ is a tower of height h built at x_S, the point of the division S closest to x_{opt}. Because the division S has the stride ϵ, then

$$s = \text{abs}(x_{opt} - x_S) < \epsilon. \tag{1.2}$$

The angle between uu' and $u'v'$ is bigger than α (by definition of α); therefore, $\frac{s}{t} = tan(\gamma) \geq tan(\alpha)$, and from this

$$t \leq \frac{s}{tan(\alpha)} < \frac{\epsilon}{tan(\alpha)} = \frac{c}{4}. \tag{1.3}$$

```
Algorithm Verify
   Input  P: terrain, k: number of towers,
          S: vector of divisions, h: height og towers
   Output x = (x1, x2, ..., xk): the position of the k solution towers

   for x = (x1, x2, ..., xk) in SxSx ... xS, with xi <> xj for i<>j
      if Cover( P, k, h, x) then
         return true
   return false
```

Figure 1.2: Pseudo-code for the verification algorithm

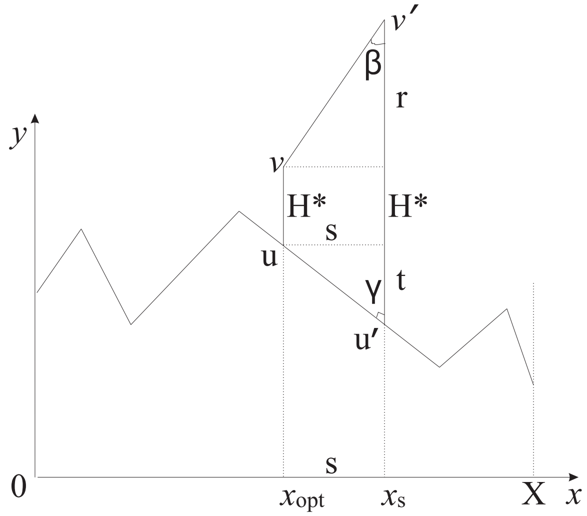


Figure 1.3: Optimal tower and approximation tower

Because $h = t + H^* + r \geq H^* + \frac{c}{2}$, it follows that $r \geq \frac{c}{2} - t \geq \frac{c}{2} - \frac{c}{4} = \frac{c}{4}$. Furthermore,

$$tan(\beta) = \frac{s}{r} < \epsilon \times \frac{4}{c} = tan(\alpha), \tag{1.4}$$

and, because both angles are in $(0, \frac{\pi}{2})$, it follows that $\beta < \alpha$. Lemma 2.2 is used to complete the proof of Lemma 2.1.

Lemma 2.2 If $\beta < \alpha$ in the setting described above, then the field of view of tower $u'v'$ includes the field of view of the optimal tower uv.

Proof of Lemma 2.2 In the proof of this result, the term *segment above terrain* denotes a segment that does not contain any point lying below the polygonal line.

Let q be a point in the field of view of the optimal tower uv, and x_q the x-coordinate of q. Consider the situation from the proof of the Lemma. There are three cases: $x_q <= x_{opt}$, $x_{opt} < x_q < x_S$, and $x_S \leq x_q$.

Because q is in the field of view of uv, then vq is a segment above the terrain in all cases. The segment vv' is also above terrain. If a tip of the terrain between the towers uv and $u'v'$ intersect vv', then that region of terrain will contain a segment that forms with the y axis an angle smaller than $\beta < \alpha$. This contradicts the choice of α. Using the same argument it can be shown that, in the first case (Figure 1.4), the segment $v'q$ is above vq and vv', and thus above terrain. From this, it follows that q is in the field of view of $u'v'$.

When q is between the two towers, it must be visible from v'. Otherwise, a segment of the terrain forms an angle sharper than β, which is impossible.

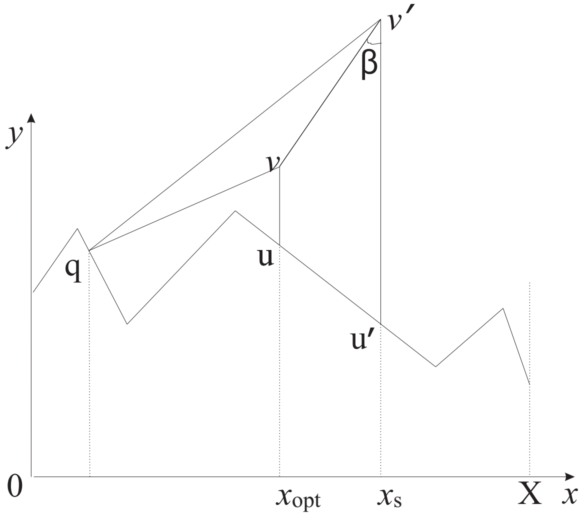


Figure 1.4: Field of view of the optimal tower included in that of the approximation tower

When $x_S \leq x_q$, because $u'v'$ extend beyond the level of v, q must also be in the field of view of $u'v'$. This concludes the proof of the Lemma 2.2.

Lemma 2.2 ensures that, for an $h \geq H^* + \frac{c}{2}$, the algorithm *Verify* finds k towers of height h that are located at locations from the division S and cover visually the entire terrain (each tower has a field of view which includes the field of view of an optimal tower). This proves Lemma 2.1.

Lemma 2.1 is used to prove claim 1 of the Theorem. When *Approx* finishes, $left$ identifies the smallest height for which *Verify* returns "true", while $right$ identifies the largest height for which *Verify* fails. Clearly, $h_{right} + \delta = h_{left}$. It is also true that $h_{right} < H^* + \frac{c}{2}$ because *Verify* fails for h_{right}. Putting the two results together,

$$h_{left} = h_{right} + \delta < (H^* + \tfrac{c}{2}) + \tfrac{c}{2} = H^* + c, \tag{1.5}$$

which proves claim 1 of the Theorem.

In order to prove the claim 2 of the Theorem, let $C_{cover}, C_{verify},$ and C_{approx} denote the computational complexity of the three algorithms involved in the solution algorithm *Approx*.

$C_{cover} = n$, the number of segments of the polygonal line P because the algorithm checks the visibility of each segment in constant time. It is easy to see that,

$$C_{verify} \leq |S|^k \times C_{cover} = (\tfrac{X}{\epsilon})^k \times n = (\tfrac{4X}{c \times tan(\alpha)})^k \times n, \tag{1.6}$$

where $|S|^k$ is an upper bound for the number of iterations performed by *Verify*.

Approx applies the algorithm *Verify* for $log(right)$ times; therefore,

$$C_{approx} = \log(right) \times C_{verify} \leq \log(\tfrac{2H_1}{c}) \times (\tfrac{4X}{c \times tan(\alpha)})^k \times n. \tag{1.7}$$

Note that C_{approx} is upper bounded by a function linear in n and polynomial in $\frac{1}{c}, X,$ and $tan(\alpha)$.

1.3 Connected Visibility Problem

Let us reconsider the initial problem of optimizing the height of k watchtowers in two dimensions. We introduce an extra constraint: each watchtower will be responsible for seeing only a connected region surrounding its base point. This constraint could arise if a guard did not wish to stay in constant communication with other watchtowers in order to know what was occurring close to his tower, or if we wanted to optimize visibility (as a guard will be able to see regions closer to his tower more easily than sections further away). It will be shown in this paper that this method has time complexity $O((n \log n)^k)$, where n is the number of segments in the terrain.

To treat the problem for k watchtowers, we divide our problem into k parts (considering individually the placement of each watchtower in its own domain - since the one-watchtower case can easily be solved by linear programming). We need a policy for choosing the watchtowers' separate domains. To find these domains, we adjust their boundaries dynamically, moving the endpoints over the x-axis and constructing locally optimal configurations.

In order to solve this, we will first develop a discretization of the x-axis based on inspection of the vertices of the terrain and the intersection points of the upper envelope (which is the lower limit of visibility of the whole terrain).

Theorem 3.1 The optimal position of one watchtower in one domain can only be at a vertex of the terrain or an intersection point in the upper envelope for that domain.

Proof. By linear programming, we know that the only critical points of the upper envelope will be at its intersection points, and by inspection, we know that the vertices of the terrain are the points closest to the upper envelope (they are local maxima within the terrain - and therefore the height needed to build a tower from the terrain to the upper envelope would be a local minimum at each of these vertices).

So, when placing one watchtower in one domain, we need only consider placing it at a vertex of the terrain or of the upper envelope. This gives us a way to discretize the x-axis: let us call these points x_1 through to x_N and divide the x-axis into intervals with $x_1, ..., x_N$ as endpoints.

Claim 3.1 $N \leq 2n$.

Proof: Since the upper envelope is constructed only by extensions of the n segments of the terrain, it could only have n possible different sections (n different slopes, or intersection points). Therefore, $N \leq n + n = 2n$.

Claim 3.2 The portions of the terrain and of the derived upper envelope between consecutive division points are straight lines.

Proof: This is a property of our choice of intervals.

Definition: Let us denote by h_x the minimum height of a single watchtower to which all of the terrain from 0 through x is visible.

Dynamics:

Overall, in order to optimize the partitions of the terrain, we must examine how the height of a tower will change as we increase the boundaries of the region it must guard. So, let us look at the simplest case: how will the height change as the region increases in one interval from x_{i-1} to x_i.

Lemma 3.1 As x (the boundary of our partition) increases along a subinterval $[x_{i-1}\ , x_i]$, h_x either decreases linearly or remains constant.

Case 1. If the slope of upper envelope segment is greater than that of the terrain segment, h_x must remain constant.

Proof. Since we have no intersection points or vertices within the interval, and since the height of the terrain is getting further away from the upper envelope, the endpoint x cannot be a location of a tower.

Case 2: If the slope of upper envelope segment is less than that of the terrain segment, and the terrain segment, increased by a height of h_{opt} (the optimal height up until x_{i-1}) intersects with the upper envelope segment then, from this intersection point until the endpoint x_i, h_x will decrease with a slope of S_T - S_{UE} (slope of terrain minus slope of upper envelope) (Figure 1.5).

To construct h_x as x varies continuously over the entire terrain, we need only consider at most $4n$ points: all $x_1, \ldots, x_N$ and all intersection points defined in Case 2 of the above lemma. Hence our problem is discrete.

The time complexity of constructing h_x in this manner is $O(n \log n)$ - $O(n)$ possible points to consider for h_x and within each of the $4n$ possible intervals, the construction of the upper envelope is of order $\log n$ (since to increase x from x_{i-1} to x_i, we need only consider one extra segment in the upper envelope).

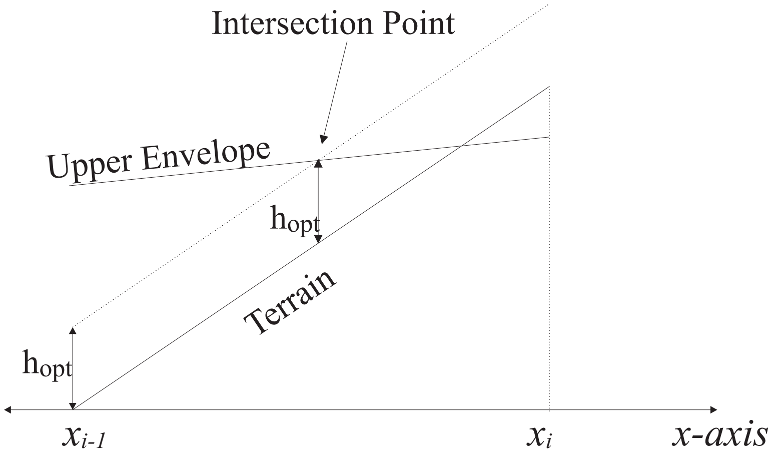


Figure 1.5: Case 2: example of decreasing height along one segment.

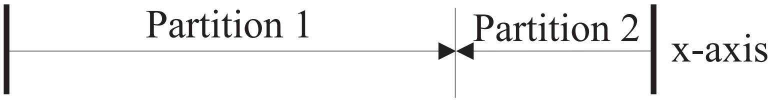


Figure 1.6: Dynamic partitioning for 2 watchtowers

1.3.1 2 watchtower problem

To place 2 watchtowers in the terrain, we will need to consider two dynamic regions instead of one. The proposed approach is to let $h_1(x)$ be the dynamic minimum height of tower 1 and $h_2(x)$ be that of tower 2. Now, let the region of $h_1(x)$ increase as x increases (i.e., it grows from $x = 0$ to the right) and let the region of $h_2(x)$ grow from $x = x_N$ to the left (Figure 1.6).

Take the maxmin of h_1 and h_2. The x value of max/min(h_1,h_2) will be the best location of the partition of the terrain into two regions. When we know the location of the partition, we can locate the position of the watchtowers from our previous computations and we know that the minimum height of these towers = max/min (h_1, h_2).

Since we are simply computing two $h(x)$, the order to time complexity of the two watchtower problem is still $O(n \log n)$.

1.3.2 k-watchtower problem

As in the 2-watchtower problem, we can simply considering dynamic partitions again, but this time consider k dynamic partitions and the heights $h_1, \ldots, h_k$ associated with each. We consider these partitions by first dividing the terrain into 2 partitions, then subdividing domain1 into 2 partitions, then subdividing again in this manner until we have k partitions. Take the min/max of each h within a subdivision, as we did for two watchtowers. This gives a method of time complexity $O((n \log n)^k)$.

1.4 Colouring algorithm in $2D$ k-watchtower problem

Problem(k-watchtower problem with whole segment visibility) Given a terrain P with n segments and a positive integer k, find the location of k towers such that every segment is visible from at least one of the tower and the maximum height of towers is minimize.

We propose *colouring* algorithm for solving the problem above.

The steps of this algorithm are as follows.

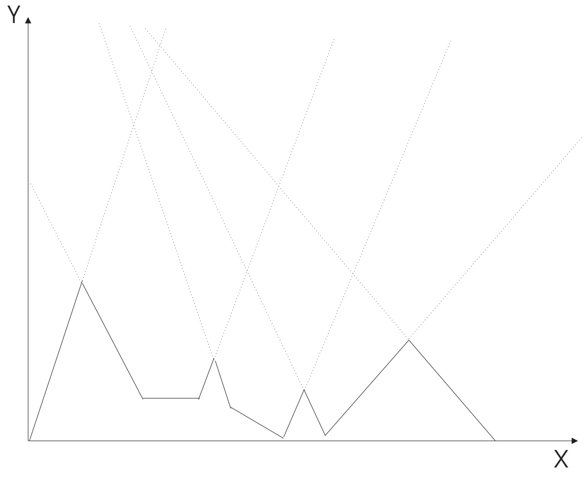


Figure 1.7: Extending edges

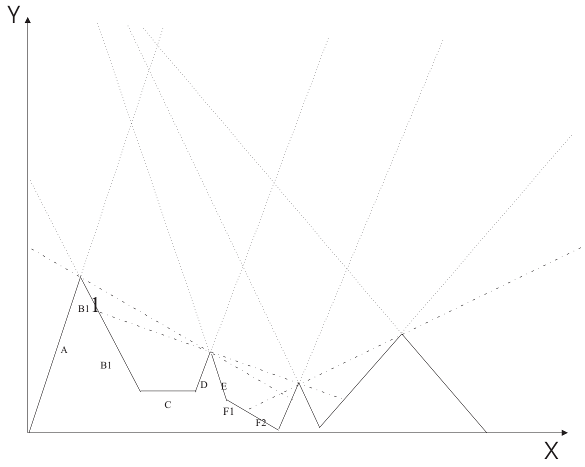


Figure 1.8: Colouring

1. Extend each segment of the terrain to a complete line (Figure 1.7).

2. Introduce additionally all segments whose endpoints are vertices of P and which lie completely above P. Together with the lines drawn in step 1, these segments induce a partition of the region of the plane above P. The sets of points on P visible to a point within such a region depend only on the region - not upon the particular point selected. Therefore, the finest partition of P induced by visibility can be completely characterized by membership relative to a set of at most $m = O(n^2)$ intervals. Moreover, the smallest planar regions of the arrangement so induced are $r = O(n^4)$ in number (Figure 1.8).

3. Indexing the set of regions by a variable i varying over index set $1, ..., r$, identify the set of terrain segments visible to each region. This can be accomplished by the visibility algorithm of *Guibas et al.* , which runs in $O(n)$ time.

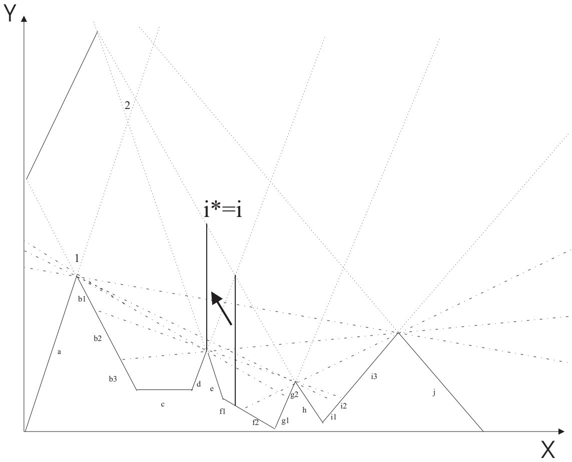


Figure 1.9: Optimization

Index i	Cover Segment (colouring)
1	a,b1,b2,b3,c,d,g2,i2,i3
2	a,b1,b2,b3,c,d,e,f1,f2,g1,g2,i2,i3
...	...
i	b1,b2,b3,c,d,e,f1,f2,g1,g2,h,i1,i2,i3
...	...

Now for each region, determine the minimum height of any tower with one endpoint on the boundary of that region and the other on the terrain. Each of these minima can be determined by linear programming in $O(n^2)$ time.(Figure 1.9)

Among all tops of towers which achieve the linear programming optimum within a given region, choose the leftmost, rightmost, and highest. Let I be the set of all such points, C the convex hull of I, and L the set of extreme points of the lower boundary of C.

This leads to an overall time complexity of $O(n^6)$ for determining L. This determination involves only the terrain, and does not involve k.

Next, consider all k-subsets of L; there are $C(|L|, k) = O(n^{4k})$ of these. Test each k-subset to decide whether the union of the corresponding k collections of visibility segments covers P. For each subset producing a cover of P, determine the largest of the corresponding tower heights inherited from Step 3 above. (This maximum tower height is associated with its originating k-subset.) After running through all feasible k-subsets, choose the k-subset which produces the least maximum height.

Theorem 4.1 The method described above produces the desired optimal height for the k-watchtower problem with whole segment visibility in $O(n^{4k+1} + n^6)$ time and $O(n^4)$ space.

We can extend the colouring algorithm in order to solve a similar problem in three dimensions and also achieve polynomial running time.

1.5 Conclusions

We have introduced three algorithms for solving restricted versions of the k-watchtower problem. The first algorithm finds an approximate solution for k-watchtower problem provided that the polygonal line does not contain vertical segments. The solution found by the algorithm is guaranteed to be accurate within a specified additive error. This algorithm runs in polynomial time, and the time upper bound is proportional to the reciprocal of the error. The second algorithm solves the k-watchtower problem

problem by finding the shortest k towers whose local visibility regions cover the entire terrain. Finally, the last algorithm solves the k-watchtower with whole segment visibility in two dimensions. This can be extended in three dimensions with polynomial running time.

References

[1] L. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. Tarjan, Linear time algorithms for visibility and shortest path problems inside triangulated simple polygons, Algorithmica, 2(2):209–234, 1987

Chapter 2

Problems in Portfolio Analysis

Participants: John Chadam (Mentor), Mehmet Atilla Begen, Ali Ghodsi Boushehri, Yuriy Kazmerchuk, Selly Kane, Viktoria Krupp, Eric Machorro, Eva-Marie Nosal, Limei Sun.

PROBLEM STATEMENT: The group considered several problems in portfolio analysis. In particular, the group generated computer codes for determining the optimal portfolio which minimizes risk for a given return. A data set was used to provide specific examples with and without shorting. In addition, the group studied how to price options on portfolios. Some specific problems which were addressed including comparing the Black and Scholes price of European-style option in the Gaussian and non-Gaussian cases. An Edgeworth expansion was used in the latter case and the magnitude of the correction was obtained for a specific data set. Finally, the values of European put option on the sum of two assets were computed directly using a Monte-Carlo simulations and an Index approximation.

2.1 Portfolio optimization

The value p of a portfolio consisting of N assets having unit prices $S_i,\ i = 1..N$ and a bond with value B can be written as

$$p = \theta_1 S_1 + ... + \theta_N S_N + bB,$$

where the proportions θ_i and b satisfy $\sum \theta_i + b = 1$.

Let's introduce some definitions. At first, we consider a portfolio without risk-free assets, i.e. $b = 0$ in this case. We call μ a *mean return* on the portfolio and σ^2 *a variance of return* if:

$$\mu = \sum_{i=1}^{N} \theta_i E_i$$

and

$$\sigma^2 = \sum_{i,j=1}^{N} \theta_i V_{ij} \theta_j = \theta^T V \theta$$

with E_i is the mean return of share S_i, $E[dS_i/S_i]$, and the matrix V is the covariance of the returns, $var[dS_i/S_i]$.

We calculate the mean-variance of an *optimized* portfolio as a solution of the following problem:

$$\min \theta^T V \theta$$

subject to the constraints:

$$\sum_{i=1}^{N} \theta_i = 1,\ \sum_{i=1}^{N} \theta_i E_i = \mu$$

In this formulation the θ_i could be negative representing short-selling. This problem was solved analytically using Lagrange multipliers. The above problem without short selling requires that the proportions $\theta_i \geq 0$ and the problem can only be solved numerically in this case. For a data set consisting of $N = 8$ risky assets both solutions are summarized in Figures 2.1 and 2.2 below.

2.2 Normality check

The basic assumption underlying the Black and Scholes approach to option pricing is that the underlying asset values follow a Geomteric Brownian motion. Since this may not be obtained in practice for a single asset, it is important to address the limitations of this GBM assumption. To this end we begin by appling statistical tests to check the normality of the returns. The Kolmogorov-Smirnov Goodness-of-Fit test statistic was used and various graphs (QQplot, histogram) were produced. Based on the statistical tests wherein the leverage point outliers were removed, seven shares were found to be normal and one found to be non-normal. In particular, the seventh share "FOSFX: Fidelity Overseas" did not have a normal distribution. The goodness-of-fit result of the fifth share "FSAVX: Fidelity Select Industrial Equipment" and the seventh share are shown as follows to illustrate this.

```
> ks.gof(data$V8, dist=’normal’)

        One sample Kolmogorov-Smirnov Test of Composite Normality

data:  data$V5 ks = 0.041, p-value = 0. alternative hypothesis:
  True cdf is not the normal distn. with estimated parameters
sample estimates:
  mean of x standard deviation of x
 0.06292663                1.611786
```

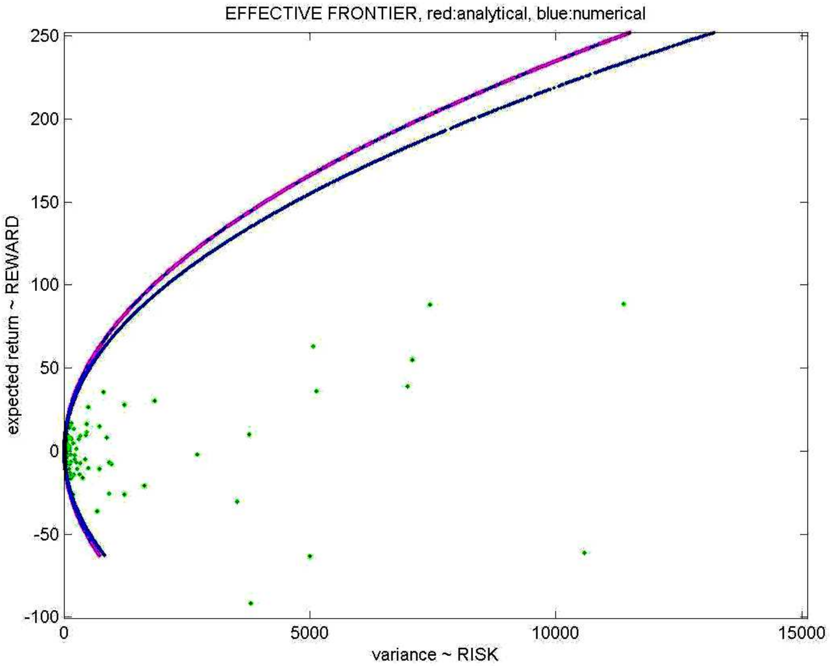


Figure 2.1: Optimal portfolio with shorting I.

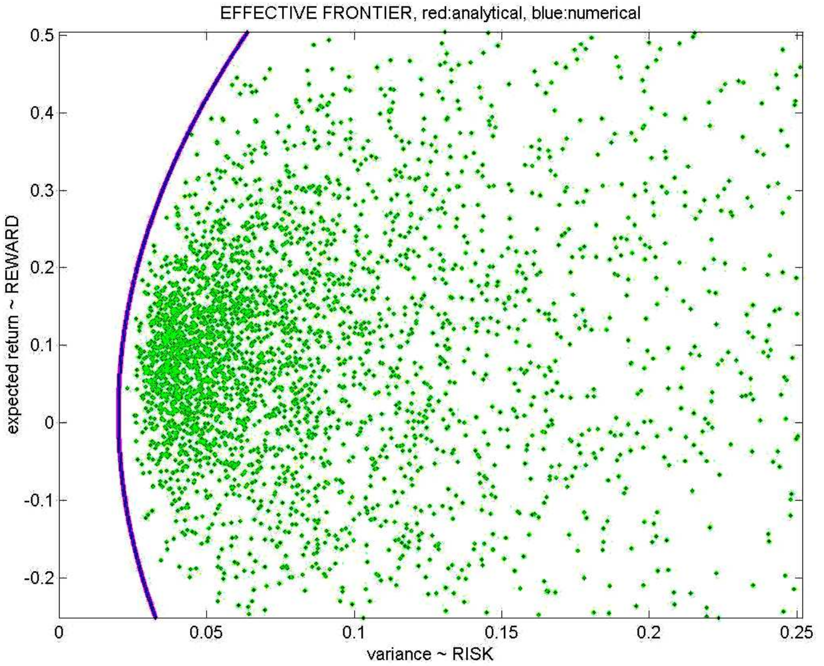


Figure 2.2: Optimal portfolio with shorting II.

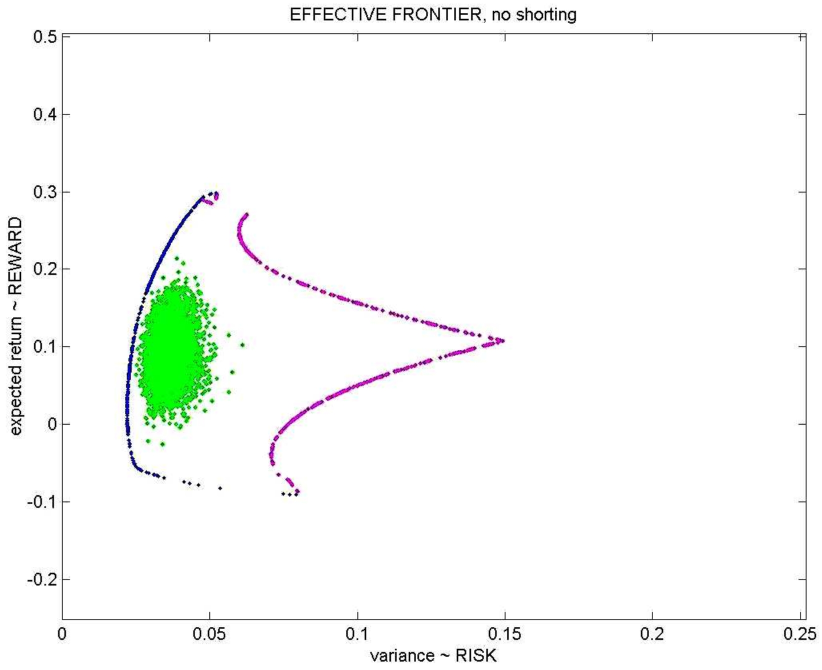


Figure 2.3: Optimal portfolio without shorting.

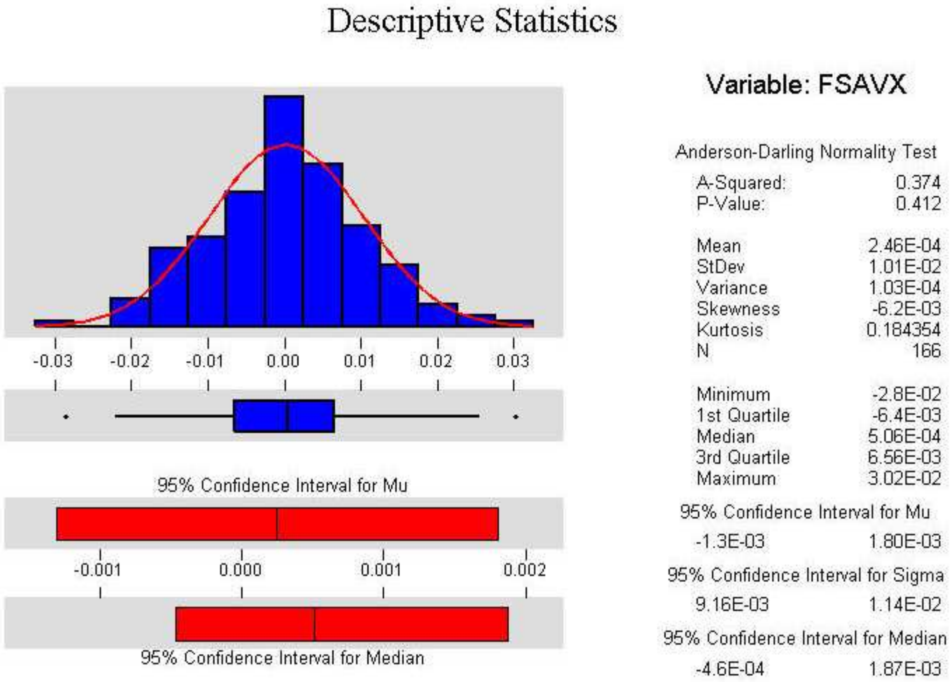
```
> ks.gof(b, dist='normal')

        One sample Kolmogorov-Smirnov Test of Composite Normality
data:  b ks = 0.0786, p-value = 0.0127 alternative hypothesis:
  True cdf is not the normal distn. with estimated parameters
sample estimates:
     mean of x standard deviation of x
 -0.0001885148              0.01173404
```

The p-value for share 'FSAVX' and 'FOSFX' are 0.21 and 0.0127 respectively. Hence, with a significance level of $\alpha = 0.05$ we accept the null hypotheses H_o of normality assumption of share 'FSAVX' ($p \leq \alpha$) and reject the assumption for share 'FOSFX' ($p \geq \alpha$).

The decision to reject H_o in the case of FOSFX is further supported by calculation of the standardized skewness (-0.30189) and kurtosis (3.6242489) both of which exceed the $\alpha = 0.05$ critical values. These were the only data of the 8 found to be non-Gaussian.

The two histograms (figure 2.4 and figure 2.5) draw a typical contrast between the funds that were found to be sufficiently normally distributed and the skewed (hence non-Gaussian) distribution of FOSFX.



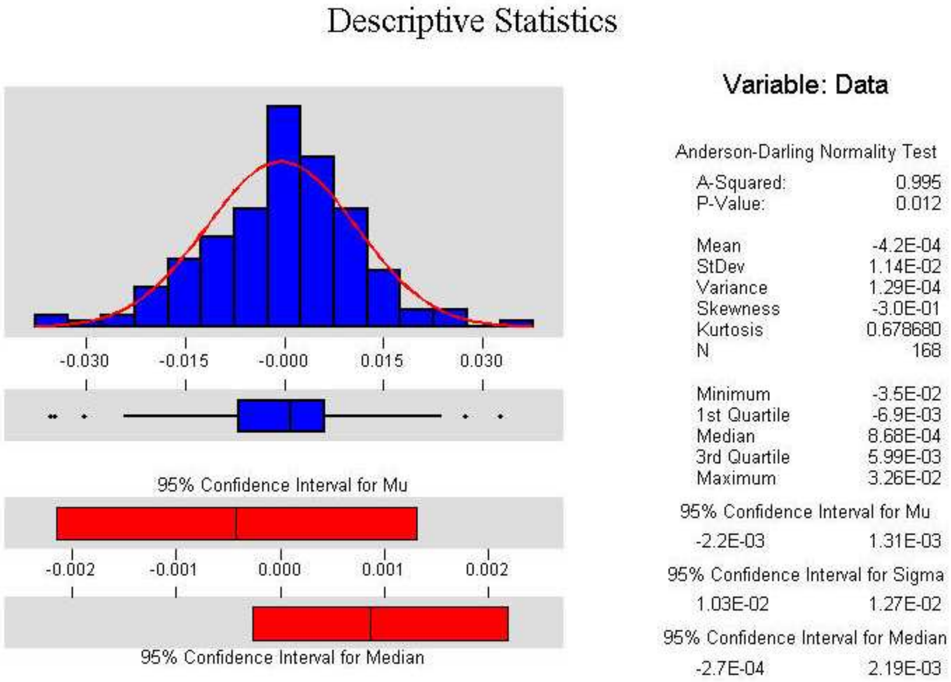
```
Variable: FSAVX

Anderson-Darling Normality Test
  A-Squared:              0.374
  P-Value:                0.412

  Mean                 2.46E-04
  StDev                1.01E-02
  Variance             1.03E-04
  Skewness             -6.2E-03
  Kurtosis             0.184354
  N                         166

  Minimum              -2.8E-02
  1st Quartile         -6.4E-03
  Median               5.06E-04
  3rd Quartile         6.56E-03
  Maximum              3.02E-02
 95% Confidence Interval for Mu
  -1.3E-03             1.80E-03
95% Confidence Interval for Sigma
  9.16E-03             1.14E-02
95% Confidence Interval for Median
  -4.6E-04             1.87E-03
```

Figure 2.4: The histogram for the share FSAVX.



```
Variable: Data

Anderson-Darling Normality Test
  A-Squared:              0.995
  P-Value:                0.012

  Mean                 -4.2E-04
  StDev                1.14E-02
  Variance             1.29E-04
  Skewness             -3.0E-01
  Kurtosis             0.678680
  N                         168

  Minimum              -3.5E-02
  1st Quartile         -6.9E-03
  Median               8.68E-04
  3rd Quartile         5.99E-03
  Maximum              3.26E-02
 95% Confidence Interval for Mu
  -2.2E-03             1.31E-03
95% Confidence Interval for Sigma
  1.03E-02             1.27E-02
95% Confidence Interval for Median
  -2.7E-04             2.19E-03
```

Figure 2.5: The histogram for the share FOSFX.

2.3 Monte-Carlo simulations of the stock prices with application to the option pricing

2.3.1 One dimensional case

Suppose, the stock prices satisfy the following Stochastic Differential Equations:

$$dS_i = rS_i dt + \sigma_i S_i dW_t^{(i)}, \ for \ i = 1, .., 8 \tag{2.1}$$

where r is the risk-free rate (or *drift*), σ_i is the standard deviation of stock price return (or *volatility*) and $\{W_t^{(i)}\}_{i=1}^{8}$ is the 8-dimensional Brownian motion with partly correlated components. Each component is normally distributed with zero mean and the variance t. Here the growth rate μ_i for individual stocks S_i are replaced by r to anticipate the risk-neutral evaluation of options. Using Ito's lemma one finds that stock prices follow a *Geometric Brownian motion* which is expressed by:

$$S_i(t) = S_i(0)e^{\{(r-\sigma_i^2/2)t+\sigma_i W_t^{(i)}\}}$$

Our task is to *simulate* stock prices using the representation above. Therefore, consider:

$$S_i(t) = S_i(0)e^{\{(r-\sigma_i^2/2)t+\sigma_i \sqrt{t}\phi\}} \tag{2.2}$$

where ϕ is $N(0,1)$. Numerically, we take a large number of samples (e.g. 100,000) of ϕ and substitute them into (2.2). Hence, we obtain a certain number of samples of $S_i(t)$. Taking an average of them we get a *simulated* price of the stock S_i at the time moment t.

Now, suppose we need to evaluate an initial value of a European put option with payoff $max(E-S,0)$ at time moment T, where E is the strike price of the option and S is the stock price at time T.

Having already simulated stock price $S(T)$ as above, we calculate the option price by discounting the payoff function:

$$V = e^{-rT}max(E - S(T), 0)$$

This is a risk-neutral price of the option.

2.3.2 Monte-Carlo simulations of two correlated stock prices

Monte-Carlo simulation is a natural method for the pricing of European-style contracts that depend on many underlying assets. Suppose, we have a European put option with the payoff $max(E-(S_1+S_2),0)$. In order to simulate the prices of two correlated stocks which satisfy the equations (2.1) we need to simulate two correlated normally distributed random variables ϕ_1 and ϕ_2 s.t.:

$$E[\phi_1\phi_2] = \rho_{12}$$

We generate them using a Cholesky factorization. Suppose, we have already generated *uncorrelated* normally distributed variables ε_1 and ε_2. We can use these variables to obtain variables with the given correlation through the transformation

$$\phi = M\varepsilon \tag{2.3}$$

where ϕ and ϵ are the columns vectors with ϕ_i and ϵ_i in the ith row. The matrix M is special and must satisfy

$$MM^T = \Sigma$$

with Σ being the given correlation matrix.

It is easy to show that this transformation will work. From (2.3) we have

$$\phi\phi^T = M\varepsilon\varepsilon^T M^T.$$

Taking expectations of each entry in this matrix equation qives

$$E[\phi\phi^T] = ME[\varepsilon\varepsilon^T]M^T = MM^T = \Sigma.$$

The Cholesky factorization gives one way of choosing this decomposition. It results in a matrix M that is lower triangular.

2.4 Pricing a European put on a portfolio on multi-assets: an index approximation

The main problem of pricing multi-asset options rests mainly on the fact that summing geometric Brownian motions does not necessarily give a GBM. In this part we assume that the portfolio and each of the assets follow a geometric Brownian motion under the risk neutral probability which will allow us to find The Black and Scholes put European price of the portfolio. More specifically we can obtain the above by assuming that the proportions of the individual stocks in the portfolio are required to be constant over time as is in the case of some mutual funds. We will in the first part price a put on a portfolio of two assets and in the 2nd part we will generalize the method for a portfolio that has multi-asset (more than two). The result obtained will be compared to those obtained by a direct Monte Carlo simulation of the full two-asset problem to check the accuracy of our approximation. We have made some financial assumptions in order to provide a real application of this method. In particular, we will assume that the vector of the portfolio returns will be Gaussian stochastic process.

2.4.1 Put option on a 2 assets Portfolio

The portfolio P is composed of the sum of the two assets S_1 and S_2 that are correlated. We assume that for any i S_i follows a geometric Brownian motion under the risk neutral probability $\mathcal{P}$. Assume that $\theta_1 = S_1/(S_1+S_2)$ and $\theta_2 = S_2/(S_1+S_2)$, which we assume to be constant. The last assumption is consistent with current mutual fund management policy. Then it can be shown that this portfolio follows geometric Brownian motion under the same risk neutral probability.

$$\frac{dP_t}{P_t} = rdt + \sigma dW_t$$

for σ which depends on σ_1 and σ_2 in the following way:

$$\sigma = \sqrt{\theta_1^2\sigma_1^2 + 2\rho_{12}\theta_1\theta_2\sigma_1\sigma_2 + \theta_2^2\sigma_2^2}$$

Therefore, the price of this type of the option could be obtained by applying the Black and Scholes formula to a new set of parameters r, σ, E, T and the initial price $S_1(0) + S_2(0)$. This price is compared to direct two-dimensional MC simulation (Section 2.3.2) in Figures 2.6 and 2.7.

2.4.2 Put Option on N assets portfolio

Let's consider the same assumptions as above. The vector of return $(\frac{dS_1}{S_1}, ..., \frac{dS_N}{S_N})$ is assumed to be a Gaussian stochastic process and for any i the unit stock price S_i satisfies (2.1). This yields that:

$$\sigma = \sqrt{\sum_{i=1}^{N}\theta_i^2\sigma_i^2 + \sum_{i\neq j}\rho_{ij}\theta_i\theta_j\sigma_i\sigma_j}$$

So, the option price could be obtained by applying the same method as in 2-dimensional case.

2.5 Pricing a non-Gaussian distributed share

The strong assumption in Black-Scholes pricing that data follows a geometric Brownian motion has been suggested as an explanation for the differences between the model prices and market prices. In particular, because the assumption of geometric Brownian motion does not hold in many cases, it is desirable to adjust the model for such cases. To do this, we approximate the underlying (true) distribution with the lognormal (approximate) distribution and add correction terms. The correction terms are found from a series expansion, called the Edgeworth series expansion, of the given distribution in terms of the

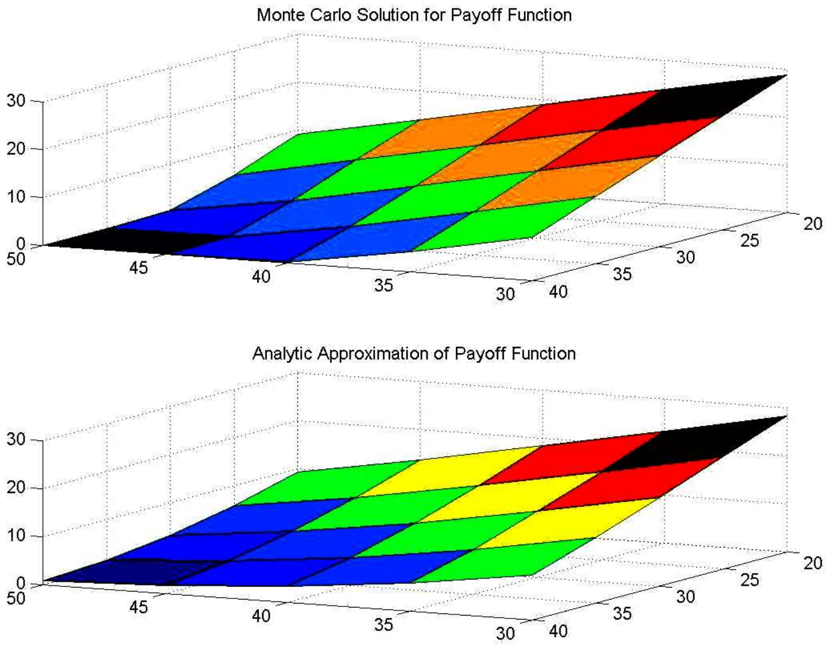


Figure 2.6: Comparison of Monte-Carlo method and analytical approximation.

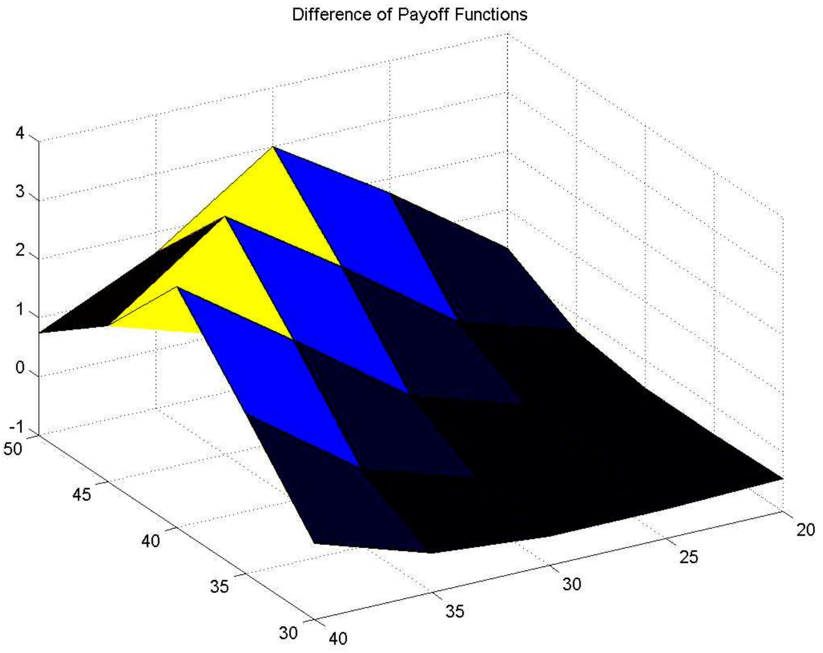


Figure 2.7: Error between two methods.

lognormal distribution (similar to a Taylor series expansion). It has coefficients that are simple functions of the moments of the true and approximating distributions. As we use only the first three adjustment values (which depend on variance, skewness, and kurtosis), our results are still approximate but they should capture most of the influence of the underlying distribution on the option pricing.

Denote the true probability distribution by F(s) and the approximate lognormal distribution by A(s) and assume that dA(s)/ds = a(s) and dF(s)/ds = f(s) exist, i.e. that the distributions have continuous density functions.

The first four cumulants can be found to be [1, p.350]

$$\begin{aligned} K_1(F) &= \alpha_1(F), \quad K_2(F) = \mu_2(F), \\ K_3(F) &= \mu_3(F), \quad K_4(F) = \mu_4(F) - 3\mu_2(F)^2 \end{aligned}$$

where

$$\alpha_j(F) = \int_{-\infty}^{\infty} S^j f(S) dS$$

is the j^{th} moment of distribution F and

$$\mu_j = \int_{-\infty}^{\infty} (S - \alpha_1(F))^j f(S) dS$$

is the j^{th} central moment of distribution F

The first cumulant is the mean, the second is the variance, the third is a measure of skewness, and the fourth is a measure of kurtosis. Analogous notation is used for moments and cumulants of A.

The Edgeworth expansion for f(s) in terms of a(s) can be proven to be [1, p.350]

$$\begin{aligned} f(S) = a(S) &+ \frac{K_2(F) - K_2(A)}{2!}\frac{d^2a(S)}{dS^2} - \frac{K_3(F) - K_3(A)}{3!}\frac{d^3a(S)}{dS^3} + \\ &+ \frac{((K_4(F) - K_4(A)) + 3(K_2(F) - K_2(A))^2)}{4!}\frac{d^4a(S)}{dS^4} + \varepsilon(S) \end{aligned}$$

where $K_1(A) \equiv K_1(F)$ and $\varepsilon(S)$ is the residual error.

Consider a put option with maturity time t (in years), strike price E, and underlying stock value of $S(0)$ at time 0. Then the value for the put option, $P(F)$ is

$$P(F) = e^{-rt} \int_0^E (E - S) f(S) dS$$

As we have seen, FOSFX (7th column) does not follow geometric Brownian motion. We apply the method outlined above to find the value for the put option $S(0) = 46.1$, $r = 0.06$, and $t = 1/3$ (4 months).

We are given $L = 85$ is the duration (in business days) of the put option being considered and $N = 170$ is the number of days for which daily return data is available in the form $\{\frac{\triangle S_i}{S_i}\}_{i=1\cdots 169}$. From this format the data were transformed to the form $\{S_i\}_{i=1\cdots N}$. That is to say, the data format was converted from the daily return rate $\{\frac{\Delta S_i}{S_i}\}_{i=1\cdots N}$ to the daily underlying asset value $\{S_i\}_{i=1\cdots N}$ where the initial asset price $S_1 = 46.1$ on the corresponding date was available at *http://www.fidelity.com.*

A second transformation was made to facilitate the estimation of the cumulants of the "true" distribution which will be based on the approximating lognormal distribution typical of a pure GBM option pricing scheme: the data was converted from $\{S_i\}_{i=1\cdots N}$ to $\{log\frac{S_{i+L}}{S_i}\}_{i=1\cdots L}$.

From this "transformed data" the sample moments were estimated by

$$\mu_1(F) \approx \frac{1}{L}\sum_{i=1}^{L} log\frac{S_{i+L}}{S_i}$$

$$\mu_J(F) \approx \frac{1}{L}\sum_{i=1}^{L}[log(\frac{S_{i+L}}{S_i}) - \mu_1(F)]^J$$

Note: the parameters (μ, σ^2) of the approximating lognormal distribution are estimated by $(\hat{\mu}, \hat{\sigma}^2)$ using the original data $\{\frac{\triangle S_i}{S_i}\}_{i=1 \cdots N}$ in the following manner:

$$\sigma^2 \approx 252\hat{\sigma}^2_{daily}$$

where

$$\hat{\sigma}^2_{daily} \equiv \frac{1}{N-1}\sum_{i=1}^{N}(\frac{\triangle S_i}{S_i} - \hat{\mu})$$

and

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N}\frac{\triangle S_i}{S_i}$$

Values used were found as follows:

$$\alpha_1(F) = ln(S_0) + rt$$
$$a(S) = \frac{1}{S\sigma\sqrt{2\pi t}}e^{-(log(S)-(log(\alpha_1(A))-\frac{\sigma^2 t}{2}))^2/(2\sigma^2 t)}$$

The results are given in Table 2.1.

Strike price	35	40	45	50	55
Black-Scholes	0.004764	0.14547	1.140137	3.855774	8.002353
No correction terms	0.003874	0.132113	1.099248	3.815515	7.984549
With correction terms	0.00360	0.12612563	1.069307851	3.75421277	7.90624269

Table 2.1: European Put Option Price of Stock7 on Sep 1st.

To partially justify dropping the error, we noted that the adjustment terms become almost negligible. For example, for K = 55, the first put given by the lognormal approximation was 7.98 dollars, the first correction term was 7.84 cents, the second was 0.00821 cents, and the third correction term was only 0.00198 cents.

2.6 Conclusions

The group studied two problems in portfolio analysis - to find the mean-variance portfolio which minimizes risk for a prescribed return and to approximate corrections to the Black-Scholes-Merton price for options due to non-Gaussian effects. A solution for the first problem was found with and without shorting as well as with and without inclusion of a riskless asset in the portfolio. The second problem is of interest because in practice most assests do not evolve according to a log-normal process and, even if they do, the sum of such processes (a portfolio) does not. Using an Edgeworth expansion we calculate the correction terms for a European put option on a single non-Gaussian asset. In addition we compute the values for a European put on the sum of two log-normal asset using an ‘index’ approximation. This is compared to values computed directly using Monte-Carlo techniques. It would be intersting to apply the Edgeworth expansion methods to this latter case.

References

[1] Jarrow, R., and Rudd, A., ‘Approximate Option Valuation for Arbitrary Stochastic Processes’, Journal of Financial Economics 10 (1982) 347 - 369.

[2] Branstein Musiol, and Semendjajeu Muhlig ‘Taschenbuch Der Mathematik’, Verlag Harri Deutsch 1997

[3]Sachs, Lothar, ‘Applied Statistics’. 1978 (Springer series in Statistics). 5th edition c1982. pg 324-328.

[4]Wilmott, P. Howison, S. Dewynne, J. ‘The Mathematics of Financial Derivatives: A Student Introduction’. Cambridge Univ. Press. 1995. 1997 reprint. c1995

[5]Luenberger, D. ‘Investment Science’ 1998. Oxford Univ. Press.

Chapter 3

Modelling a Metal Spray Forming Process

Participants: Ian Frigaard (Mentor), Mariana Carrasco Teja, John Harlim, Theodore Kolokolnikov, Melvin Leok, Allan Majdanac, Matthias Mück, Jason Slemons, and Qutaibeh Katatbeh.

PROBLEM STATEMENT: Spray-forming is a metal manufacturing process which is capable of producing large bulk deposits of various metal alloys. With careful control, rapidly solidified near-net shape deposits can be produced which have significantly improved microstructural and mechanical properties. In the billet spray-forming process a molten metal stream is first atomized by high speed gas jets and is then deposited onto a circular collector plate. The collector plate is positioned some distance from the atomizer, it rotates about a vertical axis and is withdrawn slowly downwards at a controlled speed. Usually, the metal spray is directed in towards the rotational axis and oscillates, so as to distribute the metal in a prescribed way. The main objective of this report is to model the billet growth mathematically and predict the dynamic features.

3.1 Introduction

Spray forming processes for metals involve the atomizing of a molten metal stream by means of high speed gas jets. The atomized metal is then sprayed by the jets and collected below on a disk, which is both rotating and moving vertically. This spraying is continued, so as to produce desirable shapes, known as *billets*. Ideally the billet will be cylindrical, with minimal deviations from this shape being tolerated as part of the forming process.

In this report, we present a mathematical model based on conservation laws, which are used to derive the equation of evolution of the billet surface. Since the key control parameter involved in the process is the velocity of the collection plate, we try to determine the shape of the surface by controlling this velocity and also by attempting to approximate the spray distribution leaving the atomizer. Between these two parameters, a reasonable model has been developed.

A schematic representation of the spray-form billet production is shown in Figure 3.1. The molten metal spreads rapidly towards the flat collector disk. This disk rotates about a vertical axis and gradually moves downwards, at a controllable velocity.

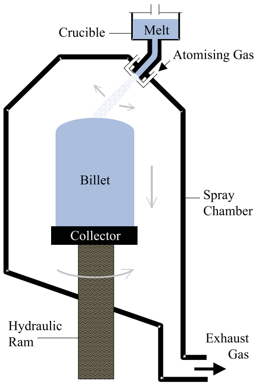


Figure 3.1: Schematic of a billet spray forming process.

3.2 Mathematical Model

The analysis of this problem is based on the conservation of mass on the surface of the billet. Namely, the rate of mass deposition per time unit on some arbitrary element $\hat{A}$ of the billet surface is simply equal to the mass flux through the surface element

$$\int_{\hat{A}} \hat{\vec{v_s}} \hat{\rho} d\hat{a} = -\int_{\hat{A}} \widehat{\vec{G}} \cdot \vec{n} d\hat{a},$$

where $\widehat{\vec{G}}$ is a directed mass flux, $\hat{v_s}$ is the velocity of the surface in the direction of the outward normal, $\vec{n}$, $\hat{\rho}$ is the density of billet, and all "hatted" variables are dimensional quantities. By representing the surface of the solidified billet as a level set

$$\widehat{\vec{F}}\left(\hat{\vec{x}}, \hat{t}\right) = 0,$$

where $\widehat{\vec{F}} : \mathbb{R}^3 \times \mathbb{R}^+ \to \mathbb{R}$, we obtain the following relation for the normal to the surface

$$\vec{n} = \frac{\widehat{\nabla} \widehat{\vec{F}}}{\left| \widehat{\nabla} \widehat{\vec{F}} \right|},$$

where $\nabla = \nabla_{\hat{\vec{x}}}$ for notational simplicity.

Furthermore, the time rate of change of $\widehat{\vec{F}}$ is given by

$$\frac{\partial \widehat{\vec{F}}}{\partial \hat{t}} = -\widehat{\nabla}\widehat{\vec{F}} \cdot \frac{d}{d\hat{t}}\hat{\vec{x}} = -\hat{\vec{v_s}}\left|\widehat{\nabla}\widehat{\vec{F}}\right|,$$

and substituting, we obtain the equation

$$\int_{\hat{A}} -\frac{\partial\widehat{\vec{F}}/\partial\hat{t}}{\left|\widehat{\nabla}\widehat{\vec{F}}\right|}\hat{\rho}\,d\hat{a} = -\int_{\hat{A}} \widehat{\vec{G}}\cdot\frac{\widehat{\nabla}\widehat{\vec{F}}}{\left|\widehat{\nabla}\widehat{\vec{F}}\right|}d\hat{a}.$$

Since $\hat{A}$ is arbitrary, and by evoking the continuity of $\widehat{\vec{F}}$, we obtain the equivalent differential equation

$$\frac{\partial\widehat{\vec{F}}}{\partial\hat{t}}\hat{\rho} = \widehat{\vec{G}}\cdot\widehat{\vec{\nabla}}\widehat{\vec{F}}.$$

In general, the mass flux expression $\widehat{\vec{G}}$ will introduce a time delay, but as the spatial scales are small relative to the velocity of the atomized metal jet, we will neglect the time lag between the atomization and the deposition event. Furthermore, this allows us to assume that dispersion of the gas jet is small, and to good approximation, the cross-sectional distribution of particles in the jet in the absence of deposition is independent of the distance from the nozzle.

The high shear flow associated with the atomization yields a ballistic trajectory for the metal jet, and we will assume that deposition occurs at the first intersection of the metal jet with the surface of the billet. If we assume that the surface is convex, then the point of first intersection can be identified by the sign of the $\widehat{\vec{G}}\cdot\widehat{\vec{\nabla}}\widehat{\vec{F}}$ term, and the non-deposition on the point of second intersection can be realized by a Heaviside function multiplying the $\widehat{\vec{G}}\cdot\widehat{\vec{\nabla}}\widehat{\vec{F}}$ term. The "shadow" effects have not been considered in the numerical analysis.

Assume a radially symmetric distribution of mass flux, $\hat{g}(\hat{r}')$, with respect to the spray direction $\hat{\vec{k}}'$ of the atomizer nozzle such that

$$\int_0^{2\pi}\int_0^{\infty}\hat{g}\left(\hat{r}'\right)\hat{r}'d\hat{r}'d\theta = 1,$$

where (r', θ', z') refers to the coordinate system attached to the atomizer. Let $\hat{\vec{x_a}}(\hat{t}) = R_{\hat{\omega}\hat{t}}(\hat{R}_a, 0, \hat{z_a})^T$ be the position of the spray at time $\hat{t}$, where $R_{\hat{\omega}\hat{t}}$ is the rotation matrix about z_1-axis at angle $\hat{\omega}\hat{t}$. With this notation the mass flux vector field reads

$$\hat{\vec{x}}_1 \mapsto \widehat{\vec{G}}_1\left(\hat{\vec{x}}_1, \hat{t}\right) = \widehat{\dot{M}}\left(\hat{t}\right)\hat{g}\left(\left|\left(\hat{\vec{x}}_1 - \widehat{\vec{x_a}}(\hat{t})\right)\times\widehat{\vec{k}'}(\hat{t})\right|\right)\widehat{\vec{k}'}(\hat{t}),$$

where $\widehat{\dot{M}}(\hat{t})$ is the mass flow rate from the nozzle and $\widehat{\vec{k}'}(\hat{t}) = R_{\hat{\omega}\hat{t}}(-\sin(\alpha(\hat{t})), 0, -\cos(\alpha(\hat{t})))^T$ is the spray direction with a declination angle $\alpha(\hat{t})$. We scale the problem using the following dimensionless variables

$$\begin{aligned}
\hat{\vec{x}} &= \hat{R}_0\vec{x} \\
\hat{\vec{F}} &= \hat{R}_0\vec{F} \\
\hat{\vec{G}} &= \frac{\widehat{\dot{M}_0}}{\pi\hat{R}_0^2}\vec{G} \\
\hat{U}_0 &= \frac{\widehat{\dot{M}_0}}{\hat{\rho}\pi\hat{R}_0^2} \\
\hat{\vec{U}} &= \hat{U}_0\vec{u}(t) \\
\hat{T}_0 &= \frac{2\pi}{\hat{\omega}}
\end{aligned}$$

where $\hat{R}_0$ is the desired radius of the billet, $\hat{U}_0$ is a characteristic withdrawal velocity of the plate, and $\hat{T}_0$ was scaled relative to the rotation period of the billet. This corresponds to taking a timescale on which all transient surface movement should be observable.

In the stationary billet coordinates with the scaled variables, the equation is given by

$$\frac{\partial\vec{F}}{\partial\hat{t}} = \epsilon\left(u\frac{\partial\vec{F}}{\partial z_1} + \dot{M}\left(t\right)g\left(r'\right)\vec{k}' + O\left(\frac{R_a\omega}{V_{s,0}}\right)\right)\cdot\nabla\vec{F}, \tag{3.1}$$

where the coefficient $\epsilon = \frac{\frac{2\pi}{\omega}}{\frac{R}{\hat{U}_0}}$ corresponds to the ratio of time scales in the problem.

There are two time scales in the problem: one for the rotation of the billet, and the other for the vertical growth of the billet. The order of the rotation time scale is much smaller than that of the growth time scale. The resulting equations have been scaled using the rotation period, and the results we are interested in are on the time scale of the billet growth. In order to compare effects, the equations are averaged over the rotation time scale.

Let $\eta = \epsilon t$ be the scaled rotation period, where ϵ is small positive constant as previously defined. This new time scale, η, is on the order of the billet growth. Therefore, all parameters in the problem are of the same order, and hence, can be compared. The resulting equation is

$$\frac{\partial\vec{F}}{\partial\eta} = u(\eta)\frac{\partial\vec{F}}{\partial z_1} + \left(\dot{M}(\eta)\bar{\vec{g}}\right)\cdot\nabla\vec{F}, \tag{3.2}$$

where $\bar{\vec{g}} = \frac{1}{T}\int_0^T g(r)\vec{k}dt$, is the time averaged distribution of the mass flux, and $\dot{M}(t)$ and $u(t)$ are replaced by $\dot{M}(\eta)$ and $u(\eta)$, respectively. Converting to cylindrical-polar coordinates, expanding the inner product, and assuming that the problem becomes symmetric with respect to the z_1-axis after the averaging process, i.e. $\vec{F}(r,\theta,z_1,t) = \vec{F}(r,z_1,t)$, the resulting equation is

$$\frac{\partial\vec{F}}{\partial\eta} = u(\eta)\frac{\partial\vec{F}}{\partial z_1} + \dot{M}(\eta)\left(\bar{g}_r\frac{\partial\vec{F}}{\partial r} + \bar{g}_{z_1}\frac{\partial\vec{F}}{\partial z_1}\right), \tag{3.3}$$

To help predict the behavior of the surface evolution, the characteristics of the differential equation (3.3) are examined. The characteristic equations are

$$\begin{aligned}
\frac{dr}{d\eta} &= -\dot{M}(\eta)\bar{g_r}(r,z_1) \\
\frac{dz_1}{d\eta} &= -\dot{M}(\eta)\bar{g_{z_1}}(r,z_1) - u(\eta)
\end{aligned} \tag{3.4}$$

The assumption of ballistic spraying, together with the continuity of the mass flux, yields

$$\frac{\bar{g_r}}{dr} + \frac{\bar{g_{z_1}}}{dz_1} = 0 \tag{3.5}$$

Linear analysis of the system (3.3) in addition to equation (3.5) determines that there are at least two saddle points on the z_1-axis. Only one of these equilibrium points is important to the behavior of the billet growth, as any others are physically located inside of the billet. From equation (3.3), the saddle point is determined by the following condition

$$\bar{g}_{z_1}(0, z_1) = -\mu \tag{3.6}$$

where

$$\mu = \frac{u(\eta)}{\dot{M}(\eta)}. \tag{3.7}$$

After non-dimensionalizing our variables, we find that $\mu = 1$ corresponds to the required billet radius which, after scalings, is 1.

Phase plane analysis of the system shows that a steady state distribution is attainable, and that all trajectories eventually lead to this steady state. These results motivate the analysis of the steady state equation, which can be derived from (3.3), by making the assumption that $F(r, z, \eta) = z + f(r, \eta)$. The steady state equation is found by eliminating all time dependence from the above assumption

$$\frac{df}{dr} = \frac{\mu + \bar{g}_z}{\bar{g}_r}. \tag{3.8}$$

The subsequent numerical simulations are based on the solution of equation (3.8).

3.3 Results

To see a dependence of the the shape of a stable billet configuration on the (scaled) withdrawl velocity μ we first restrict ourselves to the case of a single declination angle $\alpha = 30^0$ (Fig. 3.2a), and with gaussian distribution of the material within this ray. First we compute the vector field $\bar{g}$ of the mass flux averaged over the rotation about the vertical axis and use that as input to compute the solutions of (3.8) for steady billet formations for several values of μ. Since $\bar{g}_r(0, f) = 0$, the initial conditions $f(0) = f_0$ needed for the numerical integration of (3.8) are found by solving the equation (3.6).

The averaged vector field $\bar{g}$ and the curves are plotted in Fig. 3.2a. The direction of the center of the mass ray is also indicated in the figure. Note that for small velocities (say $\mu \leq 2$) the radius r_b of the billet is determined by the cutoff of the vector field $\bar{g}$. In this case, $\mu r_b^2 \approx 1$ as expected. For $\mu = 3, 4$ a big amount of mass cannot be deposited on the surface which results in a breakdown of mass conservation. This is observed in Fig. 3.3.

Note that for $\mu = 1$ the radius of the billet is approximately $r_b = 1$ which is consistent with our scalings. Further we observe that for billets with radius $r_b \geq 0.9$ the surface is concave at the center when $\alpha = 30^0$. This is undesirable because it creates non-uniformities inside the billet.

This motivated us to consider other angles α (Fig. 3.2b) as well as scanning over a sector with the ray (Fig. 3.4). From now on we restrict our attention to $\mu = 1$ which guarantees the required billet radius $r_b = 1$ (in scaled variables).

Fig. 3.2b shows different shapes corresponding to different angles α (no scanning). We observe that at a critical angle of about 40^0 the billet surface is changing from a convex to a concave shape at the center. However, in industrial applications an angle of 30^0 is usually used rather than a relatively shallow angle of 40^0 (this is to avoid the slippage of material past the surface of the billet).

To compensate a concave shape for an angle of 30^0, we simulated the scanning over a sector $[\alpha_1 = 30^0, \alpha_2]$ for several values of α_2. Fig. 3.4a shows the resulting vector field $\bar{g}$ and the resulting billet shape with $\alpha_1 = 30^0, \alpha_2 = 45^0$. Fig. 3.4b shows different billet shapes for a fixed $\alpha_1 = 30^0$ and α_2 as indicated. As expected, for α_2 large enough (about 40^0), the averaging of the mass flux over this sector insures convexity of the billet. For a very large spread of the sector (say $\alpha_2 = 45^0$) the plateau on top of the billet is shrinking.

Our analysis of the spraying process made us understand how the various parameters influence the shape of the billet. For example, within our model it is possible to produce billets of desired radius.

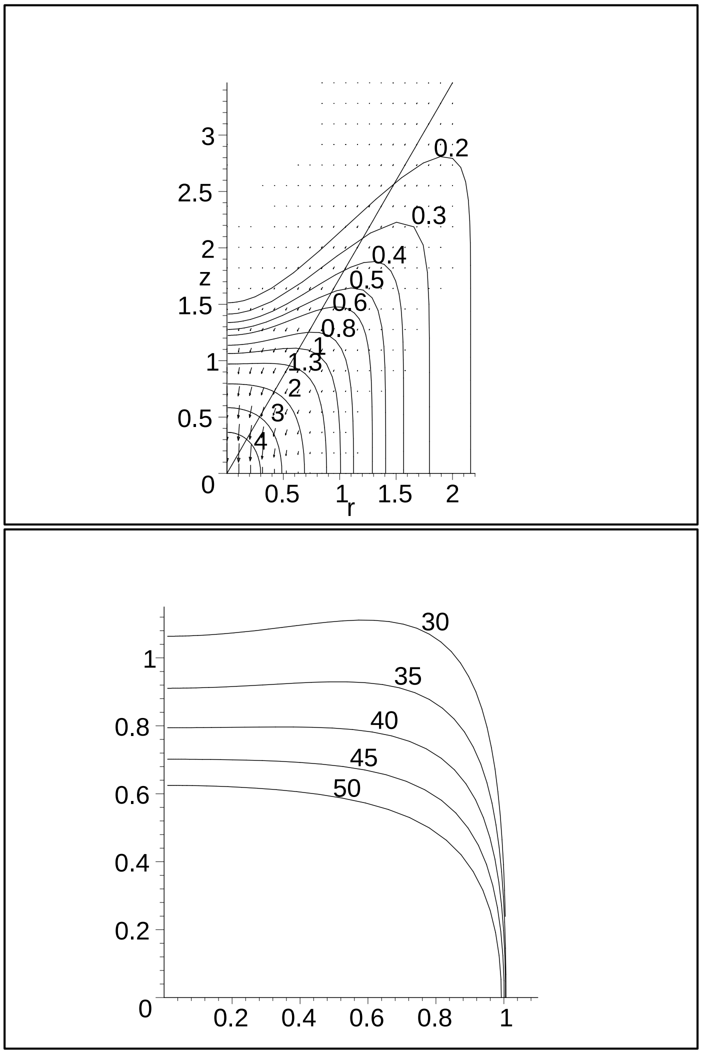


Figure 3.2: (a) Dependence of the shape of the billet on (scaled) withdrawl velocity μ (indicated above the curve). Spray angle is fixed at 30^0 and no scanning. (b) Dependence of the shape of the billet on the spray angle (no scanning). Scaled velocity is fixed at $\mu = 1$.

Billets with concave surfaces can be avoided either by increasing the declination angle or by scanning over larger sectors. By choosing parameters appropriately, one can even produce a billet of "optimal" shape such that the top is as flat as possible.

μ	0.2	0.3	0.4	0.5	0.6	0.8	1.0	1.3	2.0	3.0	4.0
r_0	2.16	1.80	1.57	1.41	1.29	1.12	1.01	0.88	0.69	0.49	0.30
μr_0^2	0.93	0.97	0.99	0.99	1.00	1.00	1.02	1.01	0.95	0.72	0.36

Figure 3.3: Relationship between μ and the billet radius r_b.

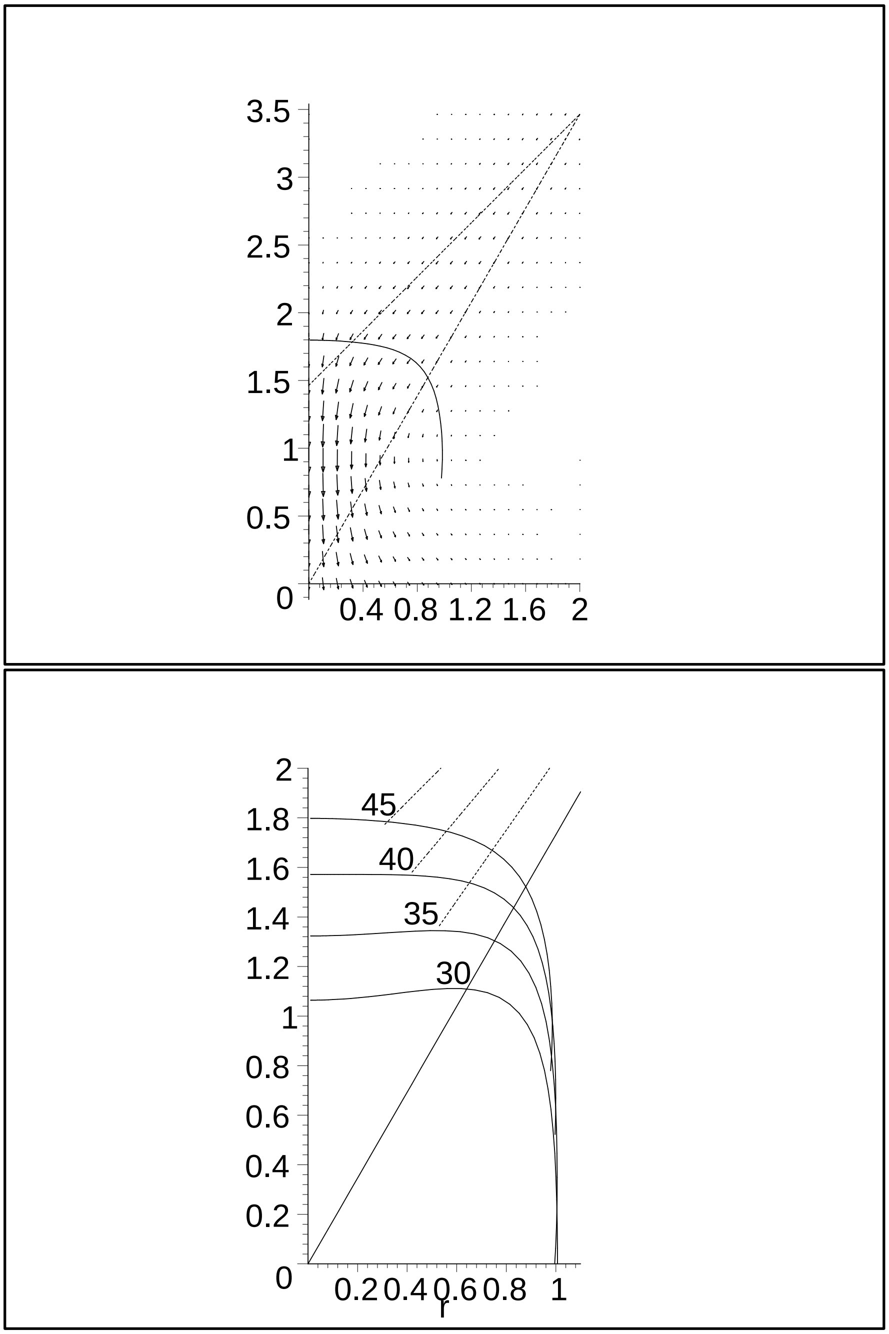


Figure 3.4: (a) Vector field $\bar{g}$ and the shape of the billet for $\alpha_1 = 30^0$ and $\alpha_2 = 45^0$. (b) Shapes of the billet for α_2 as shown and α_1 fixed at 30^0.

Chapter 4

eb Hosting Service Level Agreement

Participants: Lisa Korf (Mentor), Monica Cojocaru, Yashar Ganjali, Seungwon Jeon, Ramin Mohammadalikhani, Carmeliza Navasca, Alberto Nettel, Asa Packer, Sarah Sumner

PROBLEM STATEMENT: In this paper we propose a model for measuring the quality of service (QoS) in a Web-hosting facility. We assume that there is an agreement between the provider and a client (or customer), regarding the price of different levels of service, known as service level agreement (SLA). The client we refer to is a company.

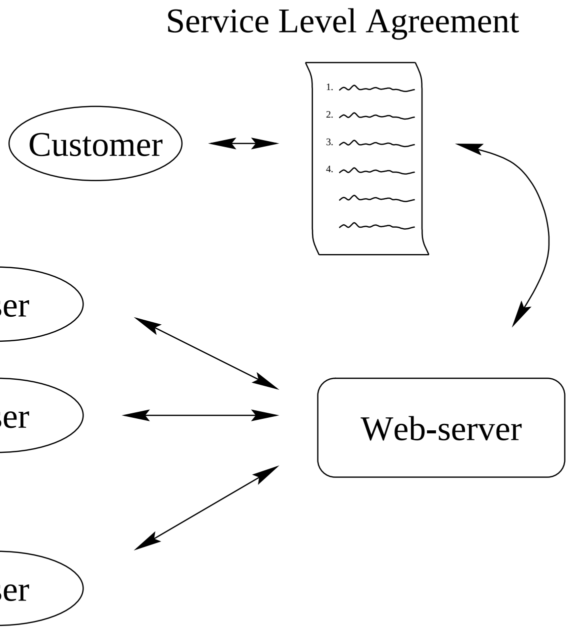


Figure 4.1: Service Level Agreement

The Web-server provides the space for the Web-pages, text documents, audio and video files etc. of the customer. Each customer has a number of users that request access to the documents on the Web-server. The Web-server has to provide a service that meets the requirements of the SLA (Figure 4). The SLA states that some QoS measurement lies within some bound for a given percentage of requests averaged over a given long period of time.

4.1 The framework

In a very simplified model, a Web-server is connected to a user via a link with a known bandwidth. The user sends a sequence of requests for the files located on the server (Figure 4.1 (A)). The Web-server decides which requests it is going to serve and simply discards all other requests. Other than choosing which requests to serve, there is another important decision which the server has to make and that is how to allocate its resources (the bandwidth, CPU time, and so on) to the requests which are going to be served.

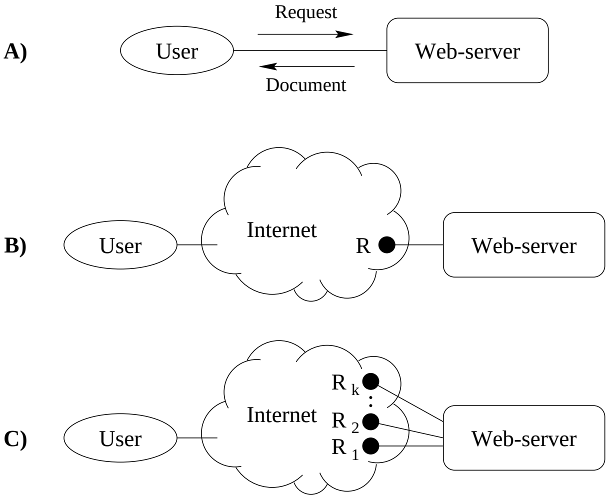


Figure 4.2: Models of network

In reality, usually the server and the user(s) are connected throught a series of intermediate routers and the quality of service provided by the Web-server to the user is affected by the quality of service provided by those routers. In a more realistic model we should also consider how the quality of service provided by the Web-server is affected by the quality of service provided by the intermediate routers (Figure 4.1 (B)). Some important parameters here are the average delay of messages, the loss rate, the throughput, and so on. For simplicity we could assume that the server is connected to a single router with known parameters. Finally, we can consider a model which seems to be the closest to the real networks in which the Web-server is connected to a number of routers each providing a (possibly) different quality of service (Figure 4.1 (C)). The Web-server can decide (based on the quality of service the routers are supposed to provide) which of the adjacent routers should be used to serve a specific request. **In this paper we are treating only the scenario in Figure 4.1A**).

4.2 Dynamics

In this section we propose a model for the dynamics of the activities provided by a Web-hosting facility under a certain SLA. We derive from here a controlled optimization problem for maximizing the revenue of the provider subject to penalties. Our model will be a discrete time one. The state variable is the number of requests of different classes for connecting to the network.

Denote by $[0, T]$ the time interval for the problem with the step Δt. In the previous section we described how the Web-hosting facility functions. Consider the system with a known maximum bandwidth C. User requests arrive at random times and a request will take a certain response time (RT) to be served. We will assume that the arriving requests belong to different classes, which are indexed from $i \in \{0, ..., J\}$. For simplicity, we will consider only two classes.

4.2.1 Notation

1. Let X_t^i be the number of requests of a certain class i at the moment t, where $i \in \{0,..,J\}$. We differentiate between the number of requests being served at the moment t and the ones that are waiting in the queue. Therefore, let us define

$$X_t^i = \begin{cases} X_t^{i,1} = \{ \text{ the number of requests in waiting}\} \\ X_t^{i,2} = \{ \text{ the number of requests being served}\} \end{cases}$$

2. To be able to model the QoS we need to keep track of how many requests have been served and how many incoming requests have initiated at a given moment of time t, arbitrarily fixed in $[0, T]$. Denote by b_t^i the number of arriving requests and by s_t^i the number of served requests at the moment t.

3. Denote by u_t^i a decision control to allocate a certain amount of bandwidth at time t for a request of class i. The control is defined as follows

$$u_t^i = \begin{cases} u_t^{i,1} = \{ \text{ the number of activated requests of class } i \text{ served at } t \ \} \\ u_t^{i,2} = \{ \text{ the number of rejected requests of class } i \text{ at } t\} \end{cases}$$

4. Denote by r_t^i the resulting revenue per request of class i at moment t.

4.2.2 General assumptions

1. In each interval of time, the number of new requests considered for service is variable (not necessarily 1).

2. The assignment of bandwidths occurs at the starting point of each time unit.

3. The amount of bandwidth allocated to each request remains fixed in our model, until the request is completely served.

4. The allocation policy adopted here is that to each incoming request of class i a certain amount of bandwidth is assigned, up to the maximum capacity possible for class i, C^i.

5. Unserved requests are lost without further impact on the system.

4.2.3 Equations of the dynamics

We can formulate now the equations describing the dynamics of the system passing from one generic state $t-1$ to the next state t as follows

$$X_t^i = X_{t-1}^i + \begin{bmatrix} -1 & -1 \\ 1 & 0 \end{bmatrix} u_t^i + \begin{bmatrix} b_t^i \\ -s_t^i \end{bmatrix} \tag{4.1}$$

The first row of the equation (4.1) represents the dynamics of the requests of class i in waiting and the second row represents the dynamics of the requests of the same class being served.

4.2.4 Optimality equation

The SLA states that some QoS measurement lie within some bound (B^i) for a given percentage of requests averaged over a given long period of time. B^i represents the SLA for the i-th class of requests. The QoS is defined as a function of the decision at time t (i.e. u_t^i) and the state of the system at that moment (X_t^i). Whenever the QoS is out of bounds, a penalty applies to the provider, thus diminishing the revenue. One may assume that there is a known threshold R for the number of requests being served.

Therefore we can write

$$QoS_t^i(u_t^i, X_t^i) = B^i u_t^{i,2} + \frac{B^i}{R} X_t^{i,2} + \alpha_i X_t^{i,1}, \text{ and}$$

$$\beta^i[E\{\frac{1}{T}\sum_{t=1}^{T} QoS_t^i(u_t^i, X_t^i) - B^i\}]^+$$

where the last expression represents the penalties applicable to the provider whenever the QoS is out of bound. The choice of the numbers β^i needs to be made such that the penalty expression approximates the constraint set in the SLA.

Now we can formulate a finite horizon stochastic optimal control problem in discrete-time to maximize the total expected reward,

$$\max_{u_t} \sum_{t=1}^{T} E\{r_t^i u_t^i\} - \sum_i \beta^i \left[\frac{1}{T} E\{\sum_{t=1}^{T} [Q_o S_t^i - B^i]^+\}\right] + \sum_{t=1}^{T} \rho_i \left[\sum_i C^i X_t^i - C\right]$$

subject to the dynamics

$$X_t^i = X_{t-1}^i + \left[\begin{array}{cc} -1 & -1 \\ 1 & 0 \end{array}\right] u_t^i + \left[\begin{array}{c} b_t^i \\ -s_t^i \end{array}\right]$$

where the state vector $X_t^i \in \mathbf{R}^{J+1} \times \mathbf{R}^{J+1}$, the control $u_t^i \in \mathbf{R}^{J+1} \times \mathbf{R}^{J+1}$, (QoS) is defined as above, ρ_i is a proportionality constant corresponding to the bandwidth constraint of each class i and r_t^i represents the revenue for the class i at time t. We note that C_i defined previously, is the maximum bandwidth capacity possible for the class i and it is independent of time.

We assume that there are admissible controls u that transfer the system from X_1 to X_T and amongst this subset of admissible controls there is a control that maximizes the expected reward. Such a control will be called an optimal control u^*. Ultimately, we look for the values of the optimal control and the maximum reward.

4.3 Dynamic Programming Algorithm

A possible approach to solve the stochastic optimal control problem is the dynamic programming technique ([1]). The idea is to assign a value function $V_u(x_0)$ for each policy u such that it is equal to the total expected reward,

$$V_u(X_1) = E\{\sum_{t=1}^{T} c_t(X_{t-1}, u_t) + \phi(X_T)\} \tag{4.2}$$

where $\phi(X_T)$ is the terminal reward. The dynamic programming method allows us to construct the optimal policy u^* and, in consequence, calculate the optimal expected reward $V(X_1)$, where

$$V(X_1) = \max_u E\{\sum_{t=1}^{T} c_t(X_{t-1}, u_t) + \phi(X_T)\} \tag{4.3}$$

We assume a finite state space S, a finite control space A, and a policy $u_t(h_t)$ in terms of the history or path $h_t = (s_{t-1}, a_{t-1}, s)$, $s \in S$ and $a_{t-1} \in A_t$.

The Algorithm:

For each time t we can assign a "cost-to-go" function

$$V_t(h_t) = c_t(s_{t-1}, u_t) + \sum_{j \in S} p_t(s_t = j | s_{t-1}, a_t) V_{t+1}(h_t, a_t, j) \tag{4.4}$$

where j is a possible state at time $t-1$. The transition probability $p_t(s=j|s_{t-1},a_t)$ is the probability of going from state s_{t-1} at time $t-1$ to state j at time t. The optimal control at t is the minimizer of (4.4), i.e.

$$u_t^* \in \mathrm{argmax}_{a\in A}\Big\{c_t(x_{t-1},a_t)+\sum_{j\in S}p_t(j|s_{t-1},a_t)V_{t+1}(h_t,a_t,j)\Big\}.$$

We now the describe the algorithm:

1. Set $t=T$ and $V(h_T)=\phi_T(s_T)$ for all histories h_T

2. Let $t\to t-1$. For each h_t,

$$\begin{aligned} u_t^* &\in \mathrm{argmax}_{a\in A}\Big\{c_t(s_{t-1},a)+\sum_{j\in S}p_t(j|s_{t-1},a)V_{t+1}(h_t,a,j)\Big\} \\ V_t(h_t) &= c_t(s_{t-1},u_t^*)+\sum_{j\in S}p_t(j|s_{t-1},u_t^*)V_{t+1}(s_t,u_t^*,j) \end{aligned}$$

3. Go to step 2 when $t=2$.

4.4 The Simulation Model

In the model of Web server traffic outlined in Section 4.2, knowledge of the distribution of the number of new requests and the distribution of the number of completed requests within a given time period is required. It has been assumed that request arrivals follow a Poisson process, but simulations were performed to give estimates of the distribution of completed requests. It would also be interesting to know the behaviour of the system when certain key parameters, such as bandwidth available for transmission, are suddenly increased during operation. Details of the simulated system are as follows:

Requests arrive at the server following a Poisson process with a mean inter-arrival time of λ_A seconds. There are $J+1$ classes of requests, {class(0),...,class(J)}. These classes might represent requests for various types of data, such as video, audio, graphics or text. The probability that a given request is of type class(i) is p_i. The data being requested is simply a file. The sizes of the requested files are assumed to follow an exponential distribution, where the mean file size for requests of class i is μ_i bytes. All requests of class(i) are assumed to require B_i bytes/second of bandwidth for transmission. The total output bandwidth available for use by the server is B_T.

In a given time period, more requests may arrive than the system is capable of serving. Unserved requests are queued in order of arrival. When a request arrives at the server, if there is enough output bandwidth available to immediately serve it, and there are no other requests in the queue, then service of the request commences.

Service time for a request is given by file size divided by the bandwidth required. If a request must enter the queue, it must wait for all preceding requests in the queue to be served first, and then (possibly even longer) for there to be sufficient available transmission bandwidth, before its service will begin. The simulation model is illustrated in Figure 4.3.

4.5 Simulation Results

Two phenomena were investigated by simulation: the effect of increasing total bandwidth on the waiting time in the queue, and the number of requests whose service is completed in a given time period. The results may be found in Figures 4.4, 4.5 and 4.6.

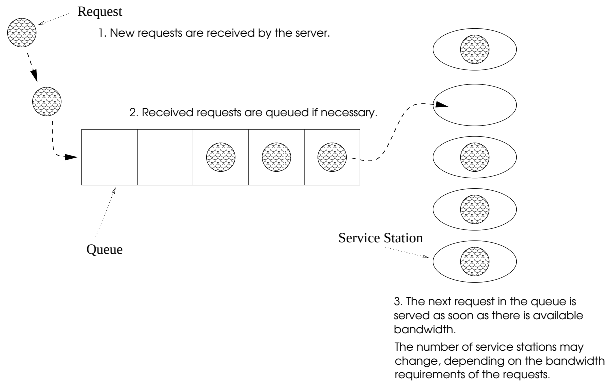

Request
1. New requests are received by the server.
2. Received requests are queued if necessary.
Queue
Service Station
3. The next request in the queue is served as soon as there is available bandwidth.
The number of service stations may change, depending on the bandwidth requirements of the requests.

Figure 4.3: Schematic of the simulated web server.

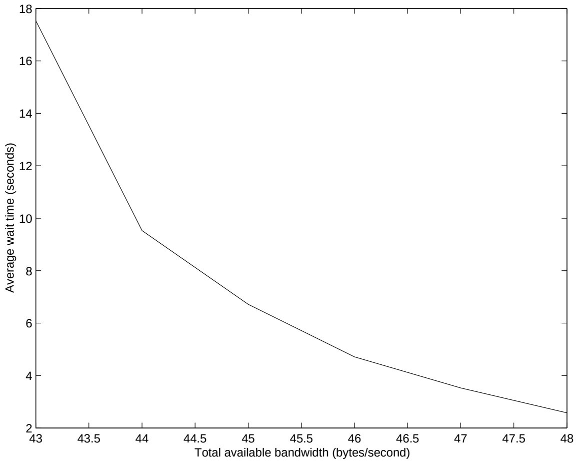


Figure 4.4: Average request waiting times versus available bandwidth. There were three different request classes, so $N = 3$. The parameter values were $\lambda_A = 1$, $\mu_0 = 10$, $\mu_1 = 30$, $\mu_2 = 60$, $B_0 = 1$, $B_1 = 2$, $B_2 = 3$, $p_0 = 0.25$, $p_1 = 0.25$, $p_2 = 0.5$, and B_T is what is plotted along the x-axis. For each value of B_T the simulation was run until 10,000 requests were served.

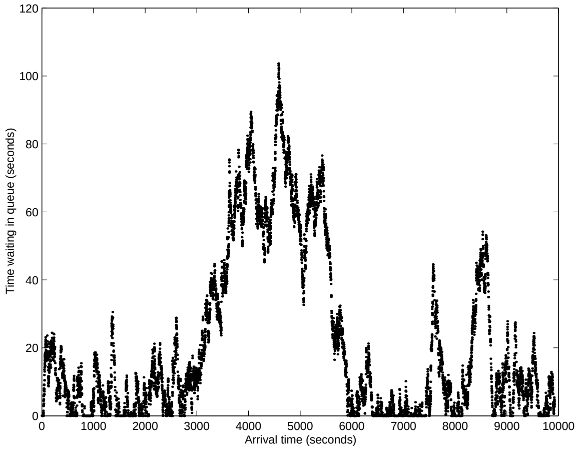


Figure 4.5: Waiting time of requests versus arrival time. In this simulation, B_T for the first 5000 requests was 24, then B_T was increased to 25. There were three classes, so $N = 3$. The parameter values were $\lambda_A = 1$, $\mu_0 = 10$, $\mu_1 = 20$, $\mu_2 = 30$, $B_0 = 1$, $B_1 = 2$, $B_2 = 3$, $p_0 = 0.25$, $p_1 = 0.25$, and $p_2 = 0.5$.

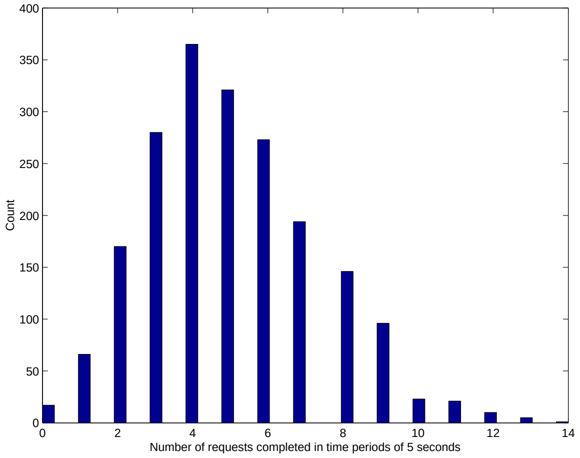


Figure 4.6: Histogram of the number of requests completed in time intervals of 5 seconds. This simulation was run until 10,000 requests were served. There were three different request classes, so $N = 3$. The parameter values were $\lambda_A = 1$, $\mu_0 = 10$, $\mu_1 = 20$, $\mu_2 = 30$, $B_0 = 1$, $B_1 = 2$, $B_2 = 3$, $p_0 = 0.25$, $p_1 = 0.25$, $p_2 = 0.5$, and $B_T = 25$.

4.6 Future work

Establishing a measure for the quality of service (QoS), for a Web hosting facility, is an extremely up-to-date problem and the authors have only approached it here by means of a very simple model. As stated in the beginning, there are a lot of possible ways to enlarge the spectrum of the model. We outline next some of these.

- To use dynamic programming techniques to solve the stochastic optimal control problem.
- To further investigate the distribution of file sizes from real data.
- To improve the model by reformulating the penalty function and the mathematical expression of the (QoS).
- To include more complex networks by extending the concepts and dimensionality of the problem.

Figure 4.7 shows the number of requests for files of different sizes on an academic Web-server[1]. This histogram shows that assuming that the size of requests has an exponential distribution is a realistic assumption.

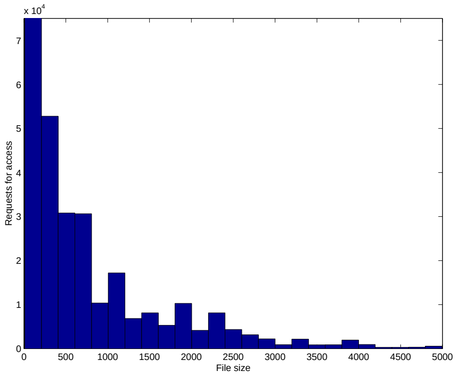


Figure 4.7: Real data file sizes distribution

References

[1] BERTSEKAS, D., **Dynamic Programming and Stochastic Control**, New York: Academic Press, 1976

[2] CHANG, Y-C., GUO, X., KIMBREL, T. and KING, A., *Optimal allocation policies for Web hosting*, IBM T.J. Watson Research Centre, P.O. Box 704, NY 10598

[3] NORTEL and BAY NETWORKS, *IP QoS-A bold new network*, September 1998, NORTEL Marketing Publications, Dept. 4262, P.O. Box 13010, Research Triangle Park, NC 27709

[4] PASCHALIDIS, I. Ch. and TSITSIKLIS, J. N., *Congestion-depending pricing of Network Services*, Technical Report, October 1998, Dept. of Manufacturing Engineering, Boston University, Boston MA 02215

[5] SUBRAMANIAN, J., STIDHAM, S. and LAUTENBACHER, C. J., *Airline yield management with overbooking, cancellation, and no-shows*, Transportation Science 33(2),1999, 147-167

[1] The data are obtained from the log file of the Web-server of the Faculty of Mathematics, University of Waterloo, June 2001

Chapter 5

Defect Analysis Using Depth from Defocus and Shape from Focus Methods

Participants: Hedley Morris (Mentor), Alex Hodge, Mahtab Kamali, Mufeed Mustafa Mahmoud, Cristina Popescu, James Rossmanith, Daniel Ryan, Ali Sanaie-Fard, Barkha Saxena.

PROBLEM STATEMENT: Newport Corporation manufactures optical equipment and in particular laser diodes. These diodes are made from semiconductor material and their operation takes place on a flat surface, approximately 200 microns square, onto which two trenches have been etched. If a number of images, at fixed focus, are taken at varying heights above the surface, the images will all be out of focus. However, the blur of each image will depend on the height above the surface. The aim of this project is to determine the diode topography from this sequence of out-of-focus images. This will enable the identification of depth anomalies that might interfere with the operation of the device. Such defects are not easily detectable by current inspection procedures.

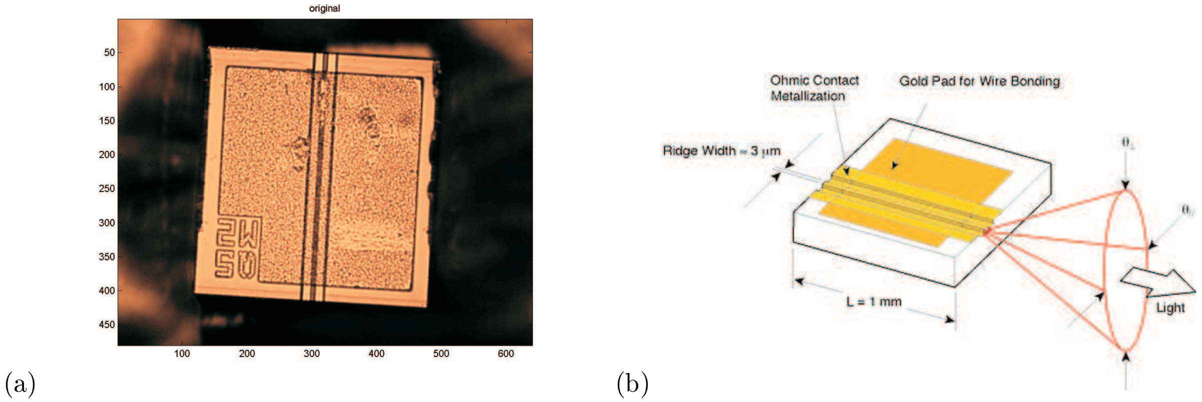


Figure 5.1: (a) A snapshot of an actual optical chip. (b) Light travels down the grooves of the optical chip. Defects in the chip may cause the light to be deflected or blocked.

5.1 Introduction

In most imaging systems, a 3D view of the real world is mapped into a 2D image. In this transformation the depth information of the image is lost and the imaging system cannot determine the full 3D structure of the image. Therefore, it is necessary to develop algorithms which can extract the 3D spatial information from a series of 2D images.

To extract the spatial information one can retrieve image characteristics by comparing two or more images of the object. If these images are obtained from placing the camera lens at different distances from the object, we refer to the depth reconstruction procedure as **shape from focus**. If the images are captured by changing the geometry of the imaging system such as change of the focal length of camera, we refer to the depth reconstruction procedure as **shape from defocus**. The shape from focus/defocus is referred to as blind de-convolution in signal processing or as image restoration in image processing.

In this specific application treated in this paper, a fiber-optic chip (shown in Figure 5.1(a)) is examined for defects by capturing 30 images at different distances from the chip with a camera. From the point of view of industry, this procedure is a relatively inexpensive way to examine the chip. The chip has to pass light through the two micro grooves (shown in 5.1(b)) cut into chip. Unfortunately, a small defect in the shape of the grooves can make the chip useless. Therefore, the objective of this paper is to estimate the spatial position of the groove by an image processing technique.

The 30 images from the chip are captured at different distances from the fiber optic chip by changing the position of the camera. The difference between two consecutive camera positions is approximately 2 nanometers. In this research project, two different methods for processing the images are examined. The first method is based on processing the array of images in the frequency domain using the Fourier transform. The second method uses a spatial transform (S-transform) which is based on a polynomial approximation of the images.

5.2 Point Spread Functions

We begin by considering a 2D picture or scene of uniform depth. The light intensity of this scene is given by $f(x, y)$. The function $g(x, y)$ describes the light intensity of an out-of-focus image of this scene. In order to understand the correlation between f and g, it is convenient to introduce the concept of a point spread function (or PSF) denoted $h(x, y)$.

Conceptually, the PSF describes how the light emitting from a point on f is distributed by the camera onto the image g. Mathematically, the PSF, $h(x, y)$, is defined as follows:

$$g = h \star f\,, \tag{5.1}$$

where $\star$ denotes the convolution operator.

In the case of an ideal pinhole camera, the PSF would be a delta function. However, in the real world, we are dealing with optical lens systems and the PSF is not this trivial. Furthermore, the PSF not only depends on the camera, but on the distance from the object to the lens; and therefore, it will be an unknown in our problem. To simplify the problem, however, some assumptions about the form of h can be made. First, h should be radially symmetric about the origin. This represents the fact that the camera should not stretch the image in some direction, or introduce some similar bias. Furthermore, we assume that our camera is a lossless system (i.e. it does not absorb any light energy in the process of collection). So, if one unit of light energy is incident on the lens, then one unit of light energy will appear in the image g,

$$\int\int h(x,y)dx\,dy = 1\,. \tag{5.2}$$

A standard approximation for h is the 2D Gaussian:

$$h_{\mathrm{g}} = \frac{1}{2\pi\sigma^2}\,e^{-\frac{x^2+y^2}{2\sigma^2}}\,. \tag{5.3}$$

Associated with the PSF is a *blur radius* which represents the radial distance that light is distributed by h. For the 2D Gaussian, the blur radius is proportional to the standard deviation σ; and therefore, throughout this paper σ will be synonymous with blur radius. Furthermore, we can then use geometric optics to relate σ to the depth D as follows:

$$\sigma = \rho\,r\,v\left(\frac{1}{F} - \frac{1}{v} - \frac{1}{D}\right), \tag{5.4}$$

where ρ, r, v, and F parameters describe the constant of proportionality between the blur radius and σ, the radius of the lens aperture, the distance from the point of perfect focus to the lens, and the focal length, respectively. Therefore, obtaining information about our PSF directly translates into depth information through camera parameters. We present below two methods for computing approximations to σ.

5.3 Method 1: A Fourier domain approach

We first consider a method based on deconvolution in Fourier space [1]. We will assume in this section that the focused image $f(x,y)$ in which we are interested and two unfocused images $g_1(x,y)$ and $g_i(x,y)$ is given by

$$g_1(x,y) = h_1(x,y)\star f(x,y) + n_1(x,y) \tag{5.5}$$

$$g_i(x,y) = h_i(x,y)\star f(x,y) + n_i(x,y)\,, \tag{5.6}$$

where $n_1(x,y)$ and $n_i(x,y)$ are random noise. In this project we will further assume that the noise is zero. Now rewriting the above equations in the frequency domain by taking a Fourier transform over the region of interest leads to the following set of equations,

$$G_1(\omega,\nu) = H_1(\omega,\nu)\,F(\omega,\nu) \tag{5.7}$$

$$G_i(\omega,\nu) = H_i(\omega,\nu)\,F(\omega,\nu)\,. \tag{5.8}$$

If we assume that the PSF is Gaussian (see Section 5.2), then the PSF and its Fourier transform are

$$h(x,y) = \frac{1}{2\pi\sigma^2}\,e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad\text{and}\quad H(\omega,\nu) = \frac{1}{2\pi\sigma^2}\,e^{-\frac{(\omega^2+\nu^2)}{2\sigma^2}}\,. \tag{5.9}$$

By combining equations (5.7) and (5.8) yields

$$\frac{G_1(\omega,\nu)}{G_i(\omega,\nu)} = e^{-\frac{1}{2}(\omega^2+\nu^2)\,(\sigma_1^2-\sigma_i^2)}\,. \tag{5.10}$$

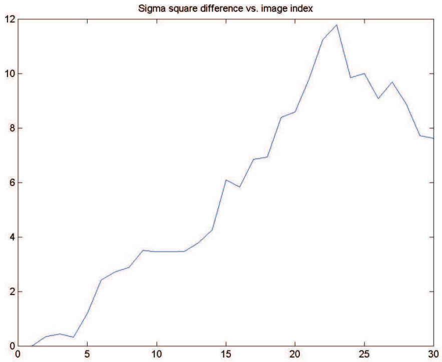


Figure 5.2: Plot of $\sigma_1^2 - \sigma_i^2$ versus the index i. We take the point where the focus changes (i.e., the index at which the maximum of the curve occurs) as a proxy for depth.

Taking the logarithm and rearranging (5.10) yields

$$\sigma_1^2 - \sigma_i^2 = \frac{-2}{\omega^2 + \nu^2} \log\left(\frac{G_1(\omega,\nu)}{G_i(\omega,\nu)}\right) . \tag{5.11}$$

A more robust formula can be obtained by integrating (5.11) over a small region in the Fourier domain:

$$C = \frac{1}{A} \int\int \frac{-2}{\omega^2 + \nu^2} \log\left(\frac{G_1(\omega,\nu)}{G_i(\omega,\nu)}\right) d\omega\, d\nu . \tag{5.12}$$

This yields to the following equation:

$$\left(\sigma_1^2 - \sigma_i^2\right) = \frac{1}{A} \int\int \frac{-2}{\omega^2 + \nu^2} \log\left(\frac{G_1(\omega,\nu)}{G_i(\omega,\nu)}\right) d\omega\, d\nu . \tag{5.13}$$

5.4 Method 2: A spatial domain approach

The second approach we consider is based not on the Fourier transform, but on the S-transform which allows us to deconvolve in physical space [2, 3]. We again use the notation of Section 5.2 to denote the unblurred image by $f(x,y)$, the PSF by $h(x,y)$, and the images by $g_i(x,y)$ where $i = 1 \ldots 30$. We begin by assuming that the image in a small region can be approximated by a bi-cubic polynomial such that

$$f(x,y) = \sum_{m=0}^{3} \sum_{n=0}^{3-m} a_{mn}\, x^m y^n . \tag{5.14}$$

Furthermore, we assume that $h(x,y)$ is a rotationally symmetric point spread function. Image $g_i(x,y)$ is obtained from the convolution of the unblurred image with the PSF,

$$g_i(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x-\xi, y-\eta)\, h(\xi,\eta)\, d\xi\, d\eta . \tag{5.15}$$

Because $f(x,y)$ is bi-cubic we can write the convolution kernel as

$$f(x-\xi, y-\eta) = \sum_{0 \le m+n \le 3} (-1)^{m+n} \frac{\xi^m\, \eta^n}{m!\, n!}\, \partial_x^m\, \partial_y^n f(x,y) . \tag{5.16}$$

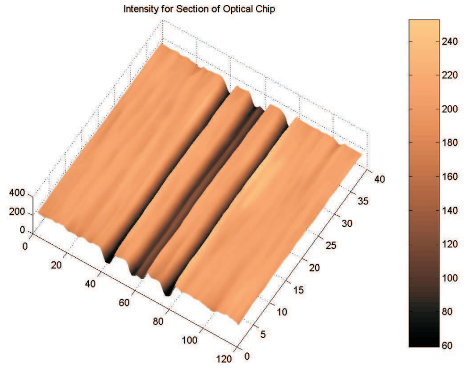


Figure 5.3: Light intensity plot for the section of optical chip used to compute a depth map.

Plugging this expression into equation (5.15) gives us that

$$g_i(x,y) = \sum_{0\le m+n\le 3} \frac{(-1)^{m+n}}{m!\,n!}\,\partial_x^m\,\partial_y^n\,f(x,y)\,h_i^{mn}\,, \tag{5.17}$$

where

$$h_i^{mn} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^m y^n\, h_i(x,y)\,dx\,dy = \int_0^{2\pi} \cos^m(\theta)\,\sin^n(\theta)\,d\theta \int_0^r r^{m+n+1}\,h_i(r)\,dr\,. \tag{5.18}$$

However, due to the periodicity of the sine and cosine, the above expression simplifies (5.17) to

$$f(x,y) = g_i(x,y) - \frac{h_i^{20}}{2}\nabla^2 f(x,y)\,. \tag{5.19}$$

Taking ∇^2 of both sides of this equation and again using the fact that f(x,y) is bi-cubic yields

$$\nabla^2 f(x,y) = \nabla^2 g_i(x,y)\,. \tag{5.20}$$

Using this information we can completely deconvolve the original integral operator and obtain the expression

$$f(x,y) = g_i(x,y) - \frac{\sigma_i^2}{4}\nabla^2 g_i(x,y)\,. \tag{5.21}$$

In the above expression, $\sigma_i^2 = 2h_i^{20}$ measures the spread of the PSF. Comparing image i to image 1 and using the fact that

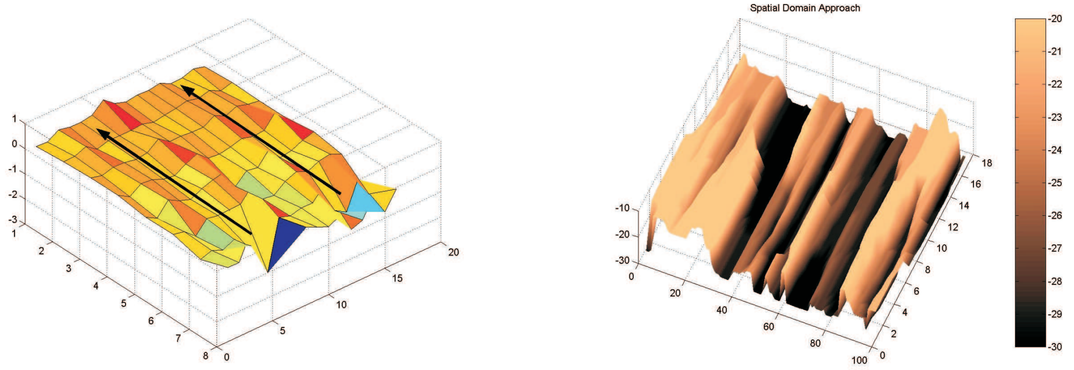
$$\nabla^2 f(x,y) = \nabla^2 g_1(x,y) = \nabla^2 g_i(x,y) \tag{5.22}$$

yields that

$$g_1(x,y) - g_i(x,y) = \frac{1}{8}\left(\sigma_1^2 - \sigma_i^2\right)\left(\nabla^2 g_1(x,y) + \nabla^2 g_i(x,y)\right)\,. \tag{5.23}$$

The difference between σ_1^2 and σ_i^2 can then be computed over a small region by integrating the above expression as follows:

$$\left(\sigma_1^2 - \sigma_i^2\right) = 8\sqrt{\frac{\int\int \left(g_1 - g_i\right)^2 dx\,dy}{\int\int \left(\nabla^2 g_1 + \nabla^2 g_i\right)^2 dx\,dy}}\,. \tag{5.24}$$



(a) (b)

Figure 5.4: Depth map proxy computed using the (a) Fourier domain approach and (b) the spatial domain approach.

5.5 Computing a proxy for the depth map

From the data set that we have been provided for this project we know that the following are true:

1. Image 1 is the furthest in distance from the object,
2. Image 1 is most out of focus,
3. $\sigma_1^2 > \sigma_i^2 \quad \forall i \neq 1$.

Therefore, the image number i for which

$$\sigma_1^2 - \sigma_i^2 \tag{5.25}$$

is a *maximum* at some spatial location is the image which is in *focus*. From this fact, we now attempt to construct a proxy depth map using a shape from focus approach. We compute $\sigma_1^2 - \sigma_i^2$ as a function of the index i and look for a maximum. A higher index will correspond to a deeper part of the object. An example of $\sigma_1^2 - \sigma_i^2$ as a function of the index i is shown for a particular pixel in Figure 5.2. In this case $i = 23$ corresponds to the focused image. Plotting the maximum i as a function of space produces a proxy for the depth map.

To test this procedure on the full problem, we now carry out the above process pixel by pixel for the light intensity map shown in Figure 5.3. The resulting depth map proxy obtained by the Fourier domain approach is shown in Figure 5.4(a) and the spatial domain approach in Figure 5.4(b). The spatial domain approach seems to produce a better result. To demonstrate that we are able to detect the channels, we average the depth map computed by the spatial domain approach along the direction of the channels, collapsing our information into the plot shown in Figure 5.5. In this plot clear dips occur in the locations where we expect the channels to be.

5.6 Obtaining depth from blur

Up to this point we were not able to compute a true depth map, but instead only a proxy for the depths using σ^2 differences. In this section we focus on obtaining true depths from our previously calculated $\sigma_1^2 - \sigma_i^2$. Re-arranging equation (5.4) gives an expression for the depth in terms of camera parameters and $\sigma_1^2 - \sigma_i^2$,

$$D = \frac{a_1 - k_2\sqrt{\sigma_1^2 - c_i^2}}{c_i^2 + a_2}\ , i = 1, \ldots, 30\,. \tag{5.26}$$

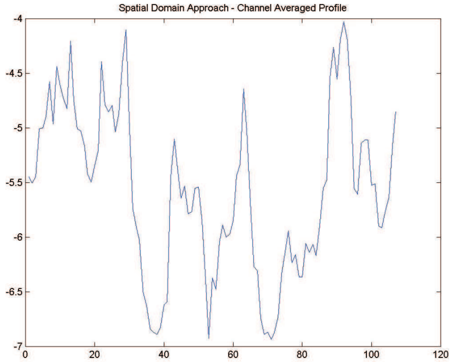


Figure 5.5: Approximate depth map averaged along channel direction.

For this equation we need the following definitions:

$$\begin{aligned} c_i &\equiv \sigma_1^2 - \sigma_i^2 \ \text{ for } \ i = 1, \ldots, 30 && (5.27)\\ a_1 &\equiv k_1 k_2^2 && (5.28)\\ a_2 &\equiv k_1^2 k_2^2 - \sigma_i^2 && (5.29)\\ \sigma_1^2 &\equiv \text{spread for the first image} && (5.30)\\ k_1 &\equiv \frac{1}{F} - \frac{1}{v} && (5.31)\\ k_2 &\equiv \rho r v\,. && (5.32) \end{aligned}$$

The true depths can now be computed from the σ^2 differences by solving the above equations for D and all the unknown camera parameters (ρ, r, v, and F) in the least squares sense. In other words, we are able to solve equation (5.26) for D and all the unknown camera parameters because we are given several images of the object at different heights (i.e., $i = 1, \ldots, 30$).

In terms of the project outlined in this paper, we were not able to apply the above method for computing true depths to the data computed in Sections 5.3 and 5.4 due to time constraints. Future work should focus on applying the above least squares analysis on the previously calculated proxy depth maps.

5.7 Conclusions

In this paper we developed two distinct methods for estimating the depth profile of a series of 2D images of a semiconductor chip. We have found that the method based on the spatial transform deconvolution method produces more accurate results than the more traditional Fourier transform approach. Although we did not have enough to time to finish the task, we also worked on developing a least squares approach for translating the depth maps produced by the deconvolution methods into physical depth maps.

Although we were able to obtain some results with the spatial transform deconvolution method, our numerical simulations fail to produce results that are accurate enough for detecting defects in the grooves of the chip. We believe that most of this is due to the fact that there exists significant noise in our data. Our data set had only a three pixel width across the channel. This makes the task of making a detailed map within the channel very difficult and allows for less filtering/smoothing of the image without destroying the channel information.

References

[1] S. Chaudhuri and A.N. Rajagopalan. *Depth from defocus: a real aperture imaging approach.* Springer-Verlag, 1999.

[2] M. Subbarao and G. Surya. Depth from defocus: a spatial domain approach. Technical report No. 92.12.03, Computer Vision Laboratory, Electrical Engineering Department, SUNY, Stony Brook, NY.

[3] D. Ziou. Passive depth from defocus using a spatial domain approach. Tech. Report, DMI, Universite de Sherbrooke, 1997.

Chapter 6

Ice Accretion

Participants: Tim Myers (Mentor), Thomas Brakel, Brian Corbett, Aude Espesset, Jihyoun Jeon, Mehdi Hadj-Karim-Kharrazi, Ali Rasekh, J. F. Williams.

PROBLEM STATEMENT: Ice accretion on surfaces is a serious problem in for any surfaces in cold condtions, such as aircraft at high altitude and structures in harsh winter environments. The problem is to model the formation of ice on surfaces from super-cooled water droplets.

6.1 Introduction

Ice accretion can cause the downing of both power lines and airplanes, unchecked it can cause a huge cost, both financial and human. Understanding the mechanism by which ice forms and how the various physical parameters affect this growth is of key importance to design of de-icing systems for aircraft and adequate structures to withstand harsh environments.

In this problem we have many competing physical phenomena and we must consider them all in order to construct a valid model. To make the problem tractable we first make many simplifying assumptions:

1. The surface is flat and uniform,
2. The incoming droplets are uniform in space and time,
3. There is little water; water motion is unimportant,
4. The surface is clean,
5. The droplets are pure liquid water,
6. Trapped air affects only the ice density, it is of no thermodynamic importance,
7. Mass losses due to evaporation are negligible,
8. The substrate and the air stream have very large thermal masses,
9. Mass is conserved,
10. Energy is conserved.

Mathematically assumptions (1) – (3) imply that a one dimensional reduction is reasonable. Assumptions (4) – (5) mean that there will be a sharp interface between the ice and water at exactly the freezing temperature of water, this also means that all heat release from the freezing of the droplets will occur at the upper ice surface. The mass balance is easier to do assuming (6). Making assumption (7) means that we do not need to consider the temperature problem in the substrate or the air as they will remain constant for all time. The last two assumptions give us equations to solve once we have decided on all the important energy balance terms.

Because we are interested in ice accretion on both land-based structures such as power cables and towers and also on airplane wings we must consider many seemingly trivial affects. Upon consultation of the aerodynamics literature [1] one finds that the relevant gain terms are the latent heat of freezing at the ice surface, the kinetic energy of the incoming drops and the aerodynamic heating due to local compression of the air. Energy is lost in proportion to the difference of the temperature of the upper surface and the air due to sublimation or evaporation, cooling due to the thermal mass of the incoming drops and surface convection. Energy is also transported by conduction. Expressions for all these mechanisms are presented in the Table 1.

Table 1: Energy balance terms

Energy inputs	1. Kinetic energy of incoming drops	$Q_k = \frac{\dot{M}W^2}{2}$
	2. Aerodynamic heating	$Q_a = \frac{rH_wW^2}{2c}$
	3. Latent heat of freezing	$Q_f = \rho_i L\dot{h}_i$
Energy outputs	1. Evaporation/Sublimation	$Q_e = \chi e_0(T - T_a)$
	2. Cooling by incoming droplets	$Q_d = \dot{M}c_w(T - T_a)$
	3. Surface Convection	$Q_s = H(T - T_a)$
Energy transport	Conduction adds or removes heat	$Q_c = \kappa\frac{\partial T}{\partial x}$

Table 2: Parameter values

Parameter	Physical meaning	Value	Units
c_a	Specific heat of air	1014	J/kg K
c_i	Specific heat of ice	2050	J/kg K
c_w	Specific heat of water	4218	J/kg K
L_F	Latent heat of fusion	3.344×10^5	J/kg
e_0	Vapour pressure constant	27.03	Pa/K
W	Wind speed	90	m/s
r	Local recovery factor	.55	
H_{aw}	Heat transfer between air and water	500	W/m^2 K
H_{ai}	Heat transfer between air and ice	500	W/m^2 K
H_{is}	Heat transfer between ice and substrate	1000	W/m^2 K
κ_i	Conductivity of ice	2.18	W/m K
κ_w	Conductivity of water	0.571	W/m K
ρ_w	Density of water	1000	kg/m^3
ρ_i	Density of ice	900	kg/m^3
χ	Evaporation coefficient	11.0	m/s
k_i	Thermal diffusivity in ice	2.18	m^2/s
k_w	Thermal diffusivity in water	0.571	m^2/s
T_a	Ambient air temperature	230 to 265	K
T_s	Substrate temperature	230 to 265	K
T_a	Freezing temperature of water	273	K
$\dot{M}$	Mass transfer rate	.045	kg/s m^2

To properly model this situation we now need only define a heat equation for each phase and then apply the appropriate energy balance at each interface. The mass balance requires that the total amount of material which has fallen remains on the surface in either liquid or solid form.

The meaning and values of all parameter values are described in Table 2. Subscripts are used to denote the phase or substance. For example, the heat lost through surface convection is given by $Q_s = H_{ai}(T - T_a)$, where H_{ai} is the heat transfer coefficient from air to ice, T_a is the fixed air temperature and T is the temperature variable.

Because we have a sharp interface the droplets freeze immediately upon impact at the upper surface. Instantly the ice will be at the substrate temperature which we assume will be well below freezing. As more droplets come in more latent heat is released and we expect the temperature at the surface to slowly increase until eventually water forms. One of the key objectives (and successes!) of this work is to determine the thickness at which this water first appears. This also suggest that we need to break the problem down into two cases, firstly when there is no water and then when both phases are present.

6.2 Model equations

From the energy and mass balances described in Section 6.1 we may write down the following equations governing our system using the geometry and notation as described in Figure 6.1. Here we are taking z to be the space direction and t for time as the independent co-ordinates. The dependent co-ordinates are explained in table

Table 3: Dependent variables

Variable	Physical meaning
$B(t)$	Thickness of the base of ice
$h(t)$	Height of the water layer
$T(z,t)$	Temperature in the ice layer
$\theta(z,t)$	Temperature in the water layer

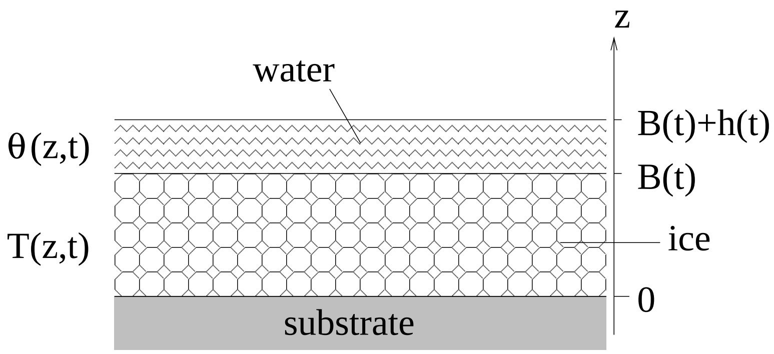


Figure 6.1: Diagram of ice accretion model

We begin the modelling with the no water case, $h = 0$.

6.2.1 No water present

Because there is no water present the conservation of mass states

$$B(t) = \frac{t\dot{M}}{\rho_i}. \tag{6.1}$$

In the ice region we have a diffusion equation for the temperature,

$$\kappa_i \frac{\partial T}{\partial t} = \rho_i c_i \frac{\partial^2 T}{\partial z^2}. \tag{6.2}$$

To be able to solve this problem we also impose the boundary conditions

$$\left.\frac{\partial T}{\partial z}\right|_{z=0} = H_{is}(T - T_s), \tag{6.3}$$

$$-\kappa_i \left.\frac{\partial T}{\partial z}\right|_{z=B} = (H_{ai} + \dot{M}c_w + \chi e_0)(T - T_a) - \left(\frac{rH_{ai}W^2}{2c_a} + \frac{\dot{M}W^2}{2} + \dot{M}L_f\right). \tag{6.4}$$

Physically the initial condition $h = 0$ states that there is initially no water and the boundary conditions (6.3)–(6.4) balance the energy loss and gain terms at the ice, substrate and ice, air interfaces repsectively.

6.2.2 Water present

We again begin by writing down the governing equations, conservation of mass and diffusion equations for the temperature in the two phases.

$$\dot{M}t = \rho_i B + \rho_w h \tag{6.5}$$

$$\rho_w c_w \frac{\partial^2 \theta}{\partial z^2} = \kappa_w \frac{\partial \theta}{\partial t} \tag{6.6}$$

$$\rho_i c_i \frac{\partial^2 T_i}{\partial z^2} = \kappa_i \frac{\partial T_i}{\partial t}. \tag{6.7}$$

This system provides only three equations in the four unknowns, $\theta(z,t)$, $T(z,t)$, $B(t)$ and $h(t)$. Considering the energy balance at the interface between the ice and water layer gives a Stefan condition [2] for the motion of the interface,

$$\rho_i L_f \frac{dB}{dt} = \kappa_i \frac{\partial T}{\partial z} - \kappa_w \frac{\partial \theta}{\partial z}. \tag{6.8}$$

The system is now closed with the addition of initial conditions in time,

$$B(t_w) = B_w, \tag{6.9}$$

$$h(t_w) = 0 \tag{6.10}$$

where t_w is the time at which water first appears and B_w the thickness, and boundary conditions in space,

$$\left.\frac{\partial T}{\partial z}\right|_{z=0} = H_{is}(T - T_s) \tag{6.11}$$

$$T(B,t) = T_f \tag{6.12}$$

$$\theta(B,t) = T_f \tag{6.13}$$

$$-\kappa_w \left.\frac{\partial \theta}{\partial z}\right|_{z=B+h} = -\left(\frac{rH_{aw}W^2}{2c_a} + \frac{\dot{M}W^2}{2}\right) + (H_{aw} + \dot{M}c_w + \chi e_0)(\theta - T_a). \tag{6.14}$$

6.2.3 Complete problem

The solution strategy is to solve the ice-only problem until the temperature at the ice air interface is the freezing temperature which indicates that water has formed. At this time, t_w, we then solve the combined problem until some large final time, t_f. Of primary interest is the thickness of the ice at time t_w.

6.3 Non-dimensionalization

Because of all the physical parameters in the problem it is difficult to discern the relevant importance of the terms. To compare the relative values we introduce non-dimensional variables and recast equations 6.1–6.14. Dimensionless variables are indicated with a superscript $\hat{}$.

We begin by rescaling the coordinates t and z. An arbitrary timescale is used such that the conservation of mass in the water only case reduces to $\hat{z} = \hat{t}$. This implies that the temporal scale τ is defined such that $\tau = t_w$ where t_w is the time at which water first appears.

$$\tau\hat{t} = t \qquad \lambda\hat{z} = z \quad \text{where } \lambda = \frac{\dot{M}\tau}{\rho_i}.$$

This choice of spatial scaling sets

$$\lambda\hat{B} = B \qquad \lambda\hat{h} = h.$$

To simplify the temperature in the ice region we rescale the temperatures as

$$\hat{T} = \frac{T - T_s}{T_f - T_s} \qquad \hat{\theta} = \frac{\theta - T_f}{T_f - T_s}.$$

With these definitions we consider the problem in the two different cases.

6.3.1 No water

The system 6.1–6.4 may be rewritten as

$$\begin{aligned}
\hat{B} &= \hat{t} \\
\epsilon_1 \frac{\partial \hat{T}}{\partial \hat{t}} &= \frac{\partial^2 \hat{T}}{\partial \hat{z}^2} \\
\epsilon_2 \left.\frac{\partial \hat{T}}{\partial \hat{z}}\right|_{\hat{z}=0} &= \hat{T} \\
\left.\frac{\partial \hat{T}}{\partial \hat{z}}\right|_{\hat{z}=\hat{B}} &= -\alpha_1 \hat{T} + \alpha_2.
\end{aligned}$$

The above parameters take the values,

$$\begin{aligned}
\epsilon_1 &= \frac{\lambda^2 \kappa_i}{\tau \rho_i c_i} \\
\epsilon_2 &= \lambda H_{is} \\
\alpha_1 &= \lambda \frac{H_{ai} + \dot{M} c_w + \chi e_0}{\kappa_i} \\
\alpha_2 &= \frac{\lambda}{\kappa_i} \left(\frac{r H_a i W^2}{2 c_a} + \frac{\dot{M} W^2}{2} + \dot{M} L_f - \frac{T_s - T_a}{T_f - T_s} (H_{ai} + \dot{M} c_w + \chi e_0) \right).
\end{aligned}$$

6.3.2 Water present

The system 6.5–6.14 may be rewritten as

$$\begin{aligned}
\hat{B} &= \hat{t} - \beta_3 \hat{h} \\
\epsilon_1 \frac{\partial \hat{T}}{\partial \hat{t}} &= \frac{\partial^2 \hat{T}}{\partial \hat{z}^2} \\
\epsilon_3 \frac{\partial \hat{\theta}}{\partial \hat{t}} &= \frac{\partial^2 \hat{\theta}}{\partial \hat{z}^2} \\
\epsilon_2 \left. \frac{\partial \hat{T}}{\partial \hat{z}} \right|_{\hat{z}=0} &= \hat{T} \\
\frac{d\hat{B}}{d\hat{t}} &= \gamma_1 \left. \frac{\partial \hat{T}}{\partial \hat{z}} \right|_{\hat{z}=\hat{B}} - \gamma_2 \left. \frac{\partial \hat{\theta}}{\partial \hat{z}} \right|_{\hat{z}=\hat{B}} \\
\hat{T}(\hat{z} = \hat{B}) &= 1 \\
\hat{\theta}(\hat{z} = \hat{B}) &= 0 \\
\left. \frac{\partial \hat{\theta}}{\partial \hat{z}} \right|_{\hat{z}=\hat{B}+\hat{h}} &= -\beta_1 \hat{\theta} + \beta_2.
\end{aligned}$$

The new parameters take the values,

$$\begin{aligned}
\epsilon_3 &= \frac{\lambda^2 \kappa_w}{\tau \rho_w c_w} \\
\gamma_1 &= \frac{\rho_i \kappa_i}{\tau \dot{M}^2 L_f (T_f - T_s)} \\
\gamma_2 &= \frac{\rho_w^2 \kappa_w}{\tau \dot{M}^2 L_f (T_f - T_s) \rho_i} \\
\beta_1 &= \lambda \frac{H_{aw} + \dot{M} c_w + \chi e_0}{\kappa_w} \\
\beta_2 &= \frac{\lambda}{\kappa_w} \left(\frac{r H_a i W^2}{2 c_a} + \frac{\dot{M} W^2}{2} - \frac{T_a - T_f}{T_f - T_s} (H_{aw} + \dot{M} c_w + \chi e_0) \right) \\
\beta_3 &= \frac{\rho_w}{\rho_i}.
\end{aligned}$$

Taking the values from Table 2 for ϵ_1, ϵ_2 we find that for $\tau \ll 400s$ both these terms may be neglected. ϵ_3 also remains small when the water layer is thin. In the next section we will consider the solutions for the problem as stated above but for $\epsilon_1 = \epsilon_2 = \epsilon_3 = 0$.

Please note the hats, $\hat{}$, over the dimensionless variables will be dropped from this point for the remainder of this and the next secion for notational convenience.

6.4 Asymptotic Solution

6.4.1 Initial Stage (no water)

In the asymptotic case the temperature profiles are linear in the thickness z, but not in time. The profile is given by

$$\begin{aligned} \frac{\partial^2 T}{\partial z^2} &= 0 \\ T(0) &= 0 \\ \left.\frac{\partial T}{\partial z}\right|_{z=B} &= \alpha_1 - \alpha_2 T. \end{aligned}$$

which is easily solved to give

$$T(z,t) = \frac{\alpha_1 z}{1 + \alpha_2 B(t)}.$$

To find the thickness, B_w at which water first appears we set $T(B_w, t_w) = 1$ which defines

$$B_w = \frac{1}{\alpha_1 - \alpha_2}. \tag{6.15}$$

6.4.2 Model with ice and water

In the asympotic regime both temperature profiles are linear.

$$\frac{\partial^2 T}{\partial z^2} = 0 \qquad \frac{\partial^2 \theta}{\partial z^2} = 0.$$

With the boundary conditions

$$T(0) = 0 \quad T(B) = 1$$

and

$$\theta(B) = 1 \quad \left.\frac{\partial \theta}{\partial z}\right|_{z=B+h} = -\beta_1 \theta + \beta_2$$

the profiles are given by,

$$T = \frac{z}{B}, \qquad \theta = 1 + \frac{\beta_2}{1 + \beta_1 h}(z - B).$$

Notice that we have solved the temperature profiles without invoking the Stefan condition! Substituing the profiles into the Stefan condition gives an ODE for the thickness of the ice layer.

$$\Gamma \frac{\partial B}{\partial t} = \frac{1}{B} - \frac{\beta_2}{1 + \beta_1 h}.$$

Once this has been computed the profiles may be recovered. Recall that the height of the ice layer and that of the water layer are related by the conservation of mass equation 6.5.

6.4.3 Asymptotic results in dimensional variables

In order to compare to experiment and the full numerical simulations we must convert our dimensionless results back into their dimensional form.

The height at which water first appears is given by,

$$B_w = \frac{\kappa_i (T_f - T_s)}{\frac{r H_{ai} W^2}{2} + \frac{\dot{M} W^2}{2} + L_f \dot{M} - (T_f - T_a)(H_a i + \dot{M} c_w + \chi e_0)}.$$

In terms of the given parameter values this works out to

$$B_w \sim 2.5mm$$

which agrees very well with experiment [3]! For these values one also finds $t_w = \tau \sim 50s$ which is well below the upper limit of $400S$.

Setting $B_w = \infty$, or $\alpha_1 = \alpha_2$ in (6.15) we can determine the temperature difference such that no water ever forms, The temperature difference beyond which no water forms is given by

$$T_a - T_s = \Delta T = \frac{\frac{rH_{ai}W^2}{2} + \frac{\dot{M}W^2}{2} + L_f\dot{M}}{H_a i + \dot{M}c_w + \chi e_0)}.$$

This gives an approximate value of,

$$\Delta T \sim -16C.$$

6.5 Numerical solution of the complete problem

The numerical solution of this problem proceeds in the same stages as the analytic solution. First the ice-only problem is considered and then once water has appeared, the combined problem is solved. Because the interface is unknown and no modelling of the exterior air region is done, we propose to use a coordinate transformation to map the physical layers, $[0, B(t)]$ and $[B(t), B(t) + h(t)]$ whose thicknesses vary with time to fixed computational intervals on $[0, 1]$. This is done by defining the coordinates

$$\begin{aligned} x &= \frac{z}{B(t)} \\ y &= \frac{z - B(t)}{h(t)}. \end{aligned}$$

This transformation causes two difficulties, it makes the equations to be solved more complicated and it is singular as the layer thicknesses tend to zero. The first is not a significant issue numerically and the second may be handled either by using an implicit method or adding an arbitrarily thin base layer. For convenience we shall employ the latter strategy. Because the timescale $\tau = t_w$ is not known a priori we must solve the full dimensional system 6.1–6.4. Again we begin with the case where water has yet to form.

6.5.1 No water present

Because the ice layer grows at a constant rate

$$B(t) = \frac{t\dot{M}}{\rho_i} + B_0$$

upon defining

$$f(x,t) = T(z,t)$$

we have a simple PDE for f,

$$f_t = \frac{\rho_i x}{\dot{M}t} f_x + \frac{\rho_i^3 c_i}{\kappa_i \dot{M}^2 t^2} f_{xx}.$$

This is solved with the boundary conditions

$$t\frac{\partial T}{\partial x}\bigg|_{z=0} = \frac{\rho_i H_{is}}{\dot{M}}(T - T_s),$$

$$-\frac{\dot{M}\kappa_i}{\rho_i} t \left.\frac{\partial T}{\partial z}\right|_{z=B} = (H_{ai} + \dot{M}c_w + \chi e_0)(T - T_a) - \left(\frac{rH_{ai}W^2}{2c_a} + \frac{\dot{M}W^2}{2} + \dot{M}L_f\right).$$

and an initial artificial layer

$$B(0) = B_0 \ll 1 \qquad (\sim 10^{-6})$$

until $f(1,t_w) = T_i$ at which point water will form at the upper surface. This relationship defines t_w. Then we move on to the coupled problem.

6.5.2 Water present

Now that the ice growth rate is no longer constant we need to solve the ODE for the free boundary as well. Defining

$$g(y,t) = \theta(z,t)$$

we need to solve the coupled system

$$f_t = \frac{B'x}{B} f_x + \frac{\rho_i c_i}{\kappa_i B^2} f_{xx},$$

$$g_t = \frac{y+B}{h} g_y + \frac{\rho_w c_w}{\kappa_w h^2} g_{yy},$$

$$\dot{M}t = \rho_i B + \rho_w h$$

and

$$\rho_i L_f B' = \frac{\kappa_i}{B} f_x - \frac{\kappa_w}{h} g_y.$$

The last equation is evaluated across the interface $x = 1$, $y = 0$. Because of the non-local nonlinearity in the system an implicit-explicit formulation should be used where the PDEs for f and g are integrated with a Crank-Nicholson scheme keeping B and h fixed, after each step B and h are updated.

6.6 Conclusions

In this report we have constructed a one-dimensional model for the growth of ice and water layers due to incoming supercooled drops. This model accounts for all significant physical effects and is hence somewhat unwieldy. Instead of analysis of the full set of equations an asymptotic reduction was made to produce a mathematically tractable model. This reduces a coupled system of PDEs to a single order ODE for the thickness of the ice layer. Once this has been numerically calculated the temperature profiles and water layer thickness may be easily obtained.

The model predicts that initially a layer of ice forms on the surface until enough latent heat has been released to melt the surface layer. A simple expression for this thickness at the onset of water formation was derived and found to agree not only with the numerical simulations but experimental data as well. From this expression one can easily see the relevant importance of the considered physical effects. Additionally we derived a critical temperature difference between the surface and air temperatures for which no water forms. This also agrees well with experimental evidence!

A general scheme for integration of the full system was attempted in MatLab but due to the large number of physical parameters, the difficulty of the problem and time constraints no satisfactory results were obtained. The general scheme is sound and, given sufficient time we believe that reliable results could be checked against the asymptotics.

References

[1] Messinger, B.L., Equilibrium temperature of an unheated icing surface as a function of air speed. Jnl. Aero. Sci. Jan. 1953.

[2] Crank, J. D., Free and moving boundary value problems. Oxford Science Publications, 1984.

[3] Myers, T.G., private communication.

Chapter 7

Estimating Risk-Neutral Probability Measures

Participants: Miro Powojowski (Mentor), Joel Hanson, Kristen Jaskie, Judy Lai, Shuqing Liang, Hassan Masum, and Rafael Meza.

PROBLEM STATEMENT: The Black-Scholes formula is commonly used to price options, due to its ease of use and comprehensibility. However, the formula assumes that the volatility of the underlying security is constant across strike prices, which is empirically not the case. For instance, the "volatility smile" refers to the fact that options which are far-from-the-money often trade at higher implied volatilities compared to options which are close-to-the-money.

It's therefore of interest to find a probability measure on option strike prices, such that using this probability measure smooths the implied volatility of the option to a constant value. This probability measure is called the Risk-Neutral Probability Measure (RNPM) .

We looked at several possible methods for finding the RNPM, and explored two in some detail: histograms and Hermite polynomials. A regression algorithm was then implemented for fitting parameterizable histograms to the observed option price data. Background, methodology, results, ideas for future work, and references follow.

7.1 Problem Background

7.1.1 Assumptions and Binomial Tree Models

Real securities are extremely complex and hence require some level of simplification. We therefore assume:

- A single-period model with two time steps: now (time 0), and some future time T.
- A risk-free rate of interest r exists (e.g. government bonds).
- No arbitrage opportunities exist (i.e. an astute investor cannot make money with zero risk through exploiting pricing imperfections).
- The securities in question are highly liquid and tradeable at will.

Suppose the stock has price S now, and can rise or fall in price into one of 2 states at time T, with prices S_{up} and S_{down} respectively: this gives the 1-step Binomial Tree model. (The model can be generalized to multiple time steps and higher-fanout trees in intuitive ways, although the computational effort involved grows rapidly due to the exponentially-increasing size of the tree.)

We can use this simple model to price the current value of an option to buy the stock at time T. To do this, we create a particular portfolio of the stock and the option which has no uncertainty as to its future value. Since this "replicating portfolio" has no risk, it must earn a rate of return equal to the risk-free rate.

Denote the current (unknown) price of the option by O; denote the payoff on the option if S goes up or down by O_{up} and O_{down} respectively. Consider a portfolio consisting of a long position in N shares of the stock and a short position in one option. We can then express the value of this portfolio under the two possible outcomes:

1. Stock goes up: S_{up} N - O_{up}.
2. Stock goes down: S_{down} N - O_{down}.

The value will be equal (and the portfolio will therefore be riskless) when these two quantities are equal, i.e. when

$$N = \frac{O_{up} - O_{down}}{S_{up} - S_{down}} \tag{7.1}$$

This could be considered as the ratio of option price change to stock price change between the two outcomes.

If the interest-free rate is r, the present value of this portfolio is found by discounting backward:

$$PV = e^{-rT}(S_{up}N - O_{up}) \tag{7.2}$$

(The first term comes from the formula for continuous compounding: $(1+r/m)^{mT}$ approaches e^{rT} in the limit.) Since the portfolio cost was S N - O, we can equate the present value of the portfolio with its current cost and solve for O:

$$O = e^{-rT}(pO_{up} - (1-p)O_{down}) \tag{7.3}$$

where p is a derived quantity equal to

$$\frac{e^{rT} - (S_{down}/S)}{(S_{up}/S) - (S_{down}/S)} \tag{7.4}$$

So, the point of this is that we can price an option using a 1-step binomial model, if we can observe the prices of the option and stock under both conditions at time T, and the price of the stock now. This is not too useful for a single time step, but if we are investing over a long period of time (and if market conditions stay similar), then we can use a series of 1-step observations to derive "expected" option prices. (Note that the option price does not depend on the probability of the stock price increasing or decreasing; this is basically because the option is being valued relative to the underlying stock and not in absolute terms.)

7.1.2 Risk-Neutral Probability

The variable p above can be interpreted as the probability of an up movement in the stock price; thus the quantity

$$(\mathrm{p}\ \mathrm{O}_{up}\ \text{-(1-p)}\ \mathrm{O}_{down})$$

is the expected payoff of the option. With this interpretation, the expected stock price at time T is

$$\mathrm{p}\ \mathrm{S}_{up}\ +\text{(1-p)}\mathrm{S}_{down}$$

Substituting the previous definition of p, we get S e^{rT}. In other words, the stock price grows at the risk-free rate.

This illustrates the principle of risk-neutrality. In a risk-neutral world, investors require no compensation for risk but are concerned only with the expected return of securities. For a completely risk-neutral investor, a government bond with guaranteed payoff is equivalent to a highly leveraged speculative portfolio, as long as both have the same expected payoff.

(NB: Risk-neutrality is obviously not true in general in the real world, where e.g. investors have finite bankrolls and are usually risk-averse toward going bankrupt. However, it may be a reasonable approximation in a sizeable range of expected payoff values for a large investor, where the investor's utility function of wealth is relatively flat.)

7.1.3 Valuing Options

What we are looking for, then, is a measure under which risk-neutral valuation holds. We applied these concepts to the valuation of European options, which can only be exercised at expiry. (In contrast, most traded options are American options which can be exercised at any time prior to expiry.) There are two main reasons for the use of European options in math finance:

1. Analytical tractability: since there is a single expiry time, the investor has only two possible actions, i.e. to exercise or not at time T.

2. The counterintuitive theoretical result that it is suboptimal to exercise American call options prematurely, due to both foregone interest on the cash used to purchase the call option and downside risk from holding the stock instead of the option. (Note that, due to our assumption of stock prices following geometric Brownian motion, this result assumes the investor has no advantage over other investors in picking undervalued stocks and predicting the future path of stock prices.)

For our case of valuing European options, the risk-neutral measure could intuitively be considered as a probability density function (PDF) of the possible option prices at expiry; integrating this PDF with the value of the option payoff gives the value of the option. Formally,

$$CallPrice = \exp(-rT)\int_K^\infty (S-K)dF(S) \tag{7.5}$$

where CallPrice denotes the market price of the call option, K the corresponding strike price, and S the price of the underlying stock. (We integrate over all stock values that give us a non-negative return on exercising the option.)

7.1.4 Our Data

We had several data files to work with:

- Option Prices (both calls and puts) for the S&P 100, S&P 500, GM, and Microsoft. As with much financial data, there are limitations in the data, e.g. lack of information on thinly traded options. (Note that some practitioners argue that more weight should be given during analysis to the implied volatility of close-to-the money options; far-from-the-money options tend to have less volatility anyway.)

- Interest rate data from 1997 to 2001. The data included, for each date, a selection of rates for different periods; it is thus possible to view the term structure of (future) interest rates at each day in the past. We can interpolate to estimate interest rates that are not explicitly given.

- Some graphs of implied volatility, and miscellaneous supporting data.

7.2 Models

7.2.1 Geometric Brownian Motion and the "Volatility Smile"

Our basic continuous case is geometric Brownian motion, where stock prices follow a random walk with positive bias (i.e. stock price changes are normally distributed with positive mean).

This implies that the distribution of asset prices in the future, conditional on current asset prices, should be lognormal (i.e. the instantaneous rate of return on the asset should be normal). As a consequence of this fact and the risk-neutrality principle, a graph of the strike price of an option against the implied volatility of the option (or the graph of observed log return against implied volatility) should be flat.

The implied volatility is a derived parameter calculated using the Black-Scholes equation for option pricing; one assumes Black-Scholes holds, observes prices, and then solves for the volatility variable in the equation. It's important to note therefore that implied volatility is not a directly observable parameter, but rather a derived parameter which is only provably valid if the model assumptions on the option's behavior hold. (Implied volatility may still be useful if the option's behavior is "close enough" to what our models say it should be ... note that defining "close enough" is also an open research problem).

Both these implications are violated in actual markets. In particular, the "volatility smile" is a phenomenon in which the implied volatility of options close to the money is less than options far from the money; these fat tails may come from larger numbers of extreme market events than predicted, or non-neutral investor risk preferences, or systematically biased expectations of future market events.

7.2.2 Martingales and Asset Pricing

A Martingale is a process for which $\mathrm{E}[\mathrm{X}_{t+1} \mid \mathrm{X}_1,..., \mathrm{X}_t] = \mathrm{X}_t$; you expect the process to generate an outcome with expected value equal to the most recent outcome, but change your expectations to match whatever actually happens. As an important example, a sum of successive IID variables, each of which has mean 0, will give a martingale. (Note that martingales are not necessarily Markov, but if they are then we have a very nice situation indeed.)

The relevance to risk-neutral valuation comes from a series of key results:

- The no-arbitrage theorem tells us that the absence of arbitrage opportunities in the market implies the existence of an equivalent measure under which discounted stock prices are martingales.

- The completeness theorem tells us that these martingale measures are unique if replicating portfolios exist for all contingent claims.

- Finally, the Fundamental Theorem of Asset Pricing tells us that a unique equivalent martingale measure exists. This measure is exactly the risk-neutral probability measure that we are searching for.

More detail can be found in e.g. (Bingham & Kiesel 1998).

7.3 Inferring the Risk-Neutral Measure

7.3.1 Our Basic Idea

Using the data described above, we have explored several methods for inferring the risk-neutral probability measure (RNPM). Two methods were looked at in some detail with regard to our sample set, Hermite polynomials and histograms (these methods define the model class used to estimate the RNPM).

The basic idea contains several steps. First, we define a model class for our risk-neutral probability measure. This model class defines the parameterized search space of functions in which we will be looking for the closest approximation to the RNPM (where "closest approximation" is defined by some loss function like least-squares).

Next, we need to actually find the particular function that best approximates the RNPM. We did this using standard regression techniques to estimate the parameters for our model, given that the model has to fit the observed option prices.

Finally, we need to check our estimated RNPM for validity. This is a statistical hypothesis testing problem for which many techniques are available; our basic approach was to compare the error terms to the expected lognormal distribution. (Note that this last step, while essential to have any faith in the inferred RNPM, is difficult to do well due to noise in the data, peculiarities in financial markets, and so on.)

7.3.2 Expansion Methods

One way to estimate the RNPM is to assume it can be approximated by:

$$f(x) = \sum_i \theta_i f_i(x) \tag{7.6}$$

for a suitable base of functions. This is just a general technique of decomposing a function into a linear combination of simpler basis functions; the θ_i are scalar parameters, and the f_i are some set of basis functions.

A simple example, which we implemented, uses a histogram approach. Each f_i is simply a histogram bin, i.e. a function which takes a constant value on some interval of predetermined width and zero value everywhere else. The θ_i then represent the height of each bin.

A more sophisticated example is given by:

$$f(x) = \sum_n \theta_n \varphi^{(n)}(x) \tag{7.7}$$

where $\varphi^{(n)}(x)$ denotes the nth derivative of $\varphi(x) = \exp(-(\frac{x}{2})^2)$

Thus this particular expansion takes the form:

$$f(x) = \varphi(x)(1 + b_1 H_1(x) + b_2 H_2(x) + \ldots) \tag{7.8}$$

where H_n denotes the Hermite polynomial of order n and the first coefficient is equal to one to ensure a function whose integral is 1.

It is important to note that in order to get a density, $f(x)$ has to be positive everywhere, a condition that is not always satisfied.

Estimation of the Coefficients.

Given a set of options with the same maturity and different strike price, we are looking for a probability measure which satisfies:

$$C_j = \exp(-rT) \int_{k_j}^{\infty} (s - k_j) dF(s) + \varepsilon_j, j = 1..n \tag{7.9}$$

where C_j denotes the observed market price of the jth option and k_j its corresponding strike price.

Assuming that $F(x)$ has a probability density $f(x)$ which can be expressed as an expansion of the form (7.6), we get the result that the price of the options is given by:

$$C_j = \exp(-rT)\sum_i(\theta_i \int_{k_j}^{\infty}(s-k_j)f_i(s)ds) + \varepsilon_j, j = 1..n, i = 1..m \tag{7.10}$$

Thus, we have to find coefficients θ_i, such that:

$$C = W\theta + \varepsilon \tag{7.11}$$

where W is a matrix defined by

$$W_{ji} = exp(-rT)\int_{k_j}^{\infty}(s-k_j)f_i(s)ds \tag{7.12}$$

Now we have a regression problem, which can be tackled in the standard way (except that we have the restriction that the resulting function has to be positive).

Two cases: Histograms and Hermite polynomials

As an initial example we adjusted a histogram to the data, i.e. we used indicator functions as our basis functions. In this case, to ensure that we get a density, we have to force the coefficients to be positive. This can be done using constrained optimization techniques, when solving the least squares problem associated with the regression.

As a second example, we explored the use of the expansion (7.8) which uses Hermite polynomials. In this case, the positivity condition is broken in some cases regardless of the sign of the coefficients, so we have to find the proper number of terms in order to get a logical result.

7.3.3 Other Methods

A number of other methods have been proposed for estimating RNPM's:

- Generalized distributions. More general distributions than the ones mentioned above.
- Mixture distributions. A combination of two or more distributions. The parameters defining the combination weights could potentially change over time, to better model dynamic option behavior as the expiry date approaches.
- Kernel smoothers and implied volatility smoothing.
- Entropy methods.
- Heuristic optimization methods.
- Implied binomial trees.
- Monte Carlo and Markov Chain approaches.

Many of these methods are surveyed in (Jackwerth 1999). Clearly, a great deal of fertile ground for exploration remains in this area.

7.4 Results and Applications

7.4.1 Statistical Inference

Our goal was to check if our models of risk-neutral measures match reality, within some confidence interval. As explained in the previous section, we had several observed parameters for our model: observed option prices, strike prices and maturity dates for each option, and the riskless (i.e. interest) rate.

The next step was to specify a statistical hypothesis that could be used for testing. Our hypothesis was that two sets of observed prices were generated by the same risk-neutral measure. Under this null hypothesis, the ratio of residuals squared should follow an F-distribution; we can thus construct an f-test that should reject the null hypothesis if large values of the f statistic are observed.

Unfortunately, a number of difficulties were encountered while carrying out this procedure. The most serious involved computational inaccuracies in the statistical packages being used (including S-Plus).

However, we were still able to implement the test for the histogram approximation method. Day-to-day changes in the RNPM were detected using our test, at a qualitatively high level of significance.

More work is needed to interpret the results of our test. Given that the RNPM has changed between two time periods, what can we infer? This requires further analysis, and correlation of changes in the RNPM with changes in market sentiment and fundamental valuation. Developing automated procedures for estimating these latter subjective quantities would be very useful, from the point of view of both hypothesis testing and interpretation of results.

7.4.2 Inferring Market Sentiment

What can we infer if the "term structure of call option prices" changes? E.g. if the observed value of an out-of-the-money call option at a specific strike price drops, two somewhat contradictory explanations are possible: i) investors have become more bearish (they expect stock prices to decrease) and hence expect that the stock price will not rise enough to exercise the options; ii) volatility in the market has been reduced, and so the chances of the stock price changing enough for the option to become in-the-money have dropped. Since increased volatility is often associated with bearish market conditions (e.g. selloffs), it takes some care to interpret changes in option prices. Developing robust quantitative estimators for such changes in sentiment seems to be an open question.

Changes in the risk-neutral distribution or in implied volatility may also imply important changes in market sentiment.

7.4.3 Future Work

Along with testing whether risk-neutrality holds and deriving a probability measure under which investors are risk-neutral, it would be useful to investigate the sensitivity of risk-neutral valuation to changes in assumptions or market conditions. This is clearly a large task, requiring a good deal of subjective evaluation and judgement in assessing market conditions and reactions; relaxing the assumptions would also make analysis more difficult.

Although general linear tests are useful in detecting changes in an RNPM, more powerful tests would be helpful in detecting only those changes which are important from a risk-management point of view. An RNPM, once estimated for a given financial instrument, can be used to price other types of financial instruments. It is therefore important to keep working to improve estimation techniques.

References

Books

- (Bingham & Kiesel 1998) NH Bingham and Rugider Kiesel. *Risk-Neutral Valuation: Pricing and Hedging of Financial Derivatives.* Springer, 1998.

- (Hull 2000) John C Hull. *Options, Futures, and Other Derivatives (4th edition).* Prentice-Hall, 2000. More analytically advanced than his other book; a good second read.

- (Hull 1998) John C Hull. *Introduction to Futures and Options Markets (3rd edition).* Prentice-Hall, 1998. Great introductory book for those new to the field; we found it helpful in understanding the financial background.

- (Nielsen 1999) Lars Tyge Nielsen. *Pricing and Hedging of Derivative Securities.* Oxford University Press, 1999.

- (Steele 2000) J. Michael Steele. *Stochastic Calculus and Financial Applications.* Springer-Verlag, 2000. Continuous-time stochastic processes, martingales and Girsanov, arbitrage and contingent claim pricing.

Papers

- (Bahra 1997) Bhupinder Bahra. Implied risk-neutral probability density functions from option prices: theory and application. From Bank of England, 1997; ISSN 1368-5562.

- (Chernov and Ghysels 1999) Mikhail Chernov and Eric Ghysels. A Study towards a Unified Approach to the Joint Estimation of Objective and Risk Neutral Measures for the Purpose of Options Valuation. At citeseer.nj.nec.com/chernov99study.html

- (Coraluppi and Marcus 1997) S. Coraluppi and S. I. Marcus. Mixed risk-neutral/minimax control of Markov decision processes. In *Proceedings 31st Conference on Information Sciences and Systems*, March 1997.

- (Duan 2000) Jin-Chuan Duan. American Option Pricing under GARCH using a Markov Chain Approximation. To appear in *Journal of Economic Dynamics and Control*, 2001.

- (Jackwerth 1999) Jens Carsten Jackwerth. Option-implied risk-neutral distributions and implied binomial trees: a literature survey. In *Journal of Derivatives*, Winter 1999; pp 66-82.

- Shreve's notes on Stochastic Calculus and Finance. Excellent resource, available for viewing or download at: www.cs.cmu.edu/∼chal/shreve.html.

- (Zhu and Avellaneda 1998)Yingzi Zhu and Marco Avellaneda. A Risk-Neutral Stochastic Volatility Model. At citeseer.nj.nec.com/zhu98riskneutral.html.

Chapter 8

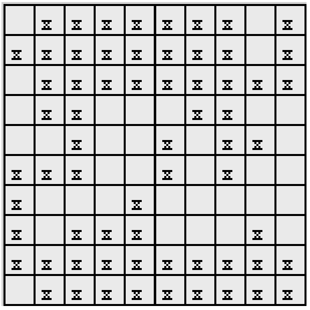
City Lights

Participants: Moshe Rosenfeld (Mentor), Tom Alberts, Angus Argyle, Andrew King, Nathan Krislock, Jill Zarestky.

PROBLEM STATEMENT: We consider a theoretical city, arranged in a grid pattern with a specified number of North-South streets (columns) and East-West avenues (rows). At each intersection of a street and avenue, there may be a light that requires power. The power switches are organized so that there is exactly one switch for every column and exactly one switch for every row. In order to save energy, we would like to minimize the number of switches turned on (and hence the unnecessary lights with power) but at the same time guarantee that all the necessary intersections have been lighted.

8.1 The Problem

Consider the ten by ten grid in figure 8.1. The squares marked by an "x" represent intersections that do not need power and the blank squares represent intersections that must be powered. In order to minimize the number of switches turned on while still covering all the necessary intersections, we begin by examining a similar problem, placement of rooks on a chessboard.



	x	x	x	x	x	x	x		x
x	x	x	x	x	x	x	x		x
	x	x	x	x	x	x	x	x	x
	x	x				x	x		
		x			x		x	x	
x	x	x			x		x		
x				x					
x		x	x	x				x	
x	x	x	x	x	x	x	x	x	x
	x	x	x	x	x	x	x	x	x

Figure 8.1: Sample grid.

Recall the allowable movements of the rook (or castle) in chess. The rook may move any number of squares in either the vertical or horizontal direction at each turn. The object is to place as many rooks as possible on a given chessboard[1] under the restriction that no two rooks should be able to take each other. In other words, there may only be one rook per row and column. It is well established that the question concerning the maximum number of rooks can be answered by the rook polynomial where the coefficient of x^k gives the maximum number of ways that k rooks may be placed. Clearly, the highest power of x with a non-zero coefficient gives the largest possible number of rooks. Unfortunately, this method does not address the issue of the switches but we may still use the basic idea of rooks to implement a solution to the city lights problem (see Grimaldi [1]).

Relevant to the lighting issue is the notion of independent rows and columns. Specifically, for each rook, we know that either the row or column occupied by the rook must be lit to ensure that the intersection represented by the rook is illuminated. From this, we may conclude that the minimum number of switches is at least as large as the maximum number of rooks placed on the grid. Furthermore, we will show that the minimum number of switches is exactly the maximum number of rooks. This is a simple consequence of Honig's Theorem and is discussed in Section 8.3.

This document is organized as follows: In Section 8.2 we will discuss our methods for solving the problem. In Section 8.3 we will present our solution of the problem and in Section 8.4 our conclusions. Finally, in Section 8.5 we will make recommendations for future study.

8.2 Methods

We used a progression of algorithms to build up to an optimal solution of the switch problem usind ideas from the rook placement problem.

1. A Greedy algorithm for the initial placement of rooks.
2. The Method we call Augmented Paths to maximize the number of rooks.
3. A systemized marking of columns and rows generated by Alternating Paths in order to minimize the number of switches.

[1] In order that the problem stated be nontrivial, we assume that a chessboard is a proper subset of the usual 8 by 8 grid, corresponding to the open squares in the streetlight grid.

The combination of these three algorithms leads to a solution in which the number of rooks is the same as the number of switches. Let us now describe the algorithms.

8.2.1 Greedy Algorithm

In the greedy algorithm, the grid is traversed row by row, starting with the topmost row. In each row, a rook is placed in the leftmost available square which does not already have a rook in the corresponding column. If the entire row is marked with x's or if all the empty squares have rooks in the associated column, then the row is skipped and we proceed to the next one. In this manner we place a sufficient number of non-interfering rooks on the grid to be sure that every lit square has a rook in either its corresponding row or column, however we cannot know if the number of rooks is or is not maximal at this stage.

8.2.2 Augmented Paths

The augmented paths algorithm finds an ordering of the rooks which allows for the maximum number to be placed on the grid. We proceed by traversing through all the blank spaces in the rows which do not contain a rook. For each appropriate space in such a row, move from the space to the rook of the same column. That each such space has a corresponding rook in it's column is a consequence of the greedy algorithm.

From this rook, we next traverse to an empty space on the same row. We will, for convenience, choose the leftmost empty space and move to the right if necessary as we proceed. There are three possible cases which we must consider.

1. The column of the leftmost space has not yet been visited by the path. We are then allowed to choose this space. If there is a rook in the column, then move to it and continue as before. If there is no rook, we may place an additional rook at the current location. Then rooks previously visited along the path constructed must be shifted 'back' along the path to accomodate the new rook. We have successfully added a new rook to the chessboard at the expense of moving a pre-existing rook to an unoccupied row.

2. The column of the leftmost space has already been included in the path at some point. We may not choose this space and instead must consider the next (counting from the left) blank space in the row.

3. There are no available spaces in the row. Either all blank spaces have been visited by the alternating paths algorithm or all the spaces have x's. We will not be able to augment the set of rooks from the blank space chosen from the original row without a rook.

Using the algorithm as stated, we find a set of non-interfering rooks for the grid which cover all the blank spaces in the sense that the greedy algorithm covered, after considering the paths from all the blank spaces in rows without a rook. Note that there may be several possible arrangements for this set for each grid. We will show later that this arrangement of rooks is in fact maximal.

8.2.3 Alternating Paths

Note that by the method of Augmenting Paths, after each traversal of the grid we have either added an additional rook or we are stuck on a rook and unable to move any further. If the latter is applicable there can exist no successful augmenting paths from ANY blank spaces on the corresponding column and the algorithm will turn on the switch for that column, then back track to the previous rook on our path in an attempt to find another augmenting path. Continue in this manner until all possibilities have been exhausted and thus it is certain that an augmenting path cannot be found from our starting square. So, for the same reason we light the column of our starting square. In this way, for each starting square in the rookless row we produce a number of lit columns characterized by the property that for each such column, its corresponding rook cannot be part of a successful alternating path.

The algorithm finishes by turning on the switches for each row which contains an unlit rook.

To summarize the method, even considering the large amount of repetition in the algorithm, it will clearly terminate eventually. At the finish, we will have a number of columns selected which must be "switched on". Then, if we select the rows which are occupied by rooks that have not been selected in column form, we will have covered all the necessary spaces. Only spaces with x's will be left, and all blank spaces will have been selected as part of the row or column selections of the rooks. Moreover, the number of switches will exactly match the number of rooks found by the augmenting path algorithm. By Honig's theorem referenced in Section 8.1 we know that this solution must be optimal. The greedy algorithm has created the initial conditions and the augmenting and alternating path algorithms have, in conjunction, found the maximal number of rooks and the minimum number of switches for our city grid.

8.2.4 The King Rook Algorithm

Another possible algorithm for finding the maximal set of rooks and minimal set of switches is as follows:

1. Create an initial rook set using the Greedy Algorithm.
2. For each row without a rook, iterate over each blank space in that row: select the corresponding column of the blank space. We are left with a collection of selected columns, each containing a rook.
3. Iterate over the columns without rooks
    (a) Iterate over the blank spaces in the column
    (b) If the blank has an unselected rook in its row, select that row.
    (c) Otherwise, create an augmenting path using the current space, the rook in its row, and the space that originally caused the rook to be lit. Remove all selections and return to step 2.
4. Light the column of every unselected rook.

The previous algorithm accomplishes the same goal as the first method described, and indeed it uses the same principles of augmenting paths. It has the advantage however, of eliminating a significant amount of the repetition involved previously. As before, the final rook set is maximal since the number of switches "on" is equal to the number of rooks. This algorithm runs in $O(n^3)$ time as the lighting of the columns or rows is $O(n^2)$ and must be repeated a maximum of n times (as many as n possible rooks).

8.3 Results

Consider the correctness of the method. From the methods construction of a solution, we know the number of columns and rows switched on equals the number of non-interfering rooks on the grid. But how do we know that all the blank squares are lit after the algorithms terminate?

Two scenarios need to be examined. In the first scenario, a blank square, S, lies in a row with no rook in the row. This blank square must lie in a column containing a rook; otherwise, the greedy algorithm would have placed a rook in the blank square. Now, since the blank squares row has no rook, the method did not find any possible augmenting path originating from our starting square S. The alternating paths algorithm would have therefore “switched on” the column containing the blank square S. And so the blank square is lit.

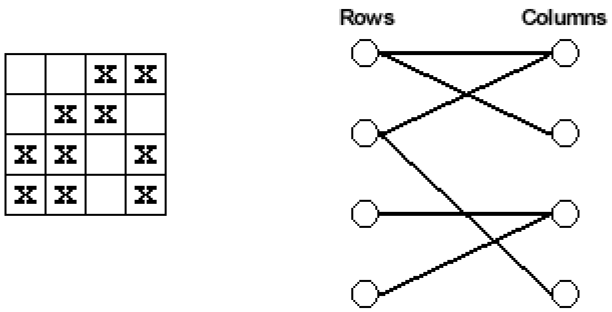
In the second scenario, a blank square, S, lies in a row with a rook R_1. The method has switched on either the row or the column (but not both) containing this rook. If the row is switched on, then the blank square S is lit. However, if the column containing the rook is switched on, then we must consider two separate cases:

- In the first case, no rook lies in the column containing the blank square S. But this cannot happen for the following reason: The column containing rook R_1 was switched on because R_1 was on an alternating path and the path was unable to move from R_1 in order to find a successful augmenting path. However, we are able to move from rook R_1 along its row to the blank square S. The blank square S is in a column with no rook. So S is the end of a successful augmenting path, and so the method would have placed a rook in the blank square S as a consequence (and shifted the other rooks on the augmenting path accordingly). Now, since the method did NOT place a rook in the blank square S, we conclude there must be a rook somewhere else in the column containing S. This leads us to the second case.

- In the second case, a rook R_2 lies in the column containing the blank square S. The column containing rook R_1 was switched on because R_1 was part of an alternating path. his implies that the blank square S was unavailable (it would not lead to a successful augmenting path). So, the column containing S must have been switched on. And so the blank square S is lit.

Thus our algorithm, which gives the same number of switches as rooks, must light all blank spaces and therfore be optimal.

In addition, we may think about this problem in terms of graphs. An alternate representation of our city grid is as a bipartite graph with rows represented as vertices on one half and columns as vertices on the other. The edges represent the intersection of a row and column where a possible rook may be placed or, referring to the original problem statement, where the light must have power. Consider figure 8.2 which demonstrates a sample grid and the corresponding bipartite graph.



		X	X
	X	X	
X	X		X
X	X		X

Figure 8.2: A grid and the associated bipartite graph.

If we approach the bipartite graph from the perspective of a maximum matching coupled with a minimum cover, then there are results which correspond to our algorithm. Specifically, the number of edges in the maximum matching is equal to the number of vertices in the minimum cover. (Honig's Theorem, J. Gross and J. Yeller [2]) The edges in a maximal matching correspond to the number of rooks placed on the grid and the minimum cover corresponds to the switch which must be turned on.

8.4 Conclusion

Thus we have constructed an algorithm which finds an optimal solution and proven its correctness. In addition, our results are backed by the literature, specifically the work related to bipartite graphs, which are clearly related to our grid problem. We feel that the solution of the two dimensional city lights problem has been sufficiently addressed and that future work should be directed towards the three dimensional problem rather than this one.

8.5 Future Work

We recommend that future efforts be directed towards the three-dimensional case of this problem. In the three-dimensional problem, the lights are arranged on the vertices of a rectangular or cubic mesh. Figure 8.3 shows a small example where the large shaded vertices are the city lights we want to turn on, and the small vertices are the lights that do not need to be lit. Like the two-dimensional case, each row, column or pillar of lights is controlled by a switch. Again, the goal is to minimize the number of switches to be turned on yet make certain the desired lights are lit.

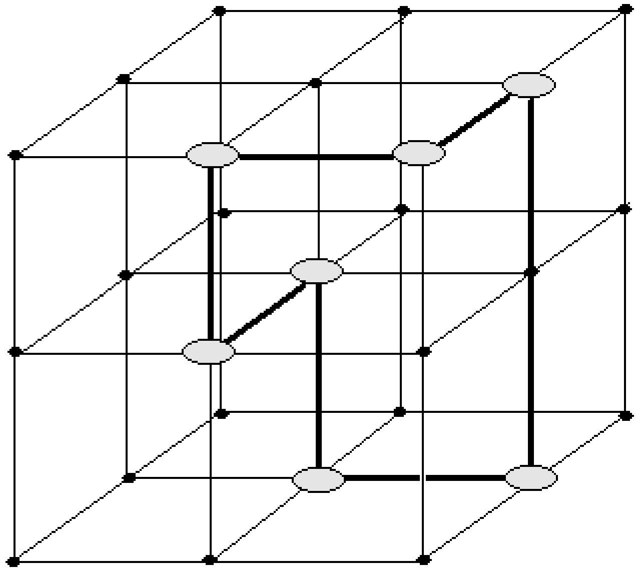


Figure 8.3: A sample 3D problem requiring more switches than rooks.

In the two-dimensional grid, the minimum number of switches equaled the maximum number of independent rooks. Is this true for the three-dimensional mesh? When placing rooks on shaded vertices in the 3D mesh, at most one rook can appear in any row, column or pillar. Figure 8.3 demonstrates a mesh where seven lights need to be lit, but the minimum number of switches exceeds the maximum number of independent rooks. At most three independent rooks may be placed in the mesh, but four switches are required to light all seven shaded vertices. So, in 3D meshes, the minimum number of switches is greater than or equal to the maximum number of rooks.

In a larger mesh, permutations of Figure 8.3 may appear several times. Then, $3k$ independent rooks would require $4k$ switches. This would cause a ratio of 4 : 3 switches-to-rooks. Other arrangements of shaded vertices in a mesh may produce a higher switches-to-rooks ratio.

An algorithm to compute the minimum number of switches for the 3D mesh has not been found. However, this problem can be translated into the Set Cover problem. In the set cover problem, a collection of subsets exists from a set S. The object is to find the smallest number of subsets whose union covers the set S. In our case, the set S is the collection of shaded vertices. Each row or column is a subset and we would need to find the minimum number of subsets which together contain all the shaded vertices. Although the general set cover problem is NP-complete, a polynomial time algorithm may exist for our special case.

References

[1] R.P. Grimaldi, Discrete and Combinatorial Mathematics: An Applied Introduction. 4th edition, 1999. Addison Wesley

[2] J. Gross and J. Yeller. Graph Theory and It's Applications. 1999. CRC Press

List of Participants

Organising Committee

Chris Bose	University of Victoria
Randy LeVeque	University of Washington
Huaxiong Huang	York University
Mark Paulhus	University of Calgary
Keith Promislow	Simon Fraser University
Ian Frigaard	University of British Columbia

Mentors

Sergei Bespamyatnikh	University of British Columbia
John Chadam	University of Pittsburgh
Ian Frigaard	University of British Columbia
Lisa Korf	University of Washington
Hedley Morris	San Jose State University
Tim Myers	University of Capetown, S.A.
Miro Powojowski	Algorithmics Corp.
Moshe Rosenfeld	University of Washington

Students

Leslie Fairbairn	Simon Fraser University
Allan Majdanac	Simon Fraser University
Tatiana Marquez-Lago	Simon Fraser University

Tom Alberts	University of Alberta
Adrian Driga	University of Alberta
Selly Kane	University of Alberta
Cristina Popescu	University of Alberta
Ling Zhao	University of Alberta

Mehmet Atilla Begen	University of British Columbia
Thomas Brakel	University of British Columbia
Mehdi Hadj-Karim-Kharazi	University of British Columbia
Theodore Kolokolnikov	University of British Columbia
Nathan Krislock	University of British Columbia
Eva-Marie Nosal	University of British Columbia
Ali Rasekh	University of British Columbia
Daniel Ryan	University of British Columbia
Ali Sanaie-Fard	University of British Columbia

Alberto Nettel	University of Calgary

Peter Anderson	University of Victoria
Angus Argyle	University of Victoria
Alexander Hodge	University of Victoria
Andrew King	University of Victoria

Mariana Carrasco Teja	University of Washington
Kristen Jaskie	University of Washington
Jihyoun Jeon	University of Washington
Viktoria Krupp	University of Washington
Rafael Meza	University of Washington
Asa Packer	University of Washington
James Rossmanith	University of Washington
Jason Slemons	University of Washington

Melvin Leok	California Institute of Technology
Hassan Masum	Carleton University
Mahtab Kamali	Concordia University
Qutaibeh Katatbeh	Concordia University
Jacky Li	Dalhousie University
Limei Sun	Memorial University of Newfoundland
Eric Machorro	Portland State University
Monica Cojocaru	Queen’s University
J. F. Williams	University of Bath
Joel Hanson	University of California, Berkeley
Carmeliza Navasca	University of California, Davis
Barkha Saxena	University of California, Santa Barbara
John Harlim	University of Guelph
Brian Corbett	University of Manitoba
Sarah Sumner	University of Ottawa
Seungwon Jeon	University of Texas at Austin
Judy Lai	University of Texas at Austin
Jill Zarestky	University of Texas at Austin
Ramin Mohammadalikhani	University of Toronto
Matthias Mück	University of Toronto
Aude Espesset	University of Western Ontario
Mufeed Mustafa Mahmoud	University of Western Ontario
Ali Ghodsi Boushehri	University of Waterloo
Yashar Ganjali	University of Waterloo
Yuriy Kazmerchuk	York University
Shuqing Liang	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: directorpims.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: J. Morrow**
  email: uw@pims.math.ca