32nd European Study Group with Industry

Final Report

edited by

JENS GRAVESEN AND POUL HJORTH

Technical University of Denmark
30 August – 4 September 1998

J. Gravesen & P. G. Hjorth (Eds.):
32nd European Study Group with Industry, Final Report.

Department of Mathematics, Technical University of Denmark,
Building 303, DK-2800 Lyngby, Denmark.
December 1998.

Available in electronic form at the URL:
`http://www.mat.dtu.dk/ESGI32`

Typeset using $\mathcal{AMS}$-LaTeX.

The study group was sponsered in part by:
The Carlsberg Bequest in Memory of Brewer J. C. Jacobsen and *The COWI Foundation.*

The painting on the cover is *Men of Industry* by P. S. Krøyer, 1902–1904.

The men of industry are gathered in the machine room of Øster Power Station, which was built in 1902. At the top, in the middle of the left hand group of three, stands the then Director of the Technical University, G. A. Hagemann, who commisioned this painting in homage to the men who introduced electricity in everyday life.

The painting can be seen at the Frederiksborg Museum in Hillerød.

Preface

European Study Groups with Industry (ESGI) is a forum where academic mathematicians work on problems directly related to industry. Workshops of this nature have taken place in Great Britain for a number of years, going back to 1963 when Prof. Alan Tayler initiated the so-called Oxford Study Groups with Industry. The coordination of the study groups is now done by the European Consortium for Mathematics in Industry (ECMI).

The format of a study group is a week long meeting (Monday – Friday) where the participating companies on the first day of the meeting each present a research problem they believe to be of a mathematical nature. Each such problem is taken up by a group of mathematicians who, together with the company representative, work towards the solution of the problem, through Thursday afternoon. Friday is used to present, in a plenary session, the results from each of the problem groups.

The reasons for the continuing success of such study groups are simple: The industrial participants get, for a very modest sum, a highly qualified 'think tank' of mathematicians to focus on their particular research problem. In addition to a full or partial resolution of the problem, the companies establish useful contacts to international researchers. The academics benefit from new ideas and challenges from the real world, providing inspiration for both education and their own research.

This is the report from the first study group held in Denmark and we are pleased to say that Danish industry received the concept with great enthusiasm. The participating companies ranged from the largest in Denmark to the very small, and the problems presented involved mathematics from a wide range of fields.

The success criterion for a Study Group is that the participating companies experience the meeting as useful, worth the effort, and that it brings them a significant step closer to the resolution of their problem.

For all of the problems in ESGI32, this criterion was met. The companies expressed their satisfaction, and some are considering what to bring for the next study group in Denmark (which is to be held in Odense, 1999). ESGI32 was, to the great relief of the organisers (as the meeting was a first-of-its-kind), an unconditional success. We look forward to many more Study Groups with Industry in Denmark.

We would like to thank the participating companies and academics for making this study group a successful and pleasant experience. In particular we thank the group of participating British mathematicians whose experience with previous study groups in a significant way contributed to the success of the meeting.

Finally we thank *The Carlsberg Bequest in Memory of Brewer J. C. Jacobsen* and *The COWI Foundation* for the generous financial support which enabled us to invite the British participants. *ECMI* made an important contribution to the social activities and *Danish Center for Applied Mathematics and Mechanics* (DCAMM) kindly donated each participant a mug. For this we are grateful to ECMI and DCAMM.

Jens Gravesen & Poul G. Hjorth,
Lyngby, 1998.

Contents

List of companies

Danfoss
DK-6430 Nordborg
Denmark
[http://www.danfoss.com/]
Representative: Stig Helmer Jørgensen

DANISCO Development Centre
Maribovej 2
DK-4900 Nakskov
Denmark
[http://www.danisco.dk]
Representative: Jannik Vindeløv

Danish Maritime Institute
Hjortekærsvej 99
DK-2800 Lyngby
[http://www.danmar.dk/]
Representative: Peter Sinding

Grundfos
DK-8850 Bjerringbro
Denmark
[http://www.grundfos.com/]
Representative: Andreas Back-Pedersen

LEGO
DK-7190 Billund
Denmark
[http://www.lego.com/]
Representative: Olga Timcenko

SCANtechnology
Rugvænget 10
DK-4100 Ringsted
Denmark
[http://isa.dknet.dk/ scantech/]
Representative: Claus Dybdal Nielsen

List of participants

Dr. Viggo Andreasen, *IMFUFA, Roskilde University, P.O. Box 260, DK-4000 Roskilde, Denmark.* [viggo@fatou.ruc.dk]

Mr. Anton Antonov, *Department of Mathematical Modelling, Technical University of Denmark, Building 305, DK-2800 Lyngby, Denmark.* [anton@unidhp1.uni-c.dk]

Dr. Andreas Back-Pedersen, *Grundfos, DK-8850 Bjerringbro, Denmark.* [abackpedersen@grundfos.com]

Mr. Anda Binzer, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.*
Presently at: *3D Facto, Fruebjergvej 3, DK-2100 København Ø, Denmark* [Anda.Binzer@3dfacto.dk]

Dr. Morten Brøns, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.* [M.Brons@mat.dtu.dk]

Dr. Janis Cepitis, *Institute of Mathematics, Latvian Academy of Sciences and University of Latvia, Latvia.* [cepitis@lanet.lv]

Dr. Rebecca Gower, *OCIAM, Mathematics Institute, University of Oxford, 24-29 St. Giles, Oxford OX1 3LB, England.*
Presently at: *Analytical Services, DfEE, Level 5F Caxton House, Tothill Street, London SW1H 9NA, England* [Rebecca.Gower@dfee.gov.uk]

Dr. Jens Gravesen, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.* [J.Gravesen@mat.dtu.dk]

Ms. Gretchen Greene, *Department of Mathematics, UCLA, 6363 Math Sciences, 405 Hilgard Avenue, Box 951555, Los Angeles, CA 90095-1555, USA.* [ggreene@math.ucla.edu]

Professor Robert Greene, *Department of Mathematics, UCLA, 6363 Math Sciences, 405 Hilgard Avenue, Box 951555, Los Angeles, CA 90095-1555, USA.* [greene@math.ucla.edu]

Mr. Morten Hagdrup, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.*
Presently at: Tele Danmark, Udviklingsdivisionen, Telegade 2, DK-2630 Taastrup, Denmark [mohag@tdk.dk]

Professor Erik B. Hansen, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.* [E.B.Hansen@mat.dtu.dk]

Professor Vagn Lundsgaard Hansen, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.* [V.L.Hansen@mat.dtu.dk]

Mr. Christian Henriksen, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.* [Christian.Henriksen@mat.dtu.dk]

Ms. Agnes E. Heydtmann, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.* [Agnes.Heydtmann@mat.dtu.dk]

Dr. Poul Hjorth, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.* [P.G.Hjorth@mat.dtu.dk]

Professor John Hogan, *Department of Engineering Mathematics, Queen's Building, University Walk, Bristol BS8 lTR, England.* [S.J.Hogan@bristol.ac.uk]

Dr. Peter Howell, *OCIAM, Mathematics Institute, University of Oxford, 24-29 St. Giles, Oxford OXl 3LB, England.* [howell@maths.ox.ac.uk]

Mr. Thomas Jakobsen, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.* [T.Jakobsen@mat.dtu.dk]

Dr. Jørn Møller Jensen, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.* [J.M.Jensen@mat.dtu.dk]

Dr. Stig Helmer Jørgensen, *Danfoss, DK-6430 Nordborg, Denmark.* [SHJ@danfoss.com]

Dr. Per W. Karlsson, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.* [P.W.Karlsson@mat.dtu.dk]

Dr. Wolfhard Kliem, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.* [W.Kliem@mat.dtu.dk]

Professor Andrew Lacey, *Department of Mathematics Heriot-Watt University, Riccarton, Edinburgh EH14 4AS, Scotland.* [A.A.Lacey@ma.hw.ac.uk]

Ms. Ahn Louise Larsen, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.* [A.L.Larsen@mat.dtu.dk]

Dr. Jens Christian Larsen, *Department of Mathematics and Physics, Royal Veterinary and Agricultural University, Thorvaldsensvej 40, DK-1870 Frederiksberg C, Denmark.* [jlarsen@dina.kvl.dk]

Professor Steen Markvorsen, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.* [S.Markvorsen@mat.dtu.dk]

Director Claus Dybdal Nielsen, *SCANtechnology, Rugvænget 10, DK-4100 Ringsted, Denmark.* [scantech@isa.dknet.dk]

Dr. Hans Bruun Nielsen, *Department of Mathematical Modelling, Technical University of Denmark, Building 305, DK-2800 Lyngby, Denmark.* [hbn@imm.dtu.dk]

Dr. Michael Pedersen, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.* [M.Pedersen@mat.dtu.dk]

Professor John W. Perram, *The Maersk Institute, Odense University, Campusvej 55, DK-5230 Odense M, Denmark.* [jperram@mip.ou.dk]

Dr. Henrik G. Petersen, *The Maersk Institute, Odense University, Campusvej 55, DK-5230 Odense M, Denmark.* [hgp@mip.ou.dk]

Mr. Thomas Randrup, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.*
Presently at: *3D Facto, Fruebjergvej 3, DK-2100 København Ø, Denmark* [Thomas.Randrup@3dfacto.dk]

Dr. Henning Rasmussen, *Department of Applied Mathematics, University of Western Ontario, London, Canada.* [a471@julian.uwo.ca]

Dr. Henrik Rasmussen, *OCIAM, Mathematics Institute, University of Oxford, 24-29 St. Giles, Oxford OXl 3LB, England.* [rasmusse@maths.ox.ac.uk]

Ms. Tiina Roose, *OCIAM, Mathematics Institute, University of Oxford, 24-29 St. Giles, Oxford OXl 3LB, England.* [roose@maths.ox.ac.uk]

Mr. Peter Røgen, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.* [Peter.Roegen@mat.dtu.dk]

Ms. Marit Hvalsøe Schou, *The Maersk Institute, Odense University, Campusvej 55, DK-5230 Odense M, Denmark.* [marit@mip.ou.dk]

Dr. Robert Sinclair, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.* [R.M.Sinclair@mat.dtu.dk]

Dr. Peter Sinding, *Danish Maritime Institute, Hjortekærsvej 99, DK-2800 Lyngby, Denmark.* [psi@danmar.dk]

Dr. Ove Skovgaard, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.* [Ove.Skovgaard@mat.dtu.dk]

Mr. Allan Sørensen, *The Maersk Institute, Odense University, Campusvej 55, DK-5230 Odense M, Denmark.* [allans@mip.ou.dk]

Dr. Mads Peter Sørensen, *Department of Mathematical Modelling, Technical University of Denmark, Building 305, DK-2800 Lyngby, Denmark.* [mps@imm.dtu.dk]

Professor Carsten Thomassen, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.* [C.Thomassen@mat.dtu.dk]

Dr. Olga Timcenko, *LEGO, DK-7190 Billund, Denmark.* [olga@digi.lego.com]

Mr. Michael Ungstrup, *Department of Mathematics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark.* [M.Ungstrup@mat.dtu.dk]

Mr. Jannik Vindeløv, *DANISCO Development Centre, Maribovej 2, DK-4900 Nakskov, Denmark.* [javi@kvl.dk]

Dr. David Wood, *OCIAM, Mathematics Institute, University of Oxford, 24-29 St. Giles, Oxford OXl 3LB, England.* [woodd@maths.ox.ac.uk]

Problems

1 Danfoss: Scroll Optimization

PRESENTED BY STIG HELMER JØRGENSEN

Air-conditioning scroll compressors are manufactured by the million today and get greater and greater market shares from traditional (reciprocating) single piston compressors.

The scroll compressor consist of two plane spiral/helix running inside each other. Normally both scrolls are the same unrolling circle involute with constant wall thickness. The compression chambers are therefore thin, oblong, and bent. Often the one scroll is fixed and the other is orbiting.

The scroll compressor has its suction port at the periphery and discharge in the spiral centre. No suction valve is needed and a discharge valve is only present to decrease power consumption a little bit.

The scroll compressor has a rather constant speed of compression chamber decrease and thus a harmless almost constant torque loading of partly the electric motor and partly the housing (action and reaction).

Consequences of the scroll geometry are long gas leakage passages and high material temperatures in the scroll centre.

Different movements of scrolls are topics of today: Co-rotating and co-orbiting scrolls.

The task could be: Chose one scroll geometry and movements of both scrolls. Find the other scroll geometry (One scroll is envelope of the other). Scroll wall thickness could be a variable too. Make investigations and see how compressor performance/efficiency is influenced. Finally, find sensibilities and optimise design if possible.

2 DANISCO: Temperature and Moisture Gradients in Sugar Silos

PRESENTED BY JANNIK VINDELØV

The distribution of heat and moisture in sugar stored in silos influence:

- The chemical and microbial stability of the sugar.
- The ease of silo emptying due to moist sugar, sugar caking and hardening
- The risk of dust explosions

The above, are important factors in sugar quality and production. In order to develop better silo operation strategies and silo designs, Danisco Sugar would like to develop a model that, with some accuracy, can predict the temperature and moisture gradients in bulk sugar.

The silos

Danisco Sugar utilises several types of silos for the storage of crystalline sugar. They differ in:

- The use of construction material (e.g. concrete or plate silos).
- The construction (e.g. volume, height, diameter, with or without a central tower and isolation etc.).
- The method of filling and emptying (e.g. first in - last out or first in - first out).
- Sugar conditioning functionality (e.g. temperature control of walls, control of the condition of the air space above sugar, ability to blow conditioned air through the bulk sugar etc.).

The silos are filled during the beet-campaign (Sept.-Dec.), the sugar is withdrawn in a regular, but not constant, basis and the silos are emptied before next years campaign.

Temperature and moisture migration

Moisture gradients are created due to moisture migration in the bulk sugar by a variety of modes:

- Diffusion due to temperature gradients.
- Diffusion due to bulk sugar quality heterogeneity .
- Natural convection due to temperature gradients.
- Forced convection due to bulk sugar conditioning (blow-through).
- Introduction and withdrawal of moisture through silo walls (e.g. concrete silos) and due to the conditioning of the air above the bulk sugar.

Temperature gradients are induced by:

- The temperature of the silo walls, top and bottom due to seasonal variations in temperature.
- The temperature of the air space above the bulk sugar due to seasonal variations or conditioning.
- The heat of bulk sugar conditioning air (blow-through) Heterogeneity of the of the bulk phase can be created due to de-mixing processes when filling and emptying silos and due to differing sugar qualities when filling. Different sugar qualities have different physical properties. The following physical properties are assumed of importance when formulating models to describe the formation of temperature and moisture gradients: bulk sugar heat capacity, heat conductivity, density, porosity, tortousity and sorption isotherms. Most will vary with the sugar quality (that is: crystal size, reducing sugar content and ash content), some with temperature and moisture.

The models

The models should be able to predict temperature and moisture gradients in sugar silos by:

- Diffusion.
- Natural and forced convection, due to:

  - Seasonal variations in outside humidity and temperature.
  - Silo dimensions and construction.
  - Silo conditioning.
  - Sugar quality.
  - The size of the bulk phase.

The basic model could have the following features:

- Mass transfer by diffusion as well as convection in bulk sugar.
- Energy transfer by conduction as well as convection in bulk sugar.
- Energy and mass transfer through bulk sugar boundaries.
- Homogenous bulk phase (one quality) of simple geometry (cylindrical).
- Moving bulk - air boundary.
- Varying boundary conditions of inner as well as outer walls, bottom and air.

A more advanced model could have the following additional features:

- Heterogeneous bulk phase.
- Complex geometry.
- Differing boundary conditions at north an south faced walls.

A numerical model is sought that can be solved on a pentium PC if written in e.g. Turbo Pascal or Fortran. Models developed for corn silos can be used as an outset, e.g:

Tanaka H, Yoshida K (1984) "Heat and mass transfer mechanisms in a grain storage silo". Engineering Sciences in the food industry. Elsevier Applied Sience Publishers, Essex, England. pp 89-98

Khankari KK, Morey RV, Patankar SV (1994). "Mathematical model for moisture diffusion in stored grain due to temperature gradients". Transactions of the ASAE 37: 1591-1604)

3 Danish Maritime Institute: Dynamic Positioning System.

PRESENTED BY PETER SINDING

The problem is how to most efficiently ensure that a ship, floating on the surface of the ocean, stays very nearly at rest with respect to the sea bed. One may employ thrusters, propellers and rudder action. More precisely, given external forces from wind, current and waves, and a specified position and motion envelope for the ship bulk, study the number, type, and placement of dynamic motion controls on the bulk with the goal of forming an optimal Dynamic Positioning System.

4 Grundfos: Mixing of Clorine in Swimmingpools.

PRESENTED BY ANDREAS BACK-PEDERSEN

GRUNDFOS would like a model, describing the problem of mixing chemicals, being dosed into water systems, to be developed. The application of the model should be dedicated to dosing aqueous solution of chlorine into swimming pools.

The problem is imagined to contain two sub-models. The first model is concerned about dosing a strong aqueous solution of chlorine into a pipe system and the second about injection of purified and chlorinated water into the swimming pool.

The contamination of the water and the chemical process reducing the chlorine content in the swimming pool could be regarded as uniform and stationary and dependent of the number of bathers.

5 LEGO: How to Build with LEGO.

PRESENTED BY OLGA TIMCENKO

The idea for the project is simple - if any 3D body is given, how can it be built with LEGO bricks?

Unit-volume (the smallest possible volume) in the LEGO universe is a so-called "generic LEGO brick". It is a brick 8mm long and wide, and 3.2mm high, and has only one position ("stud") for connecting with other bricks. Although there are a lot of different LEGO bricks, in this project the use of "family" bricks only is allowed. "Family" LEGO bricks are paralelopiped-shape bricks that can be made of "generic" LEGO bricks, by putting several of them next to each other and/or above each other.
Allowed dimensions of "family" LEGO bricks are: (if the dimensions of the generic LEGO brick are set to 1,1,1)

length	width	height
1	1	1,3,15
2	1	1,3,6,15
3	1	1,3,15
4	1	1,3
6	1	1,3,15
8	1	1,3
10	1	1,3
12	1	3
16	1	3
2	2	1,3,9
3	2	1,3
4	2	1,3,9
6	2	1,3,9
8	2	1,3
10	2	1,3
12	2	1

length	width	height
16	2	1
4	4	1
6	4	1,3
8	4	1
10	4	1,3
12	4	1,3
6	6	1
8	6	1
10	6	1
12	6	1
14	6	1
16	6	1
8	8	3
16	8	3
24	12	3

Please note that some bricks appear in different heights. Length and width of bricks correspond to their number of studs (connection points) in the horizontal plane.

The usage of LEGO DUPLO bricks is allowed also. Dimensions of DUPLO bricks, in "generic brick" measure, are:

2	4	12
4	4	6,12
8	4	6,12
12	4	6
16	4	6
20	4	6

There is one important restriction here: Due to construction, DUPLO bricks can be connected only to family bricks with an even number of generic bricks in the length and width dimensions, and only with the bricks of height at least 3.

So the task is:

For a "legoized" 3D model (i.e. a 3D model that is represented as a set of 1x1x1 generic LEGO bricks put next to each other or above each other), find an algorithm to build the model of actual LEGO bricks from the previous tables, so that model should stand connected. We assume that a brick is connected to a model if at least two of its connection points are connected to a model. (If the brick has less than three connection points, then it is enough that only one of its connection points is connected to a model).

We do not want a model to be solid, meaning "full of bricks". Whenever it is possible to make an invisible hole inside a model, we would like it to be done - to spend less bricks in building. But all the bricks should be connected to the model as described earlier. A good "rule of thumb" should be that the width of the "wall" from outside to inside of the model is 4 connection points (it can be more or less in some places, the shape of the inside hole is not important).

We can assume nothing about the shape of a legoized object except that it is connected. It can have any number of holes into it (a cup with two handles, for example).

The preferred output of this project should be an algorithm for making a computer program for determining which brick should be put in which place. One of the problems we can see now is that there is not a unique solution - i.e. every model can be built using many different bricks. A criterion that "more solid" models are preferred can be used to set some cost-function. "More solid" models are models made with bigger bricks, and with bricks that have more connection points connected (and even these two criterions can be opposite!).

I have tried to explain here a problem as it appears in "real life". I hope this is enough for you to make an exact mathematical formulation of the problem (although I am aware that, as with all "real life" problems, some terms and requirements are not defined quite precisely - we require that the outside of the final model is exactly as in the legoized model, but we do not have such precise requests about the insides of the model, not even the precise thickness of the "wall" - but, these are facts of life!)

6 SCANtechnology: Feature Recognition in 3D-Scanning.

PRESENTED BY CLAUS DYBDAL NIELSEN

A 3D-scanner measures distances from a plane or cylinder to points on 3D-objects.

The raw data is organized as measurements of a height function $z = f(x, y)$. The height-function is measured at 'scan-points' (x_{ij}, y_i) situated on 'scan-lines' where y is constant, see figure 1. Neither the scan-points nor the scanlines needs to be equidistant, and the number of scan-points on a scan-line needs not to be constant. In a scan-image we have the following typical values:

- Number of scan-lines 100-10 000.

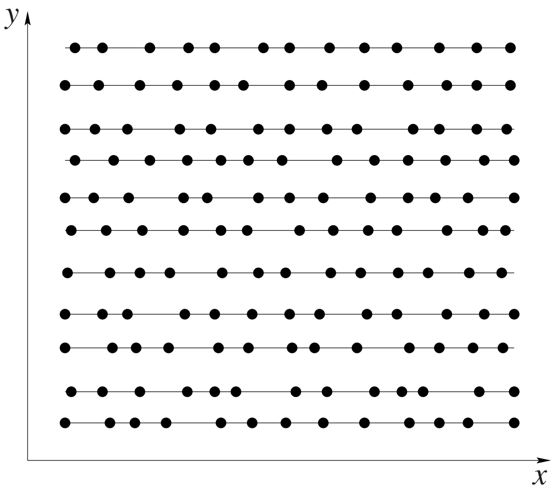


Figure 1: Distribution of scan-lines and scan-points on the reference plane

- Number of scan-points on a scan-line 100-10 000.
- Precision of y-measurements ± 0.01mm.
- Precision of x-measurements ± 0.1mm.
- Precision of z-measurements ± 0.1mm.
- Distance between scan-lines $0.2 - 5.00$mm.
- Distance between scan-points $0.2 - 2.00$mm.

At the moment data reductions is performed on each scan-line based on a curvature analysis of the height function on this line and this leads to a data-reduction of 40-80%. After the data-reduction the scan-points are triangulated and an approximation of the height function is obtained, which defines a surface. It is this triangulated surface which is used later in the process, which typical consists of the calculation of tool-paths for a milling machine, in order to make a copy of the original object.

It would be advantages if it was possible to automatically find features in the picture, such as the centerline of ridges and valleys, and peaks and holes.

The task of the study group is to

Automatically recognize features in the image which should be preserved during the data-reduction.

Work Group Reports

Danfoss: Scroll Optimization

Jens Gravesen, Christian Henriksen,
and Peter Howell

1 Introduction

The scroll compressor consists of two plane spiral/helix running inside each other, traditionally both scrolls are the same circle involute with constant wall thickness, see figure 1.

When the moving scroll follows a circular orbit the chambers moves inwards and the air or fluid trapped in the chambers is compressed. In the standard circle involute scroll compressor the volume of the compression chambers grows linearly when we move outwards and the compression rate of such a compressor is sufficient for air-condition, but not for freezers or refrigerators. One could try to increase the compression rate by having more chambers, but that would increase the leakage too.

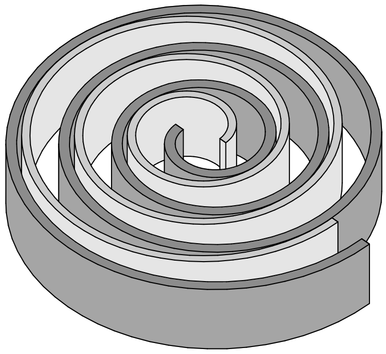


Figure 1: The standard circle involute scroll compressor.

The task from DANFOSS was to investigate how a change in the basic geometry of the scrolls and the orbit influence the compressor performance and efficiency, and then try to optimize the design.

We have not done any optimization, but we believe we have found a sound foundation for the optimization process. We can represent the geometry of the problem in a way which leads to closed expressions for all the geometric quantities which are interesting for the scroll design.

We have also analyzed the leakage between the compression chambers, and using a compressible lubrication model we have determined an expression which, appart from one integration, leads to a closed expression for the leakage.

2 The Geometry of Plane Curves

To fix the notation and terminology we give a brief introduction to the geometry of plane curves.

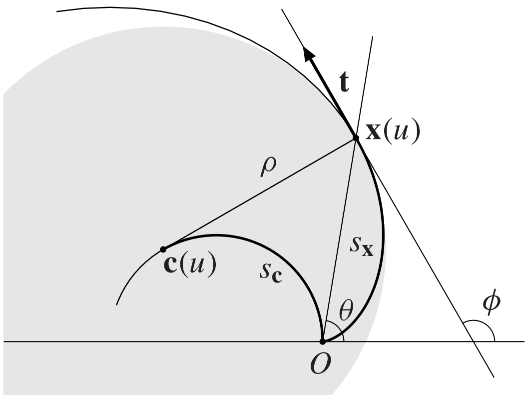


Figure 2: Some geometric concepts.

A parametrization of a regular plane curve is a differential map $\mathbf{x} : I \to \mathbb{R}^2$, such that $\mathbf{x}'(u) \neq \mathbf{0}$ all $u \in I$. The *unit tangent vector* is $\mathbf{t} = \frac{\mathbf{x}'}{|\mathbf{x}'|}$, the *unit normal vector* is $\mathbf{n} = \widehat{\mathbf{t}}$, and the *arc length* is given by $s = \int_0^u |\mathbf{x}'(t)|\, dt$.

We introduce the frame $(\mathbf{e}, \mathbf{f})$, given by $\mathbf{e}(u) = (\cos u, \sin u)$ and $\mathbf{f}(u) = (-\sin u, \cos u)$. The *tangent direction* ϕ is then given by $\mathbf{t} = \mathbf{e}(\phi)$, the *curvature* is $\kappa = \frac{d\phi}{ds}$ and the *radius of curvature* is $\rho = \frac{1}{\kappa} = \frac{ds}{d\phi}$. The *Frenet formulae* tells that $\frac{d\mathbf{t}}{ds} = \kappa\mathbf{n}$, $\frac{d\mathbf{n}}{ds} = -\kappa\mathbf{t}$, and hence $\mathbf{t}'(u) = s'(u)\frac{d\mathbf{t}}{ds} = s'(u)\kappa(u)\mathbf{n}(u)$ and likewise $\mathbf{n}'(u) = -s'(u)\kappa(u)\mathbf{t}(u)$. The *centre of curvature* is given by $\mathbf{c} = \mathbf{x} + \rho\mathbf{n}$. The *osculating circle* is the unique circle which has second order contact with the curve in the point $\mathbf{x}$, and it has centre $\mathbf{c}$ and radius ρ. The *evolute* is the the curve traced out by the centre of curvature:

$$\mathbf{c} = \mathbf{x} + \rho_{\mathbf{x}}\mathbf{n}_{\mathbf{x}}, \tag{1}$$

by differentiation we have

$$\mathbf{c}' = \mathbf{x}' + \rho_{\mathbf{x}}'\mathbf{n}_{\mathbf{x}} + \rho_{\mathbf{x}}\mathbf{n}_{\mathbf{x}}' = s_{\mathbf{x}}'\mathbf{t}_{\mathbf{x}} + \rho_{\mathbf{x}}'\mathbf{n}_{\mathbf{x}} - \rho_{\mathbf{x}}s_{\mathbf{x}}'\kappa_{\mathbf{x}}\mathbf{t}_{\mathbf{x}} = \rho_{\mathbf{x}}'\mathbf{n}_{\mathbf{x}}, \tag{2}$$

from which we see that if $\rho_{\mathbf{x}}' > 0$, then $\mathbf{c}$ is a regular curve, $\mathbf{t}_{\mathbf{c}} = \mathbf{n}_{\mathbf{x}}$, $s_{\mathbf{c}}' = \rho_{\mathbf{x}}'$, and hence $s_{\mathbf{c}} = \rho_{\mathbf{x}} + \text{constant}$. If $\rho_{\mathbf{x}}' \neq 0$ we can always obtain $\rho_{\mathbf{x}}' > 0$ by choosing the appropriate orientation. We can obviously invert this process and thus obtain $\mathbf{x}$ as an *involute* of the curve $\mathbf{c}$:

$$\mathbf{x} = \mathbf{c} - (s_{\mathbf{c}} + s_0)\mathbf{t}_{\mathbf{c}}, \tag{3}$$

and if $\mathbf{c}$ is a regular curve then $\rho_{\mathbf{x}}$ is a strictly increasing function. The following well known result tells us that an involute is a spiral:

Lemma 1. *Let* $\mathbf{x}(u)$ *be a regular curve with radius of curvature* $\rho(u)$ *and regular evolute* $\mathbf{c}(u)$. *Let* $D(u)$ *be the disk bounded by the osculating circle:*

$$D(u) = \left\{\mathbf{x} \in \mathbb{R}^2 \;\middle|\; |\mathbf{x} - \mathbf{c}(u)| < \rho(u)\right\}.$$

If the radius of curvature $\rho(u)$ *is a strictly increasing function, then the disks* $D(u)$ *forms an increasing nested sequence:*

$$u_1 < u_2 \Rightarrow \overline{D(u_1)} \subset D(u_2), \tag{4}$$

and the past and future of the curve is respectively inside and outside the disk:

$$u_1 < u_2 \Rightarrow \mathbf{x}(u_1) \in D(u_2) \wedge \mathbf{x}(u_2) \notin D(u_1). \tag{5}$$

Furthermore, choose origo $O \in \bigcap_{t \in I} \overline{D(u)}$ *and let* (r, θ) *be polar coordinates for the curve* $\mathbf{x}$, *then* θ *is an increasing function:*

$$\theta'(u) > 0. \tag{6}$$

Proof. As $\mathbf{x}(u)$ is on the boundary of $D(u)$ (4) implies (5). Now let $\mathbf{x} \in D(u_1)$, then $|\mathbf{x} - \mathbf{c}(u_1)| \le \rho(u_1)$, Furthermore $\mathbf{c}(u_1)$ and $\mathbf{c}(u_2)$ are points on the evolute and hence $|\mathbf{c}(u_2) - \mathbf{c}(u_1)| < s_{\mathbf{c}}(u_2) - s_{\mathbf{c}}(u_1) = \rho(u_2) - \rho(u_1)$. All in all we have:

$$\begin{aligned} |\mathbf{x} - \mathbf{c}(u_2)| \le |\mathbf{x} - \mathbf{c}(u_1)| &+ |\mathbf{c}(u_2) - \mathbf{c}(u_1)| \\ &< \rho(u_1) + \rho(u_2) - \rho(u_1) = \rho(u_2), \end{aligned}$$

which proves (4) and hence (5).

The angle between the line from $\mathbf{c}(u)$ to $\mathbf{x}(u)$ and the tangent $\mathbf{t}(u)$ is $\frac{\pi}{2}$, but then the angle between a line from any point in $D(u)$ to $\mathbf{x}(u)$ and the tangent is in the interval $(0, \pi)$. Because of (4) this is in particular true for the point O, and then (6) holds. □

We will later consider an orbiting curve so let us now consider a family of curves $\mathbf{x}_t(u) = \mathbf{X}(u, t)$. An *envelope* of such a family is a curve $\mathbf{y}$ which is tangential to the curve $\mathbf{x}_t$ at the point $\mathbf{y}(t)$. That means there exists a function $\mathbf{u}(t)$ such that

$$\mathbf{y}(t) = \mathbf{X}(u(t), t), \tag{7}$$

the derivative is then $\mathbf{y}'(t) = u'(t)\frac{\partial \mathbf{X}}{\partial u}(u(t), t) + \frac{\partial \mathbf{X}}{\partial t}(u(t), t)$. The derivative of $\mathbf{x}_t$ at the same point is $\mathbf{x}_t'(u(t)) = \frac{\partial \mathbf{X}}{\partial u}(u(t), t)$, so $\mathbf{y}'(t)$ and $\mathbf{x}_t'(u(t))$ are parallel if and only if

$$\left[\frac{\partial \mathbf{X}}{\partial t}\big(u(t), t\big)\, \frac{\partial \mathbf{X}}{\partial u}\big(u(t), t\big)\right] = 0, \tag{8}$$

where $[\mathbf{u}\,\mathbf{v}] = \widehat{\mathbf{u}} \cdot \mathbf{v}$ denotes the planar product, or determinant, of two vectors $\mathbf{u}$ and $\mathbf{v}$. Equation (8) determines the function $u(t)$ which then inserted in (7) gives the envelope $\mathbf{y}(t)$, see figure 3.

3 Scroll Geometry

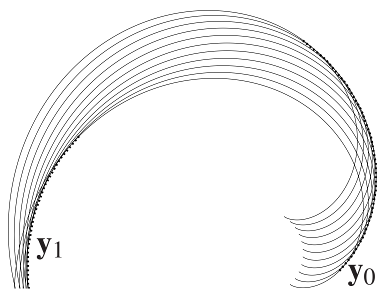


Figure 3: Two envelopes.

Suppose the curve $\mathbf{x}$ describes one side of a moving scroll wall. The side of the moving wall can be described by

$$\mathbf{X}(u,t) = \mathbf{R}\big(\psi(t)\big)\big(\mathbf{x}(u) + \mathbf{a}(t)\big),$$

where $\mathbf{a}$ denotes the translation and $\mathbf{R}(\psi)$ denotes rotation by the angle ψ. The mating side $\mathbf{y}$ of the fixed scroll wall is given by the envelope of the family of curves $\mathbf{X}$, and consequently satisfies (7) and (8). In general (8) is an unpleasant nonlinear equation, but if we use the *tangent direction* as the parameter on both curves, i.e., $u = \phi_{\mathbf{x}}$ and $t = \phi_{\mathbf{y}}$, then the equations are simplified to

$$t = u + \psi(t) + n\pi, \quad n \in \mathbb{Z}, \tag{9}$$

$$\mathbf{y}_n(t) = \mathbf{R}\big(\psi(t)\big)\Big(\mathbf{x}\big(t - \psi(t) - n\pi\big) + \mathbf{a}(t)\Big) \tag{10}$$

If $\mathbf{a}$ and ψ are periodic with period 2π, then we have $\mathbf{y}_n(t+2\pi) = \mathbf{y}_{n+2}(t)$ and we have only two candidates for the mating side of the fixed scroll wall:

$$\mathbf{y}_0(t) = \mathbf{R}\big(\psi(t)\big)\Big(\mathbf{x}\big(t - \psi(t)\big) + \mathbf{a}(t)\Big),$$
$$\mathbf{y}_1(t) = \mathbf{R}\big(\psi(t)\big)\Big(\mathbf{x}\big(t - \psi(t) - \pi\big) + \mathbf{a}(t)\Big).$$

We may reparameterize $\mathbf{y}_1$, by $t \mapsto t + \pi$, and can then write

$$\mathbf{y}_i(t) = \mathbf{R}\big(\psi_i(t)\big)\Big(\mathbf{x}\big(t - \psi_i(t)\big) + \mathbf{a}_i(t)\Big), \quad i = 0, 1, \tag{11}$$

where $\mathbf{a}_i$ and ψ_i are defined by

$$\mathbf{a}_i(t) = \begin{cases} \mathbf{a}(t) & \text{if } i = 0, \\ \mathbf{a}(t+\pi) & \text{if } i = 1. \end{cases} \qquad \psi_i(t) = \begin{cases} \psi(t) & \text{if } i = 0, \\ \psi(t+\pi) & \text{if } i = 1. \end{cases} \tag{12}$$

At time t the side $\mathbf{x}_t$ of the moving scroll wall and mating side $\mathbf{y}_0$ of the fixed scroll wall touch each other in a series of points:

$$\mathbf{y}_0(t + 2n\pi) = \mathbf{R}\big(\psi(t)\big)\Big(\mathbf{x}\big(t + 2n\pi - \psi(t)\big) + \mathbf{a}(t)\Big), \tag{13}$$

and at time $t+\pi$, the curves $\mathbf{x}_{t+\pi}$ and $\mathbf{y}_1$ touch in the points:

$$\mathbf{y}_1(t + 2n\pi) = \mathbf{R}\big(\psi(t+\pi)\big)\Big(\mathbf{x}\big(t + 2n\pi - \psi(t+\pi)\big) + \mathbf{a}(t+\pi)\Big), \tag{14}$$

and between these points the two walls bounds a chamber, (the compression chamber of the scroll compressor), see figure 4.

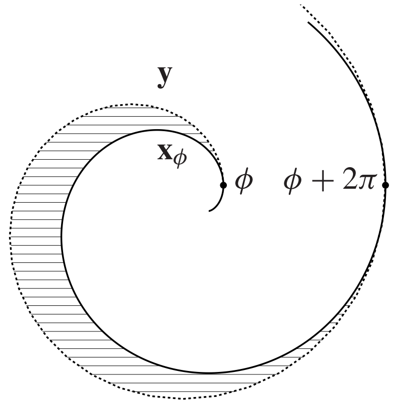


Figure 4: A compression chamber.

Excluding rotation

If we exclude rotation, i.e., let $\psi = 0$, then we just have

$$\mathbf{y}_0(t) = \mathbf{x}(t) + \mathbf{a}(t),$$
$$\mathbf{y}_1(t) = \mathbf{x}(t) + \mathbf{a}(t + \pi).$$

Circular orbit

If $\mathbf{a}$ is a circle with centre O, with radius R, and parametrized by tangent direction, then $\mathbf{a} = -R\mathbf{f}$, and we have

Lemma 2. *Let* $\mathbf{x}$ *be a regular curve, and put* $\mathbf{X}(u, t) = \mathbf{x}_t(u) = \mathbf{x}(u) - R\mathbf{f}(t)$*. The envelopes of this family are* $\mathbf{y} = \mathbf{x} \mp R\mathbf{f}$*, and they have the same evolute as* $\mathbf{x}$*.*

Proof. Let $\mathbf{c} = \mathbf{x} + \rho_\mathbf{x}\mathbf{n}_\mathbf{x}$ be the evolute of $\mathbf{x}$. If we parametrize $\mathbf{x}$ by tangent direction, then we have $\mathbf{x} = \mathbf{c} - (s_\mathbf{c} + s_0)\mathbf{t}_\mathbf{c} = \mathbf{c} - (s_\mathbf{c} + s_0)\mathbf{n}_\mathbf{x} = \mathbf{c} - (s_\mathbf{c} + s_0)\mathbf{f}$, but then $\mathbf{y} = \mathbf{x} \mp R\mathbf{f} = \mathbf{c} - (s_\mathbf{c} + (s_0 \pm R))\mathbf{f} = \mathbf{c} - (s_\mathbf{c} + (s_0 \pm R))\mathbf{t}_\mathbf{c}$, and we see that $\mathbf{y}$ is an involute of $\mathbf{c}$ too. □

3.1 Using the natural equation

If the side $\mathbf{x}$ of the moving scroll wall is given by some parametrization, then it is in general impossible to find the parametrization by tangents direction. Instead we will specify the curve by the *natural equation* in the form $s = s(\phi)$ where s denotes arc-length on the curve and ϕ denotes the tangent direction of the curve. If s is a positive, increasing and convex function, then $\rho' = s'' > 0$ and by Lemma 1 the corresponding curve is a spiral.

The *tangent vector* $\mathbf{t}(\phi)$ of $\mathbf{x}(\phi)$ is by definition of ϕ the vector $\mathbf{e}(\phi)$, and hence

$$\mathbf{x}' = \frac{ds_\mathbf{x}}{d\phi}\mathbf{t} = s'_\mathbf{x}\mathbf{e} \tag{15}$$

and

$$\mathbf{x} = \int s'_\mathbf{x}\mathbf{e}\, d\phi = s_\mathbf{x}\mathbf{e} - \int s_\mathbf{x}\mathbf{f}\, d\phi. \tag{16}$$

Observe that we can find an *explicit* expression for the parametrization of $\mathbf{x}$ if we can integrate $s(\phi)\cos(\phi)$ analytically, this is for example the case if s is a polynomial or a piecewise polynomial in ϕ. The mating sides of the fixed scroll walls are given by

$$\mathbf{y}_i(\phi) = \mathbf{R}\big(\psi_i(\phi)\big)\mathbf{x}\big(\phi - \psi_i(\phi)\big) + \mathbf{a}_i(\phi), \quad i = 0, 1. \tag{17}$$

Excluding rotation

If we exclude the rotation, then we get particularly simple equations

$$\mathbf{y}_0(\phi) = \mathbf{x}(\phi) + \mathbf{a}(\phi), \qquad \mathbf{y}_1(\phi) = \mathbf{x}(\phi) + \mathbf{a}(\phi + \pi), \tag{18}$$

$$\mathbf{y}_0'(\phi) = \big(s_{\mathbf{x}}'(\phi) + s_{\mathbf{a}}'(\phi)\big)\mathbf{e}(\phi), \qquad \mathbf{y}_1'(\phi) = \big(s_{\mathbf{x}}'(\phi) - s_{\mathbf{a}}'(\phi + \pi)\big)\mathbf{e}(\phi), \tag{19}$$

$$s_{\mathbf{y}_0}(\phi) = s_{\mathbf{x}}(\phi) + s_{\mathbf{a}}(\phi), \qquad s_{\mathbf{y}_1}(\phi) = s_{\mathbf{x}}(\phi) - s_{\mathbf{a}}(\phi + \pi), \tag{20}$$

$$\rho_{\mathbf{y}_0}(\phi) = \rho_{\mathbf{x}}(\phi) + \rho_{\mathbf{a}}(\phi), \qquad \rho_{\mathbf{y}_1}(\phi) = \rho_{\mathbf{x}}(\phi) - \rho_{\mathbf{a}}(\phi + \pi), \tag{21}$$

where ρ denotes the radius of curvature. If we assume that $\rho_{\mathbf{x}}, \rho_{\mathbf{a}} > 0$, (which we always can do by picking the right orientation) then (21) implies that $\mathbf{y}_0$ is on the 'outside' of $\mathbf{x}$ and $\mathbf{y}_1$ is on the 'inside' of $\mathbf{x}$, see figure 3.

3.2 The natural equation for a closed plane curve

Let $s_{\mathbf{a}} = s_{\mathbf{a}}(\phi)$ be the natural equation for the closed curve $\mathbf{a}$. It is clear that we must have $s_{\mathbf{a}}(\phi + 2\pi) - s_{\mathbf{a}}(\phi) = 2\pi R$, where $2\pi R$ is the length of the closed curve. It is therefore natural to put

$$s_{\mathbf{a}}(\phi) = \overline{s}_{\mathbf{a}}(\phi) + R\phi \tag{22}$$

where $\overline{s}_{\mathbf{a}}$ is a periodic function of ϕ with period 2π. As $\mathbf{a} = s_{\mathbf{a}}\mathbf{e} - \int s_{\mathbf{a}}\mathbf{f}\, d\phi$ we see that $\mathbf{a}$ is closed if and only if $\big(s_{\mathbf{a}}(2\pi) - s_{\mathbf{a}}(0)\big)\mathbf{e}(0) - \int_0^{2\pi} s_{\mathbf{a}}\mathbf{f}\, d\phi = \mathbf{0}$. Inserting (22) we get the equation $\int_0^{2\pi} \overline{s}_{\mathbf{a}}\mathbf{f}\, d\phi = \mathbf{0}$ or the two scalar equations

$$\int_0^{2\pi} \overline{s}_{\mathbf{a}}(\phi)\cos(\phi)\, d\phi = 0, \qquad \int_0^{2\pi} \overline{s}_{\mathbf{a}}(\phi)\sin(\phi)\, d\phi = 0. \tag{23}$$

That is, the Fourier expansion of $\overline{s}_{\mathbf{a}}$ is

$$\overline{s}_{\mathbf{a}}(\phi) = \sum_{n=2}^{\infty}\big(a_n\cos(n\phi) + b_n\sin(n\phi)\big). \tag{24}$$

Using only a finite trigonometric polynomial of this kind we can get a closed expression for the curve $\mathbf{a}$ parametrized by tangent direction. We finally note that the case $\overline{s}_{\mathbf{a}} = 0$ corresponds to that of a circle with radius R, $\mathbf{a} = -R\mathbf{f}$.

3.3 The volume of the compression chambers

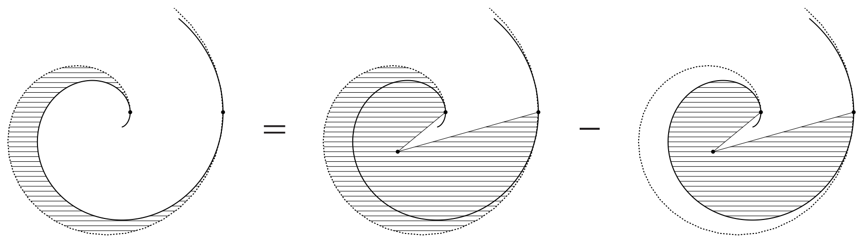


Figure 5: The volume of a compression chamber.

A compression chamber is bounded by two curve segments, so we can find the volume of the chamber as the difference between the areas spanned by a fixed point and the two curve segments, see figure 5.

When we want to find the area spanned by a point and a segment of a curve, then we first observe that the area of the triangle spanned by $\mathbf{r}$ and $\mathbf{r}'$ is $\frac{1}{2}[\mathbf{r}\,\mathbf{r}']$, so the area spanned by O and the segment of the curve $\mathbf{r}$ is given by $\text{Area} = \frac{1}{2}\int_{t_1}^{t_2}[\mathbf{r}\,\mathbf{r}']\,dt$, see figure 6, this is also a consequence of Green's theorem. Observe that a rotation around O does not change the area. So up to a sign the volume of a chamber in the scroll compressor is given by

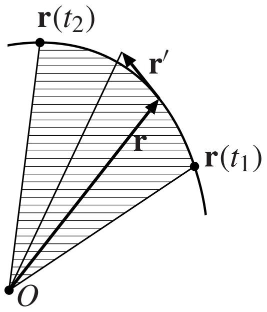


Figure 6: The area spanned by a point and a curve.

$$
\begin{aligned}
v_i(\phi) &= \frac{1}{2}\int_{\phi}^{\phi+2\pi}[\mathbf{y}_i(t)\,\mathbf{y}_i'(t)]\,dt - \frac{1}{2}\int_{\phi-\psi_i(\phi)}^{\phi-\psi_i(\phi)+2\pi}\big[\big(\mathbf{x}(t)+\mathbf{a}_i(\phi)\big)\,\mathbf{x}'(t)\big]\,dt \\
&= \frac{1}{2}\int_{\phi}^{\phi+2\pi}[\mathbf{y}_i(t)\,\mathbf{y}_i'(t)]\,dt - \frac{1}{2}\int_{\phi-\psi_i(\phi)}^{\phi-\psi_i(\phi)+2\pi}[\mathbf{x}(t)\,\mathbf{x}'(t)]\,dt \\
&\qquad - \frac{1}{2}\big[\mathbf{a}_i(\phi)\,\big(\mathbf{x}\big(\phi-\psi_i(\phi)+2\pi\big)-\mathbf{x}\big(\phi-\psi_i(\phi)\big)\big)\big].
\end{aligned} \tag{25}
$$

Excluding rotation

Without rotation we have

$$
\begin{aligned}
v_i(\phi) = \frac{1}{2}\int_{\phi}^{\phi+2\pi}&\big([\mathbf{y}_i(t)\,\mathbf{y}_i'(t)] - [\mathbf{x}(t)\,\mathbf{x}'(t)]\big)\,dt \\
&- \frac{1}{2}\big[\mathbf{a}_i(\phi)\,\big(\mathbf{x}(\phi+2\pi)-\mathbf{x}(\phi)\big)\big].
\end{aligned} \tag{26}
$$

Once more we have that if $s_{\mathbf{x}}(\phi)$ is a piecewise polynomial and $\overline{s}_{\mathbf{a}}(\phi)$ is a finite trigonometric polynomial, then we have $v_i(\phi)$ on closed form.

3.4 Circular orbit

If we specialize to the case of a circular orbit $\mathbf{a} = -R\mathbf{f}$, and no rotation, then $\mathbf{a}_0 = \mathbf{a}$ and $\mathbf{a}_1 = -\mathbf{a}$ and hence

$$\mathbf{y}(t) = \mathbf{x}(t) \mp R\mathbf{f}(t), \tag{27}$$

$$\mathbf{y}'(t) = \mathbf{x}'(t) \pm R\mathbf{e}(t), \tag{28}$$

$$\begin{aligned}
[\mathbf{y}(t)\,\mathbf{y}'(t)] &= [\mathbf{x}(t)\,\mathbf{x}'(t)] \pm R[\mathbf{x}(t)\,\mathbf{e}(t)] \mp R[\mathbf{f}(t)\,\mathbf{x}'(t)] - R^2[\mathbf{f}(t)\,\mathbf{e}(t)] \\
&= [\mathbf{x}(t)\,\mathbf{x}'(t)] \pm R\big(\frac{d}{dt}[\mathbf{x}(t)\ -\mathbf{f}(t)] - [\mathbf{x}'(t)\ -\mathbf{f}(t)]\big) \\
&\qquad \pm R[\mathbf{x}'(t)\,\mathbf{f}(t)] + R^2 \\
&= [\mathbf{x}(t)\,\mathbf{x}'(t)] \pm R\frac{d}{dt}[\mathbf{x}(t)\ -\mathbf{f}(t)] \pm 2R[\mathbf{x}'(t)\,\mathbf{f}(t)] + R^2 \\
&= [\mathbf{x}(t)\,\mathbf{x}'(t)] \pm \frac{d}{dt}R[\mathbf{f}(t)\,\mathbf{x}(t)] \pm 2Rs_{\mathbf{x}}'(t) + R^2
\end{aligned}$$

$$\rho_{\mathbf{y}} = \rho_{\mathbf{x}} \pm R \tag{29}$$

$$s_{\mathbf{y}} = s_{\mathbf{x}} + \begin{cases} R\phi \\ -R(\phi+\pi) \end{cases} \tag{30}$$

The volume of a compression chamber is given by

$$\begin{aligned}
v(\phi) &= \frac{1}{2}\int_{\phi}^{\phi+2\pi} \big([\mathbf{y}(t)\,\mathbf{y}'(t)] - [\mathbf{x}(t)\,\mathbf{x}'(t)]\big)\,dt \\
&\qquad \mp \frac{1}{2}\big[\mathbf{a}(\phi)\,\big(\mathbf{x}(\phi+2\pi) - \mathbf{x}(\phi)\big)\big] \\
&= \pm\frac{1}{2}\int_{\phi}^{\phi+2\pi} \left(\frac{d}{dt}R[\mathbf{f}(t)\,\mathbf{x}(t)] + 2Rs_{\mathbf{x}}'(t) \pm R^2\right)\,dt \\
&\qquad \pm \frac{1}{2}R\big[\mathbf{f}(\phi)\,\big(\mathbf{x}(\phi+2\pi) - \mathbf{x}(\phi)\big)\big] \\
&= \pm\frac{1}{2}\Big(R[\mathbf{f}(\phi+2\pi)\,\mathbf{x}(\phi+2\pi)] - R[\mathbf{f}(\phi)\,\mathbf{x}(\phi)] \\
&\qquad\qquad + 2Rs_{\mathbf{x}}(\phi+2\pi) - 2Rs_{\mathbf{x}}(\phi) \pm 2\pi R^2 \\
&\qquad\qquad + R\big[\mathbf{f}(\phi)\,\big(\mathbf{x}(\phi+2\pi) - \mathbf{x}(\phi)\big)\big]\Big) \\
&= \pi R^2 \pm R\Big(\big[\mathbf{f}(\phi)\,\big(\mathbf{x}(\phi+2\pi) - \mathbf{x}(\phi)\big)\big] + s_{\mathbf{x}}(\phi+2\pi) - s_{\mathbf{x}}(\phi)\Big)
\end{aligned} \tag{31}$$

3.5 The other side of the wall

So far we have only considered one pair of mating sides of the scroll walls, but in a real world scroll compressor we have of course two pairs of mating sides. In this section we will address the problem of defining both sides of the scroll wall.

Constant wall thickness

It seems natural to try to make walls of constant thickness as in the case of the standard circle involute scroll scroll compressor, see figure 1.

Let the curve describing one side of the moving scroll wall be $\mathbf{x}(\phi)$, with the natural equation $s = s_{\mathbf{x}}(\phi)$. We can write $\mathbf{x} = \mathbf{c} - (s_{\mathbf{c}} + s_0)\mathbf{t}_{\mathbf{c}}$, where $\mathbf{c}$ is the evolute of $\mathbf{x}$. Any curve $\widetilde{\mathbf{x}}$ parallel to $\mathbf{x}$ is also an involute of $\mathbf{c}$, thus

$$\widetilde{\mathbf{x}} = \mathbf{c} - (s_{\mathbf{c}} + s_1)\mathbf{t}_{\mathbf{c}} = \mathbf{x} - w\mathbf{n}_{\mathbf{x}} = \mathbf{x} - w\mathbf{f}, \tag{32}$$

where $w = s_1 - s_0$ is the distance between the curves. The thickness w has a sign and if $w > 0$, then the parallel curve $\widetilde{\mathbf{x}}$ is on the 'outside' of $\mathbf{x}$. The curvatures are related by the equation $\rho_{\widetilde{\mathbf{x}}} = \rho_{\mathbf{x}} + w$, and as $\rho = \frac{ds}{d\phi}$, the natural equation $s = s_{\widetilde{\mathbf{x}}}(\phi)$ of $\widetilde{\mathbf{x}}$ is given by $s_{\widetilde{\mathbf{x}}}(\phi) = s_{\mathbf{x}}(\phi) + w\phi$. Observe that it is only in the case of parallel curves we have such nice relations. If we put $\widetilde{\mathbf{x}}(t) = \mathbf{x}(t) + w(t)\mathbf{n}_{\mathbf{x}}(t)$ with $w(t)$ a non constant function, then $\phi_{\widetilde{\mathbf{x}}} \neq \phi_{\mathbf{x}}$, and we have do not have any explicit formula linking the curvature of $\mathbf{x}$ and $\widetilde{\mathbf{x}}$.

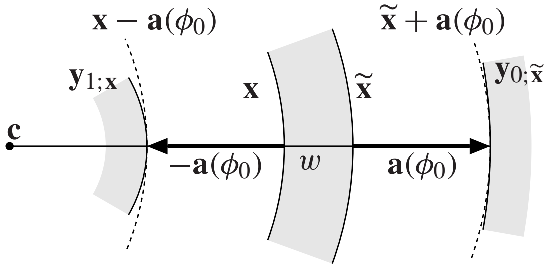


Figure 7: A scroll compressor with a moving scroll of constant width.

We now let the wall follow the orbit of $\mathbf{a}$, and restrict ourself to the case of a circular orbit, $\mathbf{a} = -R\mathbf{f}$, and with constant width $w > 0$. In that case we have a situation as in figure 7, where we see the moving scroll wall bounded by $\mathbf{x}$ and $\widetilde{\mathbf{x}}$ in the channel defined by the fixed scroll wall. By Lemma 2 we have that $\mathbf{y}_{1;\mathbf{x}}$ has the same evolute as $\mathbf{x}$ and $\mathbf{y}_{0;\widetilde{\mathbf{x}}}$ has the same evolute as $\widetilde{\mathbf{x}}$, in particular are $\mathbf{y}_{1;\mathbf{x}}$ and $\mathbf{y}_{0;\widetilde{\mathbf{x}}}$ parallel curves and we have the following result:

Lemma 3. *If a scroll wall of constant width w is moved in circular orbit with radius R, then the channel of the fixed scroll has constant width, $w + 2R$.*

The fixed scroll wall will in general *not* have constant thickness, the following result tells us when this happens:

Proposition 4. *Let $\mathbf{x}$ and $\widetilde{\mathbf{x}}$ be the sides of the moving scroll wall which moves in circular orbit and let $\mathbf{y} = \mathbf{y}_{1;\mathbf{x}}$ and $\widetilde{\mathbf{y}} = \mathbf{y}_{0;\widetilde{\mathbf{x}}}$ be the mating sides of the fixed scroll wall, as in figure 7.*

If the curves are spirals in the sense that the arclength of any of the curves is a strictly increasing convex function of the tangent direction, then the following statements are equivalent:

1. *The thickness of the scroll walls are constant.*
2. *The curves $\mathbf{x}$ and $\widetilde{\mathbf{x}}$ have a common simple closed convex evolute.*
3. *The curves $\mathbf{y}$ and $\widetilde{\mathbf{y}}$ have a common simple closed convex evolute.*
4. *The curves $\mathbf{x}$, $\widetilde{\mathbf{x}}$, $\mathbf{y}$, and $\widetilde{\mathbf{y}}$ have a common simple closed convex evolute.*

Proof. We will assume that all four curves are parametrized by tangent direction and that the circular orbit is given by $\mathbf{a} = -R\mathbf{f}$.

Assume that the wall thickness of the moving scroll wall is constant, then all four curves $\mathbf{x}$, $\widetilde{\mathbf{x}}$, $\mathbf{y}$, and $\widetilde{\mathbf{y}}$ are parallel curves by Lemma 3, and hence they have a common evolute $\mathbf{c}$. That $\mathbf{c}$ is convex, i. e., has nonvanishing curvature, follows from the fact that the scroll sides are smooth curves.

Suppose that the fixed scroll wall has constant thickness too, then the curve which together with $\widetilde{\mathbf{y}}$ defines the fixed scroll wall is of the form $\phi \mapsto \mathbf{y}(\phi + \nu)$. As the tangent direction is the same and the curves are spirals, we must have $\nu = 2\pi$. That means that the curve $\phi \mapsto \mathbf{y}(\phi + 2\pi)$ is parallel to $\widetilde{\mathbf{y}}$ and hence also to $\mathbf{y}$, but as parallel curves have common evolutes we must have that $\mathbf{c}(\phi) = \mathbf{c}(\phi + 2\pi)$ and the common evolute is closed. As $\phi + \frac{\pi}{2}$ is the tangent direction on the evolute we have that it is periodic with period 2π when parametrized by tangent direction, but then it can not have any self intersections and it must be a simple closed convex curve.

We have now shown that $1 \Rightarrow 4$, and we clearly have that $4 \Rightarrow 2, 3$. So we need only to show that $2 \Rightarrow 1$ and $3 \Rightarrow 1$. Now assume that $\mathbf{x}$ and $\widetilde{\mathbf{x}}$ are involutes of a common simple closed convex curve $\mathbf{c}$. By Lemma 3 we have that all four curves are parallel so we need only to show that $\mathbf{y}$ and the curve $\phi \mapsto \mathbf{y}(\phi + 2\pi)$ are parallel. As $\mathbf{y} = \mathbf{c} - (s_{\mathbf{c}} + s_0)\mathbf{f}$ we have that

$$\mathbf{y}(\phi + 2\pi) - \mathbf{y}(\phi) = -\big(s_{\mathbf{c}}(\phi + 2\pi) - s_{\mathbf{c}}(\phi)\big)\mathbf{f}(\phi) = -\ell_{\mathbf{c}}\mathbf{f}(\phi),$$

where $\ell_{\mathbf{c}}$ is the length of the closed curve $\mathbf{c}$. The tangent vector of both $\mathbf{y}(\phi+2\pi)$ and $\mathbf{y}(\phi)$ are $\mathbf{e}$ which is orthogonal to $\mathbf{f}$, and we see that $\mathbf{y}(\phi+2\pi)$ and $\mathbf{y}(\phi)$ indeed are parallel, and that the distance between them are $\ell_{\mathbf{c}}$.

To prove that $3 \Rightarrow 1$ we only have to interchange the role of the moving and fixed scroll wall. □

The lemma above gives us a way of constructing scroll compressors with constant wall thickness, unfortunately this does not give us much compared to the standard circle evolute scroll compressor. As $\mathbf{x}(\phi+2\pi)-\mathbf{x}(\phi) = -\ell_{\mathbf{c}}\mathbf{f}(\phi)$, the expression (31) for the volume of a compression chamber reduces to

$$v(\phi) = \pi R^2 \pm R\big(s_{\mathbf{x}}(\phi+2\pi) - s_{\mathbf{x}}(\phi)\big).$$

Furthermore if $\mathbf{c}$ is the closed evolute of $\mathbf{x}$, then $s'_{\mathbf{x}} = \rho_{\mathbf{x}} = s_{\mathbf{c}} + s_0$, so

$$\int_{\phi+2\pi}^{\phi+4\pi} s_{\mathbf{c}}(t)\,dt = \int_{\phi+2\pi}^{\phi+4\pi} \big(s_{\mathbf{c}}(t-2\pi)+\ell_{\mathbf{c}}\big)\,dt = \int_{\phi}^{\phi+2\pi} s_{\mathbf{c}}(t)\,dt + 2\pi\ell_{\mathbf{c}}$$

and hence $v(\phi+2\pi) - v(\phi) = 2\pi\ell_{\mathbf{c}}$. We see that only the size, not the shape of $\mathbf{c}$ matters, for the ratio between the volume of the chambers.

We can of course still have that only one of the scroll walls has constant width, but we would still have a problem. The volume function (31) for the two pairs of mating sides of the scroll walls will in general be different, and the compression rate would be different too.

Reflecting a pair of mating wall sides

The standard circle involute scroll compressor in figure 1 has the property that the two scroll walls are the image of each other by a reflection in a suitable point. By this reflection the two pairs of mating sides gets mapped into each other, see figure 8.

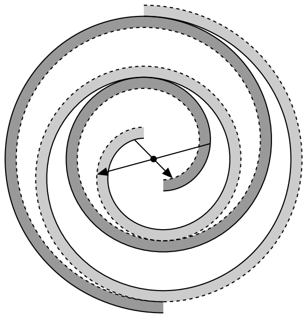


Figure 8: Reflection in a point.

We can convert this property to a principle by which we can construct the whole of the scroll walls once we have one pair $\mathbf{x}(\phi)$, $\mathbf{y}(\phi)$ of mating sides. In order to define the reflection which will give us the other pair of mating sides we need the centre of the reflection, and we only need the image of one point in order to determine the centre point.

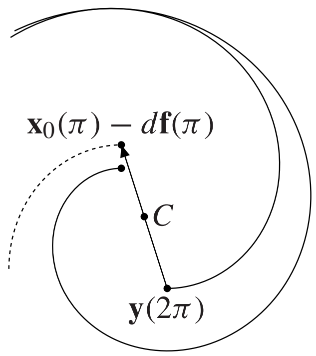


Figure 9: Reflecting a pair of mating sides in a point.

We are given one side $\mathbf{x}_t(\phi) = \mathbf{x}(\phi) - R\mathbf{f}(t)$ of the moving scroll and the mating side $\mathbf{y}(\phi) = \mathbf{x}_{\phi-\pi}(\phi)$ of the fixed scroll wall. We want the reflection to map the point $\mathbf{y}(2\pi)$ to the point $\mathbf{x}_0(\pi) - d\mathbf{f}(\pi)$ which we note lies on the normal to $\mathbf{x}_0$ opposite $\mathbf{y}$, see figure 9. The centre point C is determined by the equation $2C = \mathbf{y}(2\pi) + \mathbf{x}_0(\pi) - d\mathbf{f}(\pi)$, and then the other pair of mating scroll walls are determined by the equations $\widetilde{\mathbf{x}}_0(\phi) + \mathbf{y}(\phi + \pi) = 2C$ and $\widetilde{\mathbf{y}}(\phi) + \mathbf{x}_0(\phi - \pi) = 2C$, i.e.:

$$\widetilde{\mathbf{x}}(\phi) = \big(\mathbf{y}(2\pi) + \mathbf{x}(\pi) + (R-d)\mathbf{f}(\pi)\big) - \mathbf{y}(\phi + \pi) \tag{33}$$

$$\widetilde{\mathbf{y}}(\phi) = \big(\mathbf{y}(2\pi) + \mathbf{x}(\pi) + (R-d)\mathbf{f}(\pi)\big) - \mathbf{x}(\phi - \pi) \tag{34}$$

We have $\mathbf{y}(\phi) = \mathbf{x}_{\phi-\pi}(\phi) = \mathbf{x}(\phi) + R\mathbf{f}(\phi)$, and a short calculation shows that $\widetilde{\mathbf{y}}(\phi) = \widetilde{\mathbf{x}}_\phi(\phi - 2\pi)$. If $\mathbf{x}, \widetilde{\mathbf{x}} : [\pi, 2N\pi] \to \mathbb{R}^2$ and $\mathbf{y}, \widetilde{\mathbf{y}} : [2\pi, 2N + 2\pi] \to \mathbb{R}^2$, then at time $t \in [0, 2\pi)$ we see that the curves $\mathbf{x}_t$ and $\mathbf{y}$ touch each other in the points

$$\mathbf{y}(t + 2n\pi) = \mathbf{x}(t + 2n\pi) + R\mathbf{f}(t), \qquad \begin{cases} n = 1, 2, \ldots, N & \text{if } t = 0, \\ n = 1, 2, \ldots, N-1 & \text{if } t \neq 0, \end{cases}$$

and the curves $\widetilde{\mathbf{x}}_t$ and $\widetilde{\mathbf{y}}$ touch each other in the points

$$\widetilde{\mathbf{y}}(t + 2n\pi) = \widetilde{\mathbf{x}}(t + 2n\pi - 2\pi) + R\mathbf{f}(t), \qquad \begin{cases} n = 1, 2, \ldots, N & \text{if } t = \pi, \\ n = 2, 3, \ldots, N & \text{if } t \neq \pi. \end{cases}$$

Remember that the two pair of mating curves $\mathbf{x}, \mathbf{y}$ and $\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}$ are the reflected images of eachother and under this reflection we have the correspondens $\mathbf{x}(t + 2n\pi) \leftrightarrow \widetilde{\mathbf{y}}(t + (2n+1)\pi)$ and $\widetilde{\mathbf{x}}(t + (2n-1)\pi) \leftrightarrow \mathbf{y}(t + 2n\pi)$. This means that the sequence of chambers between $\mathbf{x}_t$ and $\mathbf{y}$ at time t is the same as the sequence of chambers between $\widetilde{\mathbf{x}}_{t+\pi}$ and $\widetilde{\mathbf{y}}$ at time $t + \pi$, we have in particular that the two 'channels' of the compressor have the same compression rate, which obviously is a desirable property.

3.6 Examples

We have considered three different natural equations: $s = \phi^2$ which gives the circle involute, and two others, $s = \phi^2 + \frac{1}{30}\phi^3$ and $s = \phi^2 - \frac{1}{50}\phi^3 + \frac{1}{400}\phi^4$. In

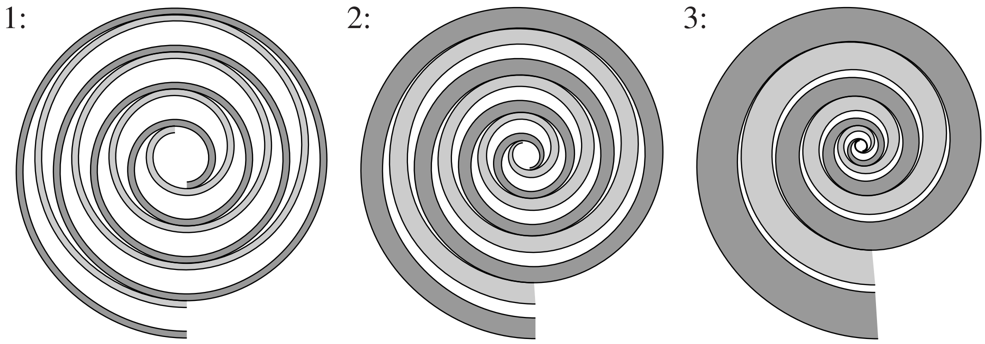


Figure 10: The three examples of scroll compressors.

all three cases we let $\mathbf{x}, \widetilde{\mathbf{x}} : [\pi, 8\pi] \to \mathbb{R}^2$ and $\mathbf{y}, \widetilde{\mathbf{y}} : [\pi, 10\pi] \to \mathbb{R}^2$ such that we at time $t = 2n\pi$, $n \in \mathbb{Z}$ have three chambers, with volume $v(2\pi)$, $v(4\pi)$, and $v(6\pi)$, where v is given by (31). We are interested in the ratio of these volumes so we have calculated $\Delta_1 = v(6\pi)/v(4\pi)$, $\Delta_2 = v(4\pi)/v(2\pi)$, and the overall volume ratio $\Delta = v(6\pi)/v(2\pi)$. Another important quantity is the *stroke volume* which is $v(6\pi)$. We can always increase this by scaling the compressor so we are really interested in the *normalized stroke volume* $\overline{V} = v(6\pi)/D^2$ where D is the diameter of the compressor. To simplify matters we use $|\mathbf{y}(10\pi) - \mathbf{y}(9\pi)|$ as an estimate of D. Finally we calculates the *leakage coefficient*:

$$l = \int_0^{2\pi} \sqrt{\kappa} \left(\frac{V_0}{V_1}\right) \left[\left(\frac{V_0}{V_1}\right)^{\gamma} - \left(\frac{V_1}{V_0}\right)^{\gamma} \right] dt,$$

see (71), where $\kappa(t) = \frac{1}{\rho_{\mathbf{y}}(8\pi - t)} - \frac{1}{\rho_{\mathbf{x}}(8\pi - t)}$, $V_0(t) = v(6\pi - t)$, $V_1(t) = v(4\pi - t)$, and $\gamma = 1.4$ as in section 7, page 28. The results are summarized in table 1 and in figure 10 we have drawn the three compressors.

	$s(\phi)$	R	d	Δ_1	Δ_2	Δ	$\overline{V}$	l
1	ϕ^2	4	2.28	1.43	1.75	2.49	0.085	0.64
2	$\phi^2 + \frac{1}{30}\phi^3$	6	2	1.68	2.14	3.60	0.041	0.67
3	$\phi^2 - \frac{1}{50}\phi^3 + \frac{1}{400}\phi^4$	6	2	2.25	2.76	6.21	0.016	1.13

Table 1: The performance of the three scroll compressors. The value $d = 2\pi - 4 \approx 2.28$ is the one which gives the circle involute scroll compressor constant wall thickness.

We should stress that example 2 and 3 are arbitrary and only are meant to illustrate that it is easy to calculate these important numbers. The calculation was done using Maple® and except for the integration in the determination of l it is possible to get the result on symbolic form starting with a natural equation on polynomial form, say $s = c_0 + c_1\phi + c_2\phi^2 + c_3\phi^3 + c_4\phi^4$. We will not present the result of that calculation, but the possibility of doing this makes an optimization and a sensitivity analysis much easier.

If we compare the natural equation in example 2 and 3 with the natural equation $s = \phi^2$ for the circle involute it is clear that s is changed most for large values of ϕ corresponding to the outer part of compressor. It would be obvious to try to change the interior part of the compressor, because then the normalized stroke volume is kept fixed, while the volume ratio may change.

4 Leakage Between the Compression Chambers

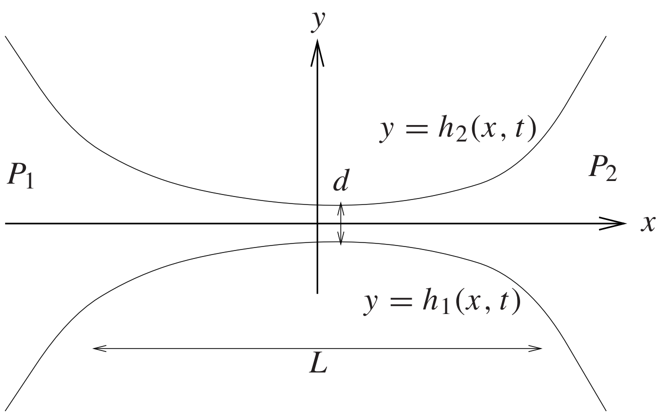


Figure 11: Schematic diagram of the gap between two chambers.

Ideally, the two halves of a scroll compressor remain perfectly in contact as they rotate. In reality it is not practical to machine them accurately enough for this to be the case, and instead there is a narrow gap. Typically the gap is around one micron across; this may be increased by wear and/or poor machining, and it is known that if it reaches around eight microns, the compressor becomes useless.

In this section we analyse the flow through this gap in an attempt to determine the leakage between neighbouring chambers in the compressor. We use compressible lubrication theory, using the fact that the gap has a very small aspect ratio. A schematic diagram of the geometry is given in figure 11. Here the gap, of typical thickness d and length L, separates two adjacent chambers containing gas at pressures P_1 and P_2. A local coordinate system is adopted with x pointing along the channel and y pointing across it. With respect to these coordinates we denote

Property	Symbol	Approx. value	Units
Viscosity	μ	10^{-5}	Pa s
Density	ρ	1	$\mathrm{Kg\,m^{-3}}$
Thermal diffusivity	k	2.5×10^{-2}	$\mathrm{J\,m^{-1}s^{-1}K^{-1}}$
Specific heat	c_v	10^3	$\mathrm{J\,Kg^{-1}K^{-1}}$
Gap thickness	d	10^{-6}	m
Contact length	L	10^{-2}	m
Rotation frequency	ω	50	$\mathrm{s^{-1}}$
Pressure drop	ΔP	10^6	Pa

Table 2: Estimated parameter values for leakage between cells in a scroll compressor.

the bottom and top of the channel (both of which are moving as the compressor rotates) by $y = h_1(x,t)$ and $y = h_2(x,t)$.

Some typical parameter values for the flow are shown in table 2. From these we can extract the important dimensionless parameters that determine the character of the flow. Firstly, the inverse aspect ratio of the channel is small,

$$\epsilon = \frac{d}{L} \approx 10^{-4}, \tag{35}$$

which will enable us to simplify the governing equations considerably. Next we deduce a typical velocity for the gas from a balance between the pressure drop and viscous drag:

$$U = \frac{d^2 \Delta P}{\mu L}.$$

Another dimensionless parameter is the ratio between this and the velocity due to rotation of the compressor:

$$\Omega = \frac{\omega L}{U} = \frac{\mu \omega L^2}{d^2 \Delta P} \approx 5 \times 10^{-2}. \tag{36}$$

Now we calculate the reduced Reynolds number using U as the velocity scale:

$$\mathrm{Re}^* = \frac{\rho U L \epsilon^2}{\mu} = \frac{\rho d^4 \Delta P}{\mu^2 L^2} \approx 10^{-4}. \tag{37}$$

Since Re^* is small, we can safely neglect inertia effects in the channel. We can also interpret Re^* as the square of a typical Mach number:

$$\mathrm{Re}^* = \frac{U^2}{c^2}, \quad \text{where } c^2 = \frac{\Delta P}{\rho},$$

and hence deduce that the flow is very much subsonic. Similarly, we calculate the reduced Peclet number using U as the velocity scale:

$$\mathrm{Pe}^* = \frac{\rho c_v U L \epsilon^2}{k} = \frac{\rho c_v d^4 \Delta P}{\mu k L^2} \approx 10^{-4}.$$

It is no accident that $\mathrm{Pe}^* \approx \mathrm{Re}^*$ since their ratio, the Prandtl number $\mathrm{Pr} = k/(\mu c_v)$, is approximately one for air.

We can immediately deduce a typical rate of leakage using the velocity scale U. The rate at which gas is lost through the channel is of order Ud, and so the cumulative loss over a cycle is typically Ud/ω. We simply have to compare this with the original area of a chamber to obtain the relative loss of gas due to leakage:

$$\text{relative loss} = \frac{d^3 \Delta P}{\mu \omega L^3} = \frac{\epsilon}{\Omega}. \tag{38}$$

The values given in table 2 suggest that this is rather small: about 0.2%. However, it is clearly highly sensitive to increases in d, and if we set $d = 8\ \mu$m, then the typical relative loss is dramatically increased to around 100%. This is in encouraging agreement with the experimental observations noted earlier.

4.1 Governing equations and boundary conditions

Our governing equations for the flow are the compressible Navier-Stokes equations (see [1])

$$\rho_t + \mathrm{div}(\rho \mathbf{u}) = 0, \tag{39}$$

$$\rho \frac{D\mathbf{u}}{Dt} = -\mathrm{grad}\, p + (\lambda + \mu)\mathrm{grad}\,\mathrm{div}\,\mathbf{u} + \mu \nabla^2 \mathbf{u}, \tag{40}$$

where $\mathbf{u}$ and p are the velocity and pressure fields, λ is the dilatational viscosity and D/Dt is the usual convective derivative. Coupled to these is the energy equation (see [1])

$$\rho c_v \frac{DT}{Dt} + p\,\mathrm{div}\,\mathbf{u} = k \nabla^2 T + \Phi, \tag{41}$$

where Φ is the dissipation,

$$\Phi = \lambda(\mathrm{div}\,\mathbf{u})^2 + \frac{\mu}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)^2. \tag{42}$$

Finally, the system is closed by specifying an equation of state. Treating the air as a perfect gas, we set

$$p = \rho R T, \tag{43}$$

where $R = c_p - c_v$ is the gas constant. It is convenient to substitute (39) and (43) into the left-hand side of (41) to obtain the energy equation in the form

$$\frac{1}{\gamma - 1}\frac{Dp}{Dt} - \frac{\gamma}{\gamma - 1}\frac{p}{\rho}\frac{D\rho}{Dt} = k\nabla^2 T + \Phi. \tag{44}$$

On the upper and lower boundaries of the channel we specify the velocity $\mathbf{u} = (u, v)$ of the gas to be the same as that of the channel wall:

$$u = u_i, \quad v = \frac{\partial h_i}{\partial t} + u_i \frac{\partial h_i}{\partial x}, \quad \text{on } y = h_i(x,t); \quad i = 1, 2. \tag{45}$$

Here the horizontal velocity u_i and position h_i of each wall are determined by the prescribed shape and motion of the scroll. We now integrate (39) with respect to y and apply the boundary conditions (45) to obtain an integrated equation representing total conservation of mass:

$$m_t + q_x = 0, \quad \text{where } m = \int_{h_1}^{h_2} \rho \, dy, \quad q = \int_{h_1}^{h_2} \rho u \, dy. \tag{46}$$

The quantities m and q represent respectively the mass density of fluid in, and flux of fluid through, each cross-section.

As boundary conditions for the temperature we assume that the channel walls are thermally insulated, so that

$$\frac{\partial T}{\partial y} = \frac{\partial h_i}{\partial x}\frac{\partial T}{\partial x} \text{ on } y = h_i(x,t); \quad i = 1, 2. \tag{47}$$

We also have to match the solution in the channel with the prescribed pressure in the chamber on either side:

$$p \to P_1 \text{ as } x \to -\infty, \quad p \to P_2 \text{ as } x \to +\infty. \tag{48}$$

4.2 Nondimensionalisation and leading-order equations

In nondimensionalising the equations (39–44) we utilize the slenderness of the geometry and the difference between the velocity scales for the fluid and for the compressor. Thus we set

$$
\begin{aligned}
&x = L\,x', \quad y = d\,y', \quad u = U u', \quad v = \epsilon U v', \quad t = \frac{t'}{\omega},\\
&h_i = d\,h_i', \quad u_i = \omega L\,u_i', \quad p = \Delta P\,p', \quad \rho = \bar{\rho}\,\rho', \quad T = \frac{\Delta P\,T'}{\bar{\rho} c_v}.
\end{aligned}
\tag{49}
$$

Henceforth we drop the primes and proceed with the dimensionless variables. The Navier-Stokes equations (40), up to order Re^* and ϵ^2, reduce to the *lubrication equations* (see [1])

$$
p_y = 0, \quad u_{yy} = p_x. \tag{50}
$$

Thus,

$$
u = \frac{p_x}{2} y^2 + Ay + B, \tag{51}
$$

where, using the dimensionless version of (45), A and B are found to be

$$
A = \frac{\Omega(u_2 - u_1)}{h} - \frac{p_x(h_1 + h_2)}{2}, \quad B = \frac{\Omega(h_2 u_1 - h_1 u_2)}{h} + \frac{p_x h_1 h_2}{2}, \tag{52}
$$

setting h to be the channel height:

$$
h = h_2 - h_1.
$$

Now consider the dimensionless version of the energy equation (44) and thermal boundary condition (47), taking only the terms at leading order in ϵ:

$$
\begin{aligned}
\frac{1}{\gamma - 1}\left(\Omega p_t + u p_x + v p_y\right) - \frac{\gamma}{\gamma - 1}\frac{p}{\rho}\left(\Omega \rho_t + u \rho_x + v \rho_y\right) - u_y^2 + O(\epsilon^2)\\
= \frac{1}{\mathrm{Pe}^*}\left(T_{yy} + O(\epsilon^2)\right),
\end{aligned}
\tag{53}
$$

$$
T_y = O(\epsilon^2) \quad \text{on } y = h_1, h_2. \tag{54}
$$

Now we use the fact that Pe^* is much smaller than one to expand T as an asymptotic expansion in powers of Pe^*:

$$
T = T_0 + \mathrm{Pe}^* T_1 + \dots .
$$

Notice that, since $\epsilon^2 \ll \mathrm{Pe}^*$, it makes sense to keep terms of order Pe^* while neglecting those of order ϵ^2.

To lowest order in Pe*, the Neumann problem (53, 54) simply tells us that

$$\frac{\partial T_0}{\partial y} = 0.$$

But from (50) we know that p is also independent of y to leading order, and thus ρ_y must likewise be zero. A relation between the as yet unknown functions $p(x,t)$ and $\rho(x,t)$ is obtained from the *solvability condition* for the inhomogeneous Neumann problem satisfied by T_1. The resulting equation is

$$\Omega p_t + \bar{u} p_x - \frac{\gamma p}{\rho}(\Omega \rho_t + \bar{u}\rho_x) = (\gamma - 1)\overline{u_y^2}, \tag{55}$$

where $\bar{\ }$ denotes the cross-sectional average:

$$\bar{\cdot} = \frac{1}{h}\int_{h_1}^{h_2} \cdot \, dy.$$

Now we simply substitute in the analytic form (51) of u to obtain

$$\Omega\left(p_t - \frac{\gamma p}{\rho}\rho_t\right) = \frac{\gamma h^2 p_x}{12}\left(p_x - \frac{p}{\rho}\rho_x\right). \tag{56}$$

A second equation linking p and ρ is obtained by substituting (51) into the dimensionless form of (46), the result of which is

$$\Omega(\rho h)_t + q_x = 0, \quad \text{where } q = \frac{\Omega \rho h(u_1 + u_2)}{2} - \frac{\rho p_x h^3}{12}. \tag{57}$$

4.3 Solution in the quasi-steady limit

Equations (56) and (57) form a closed leading-order system for ρ and p; recall that u_1, u_2 and h are prescribed functions of x and t. They can be simplified further by taking the *quasi-steady limit* $\Omega \to 0$. Then, from (56) we deduce that

$$\frac{\partial}{\partial x}\left(\frac{p}{\rho}\right) = 0, \tag{58}$$

which implies that the gas is *isothermal*. So in (57) we can set $\rho = p/(T(\gamma - 1))$ where T is constant. Thus the flux

$$q = -\frac{h^3 p p_x}{12T(\gamma - 1)} \tag{59}$$

is a function only of t. Now we simply divide by h^3 and integrate with respect to x from $-\infty$ to $+\infty$, applying the matching conditions (48), to obtain

$$q = \frac{p_1^2 - p_2^2}{24T(\gamma - 1)} \left(\int_{-\infty}^{\infty} \frac{dx}{h^3} \right)^{-1} . \tag{60}$$

This tells us the flux, *i.e.* the leakage, through the channel, and how it depends on the pressure in the chamber on either side and the geometry of the channel.

Since the minimum gap thickness is very small, the integral in (60) is dominated by the behaviour of h near its minimum. In a neighbourhood of this point, we can approximate h by a quadratic function (which in general it will be locally) say

$$h = d(t) + \frac{\kappa(t)x^2}{2},$$

where κ is the *difference between the curvatures* of the two channel walls at their closest point. Then

$$\int_{-\infty}^{\infty} \frac{dx}{h^3} = \frac{3\pi}{4d^{5/2}\sqrt{2\kappa}}.$$

5 Conservation Equations for the Chambers

Now we consider the conservation of mass and energy for a single chamber of gas. Since the Reynolds number on the scale of a chamber is large, it is common practice to assume that the gas in each chamber is turbulent and thus well-mixed. Hence we can associate a spatially-uniform temperature $T(t)$ and pressure $P(t)$ with the mass $M(t)$ of gas in the chamber, which has a given volume $V(t)$.

Firstly, note that the ratio of kinetic to thermal energy is of order

$$\frac{\text{kinetic energy}}{\text{thermal energy}} \sim \frac{\bar{\rho}\omega^2 L^2}{\Delta P} = \Omega^2 \mathrm{Re}^* \ll 1.$$

Therefore we neglect kinetic energy throughout, so that the internal energy in the chamber is simply

$$E = Mc_v T. \tag{61}$$

The internal energy is changed due to:

- The work done by changes in the volume V of the chamber which, recall, is assumed to be a prescribed function of time. The work done is given by $-P\, dV/dt$.

- The energy transported into and out of the chamber through the gaps on either side. Assuming the flow through these gaps is adiabatic, the energy transported by a mass flux q is qH, where H is the *enthalpy*, equal to c_pT.

- Dissipation and energy losses to the external environment, which we denote by $\dot{Q}$.

Putting all these together, we obtain

$$\frac{dE}{dt} = -P\frac{dV}{dt} + q_i c_p T_i - q_o c_p T + \dot{Q}, \tag{62}$$

where the subscripts i and o correspond to flow into and out of the chamber respectively. Notice that the temperature of gas flowing out of the chamber is the same as that of the gas in the chamber, T.

Henceforth we neglect $\dot{Q}$, although thermal interaction with the surroundings might be included in a more refined model. Therefore, substituting (61) into (62) and rearranging, we obtain an ordinary differential equation relating P and T:

$$V\frac{dP}{dt} = \gamma R(q_i T_i - q_o T) - \gamma P\frac{dV}{dt}. \tag{63}$$

Next we consider conservation of mass for the chamber. This simply states that the mass change is equal to the flux into the chamber minus that out:

$$\frac{dM}{dt} = q_i - q_o. \tag{64}$$

But $M = \rho V$, where the density ρ is given in terms of P and T by using the equation of state (43). Thus (64) can be rearranged to a second equation relating P and T:

$$\frac{dT}{dt} = \frac{T}{P}\frac{dP}{dt} + \frac{T}{V}\frac{dV}{dt} - \frac{RT^2}{PV}(q_i - q_0)\,. \tag{65}$$

Now the idea is as follows. Given the pressure P and temperature T in two neighbouring chambers, and the geometry of the gap between them, we can evaluate the flux from one to the other using (60). Then for each chamber we have the two differential equations (63, 65) for P and T, which are coupled to the corresponding equations for the neighbouring chambers by the fluxes q on either side.

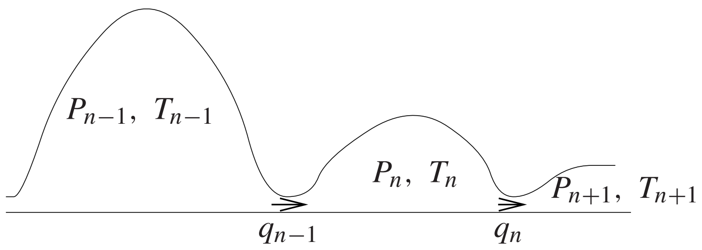


Figure 12: Schematic diagram of the coupling between neighbouring chambers.

6 The Coupled Problem

The situation is depicted schematically in figure 12. We have a series of chambers, in the n^{th} of which the gas is at pressure P_n and temperature T_n. The flux between the n^{th} and $(n+1)^{\text{th}}$ chamber is denoted by q_n, with the sign convention that $q_n > 0$ if gas flows *from* chamber n *to* chamber $(n+1)$.

From the theory of section 4, we know that in each of the gaps between the chambers the temperature is constant, but the value of that constant is determined by the sign of q: if $q_n > 0$ then the gas transported by q_n has temperature T_n, while if $q_n < 0$ it is at T_{n+1}. Thus, when we redimensionalise (60) we obtain the following expression for q_n:

$$q_n = \frac{\left(P_n^2 - P_{n+1}^2\right) d^{5/2}\sqrt{\kappa}}{9\pi\sqrt{2}\,\mu R} \begin{cases} 1/T_n & \text{if } P_n > P_{n+1}, \\ 1/T_{n+1} & \text{if } P_n < P_{n+1}. \end{cases} \tag{66}$$

Similarly, when we write down the conservation equations (63, 65) for the n^{th} chamber, whether each of q_{n-1} and q_n qualifies as flux *into* (q_i) or *out of* (q_o) the chamber depends on its sign. The resulting equations can conveniently be written in the form

$$\frac{\dot{P}_n}{P_n} = \frac{\gamma R}{P_n V_n}\left\{q_{n-1}T_n - q_n T_{n+1} + q_{n-1}^+(T_{n-1} - T_n) + q_n^+(T_{n+1} - T_n)\right\} - \frac{\gamma \dot{V}_n}{V_n}, \tag{67}$$

$$\begin{aligned}\frac{\dot{T}_n}{T_n} = -\frac{(\gamma-1)\dot{V}_n}{V_n} + \frac{R}{P_n V_n}\big\{&q_n T_n - \gamma q_n T_{n+1} \\ &+ (\gamma-1)q_{n-1}T_n + \gamma q_{n-1}^+(T_{n-1} - T_n) + \gamma q_n^+(T_{n+1} - T_n)\big\},\end{aligned} \tag{68}$$

where q^+ denotes the positive part of q:

$$q^+ = \begin{cases} q & \text{if } q > 0, \\ 0 & \text{if } q < 0. \end{cases}$$

To solve the dynamical system (67, 68) we need to apply “end conditions” at the outermost chambers. Suppose there are N chambers altogether, with the first and N^{th} open to reservoirs at *given* pressure and temperature P_0, P_N and T_0, T_N. Note that T_0 and/or T_N need only be specified if $P_0 > P_1$ and/or $P_N > P_{N-1}$, so that gas flows *into* the adjoining chambers. We also have to specify P_n and T_n for $n = 1, \ldots, N-1$ at $t = 0$. Finally, there is a complicated closure condition associated with the periodicity of the motion. Roughly speaking, it is clear that after a complete cycle, what was the n^{th} chamber has now become the $(n+1)^{\text{th}}$ chamber. The way in which this condition is implemented in practice should be made clear by the following outlined solution procedure.

1. Suppose P_n and T_n are given at $t = 0$ for $n = 1, \ldots, N$.
2. Integrate the coupled ordinary differential equations (67, 68) forward through one complete cycle, using the specified values of P_0, T_0, P_N, T_N.
3. Set
$$\begin{pmatrix} V_n \\ P_n \\ T_n \end{pmatrix}_{\text{new}} = \begin{pmatrix} V_{n-1} \\ P_{n-1} \\ T_{n-1} \end{pmatrix}_{\text{old}} .$$
4. Go to step 2.

The desired final result is a periodic solution, in which the “new” and “old” values in step 3 above are identical, that is

$$\text{periodic solution} \quad \Rightarrow \quad \begin{pmatrix} V_n(t) \\ P_n(t) \\ T_n(t) \end{pmatrix} \equiv \begin{pmatrix} V_{n-1}(t+\tau) \\ P_{n-1}(t+\tau) \\ T_{n-1}(t+\tau) \end{pmatrix},$$

where $\tau = 2\pi/\omega$ is the period of the motion. However, when there is strong coupling between the chambers it is far from clear that this periodic solution is unique or stable. Indeed, for a high-dimensional nonlinear dynamical system such as this, we might expect to see extremely complicated dynamics in general.

6.1 The small coupling limit

If the leakage is relatively small (which it should be for any worthwhile compressor), we can perturb about the zero-q solution (in which mass and entropy are preserved in each chamber)

$$P_n = P_0 \left(\frac{V_0}{V_n}\right)^{\gamma}, \quad T_n = T_0 \left(\frac{V_0}{V_n}\right)^{\gamma-1}. \tag{69}$$

Then the lowest-order fluxes are obtained by substituting (69) into the expression (66) for q_n.

Interestingly, if the compressor is in a periodic state, we only need to find q_0 to evaluate the total leakage: in its first cycle, a chamber *gains* $-q_1$ and loses $-q_0$. Then, in the next cycle, it *loses* $-q_1$ and gains $-q_2$. Over the lifetime of a chamber, all the intermediate fluxes cancel each other out, so only q_0 remains. In the small-q limit, this is readily evaluated:

$$-q_0 = \frac{P_0^2 d^{5/2}\sqrt{\kappa}}{9\pi\sqrt{2}\,\mu R T_0}\left(\frac{V_0}{V_1}\right)\left[\left(\frac{V_0}{V_1}\right)^{\gamma} - \left(\frac{V_1}{V_0}\right)^{\gamma}\right]. \tag{70}$$

Thus, assuming that d and the other parameters in (70) are constant (and beyond our control), we obtain a functional form of the total leakage:

$$\text{total leakage} \propto l := \int_0^{\tau} \sqrt{\kappa}\left(\frac{V_0}{V_1}\right)\left[\left(\frac{V_0}{V_1}\right)^{\gamma} - \left(\frac{V_1}{V_0}\right)^{\gamma}\right] dt. \tag{71}$$

We can use this functional as part of a cost function in evaluating proposed new compressor designs, for all the variables on the right-hand side of (71) can readily be evaluated for any given scroll geometry. Without performing any detailed calculations, we can immediately deduce some desirable design properties that will reduce leakage:

- The contact should be as *flat* as possible, *i.e.* κ should be minimised.
- The volume should be *reduced gradually on the first cycle*. This follows from the observation that only q_0 is relevant to the total mass loss.

7 Numerical Results

In this section we present some preliminary numerical simulations of the system (67, 68). We consider the simple configuration shown schematically in figure 13. Here, at the beginning of the cycle (diagram 1) the shaded volume of gas V_1 is taken in from the atmosphere. As V_1 decreases (diagram 2) the gas is compressed until the cycle is completed (diagram 3). Then the gas is released into a reservoir of constant volume V_2 (diagram 4). The idea is that we run the simulation through several such cycles and see what pressure we can achieve in the reservoir; this seems like a good measure of the efficacy of the compressor.

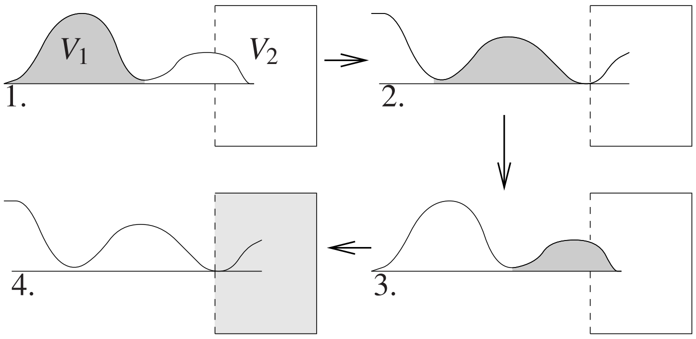


Figure 13: Schematic diagram of the compression of a single chamber (of volume V_1) which opens up into a reservoir of volume V_2.

The coupled system for the temperature and pressure in the two chambers is

$$\frac{\dot{P}_1}{P_1} = \frac{\gamma R}{P_1 V_1}\big\{q_0 T_1 + (T_0 - T_1) q_0 H(P_0 - P_1) - q_1 T_2 + (T_2 - T_1) q_1 H(P_1 - P_2)\big\} - \frac{\gamma \dot{V}_1}{V_1}, \tag{72}$$

$$\frac{\dot{T}_1}{T_1} = -\frac{(\gamma-1)\dot{V}_1}{V_1} + \frac{R}{P_1 V_1}\big\{q_1 T_1 - \gamma q_1 T_2 + (\gamma - 1) q_0 T_1 + \gamma (T_0 - T_1) q_0 H(P_0 - P_1) + \gamma (T_2 - T_1) q_1 H(P_1 - P_2)\big\}, \tag{73}$$

$$\frac{\dot{P}_2}{P_2} = \frac{\gamma R}{P_2 V_2}\big\{q_1 T_2 + (T_1 - T_2) q_1 H(P_1 - P_2)\big\}, \tag{74}$$

$$\frac{\dot{T}_2}{T_2} = \frac{R}{P_2 V_2}\big\{(\gamma - 1) q_1 T_2 + \gamma (T_1 - T_2) q_1 H(P_1 - P_2)\big\}, \tag{75}$$

where H is the Heaviside function and

$$\begin{aligned} q_0 &= \frac{d^{5/2}\sqrt{\kappa}}{9\pi\sqrt{2}\,\mu R}(P_0^2 - P_1^2)\left\{\frac{1}{T_1} + \left(\frac{1}{T_0} - \frac{1}{T_1}\right) H(P_0 - P_1)\right\} \\ q_1 &= \frac{d^{5/2}\sqrt{\kappa}}{9\pi\sqrt{2}\,\mu R}(P_1^2 - P_2^2)\left\{\frac{1}{T_2} + \left(\frac{1}{T_1} - \frac{1}{T_2}\right) H(P_1 - P_2)\right\}. \end{aligned} \tag{76}$$

With the ambient temperature T_0 and pressure P_0 given and specified chamber volume $V_1(t)$ and (constant) reservoir volume V_2, (72–76) is a well-defined problem for $\{P_1, P_2, T_1, T_2\}$. We initiate the calculations with $P_1 = P_2 = P_0$, $T_1 = T_2 = T_0$.

As outlined in section 6 at the end of each cycle we perform a replacement algorithm corresponding to (*i*) a new chamber of atmospheric gas forming in the "new" V_1; (*ii*) the "old" V_1 discharging into V_2. For the latter we find the new values of P_2 and T_2 by setting the mass and energy in V_2 after the discharge equal to the total mass and energy in V_1 and V_2 immediately prior to the discharge. Thus, after the n^{th} cycle we set

$$\begin{aligned}
P_1(n\tau+) &= P_0, \\
T_1(n\tau+) &= T_0, \\
P_2(n\tau+) &= \left[\frac{P_1V_1 + P_2V_2}{V_1 + V_2}\right](n\tau-), \\
T_2(n\tau+) &= \left[\frac{(P_1V_1 + P_2V_2)T_1T_2}{P_1V_1T_1 + P_2V_2T_2}\right](n\tau-).
\end{aligned} \tag{77}$$

In all the calculations to follow the parameter values are set as follows,

$$\begin{array}{llll}
\mu = 1, & R = 1, & \gamma = 1.4, & \kappa = 1, \\
P_0 = 1, & T_0 = 1, & V_2 = 10, & \tau = 1,
\end{array}$$

and we examine the effects of varying the "leakage parameter" d and the form of $V_1(t)$. We start with the simplest case in which V_1 is linear in t, say

$$V_1 = 1 - \alpha t,$$

where $\alpha \in (0, 1)$; $\alpha = 0$ implies a compression ratio of one, while as $\alpha \to 1$ the compression ratio goes to infinity. In figure 14 we plot the pressure in the chamber and the reservoir versus time for the case $\alpha = 0.5$, $d = 1$. We can see how P_1 increases during each cycle and is reset each time a new cycle begins. In the reservoir, P_2 varies only slightly during any cycle, and is incremented gradually at each discharge. Closer examination reveals that, because of leakage between the chamber and the reservoir, P_2 *decreases* when $P_2 > P_1$ and *increases* when $P_2 < P_1$. For these parameter values, the system appears to settle down to a periodic state after around 60 cycles, with the pressure in the reservoir enhanced by a factor of just over 2.

In figure 15 we show the corresponding behaviour of the temperature for the same parameter values. The behaviour is seen to be rather similar to that of the pressure. Henceforth we do not bother to present the results for the temperature since it is irrelevant to the total stored energy, which is proportional to P_2V_2.

In figure 16 we present the results for the pressure variations when $\alpha = 0.8$, so the compression ratio is five, compared with two in the previous calculations. As

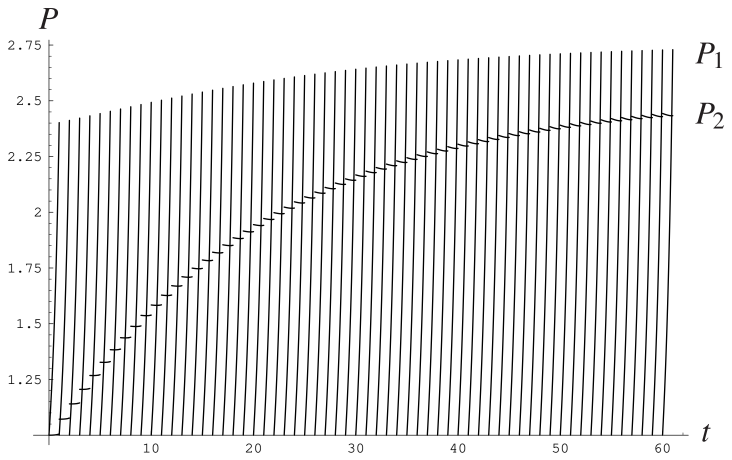


Figure 14: Pressures P_1 (in the compression chamber) and P_2 (in the pressurised reservoir) versus time. The volumes are $V_1 = 1 - 0.5\ t$, $V_2 = 10$ and the leakage parameter $d = 1$.

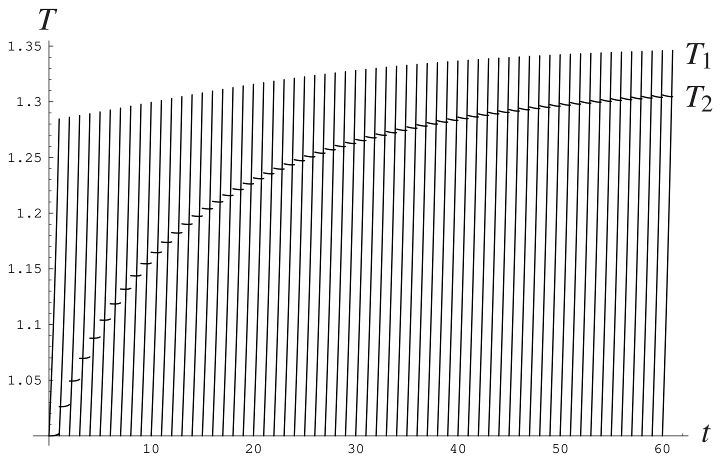


Figure 15: Temperatures T_1 (in the compression chamber) and T_2 (in the pressurised reservoir) versus time. The volumes are $V_1 = 1 - 0.5\ t$, $V_2 = 10$ and the leakage parameter $d = 1$.

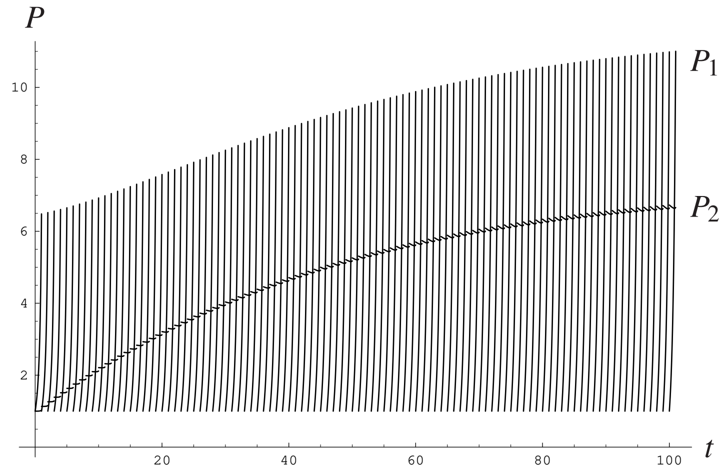


Figure 16: Pressures P_1 (in the compression chamber) and P_2 (in the pressurised reservoir) versus time. The volumes are $V_1 = 1 - 0.8\,t$, $V_2 = 10$ and the leakage parameter $d = 1$.

expected, the increased compression ratio leads to a somewhat higher final pressure, although the system also takes somewhat longer to converge to its periodic state.

In the light of figures 14 and 16 it is of interest to ask how the final pressure achieved in the reservoir depends on the compression ratio, *i.e.* on α. As a measure of this we take the average of P_2 over the 100$^{\text{th}}$ cycle:

$$\bar{P}(100) = \int_{100}^{101} P_2\,dt,$$

and plot the result versus α for various values of d in figure 17. Not surprisingly the pressure achieved increases as α is increased and as d is decreased. However, for sufficiently small d, decreasing d still further doesn't appear to have much effect; the graphs for $d = 0.25$ and $d = 0.5$ are virtually indistinguishable. This is explained by examination of the transients, which makes it clear that for these small values of d, P_2 has yet to equilibrate after 100 cycles.

So, we have shown the rather obvious results that the effectiveness of a compressor can be enhanced by increasing the compression ratio and by minimising the leakage, although these also make the convergence to maximum compression slower. Next we would like to compare compressors with the same compression ratio but different histories $V_1(t)$. Therefore we consider the family

$$V_1 = 1 - \beta t + (V_f + \beta - 1)t^2,$$

shown in figure 18, where V_f is the final volume (so the volume ratio is $1/V_f$) and β changes the volume history for a fixed V_f; $\beta = 1 - V_f$ gives the linear

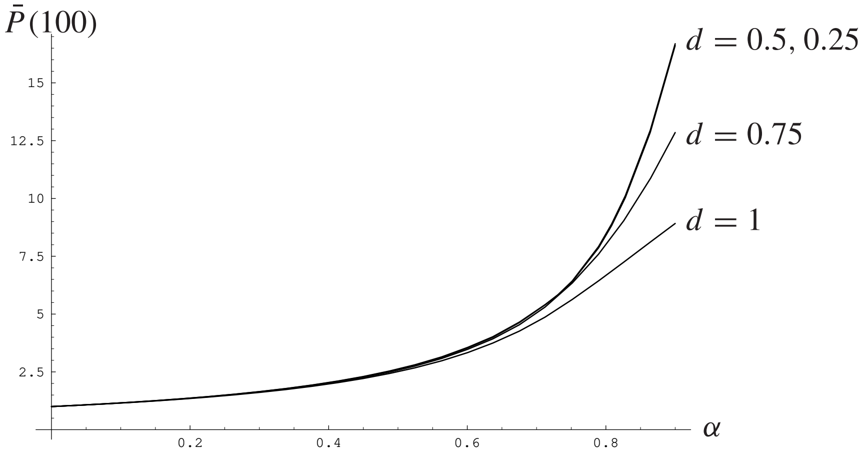


Figure 17: Average reservoir pressure over the 100th cycle versus compression parameter α. The volumes are $V_1 = 1 - \alpha t$, $V_2 = 10$ and the leakage parameter $d = 0.25, 0.5, 0.75, 1$. Notice that the curves for $d = 0.25$ and $d = 0.5$ are indistinguishable.

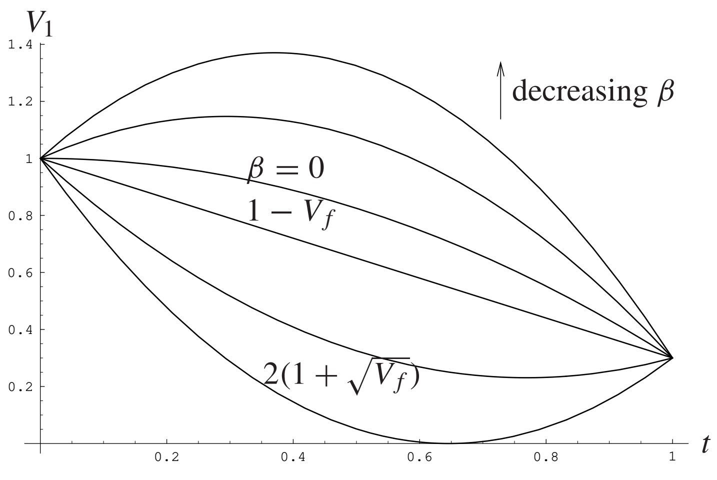


Figure 18: The family of curves $V_1 = 1 - \beta t + (V_f + \beta - 1)t^2$ for various values of β (here $V_f = 0.3$). The value $\beta = 1 - V_f$ gives a straight line joining $V_1 = 1$ at $t = 0$ to $V_1 = V_f$ at $t = 1$; $\beta = 0$ has zero gradient at $t = 0$; $\beta = 2(1 + \sqrt{V_f})$ is the value at which V_1 first reaches zero for $t \in (0, 1)$.

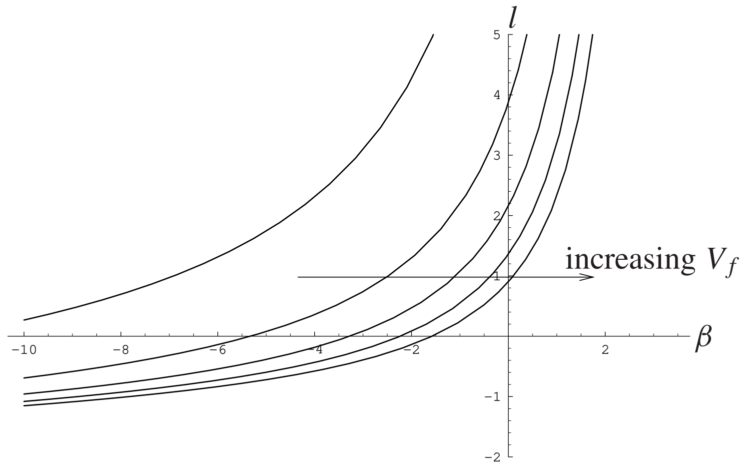


Figure 19: Theoretically predicted leakage $l = \int_0^1 (V_1^{-\gamma-1} - V_1^{\gamma-1})\, dt$ for the family of volume histories depicted in figure 18; $V_1 = 1 - \beta t + (V_f + \beta - 1)t^2$ with $V_f = 0.1, 0.2,$ 0.3, 0.4, 0.5.

$V_1(t)$, *i.e.* constant compression rate, considered previously. Broadly speaking, if β is decreased the compression is slower initially and accelerates towards the end of the cycle, and vice versa.

Firstly we check how our theoretical approximate leakage l given in (71) varies with β. In figure 19 we plot l versus β for different values of V_f. We observe that the theoretical leakage for a given volume ratio is reduced if the compression is slow at the beginning and faster at the end. This echoes the suggestion at the end of section 6 that the volume should be reduced gradually at the start. There is a surprise, however, in that l can be *negative* if β is sufficiently large and negative. This corresponds to large positive excursions in V_1 (see figure 18), as a result of which the compressor acts like a bellows, sucking extra gas into the chamber.

This conclusion is backed up by figure 20 in which we plot $\bar{P}(100)$ from our simulation versus β for different values of V_f (and with $d = 1$). Here the behaviour in general is extremely interesting; depending on the volume ratio $\bar{P}$ may be an increasing or decreasing function of β, or may vary nonmonotonically. However, for the large volume ratios which are likely to be of most practical interest, the optimal performance (*i.e.* largest possible value of $\bar{P}(100)$) is obtained by making β as large and negative as possible.

8 Conclusions

We have used the natural equation of a curve to define the geometry of the scrolls, and this allows us to obtain all the geometric properties of scroll compressor on

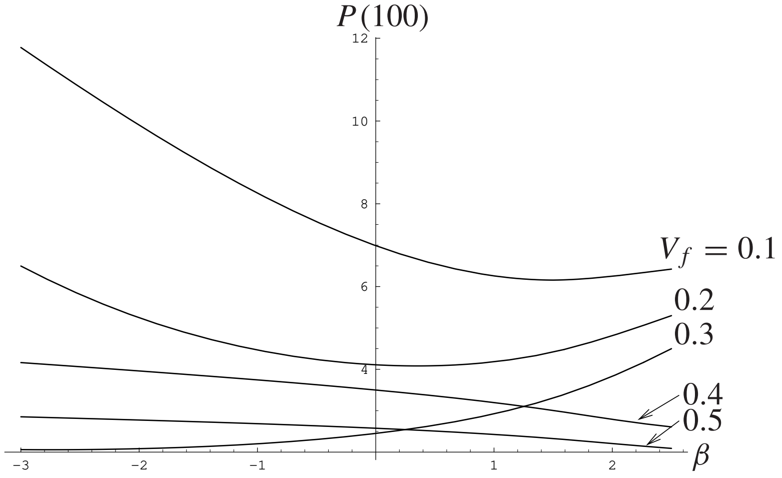


Figure 20: Average reservoir pressure over the 100[th] cycle for $d = 1$ and the family of volume histories depicted in figure 18; $V_1 = 1 - \beta t + (V_f + \beta - 1)t^2$ with $V_f = 0.1, 0.2, 0.3, 0.4, 0.5$.

symbolic form if we give the arc length as a polynomial in the tangent direction. The formulas are especially simple if we restrict ourself to compressors with a circular orbit.

We have characterized all compressors with a circular orbit and constant wall thickness, and have argued that these compressors do not offer any advantages over the standard circle involute compressor.

We have outlined a procedure which design a compressor given the natural equation of one of the sides, the radius of the circular motion, and the thickness of the scroll wall at one place. That compressor has the property that the two scrolls are the same just rotated 180° degrees relative to each other.

There is no guarantee that the scroll compressors constructed this way are physical realizable, basicly the radius of the circular motion and the thickness of the walls should be sufficiently small. It would be possible to give estimates on the bounds for these two numbers, but this has not been investigated.

We have used compressible lubrication theory to obtain a theoretical prediction of the leakage between adjoining chambers in a scroll compressor. One surprising outcome of the analysis is that the gas flowing through the narrow gap between one chamber and the next is *isothermal*. In traditional inviscid gas dynamics one would expect the temperature to decrease as the gas accelerates through the gap, while classical lubrication theory would predict an increase in temperature due to viscous dissipation. Remarkably these two effects appear to cancel each other out exactly.

Our result for inter-chamber leakage was then incorporated in a coupled model for the pressure and temperature of the gas in each chamber. The model takes the form of a dynamical system, whose general properties certainly warrant further investigation. Our analysis was limited to examination of the case in which the leakage is small. In this limit we obtained an approximate measure of the total losses due to leakage, as a functional of the geometry of the compressor. This could in future be included in a cost function for evaluating proposed new compressor designs.

We performed some numerical simulations of a simple compressor with just one chamber pumping gas into a sealed reservoir. From these we were able to test the effects on compressor performance of varying the leakage rate, the volume ratio and the volume history. The most intriguing possibility suggested by the simulations is that of making the chamber volume *increase* initially, before a rapid compression just prior to discharge. This allows the compressor to use the leakage in its favour by sucking more gas in from the atmosphere.

Many potentially important physical effects have been neglected in this study, and might be considered in the future to refine the model further. These include thermal losses through the compressor walls, and the "squeeze film" and "Couette" effects in the lubrication analysis (*i.e.* the terms multiplied by Ω in (56, 57)). Perhaps most importantly, we have restricted our analysis to a two-dimensional configuration, and thus have not considered leakage through the sides of the chambers. The results of section 4 should be applicable to these regions also.

Finally we have demonstrated how to combine the geometry and the leakage analysis to estimate the leakage of a compressor with a given geometry.

Acknowledgement

We thank DANFOSS and in particular Stig Helmer Jørgensen for providing us with an interesting problem and answering many question during the week.

Bibliography

[1] Ockendon, H. & Ockendon, J. R., *Viscous Flow*, Cambridge University Press, Cambridge, 1995.

The Group Working on the Problem

Jens Gravesen, [J.Gravesen@mat.dtu.dk]
Vagn Lundsgaard Hansen, [V.L.Hansen@mat.dtu.dk]
Christian Henriksen, [Christian.Henriksen@mat.dtu.dk]
Peter Howell, [howell@math.ox.ac.uk]
Stig Helmer Jørgensen, [SHJ@danfoss.com]
John W. Perram, [jperram@mip.ou.dk]

DANISCO: Temperature and Moisture Gradients in Sugar Silos

ANDREW LACEY AND MADS PETER SØRENSEN

1 Introduction

Danisco Sugar is the main sugar producer in Denmark. The sugar is extracted from sugar beets which are harvested during the beet campaign September – December. The refined sugar is stored in huge silos and withdrawn on a regular, but not constant, basis for distribution until the next campaign. Danisco Sugar utilizes several types of silos for the storage of crystalline sugar. They differ in construction material e.g. concrete or steel plate silos, dimension, design with or without a central tower and isolation. Various methods of filling and emptying are utilized, that is first in – last out, or first in – first out. The silos are equipped with one or more facilities for sugar conditioning. Those includes temperature control of walls, control of the condition of the air space above sugar, ability to blow conditioned air through the bulk sugar etc.

A number of problems arise when storing sugar for long periods. Due to distribution of heat and moisture the sugar can 'cake' or harden and moist sugar is difficult to empty from the silos. The chemical and microbial stability of the sugar is influenced by the temperature and moisture distribution. On the other hand under very dry conditions there is a risk of dust explosions. The object is to control the temperature and moisture content within a lower and an upper limit. Danisco Development Centre wishes to employ mathematical modelling as a tool for understanding and predicting temperature and moisture distribution in sugar stored in silos.

In moisture migration the physical processes which have to be taken into account are diffusion in the porous bulk sugar, natural convection due to temperature gradients, forced convection due to bulk sugar air conditioning, conditioning of the air above the bulk sugar and migration of moisture through concrete silo walls. The temperature distribution is influenced by seasonal variations, the temperature of the air space above the bulk sugar, remixing processes when filling and emptying silos, air conditioning and others. The following physical properties are assumed of importance when formulating models to describe the formation of temperature and moisture gradients: bulk sugar heat capacity, heat conductivity, density, porosity, tortuosity and sorption isotherms. Most will vary with the sugar quality (that is: crystal size, reducing sugar content and ash content), some with temperature and moisture.

Mathematical models for moisture and heat conduction in stored cereal grains have previously been developed [1, 2]. In modelling moisture and heat conduction in sugar our starting point is to use the approach applied in modelling moisture and heat conduction in grain and adopt the model to the case of stored sugar, including the possibility of air conditioning.

In our modelling carried out during the week of the Study Group, we used a number of simplifying assumptions, such as air flow through the sugar being dominated by forced, and constant, convection, and having processes happening fast enough, so that local equilibrium holds. (Estimates obtained at the time indicated that the temperature varies very little over the length scale of the sugar grains and their interstices.)

2 Sorption Isotherms

In the sugar bulk water can either be adsorbed onto the sugar crystalline in form of liquid water (moisture) or exist as water vapour in the interstitial air. The latent heat consumed during evaporation of moisture from the sugar to the air is denoted by ℓ (J/kg of H_2O). The moisture content W_s in the sugar is defined as the mass of moisture divided by the mass of dry sugar. The relative humidity h of the interstitial air is defined as the fraction p_v/p_s between the partial pressure of water vapour p_v and the saturation pressure of water vapour p_s. In equilibrium the relative humidity and the the moisture content W_s of sugar are related through a sorption isotherm at fixed temperature T. In mathematical terms we can express this relation in the form

$$\frac{p_v}{p_s} = f(W_s, T) \ , \tag{1}$$

where f is a function depending on the moisture content and temperature. The sorption isotherms are determined from experimental data and are normally presented as graphs showing W_s as function of the relative humidity at different temperatures. A typical example is displayed in Fig.1. We assume in the present model that the air-vapour mixture behaves as an ideal gas. The ideal gas law for vapour reads

$$p_v = \rho_v R_v T \ , \tag{2}$$

where p_v is the partial pressure of water vapour in air. The density of vapour is denoted ρ_v and R_v is the vapour gas constant in units of J/(kg °K). Here R_v is related to the universal gas constant $\hat{R} = 8317$ J/(kmol °K) through $\hat{R} = \hat{M}_v R_v$ where $\hat{M}_v$ denotes the molar mass of vapour. A similar expression holds for the partial pressure of air. The absolute humidity W_a is defined as

$$W_a = \frac{m_v}{m_a} = \frac{\rho_v}{\rho_a} \, . \tag{3}$$

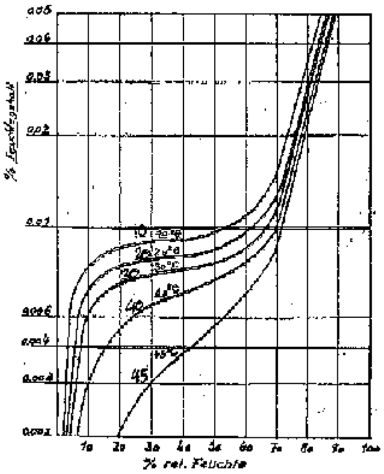


Figure 1: Sorption isotherms for equilibrium moisture relationship for sugar.

The masses m_v and m_a are the masses of vapour and dry air, respectively, in a fixed volume.

The sugar bulk is a porous medium which contains sugar crystals, liquid water, water vapour and air. The volume fraction of sugar is denoted ϕ_s. This means that the mass of sugar in a volume V_0 of the bulk is $m_s = \rho_s \phi_s V_0$, where ρ_s is the density of sugar. Similarly the volume fractions of water, air and vapour are ϕ_w, ϕ_a and ϕ_v, respectively. However, the volume fraction of air and vapour is fixed, and equals the porosity ε of the bulk; the fraction ϕ_w is believed to be normally very small.

With the above notation we can write the absolute moisture content of sugar in mathematical terms as

$$W_s = \frac{m_w}{m_s} = \frac{\rho_w \phi_w}{\rho_s \phi_s} \, , \tag{4}$$

where m_w is the mass of moisture within the sugar in a given volume V_0 of the bulk and m_s is the mass of sugar in V_0.

3 The Heat Equation

In the derivation of the equations describing heat conduction and moisture conduction, we shall assume thermodynamic equilibrium between the wet sugar and the surrounding air in the porous media. This implies that the temperature in a point of the bulk is the same for all phases and that the sorption isotherms can be used. Air flowing through the bulk sugar in order to dry or wet the sugar flows with a velocity **u** (averaged over a total volume, and not just the pores) and gives rise to convective transport of heat and moisture. The flow velocity is assumed here to be constant. We shall assume **u** to be sufficiently small so that the thermodynamic equilibrium is satisfied even in the convective case. This means that the system must reach equilibrium from an out of equilibrium case much faster than moisture is transported due to convection. With these assumptions the heat equation now reads

$$\phi_s \rho_s C_s \frac{\partial T}{\partial t} + \rho_w C_w \frac{\partial(\phi_w T)}{\partial t} + \varepsilon C_a \frac{\partial(\rho_a T)}{\partial t} + \varepsilon C_v \frac{\partial(\rho_v T)}{\partial t} \\ + \nabla \cdot (\varepsilon \rho_a C_a T \mathbf{u}) + \nabla \cdot (\varepsilon \rho_v C_v T \mathbf{u}) = \nabla \cdot (k \nabla T) + \ell \frac{\partial(\phi_w \rho_w)}{\partial t} . \quad (5)$$

The constants C_s, C_w, C_a and C_v are the specific heat of sugar, water, air and vapour, respectively. The constant k is the thermal conductivity of the porous bulk and it arises as an average of thermal conductivity over all the phases appearing in the bulk sugar. In the above equation we have used that the volume fractions of air and vapour are identical and equal to the porosity ε. Compared to sugar and water the heat capacity and volume fraction of air and vapour are so small that we shall neglect those terms in the first line of Eq.(5). However, we keep the convective terms. Invoking Eq.(4) the heat equation reduces to

$$\phi_s \rho_s C_s \frac{\partial T}{\partial t} + \phi_s \rho_s C_w \frac{\partial(W_s T)}{\partial t} + \varepsilon C_a \nabla \cdot (\rho_a T \mathbf{u}) + \varepsilon C_v \nabla \cdot (\rho_v T \mathbf{u}) \\ = \nabla \cdot (k \nabla T) + \ell \phi_s \rho_s \frac{\partial W_s}{\partial t} . \quad (6)$$

This equation involves four dependent variables, namely temperature T, the absolute moisture content in sugar W_s, and the air and vapour densities. From the ideal gas law these densities can be related to the partial presures of air and vapour. The partial presures give the relative humidity from which one can calculate the moisture content W_a in air. This is the reason why we need not involve W_a in the dynamical equations.

4 Moisture Migration

The moisture in the sugar does not migrate. The moisture migration takes place by evaporation and water-vapour diffusion through the air pores and by forced convection. Thus in the diffusion and convection equation for moisture migration, evaporation of water from the sugar acts as a source term and accordingly we have

$$\frac{\partial(\phi_v\rho_v)}{\partial t} + \nabla\cdot(\phi_v\rho_v\mathbf{u}) = \nabla\cdot\left(\frac{D_v}{\tau}\nabla(\phi_v\rho_v)\right) - \frac{\partial(\phi_w\rho_w)}{\partial t}\ . \tag{7}$$

The parameter D_v is the diffusion constant of water vapour in air and τ is the tortuosity factor of the porous media. Using the fact that the volume fraction of vapour ϕ_v equals the porosity ε and invoking Eq.(3) together with the ideal gas law we get

$$\phi_v\rho_v = \varepsilon\rho_a W_a = \varepsilon\frac{p_v}{R_v T}\ . \tag{8}$$

Inserting this expression into Eq.(7), neglecting the first term and using Eq.(4) leads to

$$\phi_s\rho_s\frac{\partial W_s}{\partial t} + \varepsilon\nabla\cdot\left(\frac{p_v}{R_v T}\mathbf{u}\right) = \varepsilon\nabla\cdot\left(\frac{D_v}{\tau}\nabla\left(\frac{p_v}{R_v T}\right)\right)\ . \tag{9}$$

In order to use the ideal gas law above, we assume local thermodynamic equilibrium of the water vapour. This may not always be a sufficiently good approximation. The final equation needed is the continuity equation for the air flowing in the pores. This equation reads

$$\frac{\partial\rho_a}{\partial t} + \nabla\cdot(\rho_a\mathbf{u}) = 0\ . \tag{10}$$

In the previous section we mentioned that the system is described by four dependent variables. These dependent variables are governed by the heat equation (6), the moisture migration equation (9), the continuity equation (10) and the sorption isotherms (1). The forced flow velocity **u** is determined from Darcy's law

$$\mathbf{u} = -\mu\nabla P - \rho_a g\mathbf{k}\ . \tag{11}$$

The presure gradient ∇P is considered an average along the entire height of the sugar pile in the silo. Accordingly, we calculate an average convection flow **u**. The constant g is the gravitational constant and $\mathbf{k}$ is the unit vector directed vertically upwards. The permeability is denoted by μ.

We need to specify initial conditions and proper boundary conditions for our problem. The boundary condition for the temperature is [4]

$$k\frac{\partial T}{\partial n} + H(T - T_{\text{out}}) = 0 \ , \tag{12}$$

which states that the heat flow across the silo walls is proportional to the temperature difference between the sugar at the silo surface and the outdoor temperature T_{out}. The proportionality constant H is the coefficient of surface heat transfer. The vector $\mathbf{n}$ is the outward unit normal at the surface of the sugar and $\partial/\partial n$ denotes an outward normal derivative. The surface heat-transfer constant will depend on the the silo wall material, degree of insulation, natural and forced convection. The boundary condition may also include a black-body radiation term, T^4-law. However, by linearization this effect can be absorbed in adjustments of H.

Boundary conditions for the moisture should also be given. Assuming no migration of moisture across the silo walls we have

$$\frac{D_v}{\tau}\frac{\partial}{\partial n}(\phi_v \rho_v) - \mathbf{n} \cdot \phi_v \rho_v \mathbf{u} = 0 \ . \tag{13}$$

Moreover, air does not flow across the silo walls and therefore $\mathbf{n} \cdot \mathbf{u} = 0$; Eqn.(13) reduces to

$$\frac{D_v}{\tau}\frac{\partial}{\partial n}(\phi_v \rho_v) = 0 \ .$$

In the case where conditioning air is blown into the sugar pile from below, the pressure p_a can be assumed fixed at the bottom and at the top of the silo of the stored sugar. From the ideal gas law we can then determine ρ_a at the bottom and at the top. Furthermore it is expected that the relative humidity of the air blown into the sugar is known and therefore p_v is also known from which ρ_v can be calculated. In the case where a considerable amount of moisture is transported through the silo walls, a suitable source term appears on the right-hand side of (13). At present we will not specify the boundary conditions in further detail as they will depend on a given silo geometry, construction materials, etc.

5 Scaling

The heat and the moisture migration equations can be written in dimensionless variables through the following scaling

$$\begin{aligned} x = k_x \tilde{x}, \qquad & y = k_x \tilde{y}, \qquad z = k_x \tilde{z}, \qquad t = k_t \tilde{t}, \\ T(x,y,z,t) = k_T \tilde{T}(\tilde{x},\tilde{y},\tilde{z},\tilde{t}), \qquad & W_s(x,y,z,t) = k_{ws} \tilde{W}_s(\tilde{x},\tilde{y},\tilde{z},\tilde{t}), \\ \rho_a(x,y,z,t) = k_{\rho a} \tilde{\rho}_a(\tilde{x},\tilde{y},\tilde{z},\tilde{t}), \qquad & \rho_v(x,y,z,t) = k_{\rho v} \tilde{\rho}_v(\tilde{x},\tilde{y},\tilde{z},\tilde{t}), \\ p_a(x,y,z,t) = k_{pa} \tilde{p}_a(\tilde{x},\tilde{y},\tilde{z},\tilde{t}), \qquad & p_v(x,y,z,t) = k_{pv} \tilde{p}_v(\tilde{x},\tilde{y},\tilde{z},\tilde{t}). \end{aligned} \tag{14}$$

The dimensionless variables are marked by a tilde and the constants have the same dimension as the original physical variables, that is k_x has dimension length, etc. In normalized variables the ideal gas law (2) for dry air can be written

$$\tilde{p}_a = \frac{k_{\rho a} R_a k_T}{k_{pa}} \tilde{\rho}_a \tilde{T} = \tilde{\rho}_a \tilde{T} \ , \tag{15}$$

by choosing the normalization constants in such a way that the factor in front of $\tilde{\rho}_a \tilde{T}$ equals unity. Out of the three independent normalization constants, we shall choose $k_{\rho a}$ and k_T according to the values in Table 1. The constant k_{pa} is then calculated from Eq.(15).

$k_{\rho a} = 1.225$ kg/m^3	$k_T = 298$ °K
$k_{pa} = 1.05 \cdot 10^5$ N/m$^2 \approx 1.$ atm.	$R_a = 287$ J/(kg °K)

Table 1.

The ideal gas law for the water vapour reads in normalized variables

$$\tilde{p}_v = \frac{k_{\rho v} R_v k_T}{k_{pv}} \tilde{\rho}_v \tilde{T} = \tilde{R}_v \tilde{\rho}_v \tilde{T} \ . \tag{16}$$

In this equation we have chosen $k_{\rho v} = k_{\rho a}$, $k_{pv} = k_{pa}$ and introduced the normalized gas constant $\tilde{R}_v$ for vapour given by

$$\tilde{R}_v = \frac{k_{\rho v} k_T}{k_{pv}} R_v \ . \tag{17}$$

Inserting the scaling (14) into the heat equation (6) we obtain

$$\frac{\partial \tilde{T}}{\partial \tilde{t}} + a_c \frac{\partial(\tilde{W}_s \tilde{T})}{\partial \tilde{t}} + a_1 \nabla \cdot (\tilde{\mathbf{u}} \tilde{p}_a) + a_2 \nabla \cdot (\tilde{\mathbf{u}} \tilde{p}_v) = \nabla^2 \tilde{T} + \frac{\partial \tilde{W}_s}{\partial \tilde{t}} \,. \tag{18}$$

The constants a_1, a_2 and the normalized air flow velocity $\tilde{\mathbf{u}}$ are given by

$$a_c = \frac{C_w}{C_s} k_{ws}, \quad a_1 = \varepsilon \frac{C_a}{C_s} \frac{k_{\rho a}}{\phi_s \rho_s}, \quad a_2 = \varepsilon \frac{C_v}{C_s} \frac{k_{\rho v}}{\phi_s \rho_s} \frac{1}{\tilde{R}_v}, \quad \tilde{\mathbf{u}} = \frac{k_t}{k_x} \mathbf{u}, \tag{19}$$

and

$$k_x^2 = \frac{k}{\phi_s \rho_s C_s} k_t, \qquad k_{ws} = \frac{C_s}{l} k_T. \tag{20}$$

All normalization constants are thus fixed except the time scaling k_t. The processes we are interested in occur over a time scale of order months. Therefore we choose k_t to be 30 days ($k_t = 2.59 \cdot 10^6$s). Typical values for the physical constants entering our problem are stated in Table 2. Some of them are given by the values found in [1]. As an example the diffusivity of water vapour D_v is calculated from [3]

$$D_v = \frac{9.1 \cdot 10^{-9} T^{2.5}}{T + 245.18} \,. \tag{21}$$

$k = 0.3$ J/(s m °K)	$\phi_s = 0.5$
$\rho_s = 1.610^3$ kg/m^3	$C_s = 1.2$ kJ/(kg °K)
$R_v = 462$ J/(kg °K)	$C_a = 1.01$ kJ/(kg °K)
$C_v = 1.88$ kJ/(kg °K)	$C_w = 4.18$ kJ/(kg °K)
$\varepsilon = 0.4$	$l = 2440$ kJ/kg
$D_v = 2.57 \cdot 10^{-5}\mathrm{m}^2/\mathrm{s}$	$\tau = 1.53$

Table 2.

The moisture migration equation (9) written in terms of the normalized variables becomes

$$\frac{\partial \tilde{W}_s}{\partial \tilde{t}} + b_1 \nabla \cdot \left(\tilde{\mathbf{u}} \frac{\tilde{p}_v}{\tilde{T}} \right) = b_2 \nabla^2 \left(\frac{\tilde{p}_v}{\tilde{T}} \right) , \tag{22}$$

and here we have introduced the constants

$$b_1 = \frac{\varepsilon k_{pv}}{\phi_s \rho_s k_{ws} R_v k_T} \quad \text{and} \quad b_2 = \frac{\varepsilon k_t D_v k_{pv}}{\phi_s \rho_s k_{ws} k_x^2 \tau R_v k_T} . \tag{23}$$

In normalized coordinates the continuity equation for air reads

$$\frac{\partial \tilde{\rho}_a}{\partial \tilde{t}} + \nabla \cdot (\tilde{\rho}_a \tilde{\mathbf{u}}) = 0 , \tag{24}$$

and the sorption isotherms are described by

$$\tilde{p}_v = \tilde{p}_v(\tilde{W}_s, \tilde{T}) . \tag{25}$$

With k_t equal 30 days the spatial scaling factor k_x takes the value 0.901 m using the physical variables stated in Table 1 and 2. As the size of a silo is large compared with k_x, this means that spatial dependencies of the temperature, humidity, etc. are predominant at the boundaries of the silo and accordingly we expect that the model can be treated as a boundary layer problem.

The distance scaling indicated by (20) comes from balancing heat conduction with rate of change of the temperature. It also gives a similar balance in the water equation, e.g. (22). Over the lengths comparable with larger silos (say height or radius of the order of 10 m), these terms cannot balance – except in boundary layers, for instance near the silo wall. However, it is found, for speeds of flow used in forced convection, that the convective term (in the upward, z, direction) still balances the time rate of change in (22). Thus, returning to dimensional variables, away from the boundary layer(s) temperature T is constant (to leading order) while water content satisfies, approximately,

$$\phi_s \rho_s \frac{\partial W_s}{\partial t} + \varepsilon u \frac{\partial}{\partial z} \left(\frac{p_v}{R_v T} \right) = 0,$$

with p_v and W_s being related through the relevant sorption isotherm; this is a nonlinear, first-order, hyperbolic equation. (With ε and μ being constant, $\mathbf{u}$ is constant and directed vertically upwards, the z direction.) In the boundary layer near the silo wall the diffusive terms in the horizontal, say x, direction (normal to the wall) now play a role and the two-dimensional problem,

$$\phi_s \rho_s C_s \frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2} , \quad \phi_s \rho_s \frac{\partial W_s}{\partial t} + \varepsilon u \frac{\partial}{\partial z} \left(\frac{p_v}{R_v T} \right) = \frac{\varepsilon D_v}{\tau} \frac{\partial^2}{\partial x^2} \left(\frac{p_v}{R_v T} \right) ,$$

has to be solved. (The space variables x and z should be scaled differently from each other in the nondimensional problem.) It is seen that T is independent of z (away from any boundary layers near the top and bottom of the sugar) and, as it satisfies the linear heat equation, can be determined explicitly. The remaining problem for the moisture is then a nonlinear convection diffusion equation (the two transport mechanisms take place in different directions). It is also inhomogeneous due to the presence of $T(x, t)$.

It should be noted that the other terms, including the effects of latent heat and of specific heat of adsorbed moisture, were found, on carrying out the scaling, to be negligible.

In solving the coupled equations (18, 22, 24, 25, 15) more generally, for a given silo geometry, a numerical approach seems to be the only adequate way to arrive at a useful and practical solution. A simple and also reliable method is the method of lines. Here the space is discretized according to $x = i\,dx$, $i = 0, 1, ..., n_x$, and similarly for the y- and z-directions. All the dependent variables are defined at discrete points (i, j, k). Spatial derivatives are replaced by central differences of second order in the discretization lengths dx, dy and dz, and inserted into the governing system of partial differential equations. The resulting system of coupled ordinary differential equations (ODEs) can now be solved numerically by using standard ode solvers. Alternative methods are e.g. the finite element method which is particularly suitable for silos with complex geometry. In reference [1] a control volume discretization procedure has been employed in order to solve moisture migration in stored grain. This approach could be adopted to our problem and formulated in such a way that silos with complex geometry can be handled.

Concerning equation (25) we need an analytical function of the sugar moisture content and temperature which approximates the sorption isotherms shown in Fig.1. For grain an example is presented in [1].

6 Concluding Remarks

The main result of the above work is the derivation of a mathematical model for moisture migration through stored sugar. The model includes air conditioning of the sugar. We have assumed thermodynamic equilibrium between the water in the sugar and the vapour in the interstitial air described by the sorption isotherms. This assumption may be invalid if air conditioning is employed with a high flow velocity, due to the time necessary to equilibrate. The flow of the conditioning air is determined from Darcy's law. This equation can be solved in conjunction with the heat and moisture equations. However, it is believed that the air flow is approximately constant, and accordingly Darcy's law is used to calculate a uniform flow velocity from the bottom to the top of the silo. Scaling of the governing equations indicates that the problem has at least one boundary layer.

The next step is to solve the model equations. For practical and realistic situations we should apply numerical methods in computer simulations. A number of methods are available. We have mentioned three of which the method of lines may be the fastest and most easy to implement on a computer. This method is suitable for silo geometries which are not too complex. Otherwise one could use the finite element method or a finite volume method for more complex geometries. The content of sugar in a silo is lovered through the year and the top surface of the sugar pile may be rather irregular.

Bibliography

[1] K. K. Khankari, R. V. Morey, S. V. Patankar, Mathematical model for moisture diffusion in stored grain due to temperature gradients. *Transactions of the ASAE* 37(5):1591–1604, 1994.

[2] H. Tanaka and K. Yoshida, Heat and mass transfer mechanisms in a grain storage silo. *Engineering Sciences in the food industry*. Elsevier Applied Science Publishers, Essex, England. 89–98, 1984.

[3] G. R. Thorpe, Moisture diffusion through bulk grain. *J. Stored Prod. Res.* 17:39–42, 1980.

[4] H.S. Carslaw and J.C. Jaeger, *Conduction of Heat in Solids*, Oxford University Press, Oxford, UK, 1959.

The Group Working on the Problem

Janis Cepitis, [cepitis@lanet.lv]

Erik Bent Hansen, [E.B.Hansen@mat.dtu.dk]

Andrew Lacey, [A.A.Lacey@ma.hw.ac.uk]

Tiina Roose, [roose@maths.ox.ac.uk]

Mads Peter Sørensen, [mps@imm.dtu.dk]

Jannik Vindeløv, [javi@kvl.dk]

Danish Maritime Institute: Dynamic Positioning System

Jens Christian Larsen, Henrik Rasmussen, and Allan Sørensen

1 Introduction

In this report, we consider the holding problem for a ship with a given arrangement of thrusters, i. e., how to ensure that a ship, floating on the surface of the ocean, stays very nearly at rest with respect to the sea bed. The aim is to determine the region of forces and torques that can be obtained by varying the directions and powers of the thrusters. In addition, for a given force and torque in this region, we would like to find, in real time, the directions and powers that should be assigned to the different thrusters.

It is important to know the holding capability when faced with the task of to keep a ship, or oil rig, at a fixed position relative to the sea bed. Unless the thrusters can be arranged such that they cancel the external forces from wind, waves, and currents, the ship/rig will drift. This can have unfortunate consequences. If an oil rig, for instance, unexpectedly starts drifting, the pipes might leak oil directly into the sea. However, knowledge of the holding capability, and the external forces in different weather conditions, makes it easier to avoid this situation.

In the section 2 we consider the problem as a discretised optimization problem and outline a procedure which give a solution to the problem. In section 3 we consider a Lagrangian formulation where we use Lagrangian multipliers to represent the side conditions. Finally in section 4 we have used Maple® to analyse a slightly different Lagrangian.

2 Solving the Problem as a Discretized Optimization Problem

In the solution described here, we assume that we are given the two components of the force F_x and F_y and the torque M_z, needed to compensate for external forces that acts on the ship, from a controller on the ship. We are not able to change the sampling period or the used strategy of this controller. We consider the following two problems:

1. The maximum holding capability problem
2. The dynamical positioning problem

2.1 The proposed solution to the problem

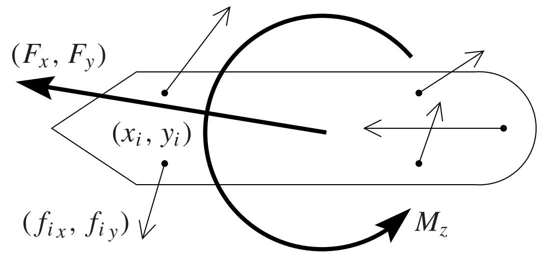


Figure 1: The required force and torque are indicated by thick arrows, while the arrangement of thrusters are indicated by thin arrows.

We call the external forces from wind, current and waves for the wind vector and denote it **W**. The vector $(F_x, F_y, M_z)^T$, describing the two components of the force and the torque needed on the boat is called the force vector and is denoted **F**. A force vector is the sum of the contributions from all the thrusters on the ship. Here we will talk about the F-space, the space of all possible vectors **F**. The relation between **W** and **F** is nonlinear, but for any given **W** there corresponds a unique vector **F**, and to each **F** it is possible to determine the possible combinations of the external forces **W**.

The first problem may be considered as a static optimization problem in the F-space. This is because the maximum holding capability problem is to determine the maximum wind that the boat is able to resist keeping a fixed position and orientation. Because of the nonlinearity between the wind vectors and the force vectors we prefer to work in F-space. That is, we want to determine the envelope E_F containing all the force vectors that can be constructed given the limitations on the thrusters. This is done by discretizing the F-space and for all possible orientations of the force vectors determine the maximum norm.

The force vector is as mentioned the sum of the contributions from all the thrusters on the ship. The ith thruster contribute with the force components f_{ix} and f_{iy} that acts on the ship on the position (x_i, y_i). Each thruster has a characteristic that depends upon the orientation of the thruster.

The determination of the maximum length of a force vector in a given direction is an optimization problem. We have as mentioned the constraints on the thrusters. Assuming convexity one can directly make use of the Kuhn-Tucker condition for an extremum and thereby obtain the maximum-force envelope E_F, see for example [1]. In the case of non-convexity one has to check the value at each extrema. The latter is what we do.

Having determined E_F one is able to consider the second problem, the dynamical positioning system. Again this is an optimization problem. If we consider the case where the thrusters are able to compensate for the applied wind, i.e. we are inside E_F, then we are looking at the following. At a given time t_i the controller compute a force vector $\mathbf{F}_i$ that is needed to control the position and orientation of the ship. We look at the related optimization problem:

$$\min \sum_{i=1}^{n} \left({f_i}_x^2 + {f_i}_y^2 \right)$$

given the constraints on the n thrusters etc. That is we want to find the most favorable solution with respect to the consumed energy. Again we are going to check the value at each extrema that are found by the Kuhn-Tucker conditions.

2.2 Limitations and enhancements

In the solution that we have suggested we are solving a discretized optimization problem. The reason for this is because the characteristic of the thrusters were discretized. We are able to find the finite set of local extrema by making use of the Kuhn-Tucker conditions.

Now, it is possible to obtain a continuous description of the characteristics of the thrusters. Therefore, in this case one might want to consider the continuous optimization problem instead. It is obviously not the case that it is a convex optimization problem. For example the angular-dependence of the effectiveness of the thrusters varies, due to jet-streamlines, from about 10% to 100%. To give a rough estimate on the holding capability, our discretized solution is perhaps sufficient, but with respect to the viewpoint of control theory, it should be noted, that the envelope that we have found is smaller than the actual one due to the discretization.

The strategy of control at the moment is first to keep the orientation correct then the x-coordinate of the position and finally the y-coordinate, but all these conditions are obtainable when working inside E_F. Therefore, it would of course be useful with more precise values on the maximum norms of the force vectors to make the control of the ship better.

There are several techniques to solve continuous non convex optimization problems, see for example [4]. When the problem is to determine E_F there are at least the following two useful methods: simulated annealing and dynamical annealing. These stochastic methods quickly converges to the interesting regions in the optimization space. Both techniques are roughly based on the idea of modeling the problem as a physical system where one is pumping energy into the system and then slowly pulling the energy out again.

If we have determined the maximum holding capability envelope it is possible to determine whether a given desired force vector is inside or outside the envelope. Now, there are some inertia in the system. When going from time t_i to time t_{i+1} the maximum angle the thrusters are able to turn is 6 degrees. What this means is, that if we at time t_i are at a given point in F-space then we are only able to operate within a finite subset inside the envelope at time t_{i+1}. But having determined E_F the choice of control strategy should be much easier.

3 A Lagrangian Formulation

In the previous section the holding capability of a ship with n thrusters was determined by searching through the $2n$-dimensional configuration space. Since the number of possible configurations in this phase space grows exponentially with n, for any given discretisation of the variables, this search through phase space typically takes $4-5$ hours of computing time even on a reasonably powerful machine.

The method of solution proposed in this section should be considerably faster. We write down a Lagrange function consisting of the function to be maximised and a number of extra terms representing the constraints on the variables. We show that extrema of this Lagrangian can be determined by finding extrema of n real-valued functions on the real line. The number of calculations needed to determine the extrema and their positions in phase space is proportional to n.

3.1 Mathematical formulation

Let (x_i, y_i) denote the position of the ith thruster relative to the centre of mass of the ship. Let θ_i denote its direction and f_i the power. Due to the presence of other thrusters and shielding by the hull, the absolute value of the force that a given thruster exerts on the centre of mass depends not only on the power f_i but also on the direction θ_i. In the model used at DMI, the effective force in a given direction θ_i is the product of f_i by an efficiency factor, $\eta_i(\theta_i)$.

Given an external force and torque acting on the centre of mass of ship, we would like to produce an opposing force and an opposing torque by means of the thrusters. Let (F_x, F_y) denote the desired force and M_z the desired torque. Our task is then to find f_i and θ_i such that,

$$F_x = \sum_i f_i \, \eta_i(\theta_i) \, \cos\theta_i ,$$

$$F_y = \sum_i f_i \, \eta_i(\theta_i) \, \sin\theta_i, \tag{1}$$
$$M_z = \sum_i f_i \left(-y_i \, \cos\theta_i + x_i \, \sin\theta_i\right).$$

Since the thrusters are only capable of finite power, we have in addition n constraints in the form,

$$f_{\min} \le f_i \le f_{\max}, \quad i = 1, \ldots, n. \tag{2}$$

The lower bound $f_{\min}$ will usually be negative, as the thrusters are capable of reversing.

In this report, we define a ship's holding capability as the collection of 'demanded' forces and torques (F_x, F_y, M_z) for which the preceding system of equations admit a solution. The determination of the holding capability is simplified by the fact that the holding capability is a star-shaped set. That is, if (F_x, F_y, M_z) is a point within the ship's holding capability, then every point on a line segment from the origin $(0, 0, 0)$ to (F_x, F_y, M_z) is also within the ship's holding capability. This is a simple consequence of the right-hand-sides of Eqs. (1) being linear in $(f_1, \ldots, f_n)$. Hence, there is no loss of generality in restricting our attention to values of (F_x, F_y, M_z) that lie on a given straight line passing through the origin.

For each pair of real constants α and β, consider the lines in the space (F_x, F_y, M_z) parameterised by F_x as follows,

$$F_y = \alpha \, F_x, \qquad M_z = \beta \, F_x. \tag{3}$$

(For the special case where $F_x = 0$, consider instead a linear relationship between F_y and M_z). By Eqs. (1), these equations can be written as follows,

$$\sum_i f_i \, \eta_i(\theta_i) \left(\sin\theta_i - \alpha \, \cos\theta_i\right) = 0, \tag{4}$$

$$\sum_i f_i \, \eta_i(\theta_i) \left(-y_i \, \cos\theta_i + x_i \, \sin\theta_i - \beta \, \cos\theta_i\right) = 0. \tag{5}$$

Using the simpler notation,

$$g_i(\theta_i) = \eta_i(\theta_i) \left(\sin\theta_i - \alpha \, \cos\theta_i\right),$$
$$h_i(\theta_i) = \eta_i(\theta_i) \left(-y_i \, \cos\theta_i + x_i \, \sin\theta_i - \beta \, \cos\theta_i\right), \tag{6}$$
$$k_i(\theta_i) = \eta_i(\theta_i) \, \cos\theta_i,$$

we get

$$
\begin{aligned}
F_x &= \sum_i f_i\, k_i\,, \\
F_y - \alpha\, F_x &= \sum_i f_i\, g_i\,, \\
M_z - \beta\, F_x &= \sum_i f_i\, h_i\,.
\end{aligned}
$$

Note that g_i, h_i, and k_i are known functions of θ_i, while the f_i are variables. The full statement of our problem is then: find the global maxima and minima of

$$F_x = \sum_i f_i\, k_i(\theta_i), \tag{7}$$

subject to the constraints

$$
\begin{aligned}
\sum_i f_i\, g_i(\theta_i) &= 0, \\
\sum_i f_i\, h_i(\theta_i) &= 0, \\
f_{\min} \le\ f_i\ \le f_{\max}, &\quad i = 1, \dots, n.
\end{aligned}
\tag{8}
$$

3.2 Local extrema

The optimization of F_x in (7) subject to the constraints in Eqs. (8) is of a classical type and has a well-known method of solution (e.g., [2], [5]). Functions describing the constraints are multiplied by undetermined constants - the Lagrange multipliers - and added to the function to be optimized, F_x. This procedure yields the following Lagrangian function,

$$
\begin{aligned}
L(f_1, \dots, f_n, \theta_1, \dots \theta_n) = F_x &+ \lambda_1\,(F_y - \alpha\, F_x) + \lambda_2\,(M_z - \beta\, F_x) \\
&+ \sum_i \mu_i\,(f_i - f_{\min}) + \sum_i \kappa_i\,(f_{\max} - f_i),
\end{aligned}
$$

where λ_1, λ_2, the μ_i and the κ_i are the undetermined Lagrangian multipliers. The advantage of optimizing the Lagrangian L instead of the original function F_x is that an optimum of the former automatically satisfies the constraints.

The μ_i and the κ_i are constraints on the domain and they are set to zero unless $f_i = f_{\max}$ or $f_i = f_{\min}$, respectively. However, λ_1 and λ_2 are not set to zero at the boundary.

Using the functions k_i, g_i and h_i that were defined above, the Lagrangian may be written as follows,

$$L = \sum_i f_i \left(k_i + \lambda_1 g_i + \lambda_2 h_i + \mu_i - \kappa_i\right) + \sum_i \left(-\mu_i f_{\min} + \kappa_i f_{\max}\right). \tag{9}$$

The first-order condition for having an optimum is that the gradient of L with respect to $(f_1, \ldots, f_n, \theta_1, \ldots, \theta_n)$ vanishes. This condition gives $2n$ equations to satisfy, one for each i,

$$k_i + \lambda_1 g_i + \lambda_2 h_i + \mu_i - \kappa_i = 0, \tag{10}$$

$$k_i' + \lambda_1 g_i' + \lambda_2 h_i' = 0. \tag{11}$$

We now consider the solution in the cases where n, $n-2 \ldots$, and so on, of the thrusters are set at either minimum or maximum power.

Case I: all n thrusters on full power

When all n thrusters are set at either minimum or maximum power, Eq. (10) contains either a μ_i or a κ_i for each $i = 1, \ldots n$. For any given choice of the directions $\{\theta_i\}$, it is always possible to choose this constant such that Eq. (10) holds. Equation (10) then puts no restriction on θ_i, and we need only consider Eq. (11).

To determine whether a given solution $\{\theta_i\}_{i=1,..n}$ of Eq. (11), for each i, is indeed a local extremum for F_x, one can consider higher-order derivatives of the Lagrangian L. Sufficient and necessary conditions for local extrema can be found in [2]; note that we consider only necessary conditions here. If this strategy of solution were to be implemented numerically, we would have to solve, for some choice of λ_1 and λ_2 (see the comment in the summary), the n equations represented by Eq. (11). However, the number of calculations needed to solve these n equations grows faster than n. Since the efficiency factor η_i of the ith thruster is influenced by the relative positions of all other thrusters, the functions k_i, g_i and k_i depend on n. Consequently, in the general case, the number of solutions for $\{\theta_i\}_{i=1,..n}$ in Eq. (11) is of order n^2 rather than n.

However, it is possible to reduce the complexity to one of order n by finding θ_i, in the case of a maximum, as follows,

$$k_i(\theta_i) + \lambda_1 g_i(\theta_i) + \lambda_2 h_i(\theta_i) = \max_{0 \le \theta < 2\pi} \left[k_i(\theta) + \lambda_1 g_i(\theta) + \lambda_2 h_i(\theta)\right]. \tag{12}$$

The algorithms used for finding extrema solve this problem in a time that depends only on the discretisation chosen and not on the number of local maxima in the functions f_i, g_i and h_i. We have thus reduced the original problem from one that grows exponentially with the number n of thrusters to one that grows linearly with the number of thrusters.

Finally, note that λ_1 and λ_2 are undetermined. This shows that the directions $\{\theta_i\}$ yielding extrema of F_x form a two-dimensional surface in the n-dimensional space of all possible directions. The surface is two-dimensional because we have imposed two constraints. These two constraints are represented by the parameters α and β in Eqs. (3).

Case II: exactly n − 1 thrusters on full power

Let us suppose that the jth thruster is not on full power. That is,

$$f_{\min} < f_j < f_{\max}, \tag{13}$$

where it is important that strict inequality holds on both sides. The constraints associated with μ_j and κ_j are now passive, and Eqs. (10) and (11) become, for $i = j$,

$$\begin{aligned} k_j + \lambda_1 g_j + \lambda_2 h_j &= 0, \\ k'_j + \lambda_1 g'_j + \lambda_2 h'_j &= 0. \end{aligned} \tag{14}$$

Once θ_j is given, the k_j, k'_j, g_j ... and so on appearing in Eqs. (14) are determined. We then have two linear equations with two unknowns, λ_1 and λ_2. In general, this system of equations can be inverted and thereby λ_1 and λ_2 are determined uniquely as functions of θ_j. The exception is when the determinant of the system is zero. Then one or both equations are negligible (either one is linearly proportional to the other or both are trivially satisfied; we do not discuss these cases in detail, as they are non-generic, but they represent no difficulty in principle). The solutions for all other values of i are determined as before, from Eq. (11), with the caveat that λ_1 and λ_2 are now the solutions of Eqs. (14). As in Case I, we obtain a two-dimensional collection of solutions, but now parameterised by (θ_j, f_j).

Note that the θ_i with $i \neq j$ are determined indirectly by θ_j through λ_1 and λ_2. Thus, depending on which thruster is not on full power, different solutions appear.

Case III: exactly n − 2 thrusters on full power

In this case, two thrusters are not on full power. If these two thrusters are number j_1 and j_2, with $j_1 \neq j_2$, then an extremum exists only if we can solve Eqs. (14) for both $j = j_1$ and $j = j_2$. If we consider first the case of $j = j_1$, then the possible choices of λ_1 and λ_2 form a closed curve in the plane, parameterised by θ_{j1}. Similarly, if we consider the case $j = j_2$, then we get another closed curve, this time parameterised by θ_{j2}. Since we have to be on both curves at the same time, the admissible values of λ_1 and λ_2 are the points of intersection. There may be any number between zero and infinity of such intersection points. (The latter happens when the efficiency functions η_{j1} and η_{j2} are identical).

Case IV: n − 3 or less thrusters on full power

Now we have to satisfy Eqs. (14) for three distinct values of j. Since we have six equations, but only five variables (the three directions and the two Lagrange multipliers), this is impossible in the generic case. Hence, we obtain a 'Sailors Lemma',

For a ship with n thrusters, where $n \geq 3$, at least $n - 2$ thrusters are needed to achieve maximal force and torque at the same time.

But the practical relevance of this lemma is questionable, to say the least.

3.3 Summary of method

The method proposed in the previous section for solving the holding problem can be summarised as follows:

1. Decide which α and β to use in Eqs. (3).
2. Given the efficiency factors $\eta_i(\theta_i)$, for $i = 1, \ldots n$, construct the functions f_i, g_i and h_i in Eqs. (6).
3. Determine the local extrema as described under cases I–III in the previous section.
4. Compare these local extrema to see which are global minima or maxima. The corresponding directions $\{\theta_i\}$ and power allocations $\{f_i\}$ are then solutions to the holding problem for this particular choice of α and β.

This procedure is repeated for as many values of α and β as desired.

As for the third point, it is important to note the following. If the functions f_i, g_i and h_i are all smooth, then it is sufficient, in case I, to optimize for only one choice of λ_1 and λ_2. Geometrically, the maximum of F_x occurs at a point where the gradient of F_x is orthogonal to the surface on which the two constraints are satisfied. The collection of points at which this conditions is satisfied is a surface parameterised by λ_1 and λ_2. Since the gradient of F_x is orthogonal to this surface, F_x is constant on the surface and the value of F_x at any point on the surface is the maximum. Similar simplifications are possible for cases II and III.

4 A Maple® Program

4.1 Statement of the first problem

We are given a floating oil rig influenced by wind and current. There are two components of the total force on the oil rig, the x component Fx, and the y component Fy, in a rectangular coordinate system xy. This force can change the position of the oil rig. In addition to this there is a moment Mz which can change the direction of the vessel.

To compensate for the changes the force and torque can give we have n thrusters at positions $(x1, y1), \ldots, (xn, yn)$ on the ship. This means we want to satisfy the equations

$$Fx = \sum_{i=1}^{n} fx_i$$
$$Fy = \sum_{i=1}^{n} fy_i$$
$$Mz = \sum_{i=1}^{n} (fx_i \, y_i - fy_i \, x_i)$$

We formulate this as an optimzation problem: Minimize the energy function

$$E = \sum_{i=1}^{n} (fx_i^2 + fy_i^2)$$

subject to the formulas above.

4.2 Solution to the first problem

We start by solving this problem. For the sake of simplicity we let $n = 4$.

First we solve for fx_1 in the first equation above and for fy_1 in the second. Finally we isolate fx_2 from the third equation.

```
> restart;
> fx1:=Fx-sum(fx[i],i=3..4)-fx2;
```

$$fx1 := Fx - fx_3 - fx_4 - fx2$$

```
> fy1:=Fy-sum(fy[i],i=2..4);
```

$$fy1 := Fy - fy_2 - fy_3 - fy_4$$

```
> eq:=Mz-sum(fx[i]*y[i]-fy[i]*x[i],i=3..4)+fy[2]
> *x[2]+fy1*x[1]-fx1*y[1]-fx2*y[2];
```

$$eq := Mz - fx_3\, y_3 + fy_3\, x_3 - fx_4\, y_4 + fy_4\, x_4 + fy_2\, x_2 + (Fy - fy_2 - fy_3 - fy_4)\, x_1 - (Fx - fx_3 - fx_4 - fx2)\, y_1 - fx2\, y_2$$

```
> fx2:=solve(eq,fx2);
```

$$fx2 := -(-Mz + fx_3\, y_3 - fy_3\, x_3 + fx_4\, y_4 - fy_4\, x_4 - fy_2\, x_2 - x_1\, Fy + x_1\, fy_2 + x_1\, fy_3 + x_1\, fy_4 + y_1\, Fx - y_1\, fx_3 - y_1\, fx_4) \Big/ (-y_1 + y_2)$$

Now the energy function E that we wish to minimize is

```
> E:=sum(fx[i]^2+fy[i]^2,i=3..4)
+fx1^2+fx2^2+fy1^2;
```

$$E := {fx_3}^2 + {fy_3}^2 + {fx_4}^2 + {fy_4}^2 + (Fx - fx_3 - fx_4 + (-Mz + fx_3\, y_3 - fy_3\, x_3 + fx_4\, y_4 - fy_4\, x_4 - fy_2\, x_2 - x_1\, Fy + x_1\, fy_2 + x_1\, fy_3 + x_1\, fy_4 + y_1\, Fx - y_1\, fx_3 - y_1\, fx_4) \Big/ (-y_1 + y_2))^2 + (-Mz + fx_3\, y_3 - fy_3\, x_3 + fx_4\, y_4 - fy_4\, x_4 - fy_2\, x_2 - x_1\, Fy + x_1\, fy_2 + x_1\, fy_3 + x_1\, fy_4 + y_1\, Fx - y_1\, fx_3 - y_1\, fx_4)^2 \Big/ (-y_1 + y_2)^2 + (Fy - fy_2 - fy_3 - fy_4)^2$$

To find the minimum we compute the following partial derivatives

```
> eq1:=diff(E,fx[3]):
> eq2:=diff(E,fx[4]):
> eq3:=diff(E,fy[2]):
> eq4:=diff(E,fy[3]):
> eq5:=diff(E,fy[4]):
```

and solve for the remaining variables fx_3, fx_4, fy_2, fy_3, fy_4.

```
> sol2:=solve({eq1,eq2,eq3,eq4,eq5}, {fx[3],fx[4],
fy[2],fy[3],fy[4]}):
```

So for small values of Fx, Fy and Mz we have found the forces the thrusters should give.

Here are the formulas for fx_3, fx_4, fy_3, and fy_4:

```
> fxx3:=subs(sol2,fx[3]);fxx4:=subs(sol2,fx[4]);
```

$$fxx3 := (3\,x_2{}^2\,Fx - x_2\,y_2\,Fy - 2\,x_2\,x_1\,Fx - 2\,x_2\,Fx\,x_3 - x_2\,y_4\,Fy + 3\,x_2\,y_3\,Fy - x_2\,y_1\,Fy - 2\,x_2\,Fx\,x_4 - Fx\,y_2\,y_3 + 3\,Mz\,y_3 - y_1\,Fx\,y_3 + Fx\,x_3{}^2 + x_1{}^2\,Fx - Mz\,y_2 - Mz\,y_4 + Fx\,y_2{}^2 - Mz\,y_1 + y_1{}^2\,Fx - Fx\,y_3\,y_4 + Fx\,y_4{}^2 + Fx\,x_4{}^2) \Big/ (3\,y_2{}^2 - 2\,y_2\,y_4 - 2\,y_1\,y_2 - 2\,y_3\,y_2 - 2\,y_3\,y_1 + 12\,x_2{}^2 + 3\,y_4{}^2 + 4\,x_3{}^2 + 3\,y_1{}^2 + 4\,x_4{}^2 - 2\,y_3\,y_4 - 2\,y_1\,y_4 - 8\,x_2\,x_4 + 3\,y_3{}^2 + 4\,x_1{}^2 - 8\,x_2\,x_3 - 8\,x_1\,x_2)$$

$$fxx4 := (3\,x_2{}^2\,Fx - 2\,x_2\,Fx\,x_3 + 3\,x_2\,y_4\,Fy - x_2\,y_3\,Fy - x_2\,y_2\,Fy - x_2\,y_1\,Fy - 2\,x_2\,x_1\,Fx - 2\,x_2\,Fx\,x_4 - Mz\,y_3 + Fx\,x_3{}^2 + Fx\,y_3{}^2 + x_1{}^2\,Fx - Mz\,y_2 + 3\,Mz\,y_4 + Fx\,y_2{}^2 - Mz\,y_1 + y_1{}^2\,Fx - Fx\,y_2\,y_4 - Fx\,y_3\,y_4 - y_1\,Fx\,y_4 + Fx\,x_4{}^2) \Big/ (3\,y_2{}^2 - 2\,y_2\,y_4 - 2\,y_1\,y_2 - 2\,y_3\,y_2 - 2\,y_3\,y_1 + 12\,x_2{}^2 + 3\,y_4{}^2 + 4\,x_3{}^2 + 3\,y_1{}^2 + 4\,x_4{}^2 - 2\,y_3\,y_4 - 2\,y_1\,y_4 - 8\,x_2\,x_4 + 3\,y_3{}^2 + 4\,x_1{}^2 - 8\,x_2\,x_3 - 8\,x_1\,x_2)$$

```
>  fyy3:=subs(sol2,fy[3]); fyy4:=subs(sol2,fy[4]);
```

$$fyy3 := -(-Fx\,y_2\,x_3 + Fx\,y_2\,x_2 - 4\,Mz\,x_2 - 4\,Fy\,x_2{}^2 + y_3\,Fx\,x_2 - y_3\,Fx\,x_3 + 4\,Mz\,x_3 - y_1\,Fx\,x_3 + 4\,x_2\,x_3\,Fy + y_1\,Fx\,x_2 - y_4\,Fx\,x_3 + y_4\,Fx\,x_2) \Big/ (3\,y_2{}^2 - 2\,y_2\,y_4 - 2\,y_1\,y_2 - 2\,y_3\,y_2 - 2\,y_3\,y_1 + 12\,x_2{}^2 + 3\,y_4{}^2 + 4\,x_3{}^2 + 3\,y_1{}^2 + 4\,x_4{}^2 - 2\,y_3\,y_4 - 2\,y_1\,y_4 - 8\,x_2\,x_4 + 3\,y_3{}^2 + 4\,x_1{}^2 - 8\,x_2\,x_3 - 8\,x_1\,x_2)$$

$$fyy4 := -(-4\,Fy\,x_2{}^2 + y_4\,Fx\,x_2 + y_1\,Fx\,x_2 + y_3\,Fx\,x_2 + Fx\,y_2\,x_2 + 4\,x_2\,Fy\,x_4 - 4\,Mz\,x_2 - y_3\,Fx\,x_4 + 4\,Mz\,x_4 - y_4\,Fx\,x_4 - Fx\,y_2\,x_4 - y_1\,Fx\,x_4) \Big/ (3\,y_2{}^2 - 2\,y_2\,y_4 - 2\,y_1\,y_2 - 2\,y_3\,y_2 - 2\,y_3\,y_1 + 12\,x_2{}^2 + 3\,y_4{}^2 + 4\,x_3{}^2 + 3\,y_1{}^2 + 4\,x_4{}^2 - 2\,y_3\,y_4 - 2\,y_1\,y_4 - 8\,x_2\,x_4 + 3\,y_3{}^2 + 4\,x_1{}^2 - 8\,x_2\,x_3 - 8\,x_1\,x_2)$$

```
>  fxx1:=subs(sol2,fx1):
>  fxx2:=subs(sol2,fx2):
>  fyy1:=subs(sol2,fy1):
>  fyy2:=subs(sol2,fy[2]):
```

We can easily substitute values for the constants by using the MAPLE command subs:

```
>  subs({Mz=4,Fx=13,Fy=5,x[1]=0,x[2]=2,x[3]=2,
x[4]=3,y[1]=1,y[2]=1,y[3]=2,y[4]=2},fyy4);
```

$$\frac{11}{12}$$

4.3 Numerical experiment

Here we perform a numerical experiment by letting Fx, Fy and Mz increase along a line through the origin and then solve for the values of the thruster forces. We have found the solution to the control problem above for small wind speeds. However at some point one of the motors will reach its maximum capability and then we have a different control problem.

We shall impose the natural restriction that each motor has only a limited amount of power according to inequalities

$$-g_i(fx_i) \leq fy_i \leq g_i(fx_i)$$

where the g_i are smooth functions. For the sake of simplicity we set

```
> x[1]:=0:x[2]:=2:x[3]:=2:x[4]:=1:y[1]:=0.2:
y[2]:=1:y[3]:=2:y[4]:=2:
```

The solution to this problem can be found by using the Kuhn Tucker Theorem. This theorem says that if we want to minimize $E(fx, fy)$ subject to the constraints

$$0 = h1(fx, fy) = Fx - \sum_{i=1}^{n} fx_i$$

$$0 = h2(fx, fy) = Fy - \sum_{i=1}^{n} fy_i$$

$$0 = h3(fx, fy) = Mz - \sum_{i=1}^{n} (fx_i\, y_i - fy_i\, x_i)$$

and

$$ki(fx, fy) = gi(fx_4) - fy_4 \leq 0$$

we have to solve the equation

$$\text{grad}\, E(fx, fy) + \sum_{i=1}^{n} \lambda_i \;\text{grad\_}hi(fx,\; fy) + \mu_1 \;\text{grad\_}ki(fx,\; fy) \qquad (*)$$

where λ_1, λ_2, λ_3, and μ_1 are real constants to be determined. This is done by the following MAPLE code.

```
> B:=matrix(8,4,[-1,0,-y[1],0, -1,0,-y[2],0,
-1,0,-y[3],0, -1,0,-y[4],H, 0,-1,x[1],0,
0,-1,x[2],0, 0,-1,x[3],0, 0,-1,x[4],-1]);
```

$$B := \begin{bmatrix} -1 & 0 & -.2 & 0 \\ -1 & 0 & -1 & 0 \\ -1 & 0 & -2 & 0 \\ -1 & 0 & -2 & H \\ 0 & -1 & 0 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & -1 & 1 & -1 \end{bmatrix}$$

In the first three columns of B we have grad h_1, grad h_2, grad h_3 and in the last we have grad k_i. We have let $H = g_i'(fx_4)$.

```
>   with(linalg):
```

```
Warning, new definition for norm

Warning, new definition for trace
```

The Kuhn Tucker equations (∗) are

```
>   eqn:={seq(evalm(B&*(vector([l1,l2,l3,m1])))[i]
>   +2*vector([fx1,fx2,fx[3],fx[4],fy1, fy[2],fy[3],
fy[4]])[i]=0,i=1..8)};
```

$$\begin{aligned}
eqn := \{&-l2 + 2\,l3 + 2\,fy_3 = 0,\ -l2 + 2\,l3 + 2\,fy_2 = 0,\ -l1 - 2\,l3 + 2\,fx_3 = 0,\\
&-l1 - 2\,l3 + H\,m1 + 2\,fx_4 = 0,\\
&-l1 - .2\,l3 + 2.500000000\,Fx + 2.500000000\,fx_3 + 2.500000000\,fx_4 - 2.500000000\,Mz\\
&- 5.000000000\,fy_3 - 2.500000000\,fy_4 - 5.000000000\,fy_2 = 0,\\
&-l2 + 2\,Fy - 2\,fy_2 - 2\,fy_3 - 2\,fy_4 = 0,\ -l2 + l3 - m1 + 2\,fy_4 = 0,\\
&-l1 - l3 + 2.500000000\,Mz - 4.500000000\,fx_3 + 5.000000000\,fy_3 - 4.500000000\,fx_4\\
&+ 2.500000000\,fy_4 + 5.000000000\,fy_2 - .5000000000\,Fx = 0\}
\end{aligned}$$

```
>   fy[4]:=H1;
```

$$fy_4 := H1$$

Later we will let $H1 = g_i(fx_4)$. For the case of one motor on maximum capability the solution is

```
>   sol:=solve(eqn,{l1,l2,l3,m1,fx[3],fx[4],fy[2],
fy[3]});
```

$$\begin{aligned}
sol := \{&\\
&fy_2 = .1428571429\,\frac{-100.\,Mz + 114.\,Fy - 214.\,H1 + 130.\,Fx + 35.\,H\,Fy - 105.\,H\,H1}{106. + 5.\,H},\\
&fy_3 = .1428571429\,\frac{-100.\,Mz + 114.\,Fy - 214.\,H1 + 130.\,Fx + 35.\,H\,Fy - 105.\,H\,H1}{106. + 5.\,H},\\
&fx_4 = -.07142857143(-210.\,Mz - 280.\,Fy + 70.\,H1 - 98.\,Fx - 422.\,H\,Fy\\
&+1313.\,H\,H1 - 75.\,Mz\,H + 80.\,Fx\,H)/(106. + 5.\,H),\\
&l3 = -1.428571429\,\frac{-30.\,Mz - 40.\,Fy + 10.\,H1 + 39.\,Fx + 7.\,H\,Fy - 28.\,H\,H1}{106. + 5.\,H},\\
&l2 = -.2857142857\,\frac{-200.\,Mz - 514.\,Fy + 314.\,H1 + 260.\,Fx + 35.\,H\,Fy - 175.\,H\,H1}{106. + 5.\,H},\\
&fx_3 = .07142857143(210.\,Mz + 280.\,Fy - 70.\,H1 + 98.\,Fx - 206.\,H\,Fy + 699.\,H\,H1\\
&-25.\,Mz\,H + 50.\,Fx\,H)/(106. + 5.\,H),\\
&l1 = .1428571429(-390.\,Mz - 520.\,Fy + 130.\,H1 + 878.\,Fx - 66.\,H\,Fy + 139.\,H\,H1\\
&-25.\,Mz\,H + 50.\,Fx\,H)/(106. + 5.\,H),\\
&m1 = .2857142857\,\frac{-50.\,Mz - 314.\,Fy + 1006.\,H1 + 65.\,Fx}{106. + 5.\,H}\}
\end{aligned}$$

```
> fX3:=subs(sol,fx[3]):
```

In the MAPLE program below we shall need all the motor forces, so we have to assign values to all the fx_i, fy_i.

```
> fY4:=H1;
```

$$fY4 := H1$$

```
> fX1:=subs(sol,fx1):
> fY1:=subs(sol,fy1):
> fX2:=subs(sol,fx2):
> fY2:=subs(sol,fy[2]):
> fY3:=subs(sol,fy[3]):
```

Just to consider an explicit example we shall use the motor constraint function

```
> gi:=x->sqrt(20^2-x^2/2);
```

$$gi := x \to \sqrt{400 - \frac{1}{2}x^2}$$

```
> H1:=gi(fx[4]);
```

$$H1 := \frac{1}{2}\sqrt{1600 - 2 fx_4^{\,2}}$$

```
> H:=diff(H1,fx[4]);
```

$$H := -\frac{fx_4}{\sqrt{1600 - 2 fx_4^{\,2}}}$$

To find the solution to the optimization problem we solve the following equation for fx_4. Once this is solved we can insert in the formulas for the fx_i and fy_i.

```
> sol1:=fx[4]=subs(sol,fx[4]);
```

$$\begin{aligned} sol1 := fx_4 = -.07142857143(&-210.\,Mz - 280.\,Fy + 35.00000000\sqrt{\%1} - 98.\,Fx \\ &+ 422.\frac{fx_4\,Fy}{\sqrt{\%1}} - 656.5000000 fx_4 + 75.\frac{Mz\,fx_4}{\sqrt{\%1}} - 80.\frac{Fx\,fx_4}{\sqrt{\%1}}) \Big/ (106. - 5.\frac{fx_4}{\sqrt{\%1}}) \\ \%1 := 1600 - 2 fx_4^{\,2} \end{aligned}$$

Here we plot the right hand side of the above equation for a particular choice of Fx, Fy and Mz. We display it below. We can solve the equation *sol1* numerically using the MAPLE command fsolve once we assign values to Fx, Fy, and Mz. This is done below.

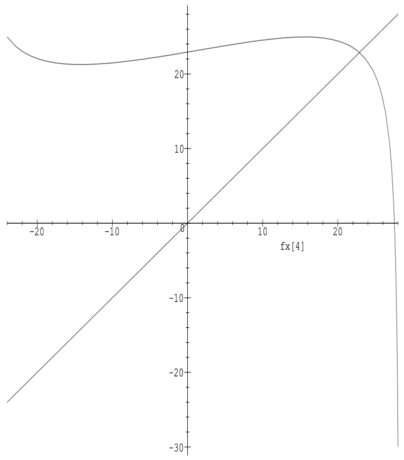
```
> plot1:=plot(rhs(subs({Fx=11*10,Fy=11*5,
Mz=11*4},sol1)),fx[4]=-24..28):
> eq12:=subs({Fx=13*10,Fy=13*5,Mz=13*4},sol1);
fX4:=fsolve(eq12,fx[4],20..28);
```

$$eq12 := fx_4 = -.07142857143 \left(-41860. + 35.00000000\sqrt{1600 - 2fx_4^2} + 20930. \frac{fx_4}{\sqrt{1600 - 2fx_4^2}} - 656.5000000 fx_4 \right) \Bigg/ \left(106. - 5. \frac{fx_4}{\sqrt{1600 - 2fx_4^2}} \right)$$

$$fX4 := 23.85218616$$

We can find the solution to equation *sol1* graphically by plotting the right hand side and left hand side of the equation. The point of intersection gives the solution.

```
>   with(plots):
>   plot2:=plot(y,y=-24..28):
>   display(plot1,plot2);
```



We can see that the plot agrees well with the numerical solution $fx_4 = 23.85$ that we found above using fsolve.

4.4 Output of the MAPLE program

We have chosen to omit the MAPLE program used to perform the numerical experiment. Below we display the output of the program, showing the motor forces found by the solution formulas above as we increase (Fx, Fy, Mz) along the line

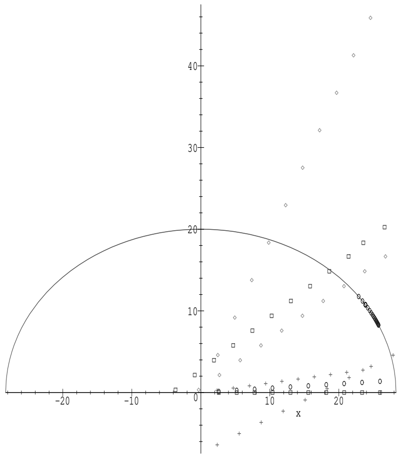
$$t(10, 5, 4), \quad t \in \mathbb{R},$$

through the origin.

```
>   x0:=evalf(sqrt(400*2));
```

$$x0 := 28.28427124$$

```
> plot1:=plot(gi(x),x=-x0..x0):
> display(plot3,plot1,plot2,plot4,plot5);
```



Motor 1: diamond, Motor 2: cross, Motor 3: box, and Motor 4 (reaching the boundary of the constraints): circle.

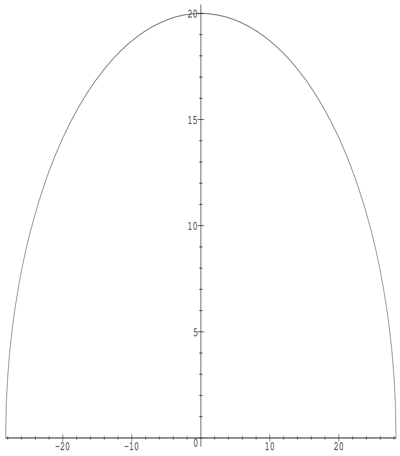
4.5 Maximum capability

In the following we shall determine the maximum capability of the oil rig, that is we shall determine the values of (Fx, Fy, Mz) that we have motor power to compensate for.

We shall use the motor constraint function

$$gi(fx_4) = \sqrt{400 - \frac{1}{2}x^2},$$

and we plot it below.



To find the maximum capability of the oil rig we discretize the values of the motor forces fx_i, fy_i and plug into the formulas for Fx, Fy, Mz. Finally we plot the resulting (Fx, Fy, Mz) values.

```
> L1:=[evalf(seq(seq(seq(seq([i*(28/3),j*(28/4),
k*(28/4),l*(28/4)],i=-3..3),j=-3..3),k=-3..3),
l=-3..3))]:
```

```
> x->(gi(x[1]),gi(x[2]),gi(x[3]),gi(x[4]),x[1],
x[2],x[3],x[4]);
```

$$x \to (\mathrm{gi}(x_1),\ \mathrm{gi}(x_2),\ \mathrm{gi}(x_3),\ \mathrm{gi}(x_4),\ x_1,\ x_2,\ x_3,\ x_4)$$

```
> L2:=map(x->[gi(x[1]),gi(x[2]),gi(x[3]),gi(x[4]),
x[1],x[2],x[3],x[4]],L1):
```

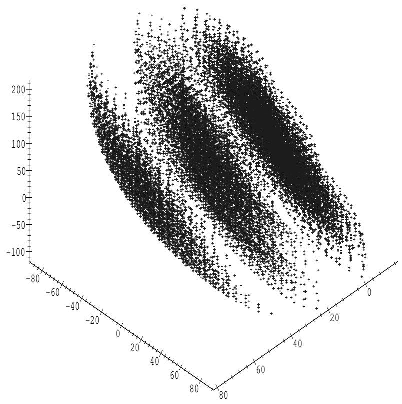
```
> x[1]:=0:x[2]:=2:x[3]:=2:x[4]:=2:
y[1]:=1:y[2]:=1:y[3]:=2:y[4]:=2:
```

```
> L3:=evalf(map(z->[sum(z[i],i=1..4),
sum(z[j],j=5..8),sum(z[k]*x[k]-z[k+4]*y[k],
k=1..4)],L2)):
```

```
> with(plots):
```

The following is a plot of the values we have motor power to compensate for in the (Fx, Fy, Mz) coordinate system. Above we only discretized part of the boundary, just to show the idea involved.

```
> pointplot3d(L3,style=point,symbol=cross,
color=blue,axes=FRAMED);
```



4.6 The case with two motors on maximum capability

When two motors are on maximum capability the Kuhn Tucker equation becomes

$$\text{grad}\, E(fx,\, fy) + \sum_{i=1}^{n} li\ \text{grad\_}hi(fx,\ fy) + \sum_{i=1}^{n} mi\ \text{grad\_}ki(fx,\ fy) = 0,$$

and this is solved below for $n = 4$. We use the notation

$$Hi = \frac{\partial}{\partial fx_i}\,\text{ki}(fx,\ fy)$$
$$Ki = ki(fx,\, fy)$$

```
>  restart;
```

The first column in the matrix B below is grad $h1$, the second is grad $h2$, the third is grad $h3$ and the last four are grad ki.

```
>  B:=matrix(8,7,[-1,0,x[1],0,0,0,H1,
-1,0,x[2],0,0,H2,0, -1,0,x[3],0,H3,0,0,
-1,0,x[4],H4,0,0,0, 0,-1,-y[1],0,0,0,-1,
0,-1,-y[2],0,0,-1,0, 0,-1,-y[3],0,-1,0,0,
0,-1,-y[4],-1,0,0,0]);
```

$$B := \begin{bmatrix} -1 & 0 & x_1 & 0 & 0 & 0 & H1 \\ -1 & 0 & x_2 & 0 & 0 & H2 & 0 \\ -1 & 0 & x_3 & 0 & H3 & 0 & 0 \\ -1 & 0 & x_4 & H4 & 0 & 0 & 0 \\ 0 & -1 & -y_1 & 0 & 0 & 0 & -1 \\ 0 & -1 & -y_2 & 0 & 0 & -1 & 0 \\ 0 & -1 & -y_3 & 0 & -1 & 0 & 0 \\ 0 & -1 & -y_4 & -1 & 0 & 0 & 0 \end{bmatrix}$$

```
>  with(linalg):
```

```
Warning, new definition for norm

Warning, new definition for trace
```

The Kuhn Tucker equations are

```
>  eqn:={seq(evalm(B&*(vector([l1,l2,l3,m1,m2,0,0])
))[i]+2*vector([fx1,fx2,fx[3],fx[4],fy1,fy[2],
fy[3],fy[4]])[i]=0,i=1..8)};
```

$$eqn := \{-l1 + x_4\, l3 + H4\, m1 + 2\, fx_4 = 0,\ -l2 - y_1\, l3 + 2\, fy1 = 0,\ -l2 - y_2\, l3 + 2\, fy_2 = 0,$$
$$-l2 - y_3\, l3 - m2 + 2\, fy_3 = 0,\ -l1 + x_2\, l3 + 2\, fx2 = 0,\ -l2 - y_4\, l3 - m1 + 2\, fy_4 = 0,$$
$$-l1 + x_3\, l3 + H3\, m2 + 2\, fx_3 = 0,\ -l1 + x_1\, l3 + 2\, fx1 = 0\}$$

Below we assign values to some of the motor forces assuming that the motors 3 and 4 are on maximum capability $fy_4 = K4(fx_4)$ and $fy_3 = K3(fx_3)$.

```
> fy[4]:=K4;
```

$$fy_4 := K4$$

```
> fy[3]:=K3;
```

$$fy_3 := K3$$

Recall that we solved for fx_1, fy_1, fx_2 in the first three equations of section 4.1.

```
> fx1:=Fx-sum(fx[i],i=3..4)-fx2;
> fy1:=Fy-sum(fy[i],i=2..4);
```

$$fx1 := Fx - fx_3 - fx_4 - fx2$$

$$fy1 := Fy - fy_2 - K3 - K4$$

```
> eq:=Mz-sum(fx[i]*y[i]-fy[i]*x[i],i=3..4)+
> fy[2]*x[2]+fy1*x[1]-fx1*y[1]-fx2*y[2];
> fx2:=solve(eq,fx2);
```

$$eq := Mz - fx_3\, y_3 + K3\, x_3 - fx_4\, y_4 + K4\, x_4 + fy_2\, x_2 + (Fy - fy_2 - K3 - K4)\, x_1 \\ - (Fx - fx_3 - fx_4 - fx2)\, y_1 - fx2\, y_2$$

$$fx2 := (Mz - fx_3\, y_3 + K3\, x_3 - fx_4\, y_4 + K4\, x_4 + fy_2\, x_2 + x_1\, Fy - x_1\, fy_2 - x_1\, K3 - x_1\, K4 \\ - y_1\, Fx + y_1\, fx_3 + y_1\, fx_4) \Big/ (-y_1 + y_2)$$

```
> Mz:=1:Fx:=2:Fy:=1:x[1]:=0:x[2]:=2:x[3]:=2:
x[4]:=2:y[1]:=1.2:y[2]:=1:y[3]:=2:y[4]:=2:
```

The solution to the Kuhn Tucker equations is

```
> sol:=solve(eqn,{l1,l2,l3,m1,m2,fx[3],fx[4],
fy[2]});
```

$$sol := \{m1 = -2.\, \frac{-248. + 230.\, K3 + 330.\, K4 + 81.\, H3\, K3 - 81.\, H3\, K4}{\%1},$$

$$l2 = -2.(-192. + 170.\, K3 + 170.\, K4 - 90.\, H3 + 189.\, H3\, K3 + 90.\, H3\, K4 - 90.\, H4 \\ + 90.\, H4\, K3 + 189.\, H4\, K4)/(\%1),$$

$$m2 = 2.\, \frac{248. - 330.\, K3 - 230.\, K4 - 81.\, H4\, K4 + 81.\, H4\, K3}{\%1},$$

$$fx_4 = (180.\, H4\, K4 + 81.\, H4\, H3\, K3 + 60. + 150.\, H4\, K3 - 80.\, H3\, K4 + 125.\, H3 - 123.\, H4 \\ - 100.\, K3 - 150.\, H3\, K3 - 100.\, K4 - 81.\, H4\, H3\, K4)/(\%1),$$

$$fx_3 = -1.(150.\, H4\, K4 + 81.\, H4\, H3\, K3 - 60. + 80.\, H4\, K3 - 150.\, H3\, K4 + 123.\, H3 \\ - 125.\, H4 - 180.\, H3\, K3 - 81.\, H4\, H3\, K4 + 100.\, K3 + 100.\, K4)/(\%1),$$

$$fy_2 = -1.(28. - 30.\, K3 - 30.\, K4 - 45.\, H3 + 54.\, H3\, K3 + 45.\, H3\, K4 - 45.\, H4 + 45.\, H4\, K3 \\ +54.\, H4\, K4)/(\%1),$$

$$l3 = 10.(-44. + 40.\, K3 + 40.\, K4 - 9.\, H3 + 27.\, H3\, K3 + 9.\, H3\, K4 - 9.\, H4 + 9.\, H4\, K3 \\ +27.\, H4\, K4)/(\%1),$$

$$l1 = 10.(-76. + 60.\, K3 + 60.\, K4 + 7.\, H3 + 24.\, H3\, K3 + 2.\, H3\, K4 + 7.\, H4 + 2.\, H4\, K3 \\ +24.\, H4\, K4)/(\%1)\}$$

$$\%1 := -100. + 81.\, H3 + 81.\, H4$$

The solution to the optimization problem is then obtained by solving the following two equations

```
>   equa1:=fx[4]=subs(sol,fx[4]);
```

$$equa1 := fx_4 = (180.\,H4\,K4 + 81.\,H4\,H3\,K3 + 60. + 150.\,H4\,K3 - 80.\,H3\,K4 + 125.\,H3 - 123.\,H4 - 100.\,K3 - 150.\,H3\,K3 - 100.\,K4 - 81.\,H4\,H3\,K4)/(-100. + 81.\,H3 + 81.\,H4)$$

```
>   equa2fx[3]=subs(sol,fx[3]);
```

$$equa2 := fx_3 = -1.(150.\,H4\,K4 + 81.\,H4\,H3\,K3 - 60. + 80.\,H4\,K3 - 150.\,H3\,K4 + 123.\,H3 - 125.\,H4 - 180.\,H3\,K3 - 81.\,H4\,H3\,K4 + 100.\,K3 + 100.\,K4)/(-100. + 81.\,H3 + 81.\,H4)$$

for fx_3 and fx_4. So we have to insert the expressions for Hi and Ki. When the two equations above are solved for fx_3, fx_4 by for instance the Newton Raphson method we can substitute the values of the remaining variables from the solution formulas in *sol*.

5 Conclusions

We have shown in this report that the holding problem for a given configuration of thrusters on a ship can be solved in a time that increases at most linearly with the number of thrusters n. This improves on the previous method of solution which, it seems, required a number of calculations that increases exponentially with n.

The holding capability has some peculiar features. Here we point out only one of these. By comparing cases I – III, we see that the solution for $(\theta_1, \ldots, \theta_n)$ changes discontinuously when one thruster goes from being on full power to slightly below full power. Consistent with this observation, the numerical results in Section 4 show a similar discontinuous change in the optimal position of the thrusters (although it is a slightly different Lagrangian which is considered).

We have made no restriction on the force (F_x, F_y) and the torque M_z. Once this general problem has been solved, it is straightforward to solve a special case where the restrictions are present. Such restrictions arise in practice because, for instance, the wind will result in both a force on the centre of mass and a torque around this point. The exact form of the relationship between force and torque depends on the ship considered. However, to see how this special case can be solved in principle, suppose that the torque is a function of the force,

$$M_z = f(F_x, F_y). \tag{15}$$

This equation describes a two-dimensional surface in the three-dimensional space (F_x, F_y, M_z). Recall that the solution to the holding problem in the general case provided us with another two-dimensional surface. The solution to the holding problem in the special case where a relation of the type in Eq. (15), holds is the intersection of the two surfaces. (By the way, it can be argued that the intersection is non-empty. Because, on physical grounds, f is continuous, $f(0,0) = 0$ and $f(x, y)$ must be unbounded, in the general case, in (x, y)).

Bibliography

[1] Chong, E.K.P. & Zak, S.H., *An introduction to optimization*, Wiley-Interscience, 1995.

[2] Courant, R. & Hilbert, D., *Methods of mathematical physics. Vol. I.* Interscience Publishers, Inc., New York, N.Y., 1953.

[3] Larsen, J. C., `http://www.matfys.kvl.dk/biomodel/notes`

[4] Petersen, H.G. Supplerende noter - AM02 IMADA Odense University, Denmark, 1995. (Lecture notes on optimization in Danish).

[5] Whittle, P., *Optimization under constraints. Theory and applications of nonlinear programming.*, Wiley-Interscience, London-New York-Sydney, 1971.

The Group Working on the Problem

Jens Christian Larsen, [jlarsen@dina.kvl.dk]

Henrik Rasmussen, [rasmusse@maths.ox.ac.uk]

Peter Sinding, [psi@danmar.dk]

Ove Skovgaard, [Ove.Skovgaard@mat.dtu.dk]

Allan Sørensen, [allans@mip.ou.dk]

Grundfos: Chlorination of Swimming Pools

Viggo Andreasen, Morten Brøns, Poul Hjorth,
John Hogan, and David Wood

1 Introduction

Grundfos asked for a model, describing the problem of mixing chemicals, being dosed into water systems, to be developed. The application of the model should be dedicated to dosing aqueous solution of chlorine into swimming pools.

The first thing to be decided is the type of model we are looking for; in particular whether this is a diffusion dominated problem (where spatial gradients and diffusion times are important) or a sink-source type of ordinary differential equation problem like the CSTR (Continuously Stirred Tank Reaktor).

Swimming pools as a rule are designed specifically to mix the re-circulated water well, i.e., much faster than the typical diffusion time associated with the pool size. Such a rapid mixing is accomplished by having the chlorinated water re-enter the pool through a large (30–80) number of jets, evenly spaced across the pool bottom. The actual mixing time in a particular public pool is monitored as a part of the routine whenever the pool water is to be drained prior to a filling with fresh water.

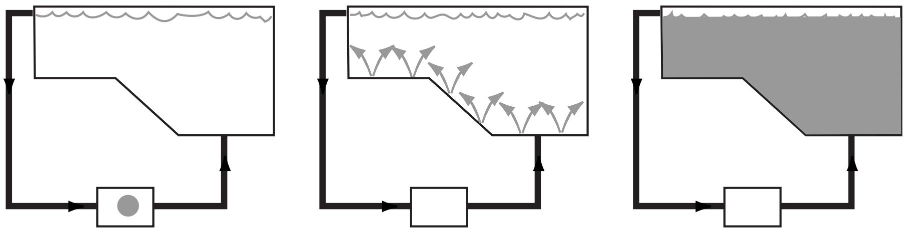


Figure 1: Mixing jets at bottom of pool.

As the final operation on the 'dirty' water, a single blob of dye is injected into the water entry system, and the subsequent spreading of the dye throughout the pool is observed and timed, see figure 1. A well functioning jet system (such as e.g., in the *Lyngby Svømmehal*) mixes the dye evenly in the entire pool in 2–3 minutes; a mixing time of less than 10 minutes is required by law in Denmark. The observed mixing time of 2–3 minutes, is far less than the diffusion time associated with pool size.

The conclusion is that, on the time scale of interest (i.e., hours), setting $\frac{\partial}{\partial x}$, $\frac{\partial}{\partial y}$, and $\frac{\partial}{\partial z} \approx 0$ is a good approximation. The pool indeed resembles the CSTR model, and can for the present purpose be modeled by a system of ordinary differential equations.

2 An ODE Model

We introduce three variables denoting the (approximatly) uniform concentrations of the following quantities:

$c(t)$ = concentration of 'free' chlorine ($ClOH$) [moles/l]
$b(t)$ = concentration of bacteria [10^9 parts/l]
$p(t)$ = concentration of organic pollutants [moles/l]

Three coupled ODE's for the time evolution of these quantities are to be set up, with source and sink terms. We begin with sources and sinks for the chlorine.

2.1 Chlorine sources and sinks

The pump and its reservoir of chlorine is a source term. The pump is connected to a chlorine gauge near the pool overflow channel, and the pump injects well separated doses of chlorine with a variable stroke frequency, i.e., a stroke rate that depends on the chlorine concentration. Figure 2 displays the stroke frequency R as a function of the concentration c.

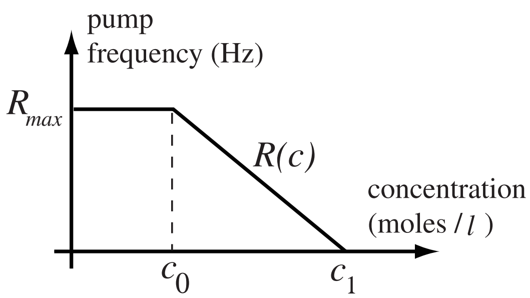


Figure 2: Chlorine injection strategy.

When the chlorine concentration is above a value c_1, the pump is off. Below c_1, the pump frequency increases linearly with decreasing c until at c_0 a maximal pump frequency R_{max} is reached. This feedback strategy will be referred to as *the ramp*, for obvious reasons. The functional form of the source term in the equation is consequently $G/V \cdot R(c)$, where

$$R(c) = \begin{cases} 0 & \text{if } c_1 < c \\ R_{max} \frac{c_1 - c}{c_1 - c_0} & \text{if } c_0 < c < c_1 \\ R_{max} & \text{if } c < c_0 \end{cases}$$

where V is the total volume of the pool, and the constant G denotes the dosage per stroke. We use a value for $G \cdot R_{max}$ of 30 moles/hour.

Next, we discuss the loss mechanisms for the chlorine. The first one of these is the chemical reactions that bind chlorine with the organic polutants p. We assume that this is a first-order chemical reaction,

$$cl + p \rightarrow clp$$

with rate constant k_4, such that the loss term is

$$-k_4 c(t) p(t)$$

The value of rate constant k_4 will be a parameter in the model.

We must also estimate the loss of chlorine due to its reaction with bacteria.

Not much is known at the microbiological level about the details of the process by which bacteria are killed by chlorine. The interaction with one molecule of chlorine is probably not enough to kill one bacteria. With the concentrations involved in the present study, there are 10^{27} chlorine molecules and 10^{10} bacteria. Even if it takes 1000 molecules of chlorine to eliminate one bacteria, the concentration of chlorine will not affected measurably.

Based on this consideration we make the assumption in this report that the chlorine kills the bacteria without actually depleting the chlorine concentration. Hence there will be no bacteria dependent sink term in the chlorine equation.

Alternativly, one could model (but this is not done in the present treatment) the bacteria as amorpheous lumps of molecules, all of which eventually react with the free chlorine to stable compounds. The details of such a reaction, the stoechiometry as well as the time scales for the various reactions, would be appropriate for models more detailed than the one considered here.

Finally, we consider the 'physical' loss of chlorine due to the filters through which the pool water is circulated, due to evaporation, or to migration into pipes and walls. This is taken to be a slow, zeroth order process, with a rate constant which we estimate from observations of the chlorination needed to maintain a constant $c(t)$ when no bathers are present (e.g., over night, or when the pools are closed to bathers for other reasons). The appropriate term has the form:

$$-k_1 c(t)$$

A typical value of k_1 is estimated from actual pool data [5]. If, in order to maintain concentration, an amount corresponding to 1/4 of the total amount of chlorine present needs to be added after a 24 hour period (with no other sink term being present) the rate constant must have a value of about $k_1 = 0.01\mathrm{h}^{-1}$.

2.2 Source and loss terms for p

The quantity p originates with the bathers [1]. In this model, each bather is assumed to stay in the pool for one hour, and in this hour to deposit roughly 5 grams of organic molecules of fairly large molecular weight (100 - 500 g/mole). See [1], p. 285. Let $N(t)$ = # of people in the pool as a function of time (i.e., throughout the day), then the source term is

$$\frac{\alpha}{V} N(t)$$

where $.01 < \alpha < .05$. In this study, we took as a specific $N(t)$ a simple sinusoidal variation throughout the day, and zero at night:

$$N(t) = \begin{cases} 80 \sin^2\left(\frac{\pi}{2} t\right) & \text{if } 0 < [t]_{24} < 12 \\ 0 & \text{otherwise} \end{cases}$$

see figure 3. A loss term for quantity p is the reaction with the free chlorine to form the compound cp:

$$-k_4 cp$$

Such a term also appears in the chlorine equation, and as already stated, the value of k_4 is regarded as a free parameter in the present model.

[1] No, the letter for this variable was not chosen arbitrarily.

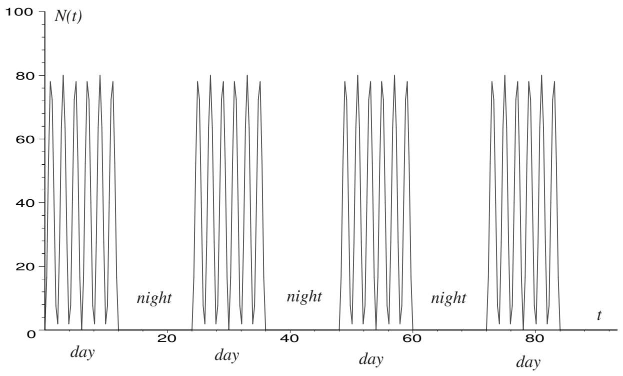


Figure 3: The assumed time dependent number $N(t)$ of bathers.

2.3 Source and loss terms for b

Bacteria are also deposited by the bathers. It is estimated [1] that each bather deposits about 10^8 bacteria per liter per hour. The source term for b is consequently

$$\beta N(t)$$

where $\beta \approx 0.1 \times 10^9/\text{h}$. Furthermore, bacteria are live and will grow by themselves (feeding off salts and and organic substances in the pool) unless kept in check by the chlorine; this process is described in the following source term for b:

$$k_3 b(t)$$

where a worst-case scenario (a two-hour doubling time) would correspond to a value for k_3 on the order of $1/3\text{h}^{-1}$.

In spite of the fact that we did not take bacteria as a loss term for chlorine, we do take chlorine to act as a loss term for bacteria; kinetically as a first-order reaction:

$$-k_2 c(t) b(t)$$

We estimate from data reported [3, 4] that the rate constant has a value of about $k_2 = 8 \times 10^6 l/(\text{mole h})$.

In principle, there is one further loss term associated with the number of bacteria themselves (due to overpopulation and deplection of nourishment, as in 'predator-prey' models). But the numbers of bacteria will (hopefully!) never get so large that this term becomes significant. So we leave it out from the b equation.

3 The ODE system

Collecting all the source and loss terms of the pervious section into a system of Ordinary Differential Equations, we have:

$$\begin{aligned}\frac{dc}{dt} &= \frac{G}{V}R(c) - k_4cp - k_1c\\ \frac{db}{dt} &= \beta N(t) + (k_3 - k_2c)\,b\\ \frac{dp}{dt} &= \frac{\alpha}{V}N(t) - k_4cp\end{aligned} \tag{1}$$

where

$$\begin{aligned} GR_{max} &= 30 \text{ moles/ h} & V &= 8 \times 10^5\ l\\ k_1 &= 0.01\ \text{h}^{-1} & k_2 &= 8 \times 10^6\ \text{l/(mol}\cdot\text{h)}\\ k_3 &= 1/3\ \text{h}^{-1} & \alpha &= 0.01 \to 0.05 \text{ moles/(pers.}\cdot\text{h)}\end{aligned}$$

Note that we have no set value for k_4. This rate constant describes in this model the rate at which chlorine is consumed by reaction with with the p quantity, and vice versa. In a more detailed model, more variables coupling through more chemical reactions might give a more detailed view. Here, with just a single coupling, we examine the qualitative changes in the evolution of the substances c, b, and p over some days, as k_4 is set to values with different orders of magnitude. Specifically, we shall examine the solutions at parameter values $k_4 = 50$, 500, 5 000, and 50 000.

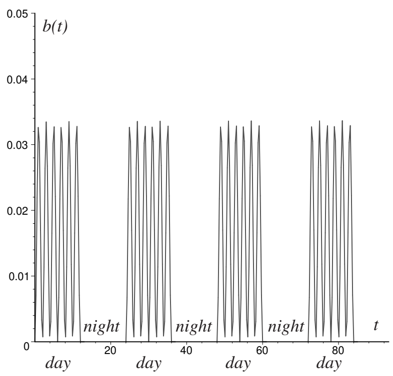


Figure 4: The time evolution of b for all examined values of k_4.

The variable b is found in this model to have an evolution that is nearly independent of such a variation in k_4. The quantity is input (via the bathers), and destroyed rapidly by the chlorine; at the end of each day, nothing remains.

This is not the case for the time evolution of the variables c and p, as illustrated in the figures below.

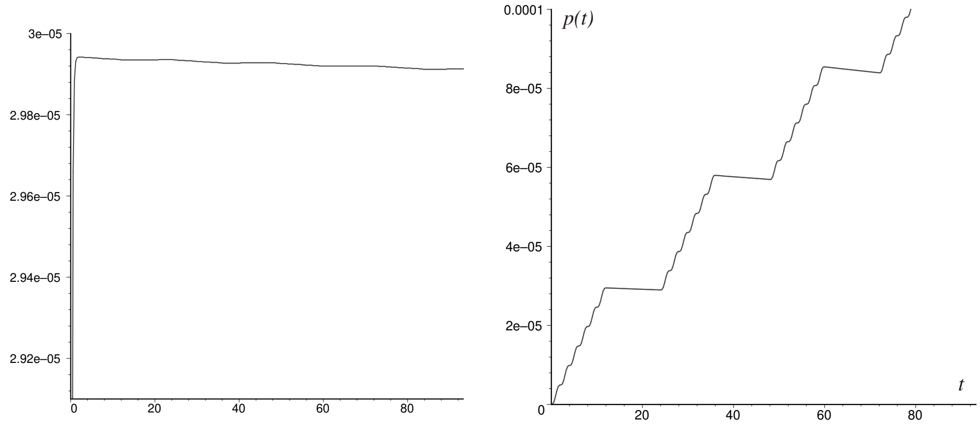


Figure 5: c(t) and p(t) for $k_4 = 50$.

For the lowest examined value of k_4, we find that c increases rapidly until a saturation value of just under 3×10^{-5} moles$/l$ is reached. This is a value where the pump input balances the 'background' loss of c. The p quantity is cumulatively added by the N function, and the $c - p$ reaction rate does not, at least on the time scale of a few days, have any influence on the accumulation of p. (only the slowly decreasing saturation level of c indicates in figure 5 that there is a $c - p$ reaction at all).

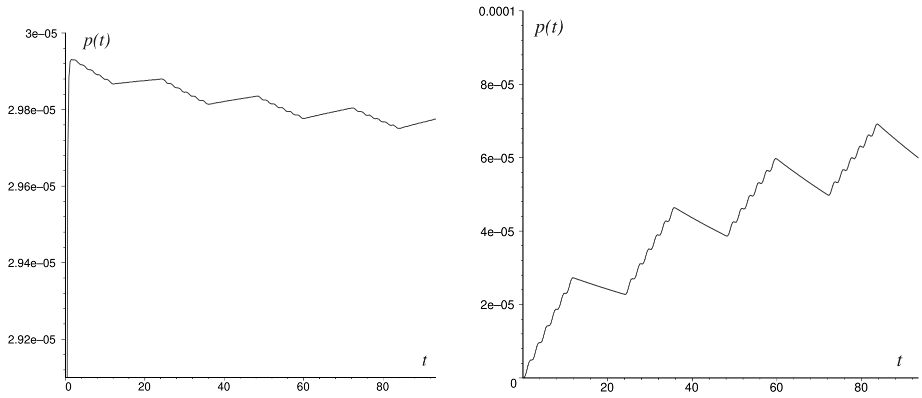


Figure 6: $c(t)$ and $p(t)$ for $k_4 = 500$.

For a ten times higher value of k_4 (see figure 6), the effect of the reaction on the chlorine level begins to be detectable; from its initial saturation level, the daytime loss of c is followed by night time return towards a saturation level. The slow decrease over a time scale of days is now more pronounced. The effect of the higher k_4 value on the p evolution is that at night the p, instead of remaining nearly constant, now is slowly being depleted by the $c - p$ reaction; but the daytime load still 'wins', and over a time scale of days the p accumulates, though possibly with an upper bound.

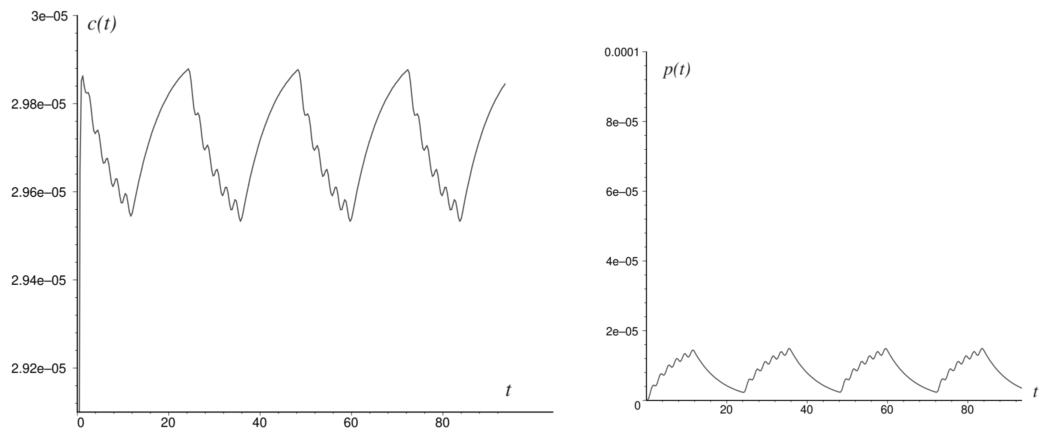


Figure 7: $c(t)$ and $p(t)$ for $k_4 = 5000$.

In figure 7, for $k_4 = 5000$, the $N(t)$ induced oscillations are quite pronounced, and the $c - p$ reaction term at this parameter value influences the solution significantly. The daytime $N(t)$-oscillations can be detected in the decreasing part of the curve. At night, the ramped pump function drives the c level up towards the saturation level, but before a steady state is reached, morning arrives, and more bathers. Nevertheless, the overall c variation on a time scale of several days is at the percentage level, a fluctuation which is acceptable by current standards.

The p evolution curve at this value of k_4 is qualitatively different from the previous two; it no longer shows a long term accumulation. Over the time scale of a single day there is a build-up, but at night the $c - p$ reaction drives the level back down; a constant average level seems to be maintained.

Finally, we can see in figure 8 the effect of a ten times higher value, $k_4 = 50.000$. The daily oscillations now take place around a lower average c value (however: note the scale, the variation is still at the percentage level). The nightly saturations with c run more nearly to completion.

The p level is now very low (we have maintained the same scale as in previous figures, for comparison), and does not show any tendency to accumulate at all. This value of the reaction constant very effectivly keeps the p level in check.

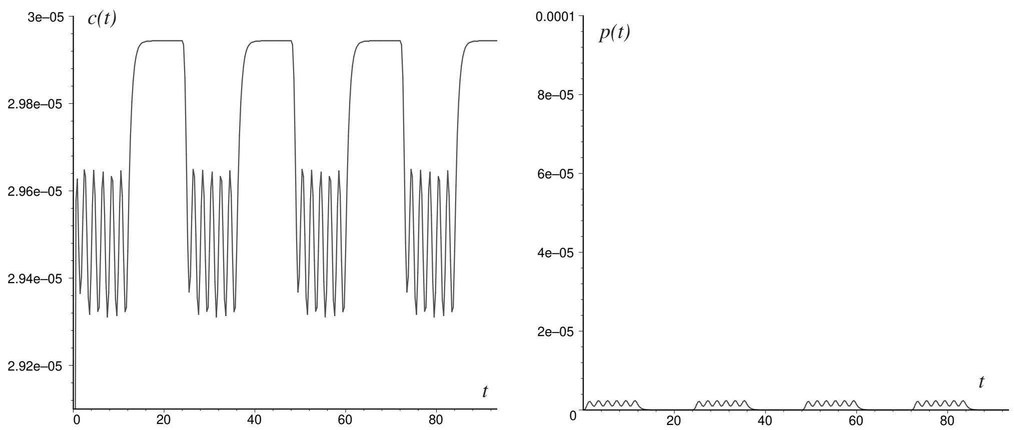


Figure 8: $c(t)$ and $p(t)$ for $k_4 = 50.000$.

4 Discussion

The chemistry in swimming pools is of considerable complexity [1, 6]. The above considerations constitute the simplest model, so to speak, in the chemistry. It contains the minimal number of variables: chlorine, one organic (b) pollutant, one inorganic (p) pollutant, and the very crudest sink, source and interaction terms for these quantities.

In particular the higher order reactions between chloride and ammonia are certainly of interest in a more detailed study, as the end compounds are potentially carcenogenic and hard to remove by filters.

Furthermore, various types of pump strategies (e.g., other response functions than the ramp) might be examined. In our opinion, the qualitative conclusions arrived at here will be invariant for a large class of ramp-like functions, but this of course should be tested.

5 Summary

We have determined that, consistent with the engineering design of pools, fluid mixing is so rapid that the time evolution of chemicals and bacteria in a pool system is well modelled by a set of coupled ordinary differential equations (ODE's).

We have constructed, as a simple model, a 3-variable ODE system, containing the evolution and coupling between one variable representing the chlorine concentration, one variable representing the concentration of bacteria, and one variable representing the concentration of organic molecules.

This model indicates that, with the chlorine as the only means to remove the organic pollutants supplied by the bathers, the rate at which the organic substances accumulate has strong dependence on the chemical reaction constant k_4. At sufficiently high values of k_4, the concentration of organic pollutants does not accumulate, on the time scale of interest.

The explicit pump strategy examined (the 'ramp') is effective in regulating the chlorine concentration.

Bibliography

[1] P. Perkins: *Swimming Pools*, (2nd edition 1978). Applied Science Publishers ltd, London, p. 285.

[2] *Håndbog om Svømmebade* (in Danish). Jydsk Teknologisk Institut 1980.

[3] K. Kristensen: *Svømmebade og sygdomsrisici.* (in Danish) Miljøprojekt nr.80, Miljøstyrelsen,, 1986.

[4] Dansk Svømmebadsteknisk Forening, Årsmøderapport 1992 (in Danish).

[5] A. Back-Pedersen, unpublished communication.

[6] B. Lang: *The Fundamentals of Private Swimming Pool Operation and Chemical Treatment.* http://www.watermaid.dyndns.com/PAGE007.HTM#Chlorination

[7] M. Noak & S. Iacoviello: *The chemistry of Chlorine Dioxide in Industrial and Wastewater Treatment Applications.* http://www.clo2.com/reading/general/indust/

The Group Working on the Problem

Viggo Andreasen, [viggo@fatou.ruc.dk]
Andreas Back-Pedersen, [abackpedersen@grundfos.com]
Morten Brøns, [M.Brons@mat.dtu.dk]
Morten Hagdrup, [mohag@tdk.dk]
Poul Hjorth, [P.G.Hjorth@mat.dtu.dk]
John Hogan, [S.J.Hogan@bristol.ac.uk]
Wolfhard Kliem, [W.Kliem@mat.dtu.dk]
David Wood, [woodd@maths.ox.ac.uk]

LEGO: Automated Model Construction

Rebecca A. H. Gower, Agnes E. Heydtmann, and Henrik G. Petersen

1 The Problem Description

The question from LEGO was, "Given any 3D body, how can it be built from LEGO bricks?"

We will assume that we have a "legoised" model, i.e., a 3D matrix containing ones in the places where unit-volume LEGO bricks should be put and zeros where there should be empty spaces. A LEGO unit-volume is a brick 8 mm long and wide and 3.2 mm high which has only one stud on the top for connection with other bricks.

So the task was to make an algorithm which takes as its input a legoised 3D model and produces a way of constructing the model from actual LEGO bricks so that the model "stands" connected.

We were restricted to the use of so-called "family" LEGO bricks and LEGO DUPLO bricks. Both kinds are rectangular prisms whose dimensions in each direction are integer multiples of the unit LEGO brick's dimensions. The lengths, widths and heights of bricks are measured quite naturally in terms of unit bricks. We were given a table of sizes of all the permitted building blocks. Due to the construction of LEGO DUPLO, these bricks can be connected to family bricks with even length and width only. Another restriction is that only family bricks of height 3 or more can be connected to the top of a DUPLO brick.

As there is more than one way of building any model of even quite small size LEGO suggested that we should seek "more stable" models. More stable models are those which are made with bigger bricks and with bricks that have more studs connected to other bricks.

Nothing can be assumed about the shape of a legoised object given as input except that it is connected. However, some characteristics of typical models and how they are commonly built were discussed.

If a model is of sufficiently large size then its interior should have a hollow space in order to save bricks. LEGO designers keep this space approximately box-shaped for simplicity and so that a supporting structure or motorised mechanism can be placed there if necessary. The thickness of the shell of such a hollow structure is between 2 and 6 unit bricks at its minimum. Faced with the task of building such a model, LEGO builders construct it roughly layer by layer. They start with a small area including part of the visible outer edge, then they work on the inner edge, that of the hollow box which is interior to the model, and finally they fill in the area between the outer and inner walls.

We found out that some "cheating" is even allowed when building a LEGO model. That is to say, one may alter the model by very small amounts if the change produces a model which still "looks good" and the change is beneficial in some other way, for example if it increases the stability.

At a later stage LEGO would like to incorporate the use of different coloured bricks, so our automated building technique should take that into consideration.

The only other requirement was that the algorithm would produce a result within a few days and thus does not need to run for enormous amounts of time to get an answer. LEGO is interested in the solution of the problem for relatively large models as are used in their parks or at exhibitions for advertising purposes. These models are typically made from approximately one million bricks. Of course, the legoised versions of these models would have a far greater number of unit LEGO bricks — the number could be expected to be at least an order of magnitude greater. In these circumstances running time is extremely important.

2 The Simplified Approach

The LEGO problem is a huge and complicated problem so in the first instance it was necessary to simplify it.

Although the use of DUPLO bricks is desirable as building with them is both cheaper and quicker, we restricted ourselves to the use of one height of brick only, the common brick of height equal to 3 minimal LEGO units. In this way we could concentrate on building layers in the model, one at a time, without concern about any brick being in more than one layer at a time. However, any algorithm which can work for bricks of height 3 could be applied without any problems to bricks of other heights if all bricks in any one layer were of uniform height. It would require some work to create an algorithm to accept bricks of mixed heights and that was beyond the scope of the study group work.

Another assumption we made was that the shape of the inner hole was given by the legoisation and we were to keep the size and shape of the legoisation regardless of the price. The reason is that to take into account what kind of "cheating" may be allowable would increase the problem size tremendously.

One layer of a model with an inner hole might look as in Figure 1.

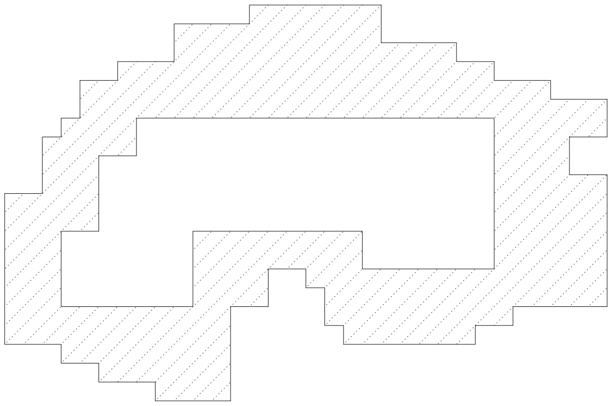


Figure 1: Example of a layer.

3 Formulation as a Combinatorial Optimisation Problem

Despite these considerable simplifications, this problem is still best tackled by a computer and any algorithm which is to be implemented by a computer must have some sort of cost function which the computer can calculate to decide if a brick placement is good or not. The concept of a “good” brick placement is something that the LEGO builders and children who play with LEGO learn from experience. It is possible for computers to mimic the learning of humans through the use of neural networks and this is something that LEGO could investigate [1] although we fear the problem might be too complicated. Instead of attempting to teach the computer a cost function this way, we chose to design one based on heuristics. We shall present this cost function (or penalty function) in Section 4.

Assuming that “the whole is *not* more than the sum of its parts”, a statement that is not quite obvious in this context, we may compute a cost function for the overall structure by adding up cost functions for all the individual bricks. We can now formulate the LEGO problem as a mathematical problem, namely: minimise this overall cost function. This is a so-called combinatorial optimisation problem, that is to minimise a function which is a map from a finite discrete set (the set of all allowable brick configurations) into the real numbers.

There are several advantages to having an overall cost function which is a sum of single-brick cost functions: The single-brick cost functions are easier to understand, and the localised nature of the single-brick cost function makes the optimisation problem very suitable for search methods based on a sequence of small changes.

4 The Penalty Function

When constructing the penalty function we had various issues in mind. The penalty function should be easy to understand conceptually and easy to modify based on experimental results. It should give a high value to structures which are undesirable and a low value to structures which have good strength. The strength of the structure depends on its connectivity so that leads to the question, "What does high connectivity mean in terms of individual bricks?"

We were provided with some examples of ideal walls of thickness 3, 4 and 6. In all cases the side view of the wall is the same and it is shown below in Figure 2.

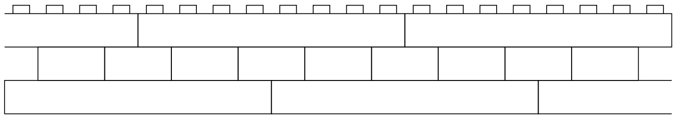


Figure 2: The side view of a wall of good structure.

The top views and end views for the walls of different thicknesses vary but they are similar to each other in many ways too. Compare Figures 3, 4 and 5.

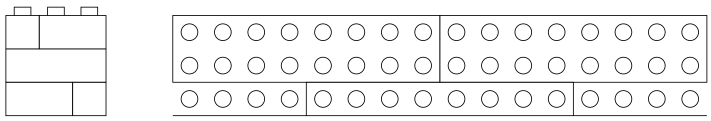


Figure 3: The end view and top view of a wall of width 3.

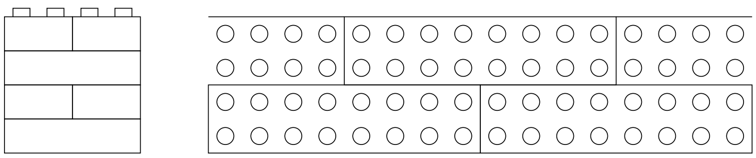


Figure 4: The end view and top view of a wall of width 4.

We were also told that a herring-bone arrangement of bricks is good for filling in thicker areas within a model. When using this structure in consecutive layers, LEGO builders would use the same arrangement of bricks but shifted by the amount of one unit brick in both directions of the plane. This structure is illustrated in Figure 6.

After examining these examples the final decision reached was that there was no single measure of connectivity and structural strength. Therefore we have a

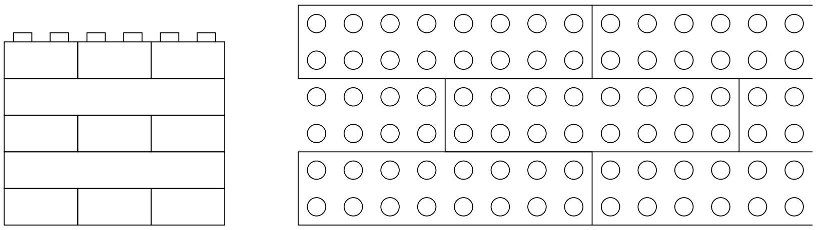


Figure 5: The end view and top view of a wall of width 6.

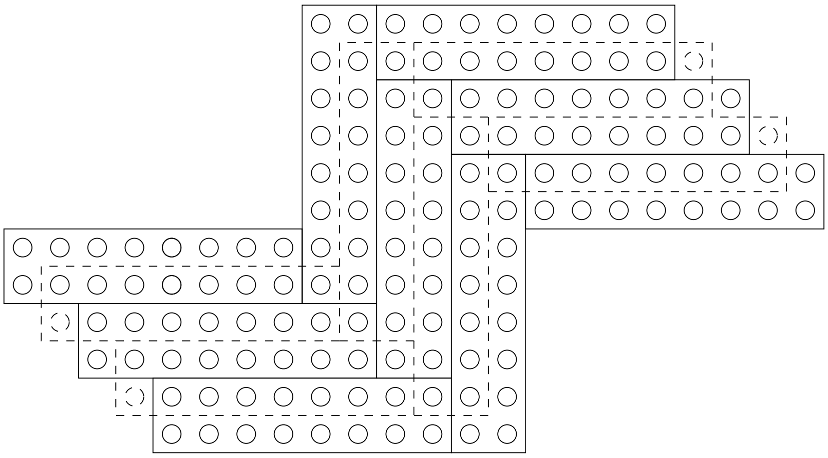


Figure 6: The top view of two layers of a herring-bone pattern. The lower layer is indicated by dashed lines.

penalty function with many terms to represent the many features which contribute to the strength of a LEGO construction.

The important factors relating to the connectivity of bricks are:

1. a high percentage of each brick's area covered, above and below, by other bricks;

2. the use of big bricks;

3. alternating directionality of bricks in consecutive layers;

4. a high percentage of each brick's vertical boundary covered by bricks in the layers above and below;

5. the joining boundary of two bricks in one layer which makes the vertical line in a "T" pattern of boundaries is adjacent to the middle of the neighbouring brick in the same layer which resides above the horizontal line in the "T" — see Figure 7 a);

6. the joining boundary of two bricks in a layer is covered above and below by a brick centred with respect to this boundary — compare with Figure 7 b).

a) b)

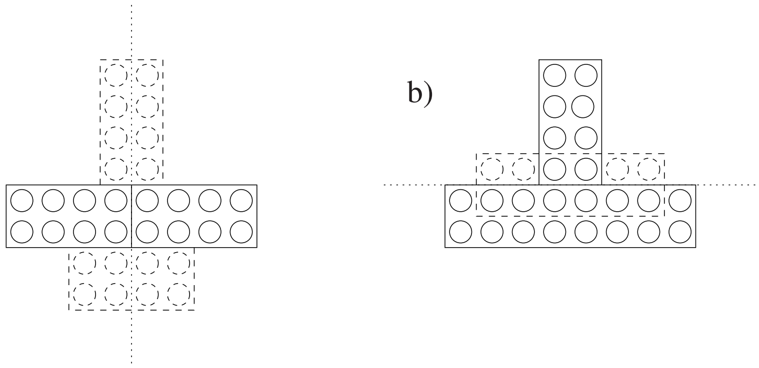


Figure 7: Horizontally a) and vertically b) centred alignment — dashed LEGO bricks indicate possible "good" placements with respect to the boundary indicated by the dotted line.

Although the first factor is important there is not so much one can do about it as that is, to a large extent, determined by the shape of the model which is fixed — at least, we assume it to be so.

In designing our penalty function we assumed that the creation of a model from the legoised information would be done one layer at a time and that once a layer was completed it would be left fixed. Although this may seem restrictive it is the practice of the professional LEGO builders and because there are so many possible ways to turn unit bricks into larger bricks we hope that this will still produce a good solution. In any case, it will help produce a solution in reasonable time.

We recommend the use of a single-brick penalty function such as the following:

$$P = C_1 P_1 + C_2 P_2 + C_3 P_3 + \cdots$$

where the C_i s are weighting constants and the P_i s are as described below.

P_1 relates to perpendicularity: This function is to penalise bricks which do not fit with factor number 3 above. The purpose of this term is to encourage a solution in which bricks on consecutive layers should be perpendicular to each other if they overlap each other when seen from above.

The Penalty P_1: Suppose a brick in layer n is placed so that it is connected to a number of bricks in layer $n-1$. For each brick in layer $n-1$ connecting to this brick in layer n, but not perpendicular to it there is a penalty of 1. (Square bricks are defined to be perpendicular to no other brick.)

The decision regarding square bricks is both natural and necessary. If they were defined to be perpendicular to everything then one way to get zero penalty from this term would be to build the entire model (so far as is possible) from square bricks but this is obviously undesirable.

As this stands it gives quite a harsh penalty to the herring-bone structure when it is used in consecutive layers as shown in Figure 6. There would be a perpendicularity penalty of 2 per brick. An alternative would be to make the penalty equal to the area of overlap with parallel bricks divided by the total area of the brick. This would reduce the penalty from this term a little. However, it could be reduced to as little as 1/8 per brick if the layers were arranged on top of each other differently, namely as in Figure 8.

P_2 relates to vertical boundaries: This function is to penalise the placement of a brick which does not fit with factor number 4 above. Boundaries between bricks on one layer should not be visible from above through the next layer.

The Penalty P_2: There is a penalty of 1 for each unit length of boundary not hidden by a brick from the layer above it. Outer boundaries, that is those on the edge of the model, are not counted.

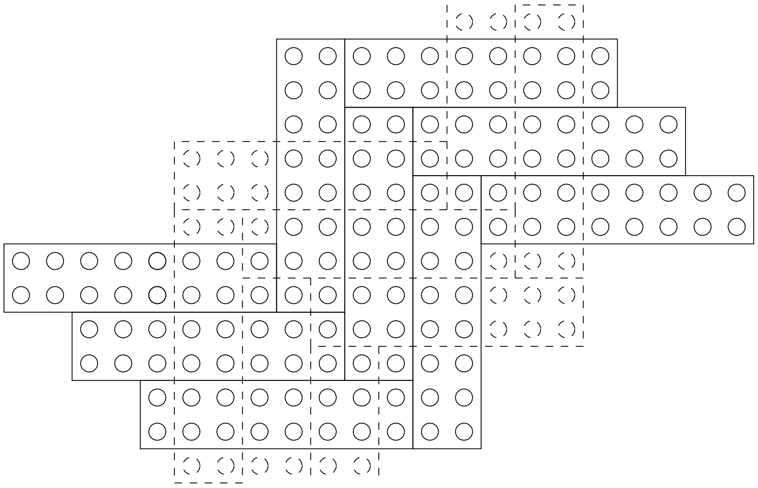


Figure 8: The top view of an alternative construction of two layers of a herring-bone pattern. The lower layer is indicated by dashed lines.

The condition about outer boundaries is necessary if the penalty function is to be used in a local search algorithm of some kind. It would make no difference in comparisons between different ways of building entire layers, or indeed, entire models. However, if this condition is not put in then a local search will try to avoid placing bricks on the edges of the model as it will give a penalty. Obviously this is undesirable as we would ideally like to arrange the bricks of a model from the outside edges inwards as LEGO builders currently do.

P_3 relates to horizontal alignment: This is to penalise bricks placed in a way that goes against factor number 5 listed above. If a brick is placed so that either its short or its long side comes up against a "T" shaped boundary formed by two other bricks in the same layer, then it should be penalised if it is not placed so that the middle of the edge is against the boundary.

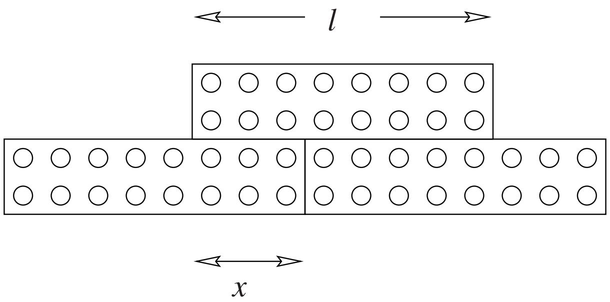


Figure 9: The top view of a situation in which the term P_3 is calculated.

The Penalty P_3: Suppose the length of the edge of the brick which is against the boundary is l and that the boundary is x units away from the end of the brick then the penalty is

$$\frac{|x - l/2|}{l/2}.$$

The division by $l/2$ is to make sure that longer bricks are not penalised by their capacity to have their half-way mark further from the boundary between the other two bricks.

The penalty function with these three terms gives zero values for the ideal walls given to us. So, although this penalty function may treat other wall constructions as being equally good as these, it will at least consider these to be optimal.

For the herring-bone structure (of either Figure 6 or 8) the value from this penalty function is higher than for a structure in which each layer looks like the top view in Figure 5 and in which the directionality of the bricks alternates from layer to layer. If care is taken to avoid a vertical boundary penalty we get a penalty of zero for this brick-wall like structure.

Note that term P_2 tends to favour the use of larger bricks as they have less boundary per unit of area. The smaller the sum of the boundary lengths in any layer the less likely it is to incur a penalty of this type. If LEGO need the penalty function to favour the use of large bricks explicitly a fourth penalty term could be added. Say, P_4 with value $1/A$ where A is the area of the brick.

One way of making sure the use of large bricks in not discouraged, which could be used in a local search, would be to take account of the area covered by each optional arrangement of bricks in any step of the local search. Then the total penalty for the bricks in each option can be divided by their area.

We have thus constructed a penalty function which we believe is a good starting point. It must be expected that (based on experiments) the user will find it necessary to alter the penalty function by modifying the weighting constants or by adding new terms to take into account other considerations such as factors number 2 and number 6. Possible modifications to take account of factor number 2 have been discussed above. Factor number 6 could be incorporated in a term in the style of P_3.

The advantage of our penalty function is that it is relatively easy to determine the necessity of such alterations and they would be relatively easy to make.

5 Using the Penalty Function — Implementation Strategies

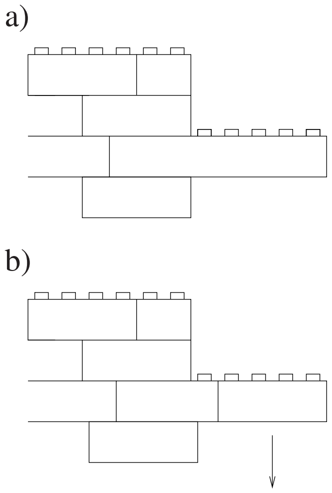


Figure 10: How a protruding part in a model should be protected by a) instead of b).

There are many ways one could make use of the above penalty function. However, in all cases there is a need to be careful at the edges of the model. For example, it is possible that in some part of a model one layer sticks out a little beyond either the layer above or below. In this case it is important that the construction of the protruding part is designed in such a way that it is part of a brick or set of bricks which are firmly attached to the model. (See Figure 10 a).) We want to exclude the possibility that the protruding part is to be made of a brick or bricks with no connection to the model as illustrated in Figure 10 b). Although such a design would incur a penalty with our function it is not forbidden. Therefore it is necessary to fix these "dangerous" edges first.

For a little extra strength and as a precaution against constructions in which bricks are only connected from either the top or the bottom we define the dangerous edges more generally as follows. The dangerous unit-volumes in layer n are those which are not overlapped (when viewed from above) by the intersection of layers $n-1$ and $n+1$. Or mathematically: Project the areas covered by layers $n-1$, n and $n+1$ into one plane and call them A, B and C respectively. Take $B \setminus (A \cap C)$ as the "dangerous", possibly disconnected areas — see Figure 11. These unit volumes should be recognised and allocated to bricks which will hold firmly in place before any other work is done on that layer and should then be considered fixed while the rest of the layer is completed.

During processing of the dangerous edges, one could use the penalty function as it stands with a high weight on vertical boundaries. This should give satisfactory connections with the layer below. Furthermore, to satisfy the wish to connect properly with the layer above also one may include a vertical boundary term relating to the above (as yet nonexistent) layer given as one big imaginary brick having the size and shape of that entire layer.

Overall, we recommend the following procedure.

1. Fix the dangerous edges by a local search or simulated annealing.

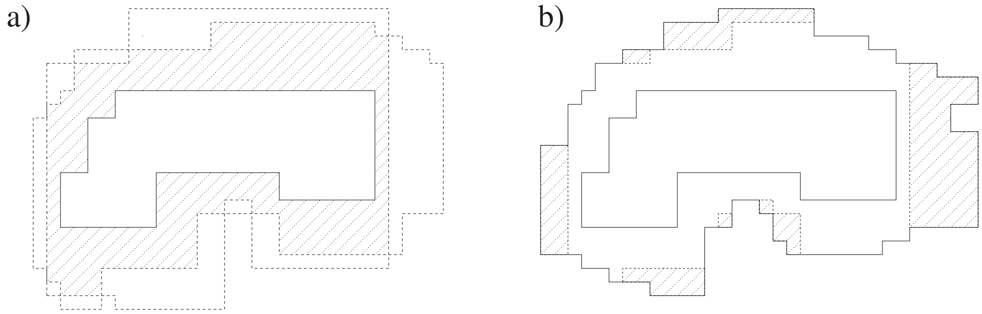


Figure 11: In a) we illustrate the outer boundaries of layer $n-1$ and layer $n+1$ with dashed lines and the shaded area is the intersection of these two layers. In b) the solid line illustrates the outer boundary of layer n and the shaded areas are the "dangerous" edges in the n-th layer. For simplicity of the inside hole, the inner boundaries of all three layers coincide.

2. Divide the remainder of the layer into regions and apply the appropriate mechanism to each region;

 (a) use of a library;

 (b) pattern filling and translating;

 (c) use of local search or simulated annealing.

By the use of a library we mean that there could be a library of methods to build certain types of features in what is known to be a good way. For example, the library could contain information about building walls of various thicknesses. The automated building programme would either have to detect those features or, in an interactive setting, a LEGO employee could mark the area of a model that is to be built by a certain method.

Pattern filling would be suitable for use in layers where there were large areas to be filled which were not suited to one of the library constructions. The filling of such an area could be done by use of something like a herring-bone structure or a brick-wall like structure or some other kind of tiling. If subsequent layers shared similar large regions to be filled with bulk pattern then each layer should be designed so that it fits in a way which is known to be good with the bulk part of the layer below. Usually this will involve use of the same pattern but translated in some way relative to the pattern in the lower layer. This is what is meant by "translating" in 2(b).

The idea behind a local search is to place either one brick or a few bricks at a time by searching through some of the possibilities for a small region. One way in which this could be done is to decide on the next small area which should be covered and then look at different options for doing this. In examining and evaluating the different options one can look ahead a few steps at a time to avoid getting into an undesirable situation later on. Having chosen the best option, according to which gave the lowest penalty, either just one brick or a few bricks could be placed.

The simulated annealing method is probably the most promising method for dealing with the regions described in parts 1 and 2(c) of the procedure. The idea is to divide these regions into medium sized subregions of size $n = 200 - 500$ studs. Each subregion is then treated separately using simulated annealing. The idea for using simulated annealing here is to take some arbitrary initial guess for brick placements on these substructures and then iterate on that by removing some small number of bricks and replacing them by a few new bricks. It is important for the success of using simulated annealing that the *change* in the cost function for each iteration can be computed rapidly.

The simulated annealing method will then lead to near optimal results for each of these subregions. The quality of the result depends on various method parameters used in simulated annealing. A description of the method and how to choose these parameters is beyond the scope of this report, excellent references treating all these problems are [2] and [3].

Apart from the values used for the method parameters, the quality of the result will depend on the size of the subregions. For very small subregions there are few degrees of freedom and a result similar to local search methods must be expected. Increasing the size of the subregions leads to better results. However, the number of necessary iterations goes quadratically with "the problem size", which is linearly related to n. Thus, for an overall structure of M studs treated entirely with simulated annealing, we get an overall computer time for all the M/n subregions of

$$T = \alpha \times (\text{number of iterations}) = \alpha\beta \left(\frac{M}{n}\right) n^2 = \alpha M n.$$

The constant α is the overhead for computing the change in the object function for one iteration. This constant depends strongly on how efficiently the software implementation is. Advanced geometrically based data structures should be used to keep track of which bricks in the layer below may overlap with any given brick, and which bricks share a boundary with a given brick. If the number of necessary (floating point or boolean) operations for an update is 1000 (a rather conservative guess), the value for α will be around $\alpha \simeq 3 \times 10^{-7}$ minutes on a Pentium Pro. The constant β gives the linear relation between the number of iterations and the problem size. This constant depends on the method parameters (that is the required quality). It is difficult to give a guess at the value of β, but for other applications such as Travelling Salesman Problems, the constant β has typical values between 10 and 100. We then get an overall computer time of $T \simeq 7000$ minutes or 5 days when $M = 10^7$, $n = 200$ and $\beta = 10$. The bottom line of this overhead discussion is that the quality of the result depends on the patience of the user and the computer power. There may be nothing to do about the users patience, but with the current rate of increase in computer power there should be a hope for high quality results within a reasonably short time. Furthermore, subregions on the same layer which do not share a boundary may be treated simultaneously. The method is thus well suited for a multiprocessor computer with a possible high speedup gained.

6 Conclusions

Although the problem is far from completely solved we have produced some ideas which can help LEGO begin the next stage of their automatic construction of models. Namely, we have produced a penalty function which can be adapted to their needs and we have discussed implementation strategies for an algorithm to use this function.

Obviously the next stage would be to test the ideas presented here by writing a computer program to implement an algorithm along these lines. Then some experimentation could be performed to determine suitable weighting constants in the penalty function.

It may be that further refinements of the penalty function or implementation strategy are required. For example, it may be necessary to add further terms to the penalty function. Other open questions are, "How should the first layer be built?" and, "Which layer should be chosen as the first?" A natural approach might be to start with the layer which has the biggest solid area.

Finally, when testing and refining has produced a satisfactory result it will be time to generalise this work and produce an algorithm capable of using bricks of different heights. The first step could be to use bricks of different heights in separate layers and when that is successful to move on to bricks of mixed heights crossing layer boundaries.

Another feature which could be incorporated into a more advanced algorithm would be to look beyond the connections with the layers immediately above and below a particular layer. At the moment there is nothing in the algorithm to favour the design illustrated in Figure 2 over an arrangement in which the vertical boundaries in every second layer are aligned. Yet the arrangement illustrated in this paper is preferable.

Acknowledgements

Olga Timcenko for many helpful comments and patiently answering so many questions during the week.

Bibliography

[1] Bishop, C. M., *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.

[2] Aarts, E. H. L. & van Laarhoven P. J. M., *Simulated Annealing: Theory and Applications.* D. Reidel Publishing Company, Dordrecht-Boston, Mass., 1987.

[3] Aarts, E. H. L., *Simulated Annealing and Boltzmann machines.* John Wiley and sons, Chichester, 1989.

The Group Working on the Problem

Anton Antonov, `[anton@unidhp1.uni-c.dk]`

Rebecca A. H. Gower, `[Rebecca.Gower@dfee.gov.uk]`

Agnes E. Heydtmann, `[Agnes.Heydtmann@mat.dtu.dk]`

Thomas Jakobsen, `[T.Jakobsen@mat.dtu.dk]`

Jørn Møller Jensen, `[J.M.Jensen@mat.dtu.dk]`

Ahn Louise Larsen, `[A.L.Larsen@mat.dtu.dk]`

Henrik G. Petersen, `[hgp@mip.ou.dk]`

Carsten Thomassen, `[C.Thomassen@mat.dtu.dk]`

Olga Timcenko, `[olga@digi.lego.com]`

SCANtechnology: Feature Recognition for Automated Sculpture Reproduction

Gretchen Greene, Robert Greene, Steen Markvorsen, Hans Bruun Nielsen, Peter Røgen, Robert Sinclair, and Michael Ungstrup

1 Introduction

This report deals with a problem in the milling process of stone sculpture reproduction. Data for the milling process are supplied through a laser scanning of the original sculpture as measurement of a height function $z = h(x, y)$. 'Scan-points' are situated $0.2-2$mm apart on 'scan-lines' with intermediate distance $0.2-5$mm. The precision of the x-, y-, and z-measurements is within ± 0.1mm. These data define a surface which can be reproduced in stone using a CAD/CAM-system, that calculates tool paths for the milling machine.

The milling process can be described as follows. After an initial rough saw and hammer step to remove most of the excess stone, five or six tools are used in order of decreasing size. A tool is run back and forth across the stone (see Figure 1), at each point penetrating to the depth where it first hits the desired sculpture surface. The smallest tool requires many passes to cover the entire block of marble, accounting for up to 80% of the 10 hours of milling time. Increased efficiency at this stage could significantly reduce total production costs.

One question is whether we can isolate areas where the smallest tool is needed. A second question is whether, within those areas, there is a better path for the tool to follow than the back and forth pattern of horizontal lines. Better may mean more efficient space coverage, but it may also be a question of aesthetics. A sculpture with evenly spaced horizontal tool marks doesn't look good (see Figure 1) and the features may stand out more clearly if the tool marks in some sense paralleled the features and followed the natural contours. To answer these questions, we looked first to identify the small tool regions, and then sought to reduce each region to a curve or curves to get a basic tool path.

What are the small tool regions? A ridge in the sculpture, like the bridge of a nose, can be cut out by fairly large tools. There is plenty of room for maneuvering. Even

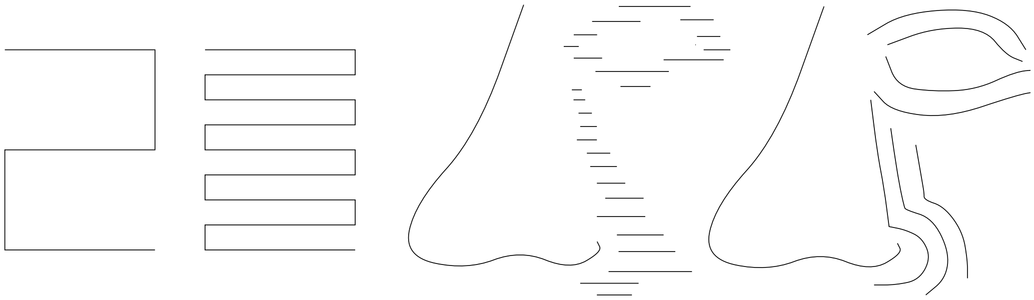


Figure 1: On the left are shown the tool paths currently used for the large and small tools respectively. On the right hand side, a sketch of the result obtained by moving the tool back and forth in horizontal lines, as compared to moving the tool along the features of the surface.

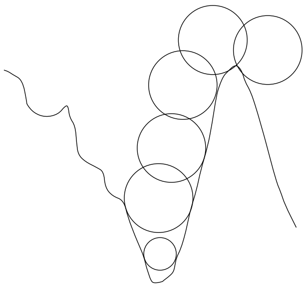


Figure 2: The figure illustrates how only small tools will fit at the bottom of valleys in the surface. The smaller tools all have spherical ends with diameters down to 3mm.

a steeply sloping side is not a problem until we hit the bottom of a valley. It is only here in the valley where a large tool cannot fit (see Figure 2). Since the smallest tools all have spherical ends with diameters down to 3mm, it seemed natural to ask: *How large a sphere would fit on the surface at each point?*. We answer this question in the following section.

2 Maximal Principal Curvatures

We model the sculpture surface as a graph (Monge patch) surface S defined by a height function h over a (usually rectangular) domain Ω in the plane $\mathbb{R}^2$:

$$S = \big\{(x, y, z) \in \Omega \subset \mathbb{R}^2 \;\big|\; z = h(x, y)\big\}.$$

At each point p of the surface S the normal line l orthogonal to the surface at p defines a bundle of planes, the normal planes of S at p, see Figure 3.

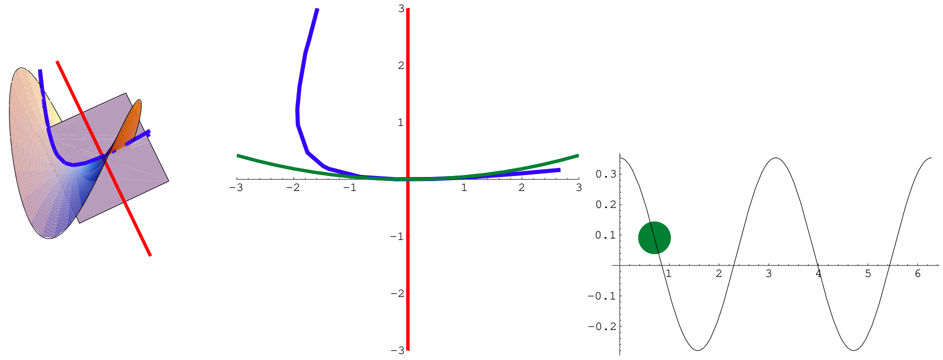


Figure 3: This graph surface (on the left hand side) is defined by $h(x, y) = \frac{1}{2}(x^2 - \frac{1}{2}y^2)$. The surface is cut by a normal plane. The planar intersection curve (in the middle) is approximated by a circle. On the right hand side, the curvature (the signed reciprocal of the radius of the approximating circle) of the intersection curve is graphed versus the turning angle of the normal plane when the normal plane is rotated around the normal line.

The intersection of a given fixed normal plane with the surface contains a (planar) curve through p. This curve has a signed curvature at p: The sign is positive if it curves upward in the direction of the $z-$axis and negative if it curves downward. The numerical value is the reciprocal value of the radius of the unique circle which is the best approximation to the curve at p.

When the plane is rotated around the normal line the corresponding curvature of the planar intersection curve will oscillate between the two so called *principal curvatures* $\lambda_1(p)$ and $\lambda_2(p)$ of the surface at p, see Figure 3.

We are mainly interested in the function

$$\max_P = \max\{\lambda_1(p), \lambda_2(p)\}$$

which measures the largest curvature of the normal intersection curves at p (as "seen by the milling tool" from above). I.e., $1/\max(p)$ is the largest possible radius of the milling drill needed to drill away the stone material right above the surface S at the point p.

The analytical expression for $\max_P(p)$ as a function of the location parameters (x, y) is readily found from the classical formulas for the Gauss curvature K and the mean curvature H of the surface at each point, see e.g. [1]:

$$K = \lambda_1 \lambda_2 = \frac{h_{xx}h_{yy} - {h_{xy}}^2}{\left(1 + {h_x}^2 + {h_y}^2\right)^2}$$

and

$$2H = \lambda_1 + \lambda_2 = \frac{\left(1 + {h_x}^2\right)h_{yy} - 2h_x h_y h_{xy} + \left(1 + {h_y}^2\right)h_{xx}}{\left(1 + {h_x}^2 + {h_y}^2\right)^{\frac{3}{2}}},$$

where h_x, h_{xy} etc. denote partial derivatives of the height function. This gives:

$$\begin{aligned} \max_P(x, y) &= H + \sqrt{H^2 - K} \\ \min_P(x, y) &= H - \sqrt{H^2 - K}. \end{aligned}$$

We are then able to plot the contour levels in the (x, y) plane of the function $\max_P(x, y)$. Thereby we localize the regions where each given drilling head (of a given radius) can be used most effectively, i.e. where large drilling heads can be used to cut away as much material as possible and at the same time get close to the surface and where small drilling heads are needed to trace out the finest details of the sculpture.

In particular we can now point out those regions in the (x, y) - plane where $\max_P(x, y)$ is larger than or equal to the curvature of any given spherical drilling head. The system of these regions - which at places may degenerate to curves or even to points - we will henceforth denote the *ρ-Feature Regions* of the surface. If we let ρ denote the radius of the spherical drilling head (e.g. $\rho \geq 1.5$mm), then for each ρ

$$\mathrm{FR}(\rho) = \left\{ (x, y) \in \Omega \;\middle|\; \max_P(x, y) \geq \frac{1}{\rho} \right\}.$$

If the radius of the second finest drilling head is 3mm - say, then all features with radius of curvature larger than 3mm have been milled out leaving the rest to the finest tool. Hence, FR(3mm) is the set of "finest" Feature Regions, i.e. the regions that have to be drilled lastly with the finest drilling tool available.

To each connected component of a given Feature Region $\mathrm{FR}(\rho)$ we may now associate a center line of that component. Our candidate for this center curve construction is described below. The set of center lines for $\mathrm{FR}(\rho)$ is called the *ρ-Feature Lines* of the surface and is denoted $\mathrm{FL}(\rho)$.

The concrete drilling of the stone with a drilling head of radius ρ should then begin by drilling along the set of curves $\mathrm{FL}(\rho)$ and then work through all of $\mathrm{FR}(\rho)$ by traversing consecutively in each component of $\mathrm{FR}(\rho)$ the neighbouring curves to the respective Feature Lines.

In the case of analytically well defined surfaces we get Feature Regions $\mathrm{FR}(\rho)$ that are clearly visible and recognizable from the original surfaces (for each choice of

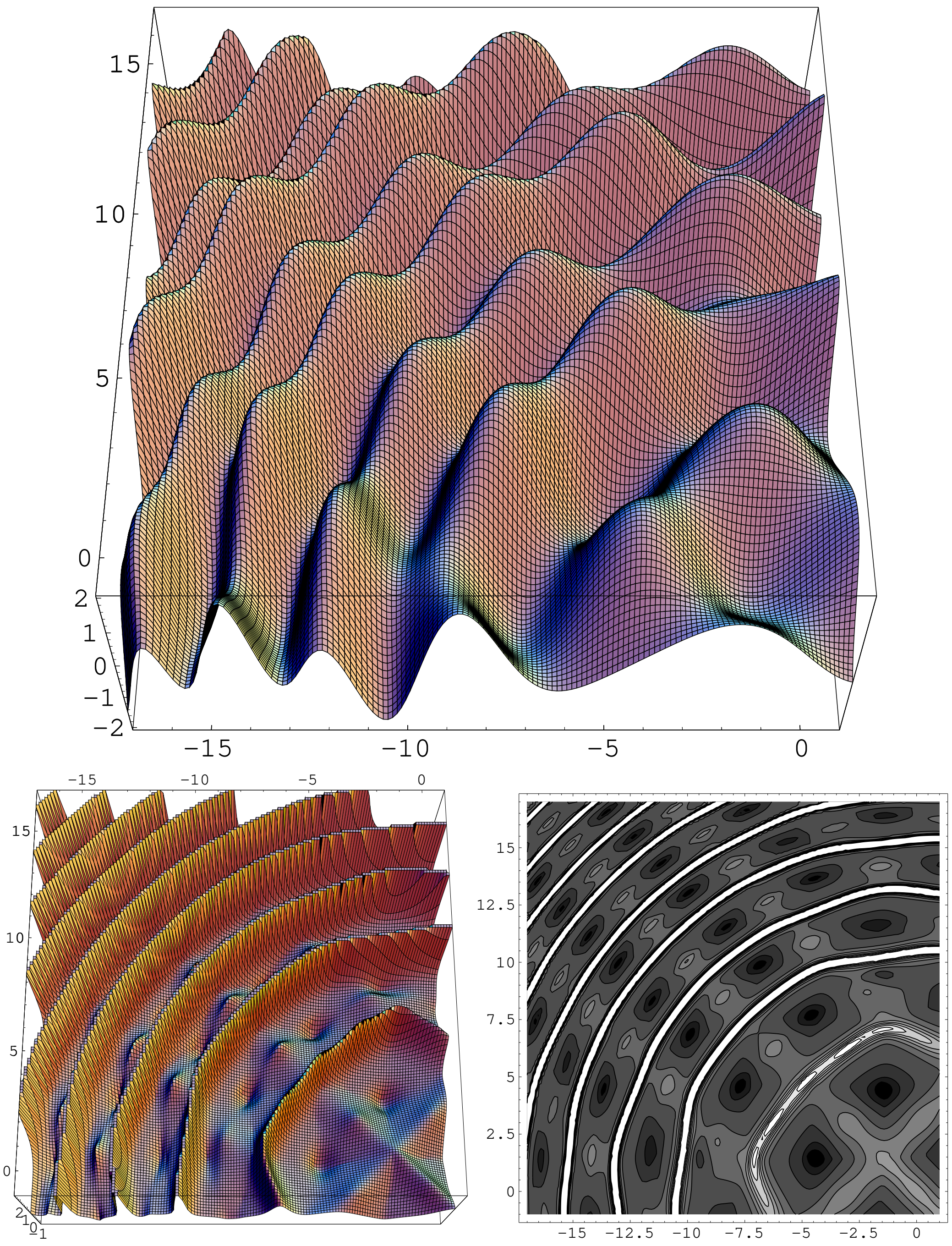


Figure 4: The Mountain Fold surface, shown on the top, is given by the height function: $h(x, y) = \sin(0.1(x^2 + y^2)) + \sin(x)\sin(y)$. At the bottom, the maximal principal curvature is shown as a height function over the domain of the surface to the left and to the right, as a contour plot, where white corresponds to high maximal principal curvature.

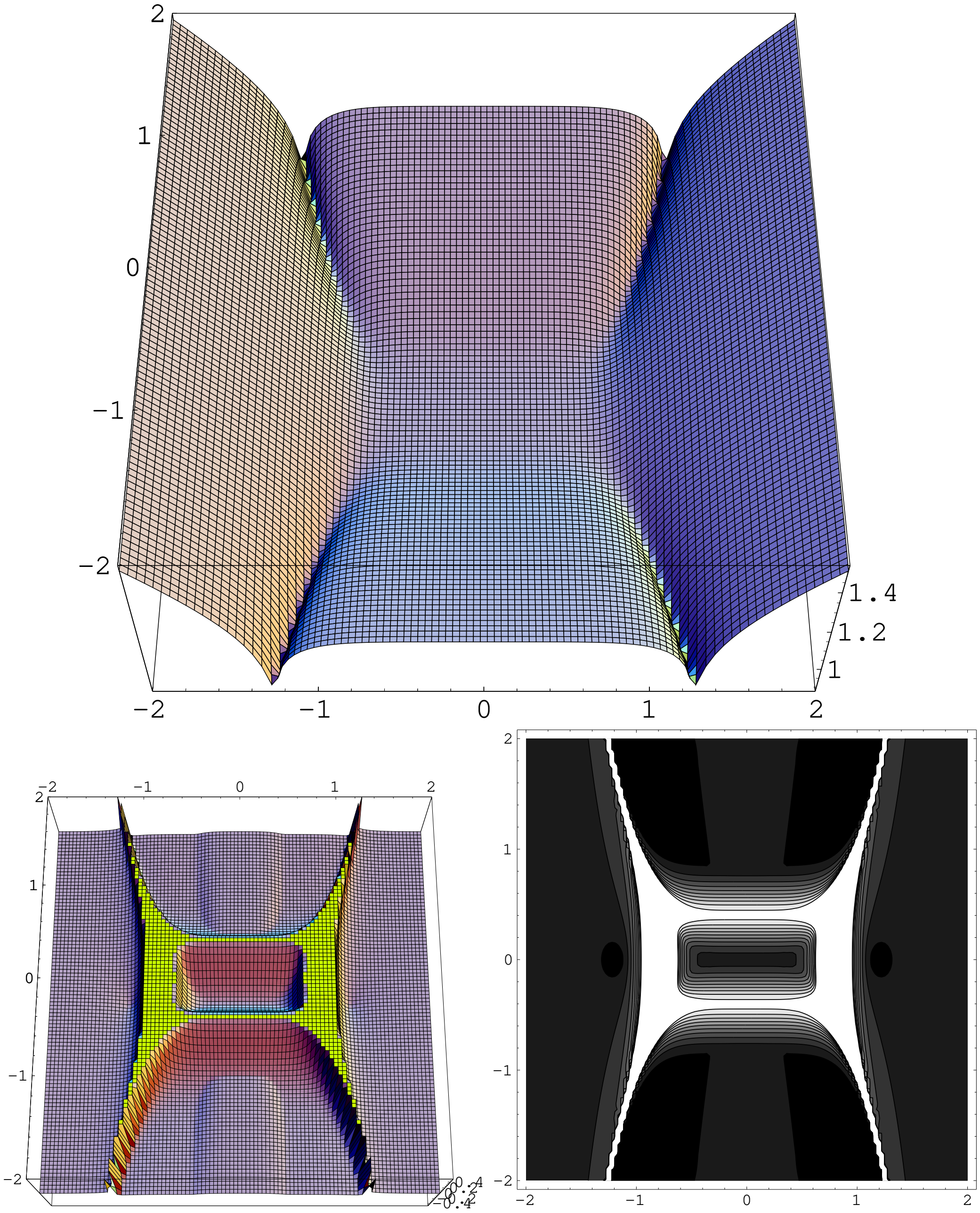


Figure 5: The Batman Face surface, shown on the top, is given by the height function: $h(x, y) = ((x^6 - y^2)^2 + 0.1)^{1/20}$. At the bottom, the maximal principal curvature is shown as a truncated height function over the domain of the surfaces to the left and to the right, as a contour plot, where white corresponds to high maximal principal curvature.

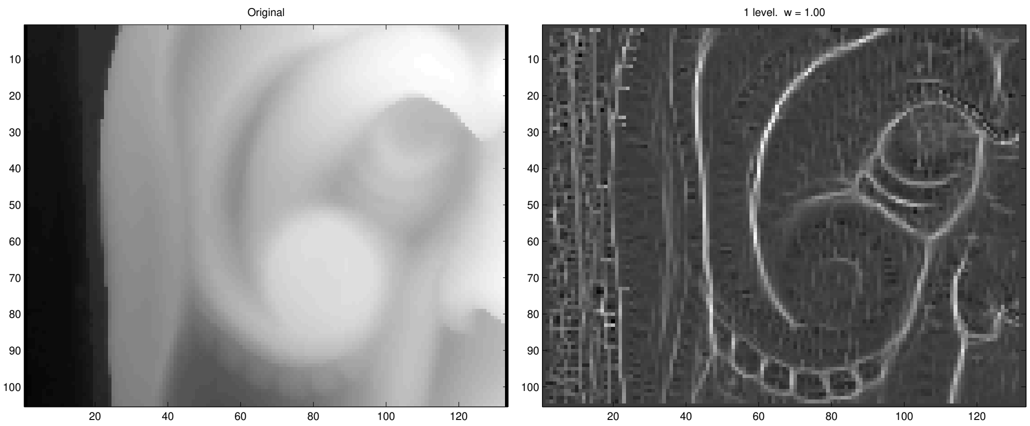


Figure 6: On the left hand side, the height of a piece of a sculpture, is shown in gray-scale over the lattice. The highest regions are white and the lowest regions are black. On the right hand side, the maximal principal curvature of the sculpture is shown in gray-scale over the lattice. High maximal principal curvature is white and the low maximal principal curvature is black.

cut off value ρ). See the two examples: Mountain Fold on Figure 4 and Batman Face on Figure 5. For each of these examples we show the surface, the graph of the function $\max_P(x, y)$ and the contour lines of the function $\max_P(x, y)$. The various Feature Regions are then the regions in the contour plot which have colours lighter than a given value.

In later sections we will therefore apply the same basic strategy to develop numerically robust Feature Lines from the discrete height function scan data. In this process we will concentrate on localizing the Feature Lines corresponding to the finest tools, since the geometry of these curves is naturally the most complicated to extract from the raw data.

3 Calculation of Maximal Principal Curvature on Sculptures

This section describes how to calculate $\max_P(x, y)$ given only a discrete height function on a regular lattice. The graph of such a function is shown in Figure 6 on the left.

As a first step the raw data is smoothed by replacing the height at each interior point by the average of the height at its eight neighbouring points, with weight one, and the height at the point itself, with weight four. This is done to reduce noise coming from the finite precision in measurements (cf. Section 1).

To calculate the principal curvatures the first and second partial derivatives of the height function are needed (see Section 2). To get these partial derivatives we associate to each interior lattice point an "osculating" second degree polynomial. A local approximation by a second degree polynomial, involves 6 parameters. These parameters are found by weighted least squares fit to the heights in the lattice point and its 8 immediate neighbours or an extended set including also the next 16 neighbours. Experiments indicate that the choice of weights and the number of included neighbours has very little influence." The second degree polynomial defines the partial derivatives needed to evaluate the formula for the maximal principal curvature given in Section 2. Finally, the resulting maximal principal curvature lattice function is smoothed as above. The result is shown in Figure 6 on the right. Note, how the white regions on the left hand side of this figure capture all the valley-features in the picture on the right. To find the ridge-features of the picture one has to calculate the minimal principal curvature.

To check the robustness of the calculation of the maximal principal curvature of the sculpture variation of the chosen weights, inclusion of the next nearest neighbours, and repeated or omitted smoothing have been tried. These alternatives give only slight changes of the calculated maximal principal curvatures. Hence, this calculation must be characterized as robust.

The “finest tool set” FR(3mm) (see Section 2), i.e., the set where the second finest tool can not get down in the clefts is shown on the left hand side of Figure 8.

4 Finding Center Lines

The idea that a region in the plane that resembles a strip has a well-defined center line is a basic geometric intuition. When we drive on a two-lane road, we accept without hesitation that it is a plausible thing for there to be a line down the middle of the road, dividing the road into two halves. But making this sort of idea precise and computing effectively exactly where the center line is turns out to be a surprisingly subtle matter. And a considerable literature has grown up around it. See e.g. the references in [2].

For regions bounded by two curves which are in some sense approximately parallel, for example, a road, it is natural to define the center line to be the locus of points that are equidistant from the two boundary curves (and in the region). In the case of a strip bounded by two parallel lines, this gives the obvious intuitive answer for the center line of the strip. And it produces visually plausible answers in more general but similar cases.

A point on the center line in this sense has the property that there are two distinct shortest connections to the boundary, one to one of the boundary curves and one to the other. This suggests that one might define the center curve of a general region to be the (closure of) the set of interior points for which there are at least two shortest connections to the boundary of the region. Alternatively in the case of smooth boundary one might consider the locus of points with at least two equilength perpendiculars to the boundary (whether or not these were shortest connections). Of course the set of points with at least two shortest connections to the boundary need not be a curve in the conventional sense. For a circle, it would be the center of the circle alone, for example. And for other regions the center line thus defined may have bifurcations. But even so the definition agrees well with visual intuition. And at least the set defined has no interior so that it is plausibly a candidate for being a curve in some sense.

5 An Approximate Method for Finding Center Lines

In the present project, we are concerned primarily with strip-like regions that are bounded by two curves that are close together compared to their extent, except at the "capped off" ends of the regions. In effect we want to look at roads with an end and a beginning. For such regions, the locus of point equidistant from the "sides" of the strip is an appropriate idea of the center line. The ends being a small part of the region can be largely ignored for our purposes.

The description of the center line points as those equidistant from the boundary curves is conceptually simple but is not particularly efficient computationally when the boundary curves are given in terms of a fairly dense but discrete set of points. The difficulty arises from the fact that in effect one would need to check equidistance for every interior point of some fine lattice and for each such point the check would involve finding the distance to all the boundary points or at least to all those that were reasonably near-by. Too many variables to be varied!!

5.1 A step by step algorithm

A more efficient, if somewhat approximate, method can be based on the following observations: First, for a parallel sided straightline boundaries strip (as shown on Figure 7), a center line point can be found by choosing an arbitrary point ("initial point" on the figure) of the interior, drawing a line (dotted on the figure) through it (not parallel to the boundary), and taking the midpoint of the intersection of that

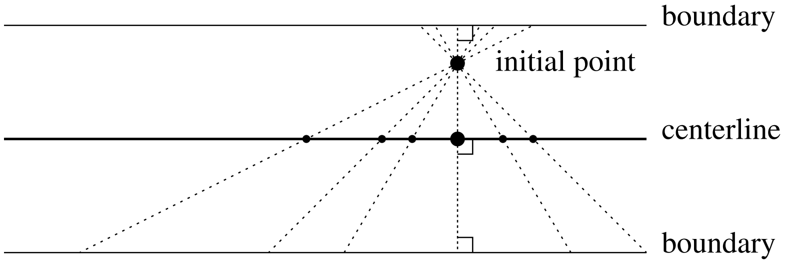


Figure 7: A parallel sided straightline boundaries strip with center line.

line with the strip. Secondly if the line through the interior point has the shortest possible intersection with the strip (among all lines through the given point), then the center line is perpendicular to that line. This is the tight-dotted situation on Figure 7.

For more general strip-like regions, these observations no longer apply precisely. But they remain good approximations. Moreover, it is plausible, in the second part, to check not all lines through the given point but some discrete, finite set of them and to take the shortest among those.

Following this general line of thought one can arrive at the following algorithm:

Step 0: Choose a point of the interior.

Step 1: For line vertical, horizontal, and at 45 degrees above and below the horizontal, find the endpoints of the intersection of the line with the region (or, more precisely, the endpoints of the connected component of the intersection that contains the chosen point).

Step 2: Choose the shortest of the intersections segments of Step 1 and find its midpoint.

Step 3: Move perpendicularly to the chosen intersection segment from its midpoint by some prechosen distance e. (The appropriate choice of e depends on the geometric smoothness of the boundary. Irregular boundaries would suggest the use of small e while smoother more nearly straight boundaries would allow larger choices of e).

Step 4: Return to Step 1 with the point found in Step 3 and proceed as before (the step 3 point replaces the initializing step 0 point).

The step 2 midpoints are the points desired as the center line. The Step 3 points are "helper " points, not necessarily center line point candidates, although for small e values they will be near-by.

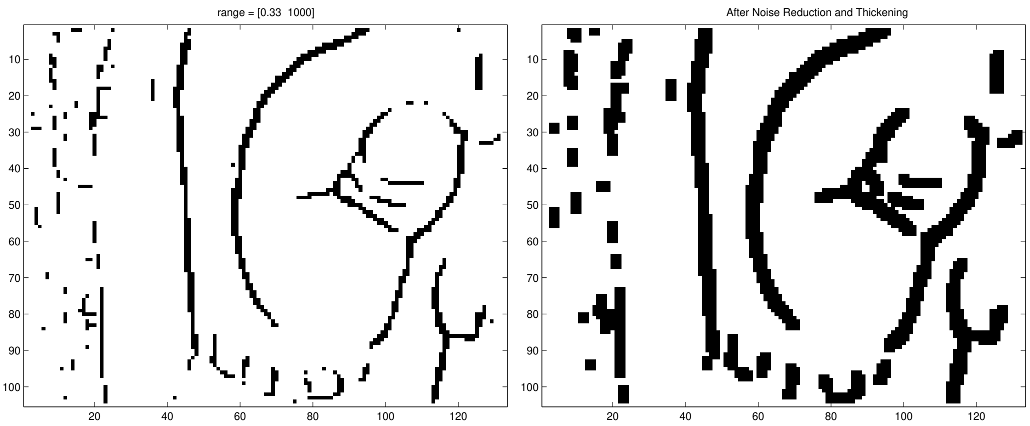


Figure 8: Left: The finest tool set. Right: A noise reduced and thickened version.

This algorithm is an approximate one. The center line candidate points of Step 2 are not necessarily actually center line points in the equidistant sense defined earlier. And the effectiveness of the approximation is not easy to analyse in generality. But experimentation (in Section 5.2) in numerous actual instances gave approximate center lines that were in fact satisfactorily close to the precise center line.

5.2 Implementation

The actual implementation assumes that the data are available on a regular square grid. Further, it assumes that the curvature data have been divided into those falling within a given interval, and those not, as done in Section 3 and shown on Figure 8 to the left hand side. The actual input is therefore a matrix of ones (which will be called points) and zeros. The algorithm has the following steps:

Noise reduction: If a point is isolated, has only one nearest neighbour, or no more than two next-to-nearest center(diagonal) neighbours, it is deleted.

Thickening: The nearest and next-to-nearest neighbours of existing points are also set to one. See Figure 8 to the right.

Identification of regions: The set of all points is separated into maximal disjoint connected (via nearest and next-to-nearest neighbours) components.

Thinning: Those regions with less than 100 points are deleted. See Figure 9 to the left.

For each region and initial directions up and down:

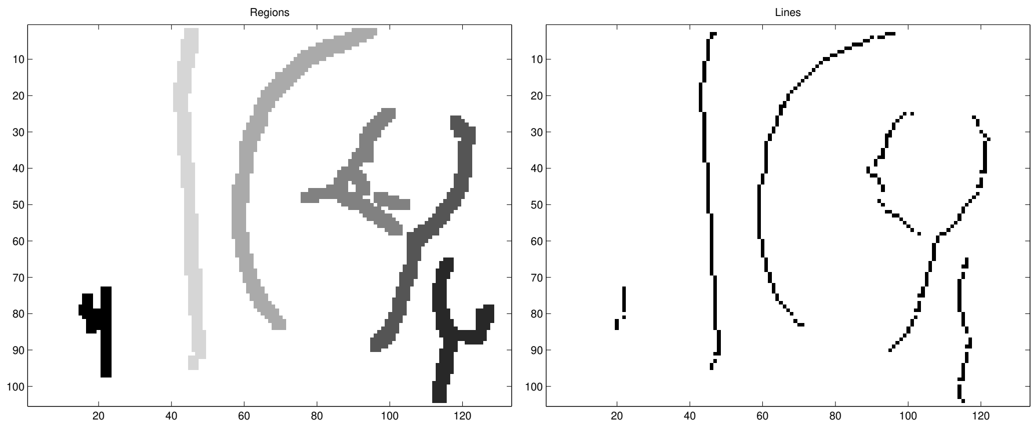


Figure 9: Left: The identified regions. Right: The first iteration of center lines.

- Find a starting point within the region.
- Draw vertical, horizontal and diagonal lines through it.
- Intersect each line with the region.
- Add the midpoint of the shortest line segment to the approximation of the line.
- Set the new current direction to be the normal to the shortest line segment whose scalar product with the current direction is positive (stop if none exists).
- Find the nearest point to the midpoint of the shortest line segment in the direction of the new current direction, and repeat using this point as a new starting point.

The resulting center lines are shown on Figure 9 to the right. There are now two problems left:

(1) To take care of the fact that the feature regions may not be simply connected, simply check for each new center line point whether it is within distance $2e$, say, of any of the points of the center line which are more than five iterations away in the past.

(2) To locate and drill bifurcating branches of the feature region tree, we do as follows: Recall the two pictures on Figure 9. For each constructed center line we remove from the feature region the collection of all points on the shortest intersection lines (constructed in Step 2 of Section 5.1) together with their neighbours within distance $1.5e$, say. Then we are left with a reduced feature region on which

we then run the algorithm again. This process clearly terminates and gives a complete centerline description for the corresponding feature region.

6 Drilling Strategy

On the basis of the center line finding algorithm, in particular from Step 2 of Section 5.1, we can easily parametrize the parallel drilling paths that the drilling tool has to traverse in order to cover the full feature regions: Let $2R$ be the length of the longest line segments found in Step 2 in Section 5.1, let $c(i)$, $i \in [0, N]$ be the center curve, and let $d(i)$, $i \in [0, N]$ be a choice of the direction of the shortest line through $c(i)$ for each $i \in [0, N]$. Define $P : [0, N] \times [-R, R] \to \Omega$ by $P(i, j) = c(i) + jd(i)$. Then the desired tool-path is

$$
\begin{array}{ccccccccc}
P(0,R) & \rightarrow & P(1,R) & \rightarrow & & \cdots & & \rightarrow & P(N,R) \\
 & & & & & & & & \downarrow \\
P(0,R-1) & \leftarrow & \cdots & & \leftarrow & P(N-1,R-1) & & \leftarrow & P(N,R-1) \\
\downarrow & & & & & & & & \\
P(0,R-2) & \rightarrow & \cdots & & & & & & \\
 & & & & & \cdots & & \rightarrow & P(N,-R).
\end{array}
$$

In order to get a reasonable tool-path it is probably necessary to check more than four lines in the plane as suggested in Step 1 in Section 5.1.

7 Conclusion

(1) For any given set of discrete scan data corresponding to a sculpture height function we have defined the function $\max_P(x, y)$, the maximal principal curvature function.

(2) For each given spherical milling tool we have, via the function $\max_P(x, y)$, defined the corresponding Feature Regions, which - to the depth and detail of the given tool - capture all visual features of the sculpture.

(3) For each given set of Feature Regions we have developed a Feature Line finding algorithm which will produce the initial track for the corresponding spherical milling tool.

(4) The Feature Line algorithm furthermore provides a full 2-dimensional parametrization of the corresponding Feature Regions. Hence it gives a strategy for

the complete milling process to the depth and detail defined by any spherical milling tool.

Bibliography

[1] M. P. do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice Hall, 1976.

[2] Peter J. Giblin and Guillermo Sapiro, Affine-invariant distances, Envelopes and Symmetry sets, *Geom. Dedicata* 71:237–261, 1998.

The Group Working on the Problem

Gretchen Greene, [ggreene@math.ucla.edu]
Robert Greene, [greene@math.ucla.edu]
Steen Markvorsen, [S.Markvorsen@mat.dtu.dk]
Claus Dybdal Nielsen, [scantech@isa.dknet.dk]
Hans Bruun Nielsen, [hbn@imm.dtu.dk]
Thomas Randrup, [Thomas.Randrup@3dfacto.dk]
Peter Røgen, [Peter.Roegen@mat.dtu.dk]
Marit Hvalsøe Schou, [marit@mip.ou.dk]
Robert Sinclair, [R.M.Sinclair@mat.dtu.dk]
Michael Ungstrup, [M.Ungstrup@mat.dtu.dk]