Odile Marcotte

Comptes rendus

Sixième atelier de résolution de problèmes industriels de Montréal

17 au 21 août 2015

Proceedings

Sixth Montréal Industrial Problem Solving Workshop

August 17 – 21, 2015

CRM-3350

CENTRE DE RECHERCHES MATHÉMATIQUES

CRSNG NSERC

Preface

C'est avec grand plaisir que je vous présente les comptes rendus du Sixième atelier de résolution de problèmes industriels de Montréal, qui a eu lieu au Centre de recherches mathématiques du 17 au 21 août 2015. Les six problèmes fournis par des entreprises étaient très différents les uns des autres, tant par les domaines concernés (ordonnancement de la production, photonique, aéronautique, visualisation, prévision de la valeur et fabrication additive) que par les techniques mathématiques utilisées pour résoudre ces problèmes (programmation mathématique, équations aux dérivées partielles, optimisation combinatoire, géométrie algorithmique, statistique et méthodes d'éléments finis).

Le CRM est particulièrement reconnaissant au Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) d'avoir financé cet atelier, par le truchement de la Plateforme d'innovation des instituts (PII), un projet des trois instituts de mathématiques canadiens visant à susciter de nouvelles collaborations entre mathématiciens et entreprises. Je voudrais aussi exprimer ma reconnaissance aux entreprises qui nous ont proposé des problèmes (Rio Tinto, Genia Photonics, Bombardier, Plotly, la Banque Nationale et Pratt & Whitney Canada), ainsi qu'aux chercheurs qui ont généreusement accepté de coordonner le travail des équipes (Charles Audet, C. Sean Bohun, Perouz Taslakian, Maciej Augustyniak et José Urquiza). Finalement je remercie chaleureusement Mlle Tina Mitre d'avoir accepté de rédiger une bonne partie de deux des rapports inclus dans ces comptes rendus.

It is with great pleasure that I introduce the Proceedings of the Sixth Montréal Industrial Problem Solving Workshop, which took place at the Centre de recherches mathématiques from August 17 *to* 21, 2015. *The six problems submitted were quite different from one another, whether one considers the fields where they arose* (*production scheduling, photonics, aeronautics, vizualisation, lifetime value prediction, and additive manufacturing*) *or the mathematical techniques used to solve these problems* (*mathematical programming, partial differential equations, combinatorial optimization, computational geometry, statistics, and finite elements methods*).

The CRM is especially grateful to the Natural Sciences and Engineering Research Council of Canada (*NSERC*) *for having supported this workshop through the Institutes Innovation Platform* (*IIP*), *a project of the three Canadian mathematics institutes whose goal is to foster new collaborations between mathematicians and industry. I am also very grateful to the companies that provided the problems studied during the workshop* (*Rio Tinto, Genia Photonics, Bombardier, Plotly, the National Bank, and Pratt & Whitney Canada*) *and the researchers who led the teams*(*Charles Audet, C. Sean Bohun, Perouz*

Taslakian, Maciej Augustyniak, and José Urquiza). *Finally I would like to thank Miss Tina Mitre, who wrote a substantial portion of two of the reports included in the proceedings.*

Montréal, le 15 août 2016 *Odile Marcotte*

Contents

1 Planning of the Maintenance Outages for a Set of Hydroelectric Units 1
Nadir Amaioua, Charles Audet, Emmanuel Bigeon, Pascal Côté, Quentin Desreumaux, Amina Ihaddadene, Youness Mir, Shahrouz Mirzaalizadeh, Daili Noureddine, Jesus Rodriguez, and Luckny Zéphyr
1.1 Introduction and Context 1
1.2 High-Level Problem Formulation 2
1.2.1 Current Situation 3
1.2.2 Problem Structure 3
1.3 Two Approaches to the Scheduling and Production Problem 4
1.3.1 Direct Approach 4
1.3.2 The Decomposition Approach 6
1.4 Discussion 9
References 10

2 Modelling and Specifying Dispersive Laser Cavities 11
C. Sean Bohun, Bryan Burgoyne, Yuri Cher, Linda J. Cummings, Peter Howell, Tina Mitre, Laurent Monasse, Judith Müller, and Stéphane Rouillon
2.1 Background 11
2.1.1 Tuneable Lasers 11
2.1.2 Wave Breaking 13
2.2 Problem Statement 13
2.3 Modelling Efforts 14
2.3.1 Hybrid Model 15
2.3.2 Non-Dimensionalization 18
2.3.3 Problem Restatement 19
2.3.4 Stationary Phase 20
2.4 Results 20
2.4.1 Simple Iteration Using an FFT 20

2.4.2 Stationary Phase Approximation 23
2.5 Summary 23
2.5.1 Conclusions 23
2.5.2 Further Questions 24
References 24

3 Optimal Partitioning of Multi-Block Structured Grids 27
Mohammad Akhavan, Patrice Castonguay, Marina Chugunova, Julio C. Góez, Mikko Kivelä, Odile Marcotte, Christophe Meyer, Tina Mitre, and Ragheb Rahmaniani
3.1 Background Information 28
3.2 Defining Objectives and Subproblems of the Project 29
3.3 Graph Design 30
3.4 Optimization Algorithms for Cluster Formation 32
3.5 Multi-Block Reformation 37
3.5.1 A Difficulty: The Volume Defined by a Set of Small Blocks Cannot Be Assumed to be a 3D-Polyhedron . 38
3.5.2 Reduction to the Case Where the Union of Blocks Can Be Considered as a 3D-Polyhedron 39
3.5.3 A Simple Greedy Heuristic 40
3.5.4 Computational Results 42
References 44
Appendix 45

4 Visualizing Huge Plots on the Web 47
Mikola Lysenko and Perouz Taslakian
4.1 Introduction 47
4.2 Progressive Loading of 2D Scatter Plots 51
4.3 Displaying 2D Line Plots 53
References 54

5 Lifetime Value in the Bank Industry 55
Maciej Augustyniak, Wyean Chan, Jean-François Forest-Désaulniers, Maïkel Geagea, Ryan Halabi, Cody Hyndman, Francis Lafontaine, Huimei Li, Kevin Luk, Hervé Mensah, Manuel Morales, Younes Ommane, Raphaël Ramora, Seyedhossein Rezaeilalami, Olivier Trottier, Shohre Zehtabian, and Farshid Zoghalchi
5.1 Introduction 55
5.1.1 Context 55
5.1.2 Problem 56
5.1.3 Objectives 56
5.2 Description of the Data 57
5.2.1 Commercial Data Set 57
5.2.2 Retail Data Set 58
5.3 Literature Review 59
5.4 Methods Considered 62

5.4.1 Self-Organizing Maps (SOMs) 62
5.4.2 Stepwise Regression 74
5.4.3 Principal Component Analysis (PCA) 74
5.4.4 Dependent Transition Probabilities through a Logistic Regression 75
5.4.5 Product Profile Matching 76
References 79

6 Finite Element Analysis of Ultralight Metallic Lattices ... 81
Thomas Briffard, Guillaume Fortier, Alireza Khademi, Yves Martin, Judith Müller, Argyrios Petras, Benoît Pouliot, Serge Prudhomme, Patrick Terriault, José Urquiza, Yingjun Wang, Tyler Wilson, and Huang Xu
6.1 The Problem 81
6.1.1 Context 81
6.1.2 Project Goal and Research Avenues 82
6.2 The Team's Results 83
6.2.1 Pratt & Whitney Specific Needs 83
6.2.2 The Literature 85
6.2.3 Examples of Computations for the Full Lattice and the Plate Lattice 85
References 91

1
Planning of the Maintenance Outages for a Set of Hydroelectric Units

Nadir Amaioua, Charles Audet, Emmanuel Bigeon, Pascal Côté, Quentin Desreumaux, Amina Ihaddadene, Youness Mir, Shahrouz Mirzaalizadeh, Daili Noureddine, Jesus Rodriguez, and Luckny Zéphyr

Abstract Operation of the hydroelectric systems is affected by planned outages for preventive maintenance. We analyze the hydroelectric operation at Rio Tinto Aluminium in order to propose solution strategies for the generator maintenance scheduling problem. We discuss a decomposition approach and heuristic method. We also present preliminary computational results for the proposed heuristic.

1.1 Introduction and Context

Rio Tinto (RT) is a multinational company that owns many aluminium smelters throughout the world. In the Saguenay – Lac-Saint-Jean (SLSJ) region RT operates four aluminium smelters that produce more than one million tons of aluminium per year. The Québec Power Operation (QPO) manages

Nadir Amaioua · Charles Audet · Amina Ihaddadene · Jesus Rodriguez
GERAD and Polytechnique Montréal

Emmanuel Bigeon
École de technologie supérieure

Pascal Côté
Rio Tinto Alcan

Quentin Desreumaux · Youness Mir
Université de Sherbrooke

Shahrouz Mirzaalizadeh
CIRRELT

Daili Noureddine
Université de Sétif 1

Luckny Zéphyr
Cornell University

the production and distribution of energy. The six hydroelectric powerhouses owned by RT (three on the Péribonka river and three on the Saguenay river) produce on average 2,080 MW per year, which amounts to almost 90% of the energy required for its aluminium melting operation. The reliability of generators is thus of utmost importance for the company.

In order to ensure the required level of reliability, QPO plans generator outages for maintenance activities. This maintenance schedule is defined several months in advance according to the available resources and maintenance policies. For the time being QPO does not have any automated system for building a maintenance schedule. The goal of this project is to formulate an optimization model for scheduling the outages for the SLSJ region and to propose algorithms for solving this model.

While building a maintenance schedule one must take several constraints into account. First, the resources that can be allocated to maintenance work are limited. Second, the required maintenance work must be carried out by various trades, each of which having a limited work force. Overtime may be used but in this case the labour costs will be higher. If a maintenance outage has a longer than average duration, it may be split into two parts when necessary. Finally, the main difficulty in building a schedule lies in the random nature of the natural inflows. Indeed, during certain periods (spring, summer, and autumn), there are large variations in the precipitations.

QPO uses a model to forecast the precipitations on the catchment basin. This model generates several scenarios for the natural water supply stored in the reservoirs. The schedule building must take these scenarios into account and minimize losses due to the outages and losses due to a lack of efficiency. Outage-related losses occur when the natural water supply (i.e., precipitations) is abundant and the reservoirs cannot store the excess water. In this case some extra turbo-generator must be switched on. If these turbo-generators are not available because of an outage, the excess water must be evacuated and "outage-related losses" will be recorded. The "lack-of-efficiency losses," on the other hand, are due to the operation of turbo-generators in areas with a high flow rate, where the transformation of the flow into power is not efficient.

The solution to the optimization problem will produce a single schedule, which minimizes the expected total losses (the sum of the outage-related losses and the lack-of-efficiency losses) given a probability distribution on the set of available scenarios.

1.2 High-Level Problem Formulation

A detailed and recent literature review of maintenance scheduling in the electricity industry is found in [3].

1.2.1 *Current Situation*

The current situation at RT is as follows. Decision-making is shared among two entities. On the one hand, the Network Control Centre (NCC) proposes a maintenance schedule by taking into account factors such as workforce and available equipment. On the other hand, the Water Resources Management group (WRM) evaluates the hydro-production costs by solving a nonlinear optimization problem which takes as an input the maintenance schedule proposed by the NCC.

If the WRM estimates that the value of energy production corresponding to the proposed schedule is not satisfactory, a new maintenance plan is requested from the NCC. Typically the new schedule consists of a slight modification to the previous one. In the current practice, the interaction between the NCC and the WRM continues until a consensus is reached. These iterations are time-consuming.

The objective of our research is to propose an alternative way (more efficient and integrated) of building schedules and production plans.

1.2.2 *Problem Structure*

We will use the following notation. Let Ω be the set of feasible maintenance schedules. For a schedule $s \in \Omega$, we define $\mathcal{X}(s)$ to be the set of feasible production variables. The problem is naturally decomposable into a maintenance scheduling problem (S) and a hydro-production (P) optimization problem. In short the maintenance scheduling problem can be defined as

$$\text{(S)} \qquad \begin{aligned} &\max_{s} p(s) - m(s) \\ &\text{s.t. } s \in \Omega, \end{aligned}$$

where $p\colon \Omega \to \mathbb{R}$ is the optimal energy production (converted into dollars) that can be achieved under schedule s, and $m\colon \Omega \to \mathbb{R}$ is the maintenance cost corresponding to the schedule. The function m is analytically known, thus easy to evaluate. The function p, however, is computed by solving the following hydro-operation problem:

$$\text{(P)} \qquad \begin{aligned} p(s) \coloneqq \ &\max_{x} f(x) \\ &\text{s.t. } x \in \mathbb{X}(s), \end{aligned}$$

where $f\colon \mathcal{X}(s) \to \mathbb{R}$ is the value of the energy production achieved under maintenance schedule s and production plan x. In the current RT conditions, problem (P) is solved using the IPOPT optimization solver [5]. The compu-

tational time for solving an instance of the production problem for a given schedule is approximately 30 seconds.

1.3 Two Approaches to the Scheduling and Production Problem

We propose two approaches for solving the scheduling and production problem. The first one makes use of the fact that $p(s)$ is readily computed by launching the IPOPT software. It consists of designing a heuristic local exploration of Ω in some neighbourhood of the initial schedule proposed by the NCC. This approach interprets the objective function $p(s)$ as a black box and does not attempt to use or extract information from the production problem (P). This will be referred to as the *Direct Search* approach.

The second approach considers the same problem decomposition, but makes use of dual values for constraints associated with specific maintenance periods in problem (P). Those dual values enable one to derive optimality cuts for the scheduling problem (S). The second approach will be refereed to as the *Decomposition* approach.

1.3.1 *Direct Approach*

Currently, RT solves a stochastic optimization problem for determining the hydro-operation plan and for estimating the value of energy production. Clearly, adding maintenance variables and constraints to this problem would increase its complexity. The direct approach would avoid this difficulty by using as a black box the mathematical program (as currently implemented) for estimating the value of energy production $p(s)$ for candidate schedules s. A neighbourhood search heuristic would be implemented for generating a sequence of maintenance schedules.

A main advantage of the Direct Search approach is that convexity and differentiability are not required for the problem functions.

Proposed Methodology

Figure 1.1 shows the interaction between the heuristic and the black box.

We propose a heuristic that, starting with a schedule s_0 as the current solution s^*, creates iteratively new schedules in the neighbourhood $\mathcal{N}(s^*)$. Each new candidate solution is sent to the black box for the estimation of the corresponding hydro-production energy value. A neighbour of the current solution is obtained through shifting by one day (forward or backward)

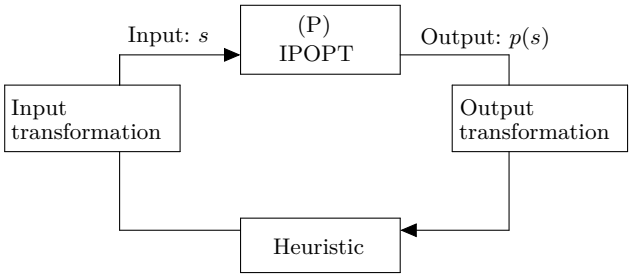

Fig. 1.1 The Direct approach

the starting date of a maintenance task. If a candidate schedule in the neighbourhood improves the current solution, it is retained as the new current solution s^* and the neighbourhood around s^* is explored. Otherwise a different schedule is created by shifting the starting date of another task.

For describing the heuristic approach we define the following notation.

d_j: Starting day of maintenance task j.

$s_{d_j \pm 1}$: Candidate maintenance schedule generated by shifting one day forward (+) or backward (-) the starting day of maintenance task j with respect to the current solution s^*.

$BB(\cdot)$: Energy value (obtained by means of the black box).

In Algorithm 1.1 several stopping criteria may be used. Here are some of them.

- The algorithm stops if no improvement is detected in the neighbourhood of the current solution (which is thus a local maximum).
- The algorithm stops if a predetermined number of iterations have been performed.

```
1  Initialize s* ← s_0;
2  while No stopping criteria met do
3      for j ∈ maintenance tasks do
4          Create a schedule s_{d_j − 1};
5          if BB(s_{d_j − 1}) improves the solution then
6              s* ← s_{d_j − 1}
7          else
8              Create a schedule s_{d_j + 1};
9              if BB(S_{d_j + 1}) improves the solution then
10                 s* ← s_{d_j + 1}
11             end if
12         end if
13     end for
14 end while
```

Algorithm 1.1 Heuristic approach

- The algorithm stops if the execution time exceeds a predetermined value.

For the test run we chose the third option as a stopping criterion.

Two Matlab scripts were written: one for creating the input text file for the black box (i.e., for the hydro-operation problem), and the other for processing the output file of the black box in order to extract the objective value and the problem solution.

Preliminary Numerical Results

RT provided us with an initial schedule and we used a remote connection in order to test the heuristic on its computers. The evolution of the objective value across iterations is shown in Figure 1.2. A 100% value represents the utopian situation in which no maintenance is scheduled. The blue curve in Figure 1.2 only reflects the improvements of the objective function value over the iterations. It shows that the final solution was identified around the optimization call number 195. In the same figure, the green line displays the values corresponding to all of the candidate schedules, including those that failed to improve the objective value. The heuristic reached an improvement over the initial schedule of approximately 0.05%, which corresponds to approximately $420,000 for the entire time period.

Figure 1.3 displays in black the initial schedule and in red the solution computed by the direct search approach. The difference between the two schedules is not significant, which can be convenient for the maintenance unit at RT.

We were able to find a better objective value by modifying slightly the initial schedule. This fact suggests that more rigorous approaches, such as the one proposed in the next subsection, could lead to further improvements.

1.3.2 *The Decomposition Approach*

Instead of exploring the neighbourhood of the current solution, this approach tries to exploit the structure of the problem in order to explore the whole solution space efficiently. Nevertheless this is a challenging problem, because in addition to the complexity of the stochastic hydro-power operation problem, the maintenance scheduling problem may have a solution space of order 2^{mst}, where m denotes the number pf maintenance tasks, s the number of scenarios, and t the number of time periods where the list of maintenance tasks can be initiated.

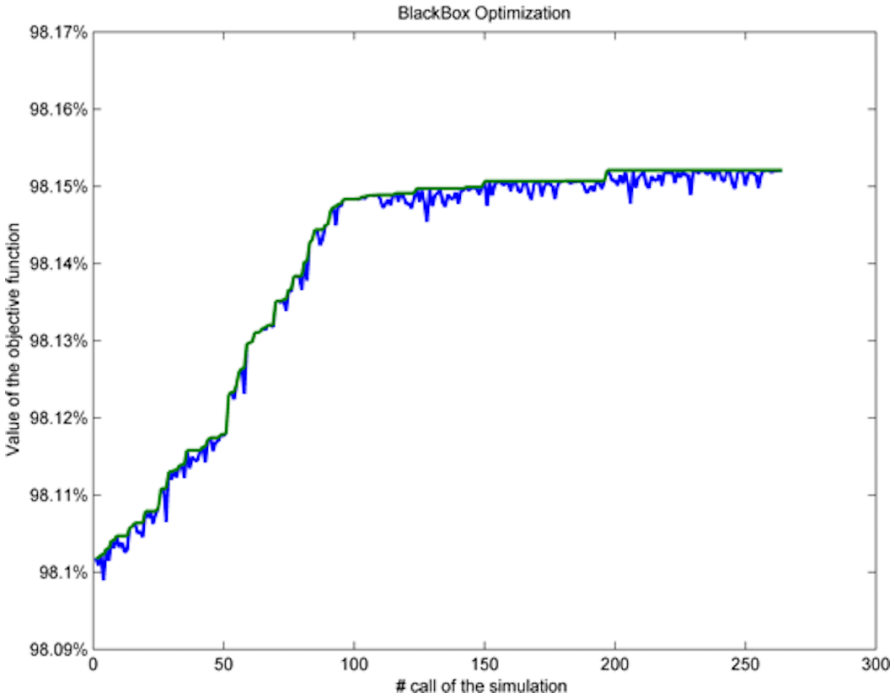

Fig. 1.2 Evolution of the objective function

Problem Formulation

The combined hydro-production planning & maintenance scheduling problem (P&S) is a stochastic nonlinear combinatorial optimisation problem whose optimal solution is difficult to compute. Decomposing this problem into maintenance decisions and production decisions (see Section 1.2.2) leads to smaller problems that are easier to solve. Following this idea we propose to implement a solution method based on Benders decomposition. For this approach, let us define the master problem (MP) as follows.

$$
\begin{aligned}
\text{(MP)} \qquad & \min_{(s,z)} z \\
& \text{s.t. } s \in \Omega \cap \mathcal{F} \\
& \quad (s,z) \in \mathcal{O}
\end{aligned}
$$

Here z denotes the objective function value of the complete problem (P&S), Ω the set of feasible decisions for the maintenance problem (S), $\mathcal{F}$ the set of maintenance decisions that are feasible for the hydro-production problem (P), and $\mathcal{O}$ the set maintenance decisions that are optimal for the complete problem (P&S).

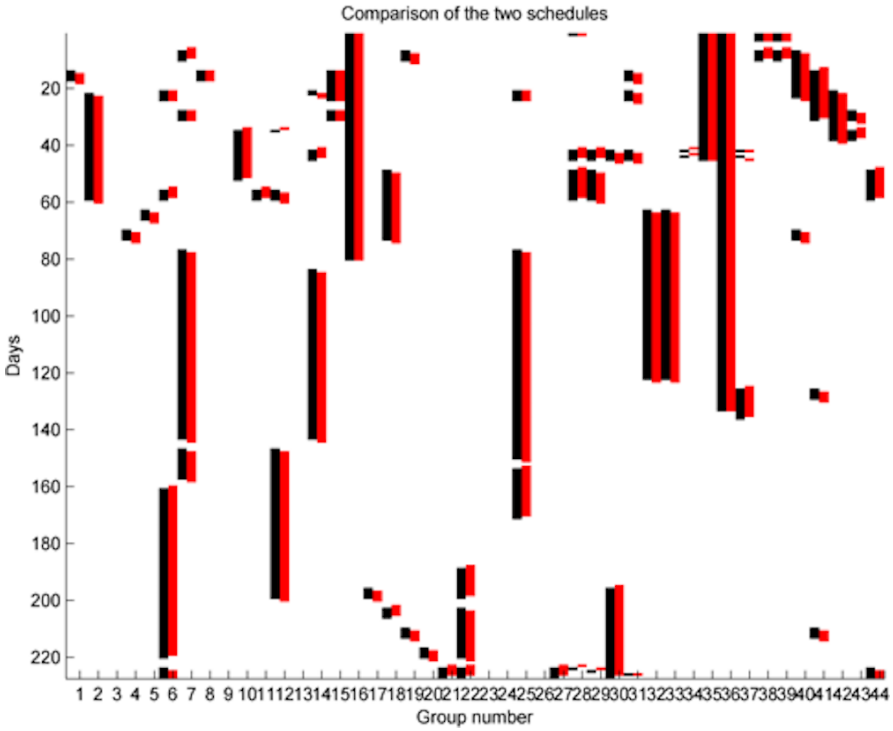

Fig. 1.3 Comparison between the initial schedule and the improved solution

The subproblem (SP) is defined as the production problem

$$\text{(SP)} \qquad p(s) := \max_{x} p(x) \quad \text{s.t. } x \in \mathcal{X}(s).$$

To connect the two problems (MP and SP), the maintenance decisions s in (SP) are fixed and given by (MP). On the other hand, $\mathcal{F}$ and $\mathcal{O}$ in (MP) are defined by feasibility cuts and optimality cuts, respectively, generated from the dual solutions of (SP).

Relaxing $\mathcal{F}$ and $\mathcal{O}$ in (MP) yields a relaxed master problem (RMP), in which cuts can be sequentially added whenever a trial maintenance plan s is found to be infeasible or suboptimal. In each iteration the subproblem (SP) provides "feedback" about the quality of the solution (e.g. dual prices and dual extreme directions). This information is used in the (MP) for including Benders cuts that can lead to a better solution. This iterative process is repeated until the convergence criteria are met, i.e., the schedule is feasible, the running time is acceptable, and the optimality gap is within a given threshold. The decomposition approach is summarized in Figure 1.4.

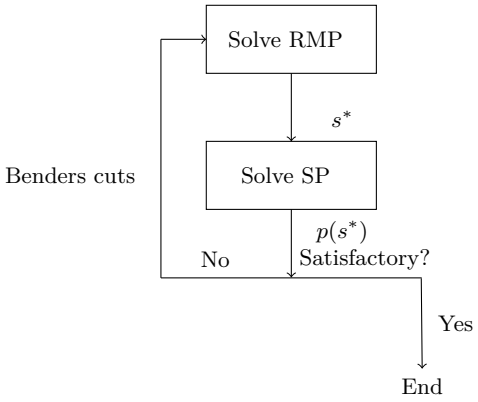

Fig. 1.4 Illustration of the decomposition approach

1.4 Discussion

The direct search optimization approach is conceptually simpler and easier to implement than the decomposition approach, and can lead more quickly to tangible numerical results. The decomposition approach, however, uses more efficiently the solution information provided by the subproblem and could thus yield global optimal solutions faster than by solving the complete P&S problem.

Future work within the direct search approach could consist of the following.

- Take into account the true scheduling constraints, including those on work force and overtime.
- Use a broader neighbourhood strategy around a schedule, together with a more sophisticated optimization solver (such as the Nomad software [4]) to solve the black-box scheduling optimization problem.
- Exploit the surrogate of function $p(s)$ under the surrogate management framework [2] of the Mesh Adaptive Direct Search algorithm [1].
- Apply a warm start with the IPOPT solver (i.e., in a given iteration, make IPOPT use the solution obtained at the previous iteration).

Future work within the decomposition approach includes the following.

- Formulate the maintenance scheduling problem and the complete hydro-operation & maintenance scheduling problem.
- Use Benders decomposition techniques to solve the scheduling and production problem.
- Develop an algorithm to solve the master problem efficiently. An important issue with respect to Benders decomposition is the efficient solution of the master problem. Column generation techniques or dynamic programming algorithms could be investigated for solving this problem.

- Use the nonlinear model of RT (instead of the current linear approximation) for the production subproblem. Indeed the linear approximation may have a significant and unwelcome impact on the solution. One way to avoid the linearization is to obtain dual variables directly from the IPOPT solver.

References

[1] C. Audet and J.E. Dennis Jr. "Mesh adaptive direct search algorithms for constrained optimization". *SIAM Journal on Optimization* 17:1 (2006), 188–217.

[2] A.J. Booker, J.E. Dennis Jr., P.D. Frank, D.B. Serafini, V. Torczon, and M.W. Trosset. "A rigorous framework for optimization of expensive functions by surrogates". *Structural and Multidisciplinary Optimization* 17:1 (1999), 1–13.

[3] A. Froger, M. Gendreau, J.E. Mendoza, E. Pinson, and L.-M. Rousseau. *Maintenance Scheduling in the Electricity Industry: A Literature Review.* Tech. rep. CIRRELT-2014-53. CIRRELT, 2014.

[4] S. Le Digabel. "Algorithm 909: NOMAD: Nonlinear Optimization with the MADS algorithm". *ACM Transactions on Mathematical Software* 37:4 (2011), 44:1–44:15.

[5] A. Wächter and L.T. Biegler. "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming". *Mathematical Programming* 106:1 (2006), 25–57.

2
Modelling and Specifying Dispersive Laser Cavities

C. Sean Bohun, Bryan Burgoyne, Yuri Cher, Linda J. Cummings, Peter Howell, Tina Mitre, Laurent Monasse, Judith Müller, and Stéphane Rouillon

2.1 Background

2.1.1 *Tuneable Lasers*

While a classical laser is thought of as having a single wavelength at which it operates, tuneable lasers are characterized by the ability to control this wavelength. There are many different types of tuneable lasers but in this report we will focus on dispersion-tuned, actively mode-locked, fibre lasers. With this particular type of laser, a highly dispersive element is combined with a modulator to select a particular wavelength. The dispersion causes

C. Sean Bohun
University of Ontario Institute of Technology

Bryan Burgoyne
Genia Photonics

Yuri Cher
University of Toronto

Linda J. Cummings
New Jersey Institute of Technology

Peter Howell
Oxford University

Tina Mitre
McGill University

Laurent Monasse
CERMICS, École des Ponts ParisTech

Judith Müller
Burlington, VT

Stéphane Rouillon
Centre de recherches mathématiques

each wavelength to have a distinctive propagation time in the cavity, and a wavelength is selected electronically by varying the repetition rate of the pulse with time modulation. Because of the way a wavelength is selected, many wavelengths can be produced within a very short time. This particular capability has direct applications to the life sciences, the biomedical industry, spectroscopy, and high-resolution imaging. Figure 2.1 shows both the external and schematic views of a tuneable laser developed by the industrial participant, Genia Photonics [2].

The cavity consists of an amplifying Erbium (Er) fibre and a highly dispersive element (CFBG) that act on the optical energy injected with a 980 nm fixed-frequency source laser. Working together, these two elements spread the energy through a wide range of frequencies that can be individually selected. By imposing a variable electronic modulation window onto this collection of dispersed frequency modes within the cavity, only those wavelengths that traverse the cavity while coinciding with the modulation frequency are se-

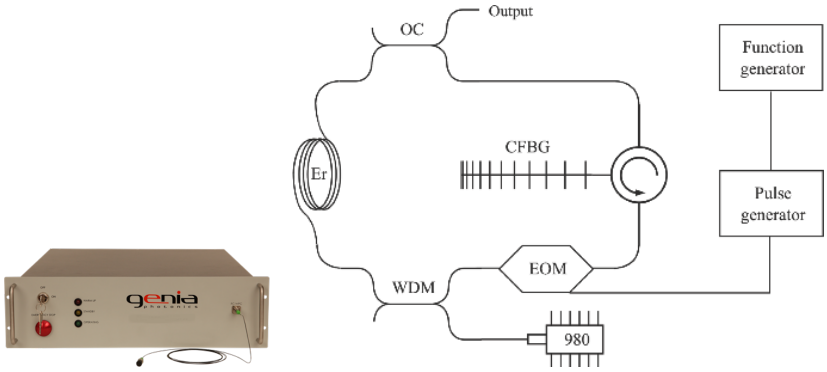

Fig. 2.1 Left: A tuneable pulsed fibre-based laser from Genia Photonics, offering a wide and quick wavelength range selectivity. Right: Schematic diagram of a tuneable laser consisting of an electrooptics modulator (EOM); optical coupler (OC); wavelength division multiplexer (WDM); and chirped fibre Bragg grating (CFBG).

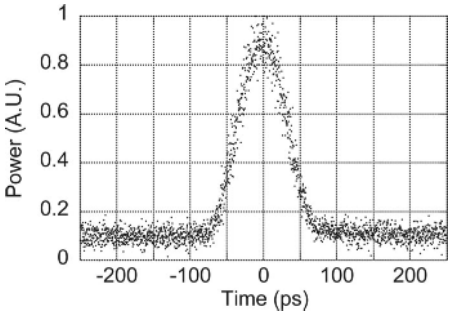

Fig. 2.2 Typical output signal from a tuneable fibre-based laser as reported in [1]. The short duration necessitates the averaging of the pulse over many iterations.

lected. The resulting pulse is sampled from the cavity with an optical coupler and utilized downstream. Figure 2.2 shows a typical pulse obtained with this technique. A more detailed discussion of the theory of operation can be found elsewhere [5, 7, 8].

An inherent difficulty in this sector is the very short duration of the pulses, $\sim$ 100 ps, that challenges the limitations of modern detection equipment. Pulse shapes have to be determined from autocorrelation measurements [6] or cross-correlation techniques [4], since the pulses are too short to be measured directly.

2.1.2 *Wave Breaking*

One of the main challenges with respect to the design of the laser is the elimination of an effect known as wave-breaking, whereby the pulse becomes unstable and ultimately unsustainable. Although the exact wave-breaking mechanics in the cavity are not entirely clear, what is observed experimentally is that there is a power threshold beyond which the nonlinearity of the fibre generates new frequency components appearing as fringes in the outer range of the pulse envelope. These different frequencies interfere with one another when they coexist at the same time within the pulse, leading to the generation of additional frequency components. The wave pulse deteriorates progressively as the fibre produces this cascade of additional frequencies. While modulating the signal in time can remove some of these frequencies, it also shapes the pulse; this makes the exact operation of the laser difficult to predict.

The functionality of tuneable lasers is made possible by the inherent nonlinearity arising from the interplay between the dispersive element, the time modulation window, and the deformation of the pulse as it travels through the fibre. This essential nonlinear behaviour, however, is also responsible for the breaking of the pulse at higher power levels. To make progress towards solving some of the design challenges for a tuneable laser, the optical nonlinearity of the glass fibre must also be considered (in any model). In the model described below each of the laser elements is considered independently as a nonlinear map. The stable operating point of the laser cavity is then described as a fixed point in the functional composition of these maps. One of the important new aspects with this technique is that the maps do not commute: the ordering of the elements in the cavity thus becomes important.

2.2 Problem Statement

Measuring the effect of the various experimental parameters is prohibitively expensive and difficult to carry out. Therefore we need high-resolution models

that predict the experimental observations. In addition, the modelling effort can provide further insight into the hierarchy of the processes involved and identify which processes dominate the observed behaviour.

Much of the team work revolved around the discussion of the wave-breaking effect, parameter sensitivity, and the determination of the wave profile. These issues can be summarized with four crucial questions:

1. Why does the wave break?
2. What is the range of the effective parameters yielding a wave with specific properties?
3. How do the parameters interact with one another?
4. Can a particular wave profile be specified?

In the work described below, each of these questions was examined using insights based on the structure of the various elements in the cavity and their interactions for a particular ordering.

2.3 Modelling Efforts

Traditional modelling efforts have focused on using a single equation describing the evolution of the average electric field. If one ignores the nonlinearities in the fibre itself and assumes small perturbations in the signal during its propagation within the cavity, then one is led to the expression [3]

$$\frac{\partial A}{\partial z} = -\mathrm{i}\frac{\beta_2}{2}\frac{\partial^2 A}{\partial T^2} - \frac{\epsilon}{2}T^2 A + \frac{g}{2}A.$$

In this relationship, $A = A(T, z)$ is the complex-valued electric field and is described as a function of T, the time in the frame of the moving pulse, and z, the number of roundtrips in the cavity. The various parameters (β_2, ϵ, and g) respectively control the second order dispersion, the bandwidth of the modulation, and the gain within the cavity.

Solutions of this model describe indeed some of the characteristics of the pulse, but it provides limited insight into the role of the various elements within the cavity, because of the effects of averaging. For example, there is a lack of separation between the dispersion within the fibre and the dispersion of the CFBG. Moreover, the model treats the gain as a single tuneable parameter, neglecting the effect (observed experimentally) of a gain that saturates with the pulse energy. Another effect lost in the averaging is the unique behaviour of the propagation within the fibre, without the other nonlinearities playing a role. For the lengths under consideration, only the phase of the field is changed within the fibre, while the energy is preserved.

One possible way to enhance the predictions of the single-equation model is to consider a sequence of four models, each of which has its parameters tuned to a particular element of the cavity: Dispersion, Modulation, Gain,

and the Output coupler/loss. Research along these lines has been shown to yield models reproducing the experimental results in a more precise fashion, especially at higher pulse energies [3]. In this report we have suggested and described a hybrid version of this idea, in which the nonlinearity of the individual components is embraced and used to derive a nonlinear map for each of the processes. This approach has the advantage that higher-order effects can be included into the various components in a systematic manner.

2.3.1 *Hybrid Model*

We treat the cavity as consisting of five distinct processes $\{\text{Gain}, \text{Loss}, \text{Modulation}, \text{Fibre}, \text{Dispersion}\}$, each of which generates a nonlinear map acting on the waveform. For this report a fixed ordering of these elements is considered, for the purposes of illustration. It would be straightforward, however, to modify the model by changing the order of the elements or including multiple copies of some elements.

Each of the elements is characterized by a dominant physical process and this behaviour was built into the derivation of each map. The following features were taken into account.

- The gain should saturate with the pulse energy.
- The modulation must reflect the experimental profile that was chosen.
- The fibre should be a map with nonlinear phase evolution and minimal dispersion.
- The dispersion element should correspond to a known built-in frequency dependence.

With these features as a guide, we now give a map for each of the components.

2.3.1.1 Saturated Gain

Gain within the optical cavity results from the gain fibre saturating with the amount of pulse energy. To model this effect we propose two characteristic parameters: a small signal gain g_0 and a saturation energy E_{sat} setting the energy level where the gain begins to saturate. Using these two parameters an appropriate model for the propagation of the pulse through the gain element becomes

$$\frac{\partial A}{\partial z} = \frac{g_0 A}{2(1 + E/E_{\text{sat}})}, \qquad E(z) = \int_{-\infty}^{\infty} |A(T, z)|^2 \, \mathrm{d}T. \tag{2.1}$$

Multiplying by $\bar{A}$, the complex conjugate of A, and integrating over T yields the relationship

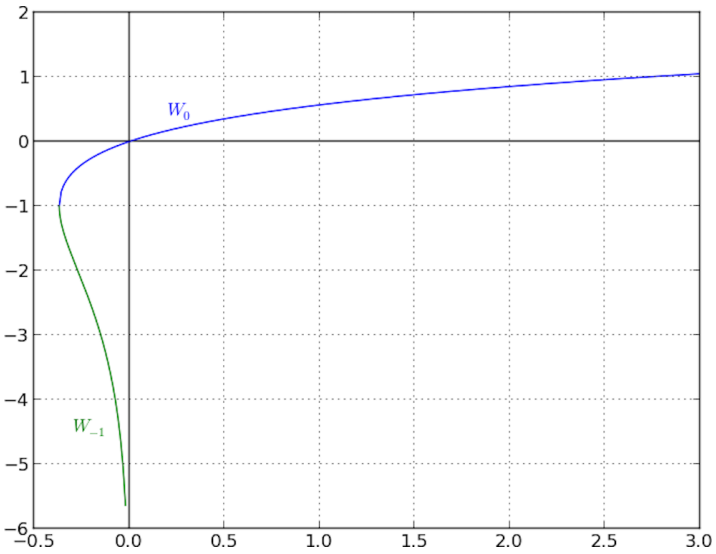

Fig. 2.3 The first two branches of the Lambert function.

$$\frac{\mathrm{d}E}{\mathrm{d}z} = \frac{g_0 E}{1 + E/E_{\mathrm{sat}}}, \tag{2.2}$$

which can be solved implicitly for the energy as a function of z. Setting $E(0) = E_0$ yields the relationship

$$\frac{E(z)}{E_{\mathrm{sat}}} \mathrm{e}^{E(z)/E_{\mathrm{sat}}} = \frac{E_0}{E_{\mathrm{sat}}} \mathrm{e}^{E_0/E_{\mathrm{sat}}} \mathrm{e}^{g_0 z} \tag{2.3}$$

for any z within the gain fibre and by inverting (2.3) (using the Lambert W-function) we obtain

$$\frac{E(z)}{E_{\mathrm{sat}}} = W_0\left(\frac{E_0}{E_{\mathrm{sat}}} \mathrm{e}^{E_0/E_{\mathrm{sat}}} \mathrm{e}^{g_0 z}\right). \tag{2.4}$$

Finally we note that

$$\frac{1}{E}\frac{\mathrm{d}E}{\mathrm{d}z} = \frac{g_0}{1 + E/E_{\mathrm{sat}}} = \frac{1}{|A|^2}\frac{\partial}{\partial z}|A|^2 \tag{2.5}$$

holds and conclude that the gain fibre affects the amplitude of the pulse according to the relation

$$A_1(z,T) = A_0(T)\left(\frac{E_{\mathrm{sat}}}{E_0}\right)^{1/2}\left(W_0\left(\frac{E_0}{E_{\mathrm{sat}}} \mathrm{e}^{E_0/E_{\mathrm{sat}}} \mathrm{e}^{g_0 z}\right)\right)^{1/2}, \tag{2.6}$$

with $A_0(T)$ being the input pulse profile at $z = 0$.

Therefore the net effect of the gain component as a map from the input $A(T)$ to the output $G[A](T)$ is

$$G[A] = A\left(\frac{E_{\text{sat}}}{E}\right)^{1/2}\left(W_0\left(\frac{E}{E_{\text{sat}}}\,\mathrm{e}^{E/E_{\text{sat}}}\mathrm{e}^{g_0\ell_G}\right)\right)^{1/2}, \tag{2.7}$$

$$E = \int_{-\infty}^{\infty}|A|^2\,\mathrm{d}T, \tag{2.8}$$

where ℓ_G is the length of the gain component.

2.3.1.2 Saturated Loss

The whole circuit has a net loss described by the map $L[A] = A\mathrm{e}^{-\alpha L/2}$, where α denotes a loss coefficient and L the length of the whole circuit.

2.3.1.3 Modulation

The modulation component simply multiplies A by a specified function of $\mathcal{T}$:

$$M[A] = \mathcal{T}(T)A, \tag{2.9}$$

where the transfer function is given by the relation

$$\mathcal{T}(T) = \frac{1}{2} - \frac{\nu}{2}\cos\bigl(\mu\pi\mathrm{e}^{-T^2/T_M^2}\bigr), \tag{2.10}$$

with T_M being the width of the modulation window. The parameters μ and ν are both equal to one in the ideal case.

2.3.1.4 Fibre (no dispersion)

The fibre dispersion is negligible when compared to the dispersion of the grating. Neglecting the fibre dispersion yields a fibre amplitude that satisfies the equation

$$\frac{\partial A}{\partial z} = \mathrm{i}\gamma|A|^2A, \tag{2.11}$$

where γ is in principle a known parameter. In this limit, the fibre only affects the phase and the solution for A is given by

$$A(z,T) = A_0(T)\mathrm{e}^{\mathrm{i}\gamma|A_0(T)|^2z}, \tag{2.12}$$

where once again, $A_0(T)$ is the profile input at $z = 0$. Therefore, the net effect of the fibre component is described by the map

$$F[A] = A\mathrm{e}^{\mathrm{i}\gamma|A|^2\ell_F}, \tag{2.13}$$

where ℓ_F is the length of the fibre.

2.3.1.5 Dispersion

The dispersion is specified by its action in the frequency domain. If we define the Fourier transform of A as

$$\mathcal{F}[A](\Omega) = \int_{-\infty}^{\infty} A(T)\mathrm{e}^{\mathrm{i}\Omega T}\,\mathrm{d}T, \tag{2.14}$$

then the effect on $\mathcal{F}[A]$ is given by

$$D\colon \mathcal{F}[A] \mapsto \mathcal{F}[A]\mathrm{e}^{\mathrm{i}\bar{\beta}_2\Omega^2}, \tag{2.15}$$

where $\bar{\beta}_2$ is a known dispersion coefficient. This can be expressed as a convolution of A with the dispersion action to yield

$$D[A] = \frac{\mathrm{e}^{\mathrm{i}\pi/4}}{2\sqrt{\pi\bar{\beta}_2}} \int_{-\infty}^{\infty} A(\tau)\mathrm{e}^{-\mathrm{i}(\tau-T)^2/4\bar{\beta}_2}\,\mathrm{d}\tau. \tag{2.16}$$

2.3.2 *Non-Dimensionalization*

Looking back at the derivation of the various maps, we can shed light on their structure by choosing appropriate scalings for the time T, the energy E, and the amplitude A. In particular, we choose to scale the time with respect to the period of modulation T_M, the energy with respect to the saturation value E_{sat}, and the amplitude in a fashion consistent with these two choices. We obtain

$$T = T_M\widehat{T}, \qquad E = E_{\mathrm{sat}}\widehat{E}, \qquad A = \left(\frac{E_{\mathrm{sat}}}{T_M}\right)^{1/2}\hat{A}. \tag{2.17}$$

When dropping the hats, the normalized maps take the form

$$G[A] = \frac{1}{E^{1/2}}\big(W_0(aE\mathrm{e}^{E})\big)^{1/2}A, \qquad L[A] = hA, \tag{2.18}$$

$$M[A] = \left(\frac{1}{2} - \frac{\nu}{2}\cos\big(\mu\pi\mathrm{e}^{-T^2}\big)\right)A, \qquad F[A] = \mathrm{e}^{\mathrm{i}b|A|^2}A, \tag{2.19}$$

$$D[A] = \frac{\sigma\mathrm{e}^{\mathrm{i}\pi/4}}{\sqrt{\pi}}\int_{-\infty}^{\infty} A(\tau)\mathrm{e}^{-\mathrm{i}\sigma^2(\tau-T)^2}\,\mathrm{d}\tau, \tag{2.20}$$

where each component has a corresponding non-dimensional parameter that characterizes its behaviour:

$$a = \mathrm{e}^{g_0\ell_G} \sim 30, \qquad h = \mathrm{e}^{-\alpha L/2} \sim 0.1, \tag{2.21}$$

$$b = \frac{\gamma\ell_F E_{\mathrm{sat}}}{T_M} \sim 10, \qquad \sigma = \frac{T_M}{2\sqrt{\bar{\beta}_2}} \sim 5. \tag{2.22}$$

Each of these can be individually tuned by varying either the length or the material properties of the various components within the laser cavity. An interesting point is that the modulation, gain, and loss terms reduce to multiplicative operators acting on the amplitude, whereas the fibre and dispersion mix the amplitude and phase information (the former at a localized point and the latter across the whole spectrum).

2.3.3 *Problem Restatement*

Forming a circuit within the cavity consists of concatenating the various elements and joining them with various lengths of fibre. Using the blocks that have been derived above, the output of each component is fed directly into the input of the next so that circuit formation is associated with functional composition of the maps. For example, consider a cavity with a single dominant fibre component and then loss, modulation, and gain elements that feed directly into a dispersion element. Here is a schematic representation of the circuit.

$$\cdots \text{Gain} \to \text{Disp.} \to \text{Fibre} \to \text{Loss} \to \text{Mod.} \to \text{Gain} \cdots$$

This corresponds to the iterated nonlinear map

$$\mathcal{G}[A] = M\Big[L\Big[F\Big[D\big[G[A]\big]\Big]\Big]\Big] \tag{2.23}$$

for each cycle around the cavity.

With this particular sequence of components, the determination of the invariant profile amounts to finding a fixed point in the map $A \mapsto \mathcal{G}[A]$. Questions about the instability caused by wave-breaking (observed experimentally) correspond to questions about loss of stability of the fixed point.

2.3.4 *Stationary Phase*

The non-dimensional parameters b and σ associated with the fibre and the dispersion (respectively) are consistent with the experimental observation that the phase of A changes rapidly as a function of T while the amplitude varies slowly. This remark allows one to use the method of the stationary phase to approximate the convolution.

Letting $A(T) = |A(T)|\mathrm{e}^{\mathrm{i}\sigma^2\psi(T)}$, we find in the limit as $\sigma \to \infty$ that the dispersion, to leading-order, takes the form

$$D[A] \sim \left(1 - \frac{\psi''(\tau_*)}{2}\right)^{-1/2} A(\tau_*)\mathrm{e}^{-\mathrm{i}\sigma^2(\tau_* - T)^2}, \tag{2.24}$$

where t_* satisfies the condition

$$\psi'(\tau_*) = 2(\tau_* - T). \tag{2.25}$$

2.4 Results

Initial numerical computation of the fixed point shows that iteration of this map can indeed result in a stable profile and that at higher energies, an instability seems to be associated with the breakdown of the pulse. These results reflect the experimental observations and are encouraging for this new model.

2.4.1 *Simple Iteration Using an FFT*

In this first method the Fast Fourier Transform is used to compute the dispersion so that

$$D(A) = \mathcal{F}^{-1}\left(\mathrm{e}^{\mathrm{i}\omega^2(/4\sigma^2)}\mathcal{F}(A)\right) \tag{2.26}$$

holds. The other components are left unchanged and the modulation is considered to be ideal ($\mu = \nu = 1$), so that

$$G(A) = \frac{1}{E^{1/2}}\left(W_0(aE\mathrm{e}^E)\right)^{1/2} A, \qquad L(A) = hA, \tag{2.27}$$

$$M(A) = \frac{1}{2}(1 - \cos(\pi\mathrm{e}^{-T^2}))A, \qquad F(A) = \mathrm{e}^{\mathrm{i}b|A|^2}A, \tag{2.28}$$

and the amplitude is discretized by taking N points within a numerical time window $(-T_m, T_m)$. The energy E was computed by using a simple

trapezoidal numerical integration of $|A|^2$ over the time window. For the simulations $N = 16384$, $T_m = 8$, and the nominal values of $a = 30$, $h = 1$, $\sigma = 5$ were chosen. We assigned several values to b in order to contrast the non-breaking and the breaking waves.

2.4.1.1 Non-Breaking

Choosing $b = 0.01$ (effective fibre length) results in a stable pulse with an observed kurtosis of 2.645, less peaked than a Gaussian with a kurtosis of 3. The convergence to the final energy over 200 iterations and the instantaneous amplitude squared, $|A|^2$, phase, ω, and negative chirp $\mathrm{d}\omega/\mathrm{d}T$ are shown in Figure 2.4. The real and imaginary components of the pulse and the associated frequency spectrum are displayed in Figure 2.5.

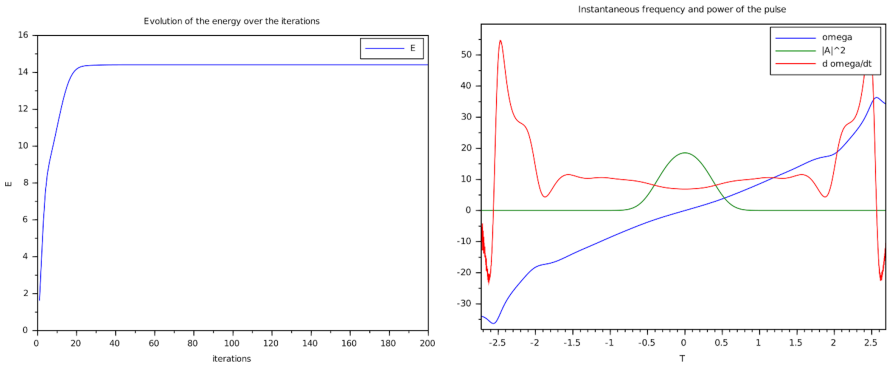

Fig. 2.4 No breaking in the wave. Left: Energy over 200 iterations, showing no wave-breaking. Right: The corresponding instantaneous frequency and pulse power.

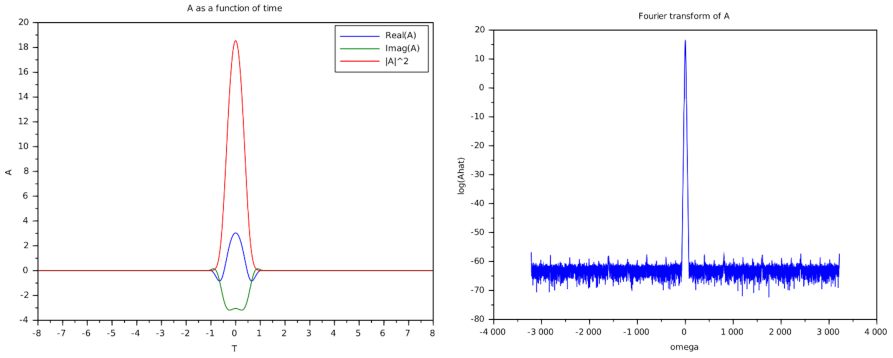

Fig. 2.5 No breaking in the wave. Left: Real and imaginary components of the amplitude as functions of time. Right: Instantaneous amplitude squared, $|A|^2$, phase, ω, and negative chirp ω'.

2.4.1.2 Breaking

Increasing the value of b causes the pulse shape to flatten and eventually it breaks (at $b = 0.1$), where the kurtosis has dropped to 2.37 before the onset of the instability. Figures 2.6 and 2.7 detail the characteristics of the pulse at this point.

The beginnings of a stability diagram for the parameter set can be mapped out by modifying each parameter in turn. Given the assignments $a = 30$ and $h = 1$, we have observed the following upper bounds of stability for b at several values of σ: $\sigma = 2$: $b \lesssim 0.008$; $\sigma = 5$: $b \lesssim 0.015$; $\sigma = 10$: $b \lesssim 0.02$.

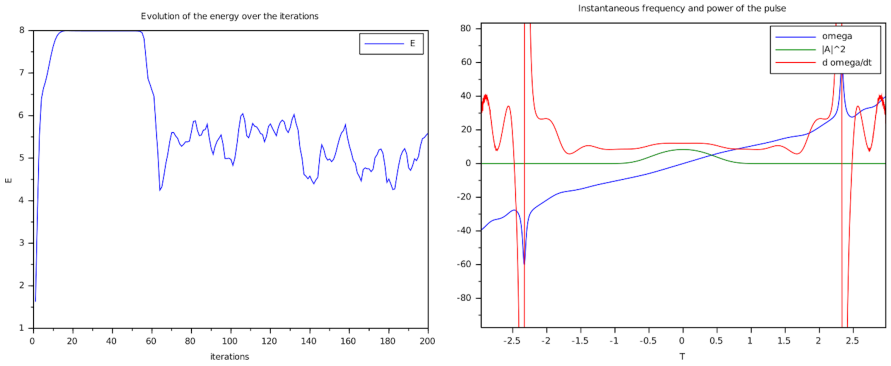

Fig. 2.6 Breaking in the wave. Left: Energy over 200 iterations, showing wave-breaking. Right: The corresponding instantaneous frequency and pulse power.

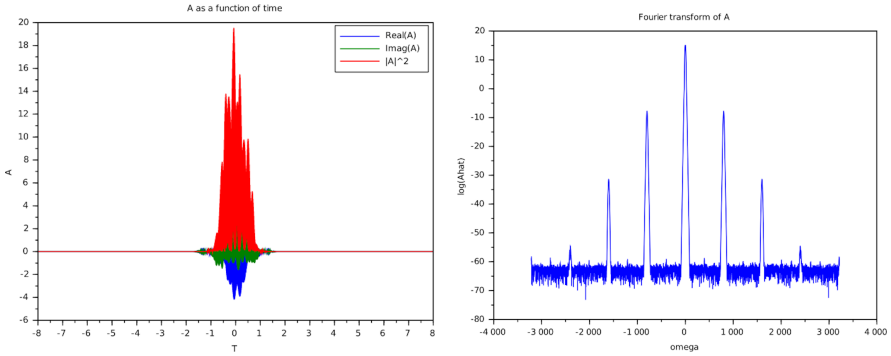

Fig. 2.7 Breaking in the wave. Left: Real and imaginary components of the amplitude as functions of time. Right: Instantaneous amplitude squared, $|A|^2$, phase, ω, and negative chirp ω'.

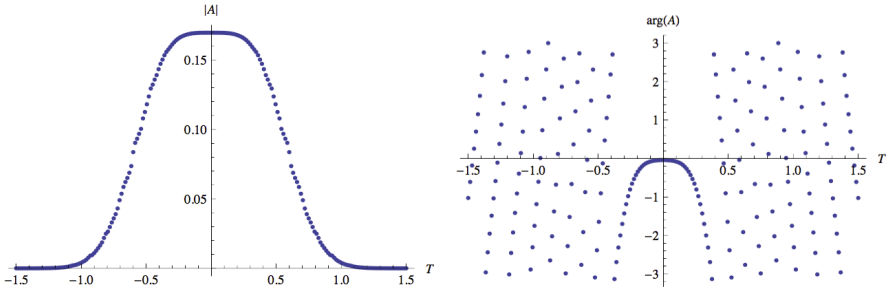

Fig. 2.8 The amplitude of the stationary phase in the dispersion.

2.4.2 *Stationary Phase Approximation*

We tried to use the stationary phase approximation for the dispersion, where the following values were chosen: $a = 30, b = 10, h = 0.25, \sigma = 5$. The approximation uses the stationary phase representation of the dispersion and assumes that ψ'' is essentially constant, so that the point where it is evaluated does not matter. As for the gain, loss, modulation, and fibre blocks, the components are used in the manner described previously. Figure 2.8 shows the results of this second approximation, which are very similar to the results obtained with the full Fourier transform technique.

2.5 Summary

2.5.1 *Conclusions*

The various schemes all show that there is a set of parameters that does mimic the shape of the pulses observed in the laboratory. Pulses can be shaped by modifying the parameters but some more work must be carried out to determine the viable range preventing the wave from breaking. Further analysis is required to describe in detail the effects observed with the proposed hybrid model, but the results are tantalizing.

Although they are preliminary, our results shed some light on the manner in which a stationary profile is generated. If one considers the fibre and dispersion only, one sees that

$$F[A] = A\mathrm{e}^{\mathrm{i}b|A|^2}, \qquad D[A] \sim A(\tau_*)\mathrm{e}^{-\mathrm{i}\sigma^2(\tau_*-T)^2}. \tag{2.29}$$

This implies that if one ignores higher-order effects, the shape of the profile is driven by $\mathrm{e}^{-\mathrm{i}\sigma^2(\tau_*-T)^2}$, which comes from the dispersion element. Nonlinear

effects in the fibre could be compensated by a modified phase resulting from the dispersion element.

2.5.2 *Further Questions*

With this initial analysis indicating that the hybrid model can reproduce the experimental wave-breaking phenomena, the industrial partner suggested new research directions that would enhance the capabilities of the tuneable lasers found on the market. In particular, there is extreme interest in:

1. Determining the "best" pulse shape so as to obtain the most linear chirp possible. This would enable efficient pulse compression;
2. Using the internal characteristic frequencies of the cavity determined by spreading the pulse out and modulating so as to clip the pulse. This information may be useful in determining a number of the effective parameters in the single-equation model.

References

[1] B. Burgoyne, A. Dupuis, and A. Villeneuve. "An experimentally validated discrete model for dispersion-tuned actively mode-locked lasers". *IEEE Journal of Selected Topics in Quantum Electronics* 20:5 (2014), 390–398.

[2] B. Burgoyne and A. Villeneuve. "Programmable lasers: design and applications". *Fiber Lasers VII: Technology, Systems, and Applications*. (San Francisco, CA, 2010). Ed. by K. Tankala and J.W. Dawson. Proceedings of SPIE, volume 7580. Bellingham WA: SPIE, 2010, 758002.

[3] B. Burgoyne and A. Villeneuve. "A discrete master equation for dispersion-tuned fiber lasers". *CLEO: Applications and Technology*. (San Jose, CA, 2012). Washington, DC: OSA Publishing, 2012, JW2A.83.

[4] C. Iaconis and I.A. Walmsley. "Spectral phase interferometry for direct electric-field reconstruction of ultrashort optical pulses". *Optics Letters* 23:10 (1998), 792–794.

[5] S. Li and K. Chan. "Electrical wavelength-tunable actively mode-locked fiber ring laser with a linearly chirped fiber Bragg grating". *IEEE Photonics Technology Letters* 10:6 (1998), 799–801.

[6] H. Nakatsuka, D. Grischkowsky, and A.C. Balant. "Nonlinear picosecond-pulse propagation through optical fibers with positive group velocity dispersion". *Physical Review Letters* 47:13 (1981), 910–913.

[7] S. Yamashita and M. Asano. "Wide and fast wavelength-tunable mode-locked fiber laser based on dispersion tuning". *Optics Express* 14:20 (2006), 9299–9306.

[8] S. Yamashita, Y. Nakazaki, R. Konishi, and O. Kusakari. “Wide and fast wavelength-swept fiber laser based on dispersion tuning for dynamic sensing”. *Journal of Sensors* 2009 (2009), 572835.

3
Optimal Partitioning of Multi-Block Structured Grids

Mohammad Akhavan, Patrice Castonguay, Marina Chugunova, Julio C. Góez, Mikko Kivelä, Odile Marcotte, Christophe Meyer, Tina Mitre, and Ragheb Rahmaniani

The problem proposed by Bombardier Inc. consisted of minimizing the solving time of fluid flow equations and balancing workload on parallel processors.

Section 3.1 of this report presents the context of the project; contains a description of the Bombardier company, a brief overview of computational fluid dynamics, and Navier–Stokes equations; and ends with the presentation of the categories of grids used for solving fluid-flow equations. Section 3.2 contains our optimization goals and our decomposition of the project into three main subproblems, each having a particular focus. In Section 3.3 we present the first one of the list, describing our approach for designing a graph with weighted nodes and edges. Section 3.4 is dedicated to the second subproblem and focuses on cluster-optimization algorithms. Section 3.5 presents the third and last step of our project, which is the design of block-formation algorithms

Mohammad Akhavan · Ragheb Rahmaniani
Polytechnique Montréal

Patrice Castonguay
Bombardier

Marina Chugunova
Claremont Graduate University

Julio C. Góez
Université de Montréal

Mikko Kivelä
University of Oxford

Odile Marcotte
CRM and UQAM

Christophe Meyer
UQAM

Tina Mitre
McGill University

given a number of available processors. Preliminary results accompany each of the subproblems.

3.1 Background Information

We first consider the industrial context of this project: Bombardier is one of the leading manufacturers in the aircrafts and rail industry, with over 74,000 employees. Their transport solutions include but are not limited to business jets, commercial planes, high-speed trains, light rail, and trolley buses. Since the company focuses on the design, the manufacture, and the reparation of these structures and their components, computational fluid dynamics is a main area of research and specialization of the company.

By definition, computational fluid dynamics uses tools from numerical analysis for simulations of viscous surfaces. The physical models are represented by Navier–Stokes equations, which are non-linear partial differential equations describing the physics and the motion of the fluidic substances in a region of space and time.

To initialize the problem, the equations of motion require boundary conditions defining the geometrical and physical bounds of the modelled object. Next, the volume on which the equations must be solved requires discretization and becomes a computational mesh of hexahedral cells in three-dimensional space.

Finding innovative numerical tools for the Navier–Stokes equations, however, is not the topic of our project. There is a wide area of research in the field, and Bombardier is currently using the algorithmic tool entitled FANSC,[1] which was developed in 1999. With the advent of parallel computations and high-efficiency multi-processors, an optimization challenge arises: how do we minimize the computational time by balancing the workload between the processors and considering the work of each cell?

Depending on the characteristics of the object or the problem to be solved, the mesh used for the solver may be structured, unstructured, or a hybrid of these two kinds of meshes. A structured grid, as seen in Fig. 3.1(A), contains regular connectivities between quadrilaterals or hexahedra, in two- or three-dimensional space, respectively. Unstructured grids, like the one shown in Fig. 3.1(B), are composed of triangles or tetrahedra and do not allow a one-to-one mapping into a structured domain. Because of efficiency considerations and in order to use the convergence properties of the first type of grid, fluid dynamics equations are often solved on multi-block structured grids, which are large areas composed of structured domains with potentially different dimensions (see Fig. 3.1(C)).

[1] **F**ull **A**ircraft **N**avier–**S**tokes **C**ode

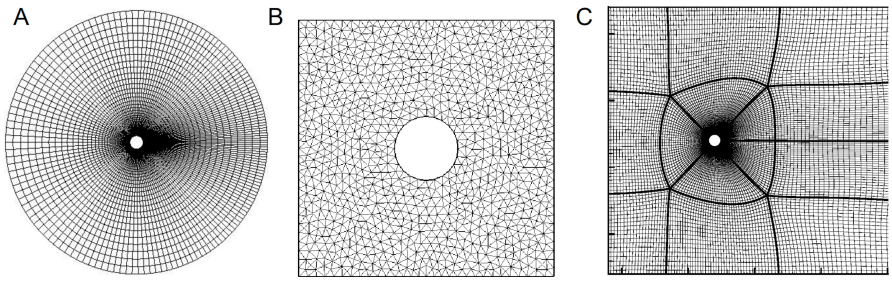

Fig. 3.1 Example of two-dimensional grids used for solving equations representing fluid flow dynamics: A. Structured domains composed of quadrilaterals, B. Unstructured grids composed of triangles, C. Multiblock structured grids, where each block represents a structured grid as in case A but the connectivities between the blocks exhibit irregularities similar to those in case B.

Since each block (in the last scenario) is a structured grid on its own, the problem becomes computationally tractable. The challenge in the next step is the distribution of the tasks. A complete algorithm for the development of multi-block structured Navier–Stokes is found in [7], and requires that a block have a minimum of 8 cells in each direction. The following question arises: is the solution presented in this article optimal if we allow splitting and merging of the initial blocks? Can we generalize the solution and parallelize the workload, given a particular number of available processors?

3.2 Defining Objectives and Subproblems of the Project

As previously mentioned, we are working with a multi-block structured grid and the overall objective is to find the optimal way of allocating particular groups of blocks to a certain number of available processors.

The constraints of our problem, which will be formulated in a mathematical fashion in Section 3.4, are the following:

- The computational load must be balanced, i.e., the difference between the numbers of cells allocated to any two processors must be as small as possible;
- The maximum communication cost between a processor and all the other processors must be as small as possible;
- The number of blocks allocated to a given processor must not be too large;
- None of the blocks allocated to a given processor must be too small.

To propose a complete solution, we decompose our problem into three main stages:

1. Refine the original partition so as to obtain small blocks and create a graph with weights on nodes and edges;

Table 3.1

Subproblem	Input	Output
1. *Graph design*	Connectivity matrix, as used by FANSC software	Graph with weighted vertices and edges
2. *Cluster optimization*	Graph with weighted vertices and edges	Cluster formed by given optimization constraints
3. *New block formation*	Cluster formed by given optimization constraints	New multi-block structured-grid for the given set of processors

2. Partition the graph into clusters (i.e., groups of blocks assigned to a given processor);
3. Reorganize the blocks within a given cluster so as to obtain relatively large blocks that are grid-like and are not too "thin."

3.3 Graph Design

Our method was initially tested on a generic example provided by Bombardier. We have read the code 3.1, which can be found in the appendix, and produced the three-dimensional structure in Fig. 3.2 by subdividing the structure into cells of dimension $8 \times 8 \times 8$ and colouring each initial block. We have considered the size of each of the five blocks, the position of the blocks in three-dimensional space, and the connectivities between the adjacent faces (note that the green, yellow, and red blocks should be connected and form a ring). The figure presented here was made using `MATLAB`. Since the second step of our solution required the connectivities to be described as graphs with weighted edges and vertices, we had to describe the figure from a new perspective.

Afterwards we created weighted graphs for the three shapes shown in Fig. 3.3. For example, in order to obtain the weight of each node (or $8\times8\times8$ cell) forming the cube in Fig. 3.3(A), we consider that a corner has 3 neighbours, an edge four neigbours, the inside of a face five neighbours, and a node inside the cube six neighbours (as in Table 3.2). We also consider the weight of each edge connecting two nodes, or cells, to be the sum of the volumes of these two nodes. In more elaborate examples, some cells might not have the shape $8 \times 8 \times 8$, but they should be chosen so that they are as close as possible to this cuboid shape. In Fig. 3.3(B), cells or nodes creating the adjacent face would contain an extra neighbour. A similar rationale can be used for case C of Fig. 3.3.

After creating a graph with weighted vertices and weighted edges, the next step consists of optimizing the allocation of cells or blocks to processors, with each cluster of cells or blocks corresponding to a single processor. For the

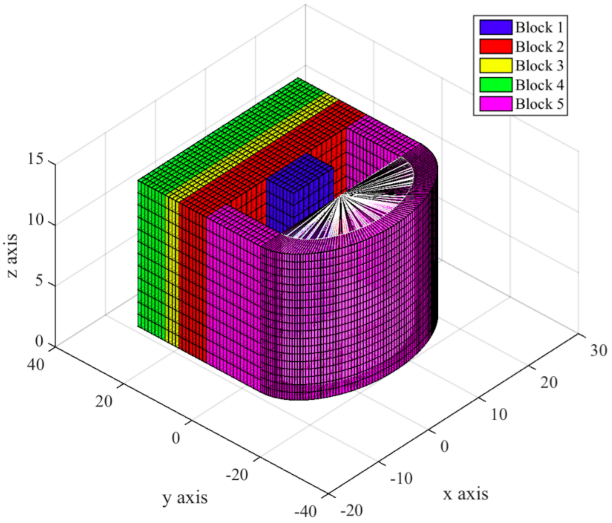

Fig. 3.2 Three-dimensional plot of the surface provided

Table 3.2 Excerpt from the *.txt file containing the graph of a $7 \times 8 \times 9$ block, composed of $8 \times 8 \times 8$ cells.

Node	Neighbours
1	$(2, 8, 57)$
2	$(1, 3, 58, 114)$
3	$(2, 4, 59, 115)$
4	$(3, 5, 60, 116)$
5	$(4, 6, 61, 117)$
6	$(5, 7, 62, 118)$
7	$(6, 14, 63)$
8	$(15, 1, 64, 9)$
9	$(8, 10, 16, 2, 65)$
10	$(9, 11, 17, 3, 66)$

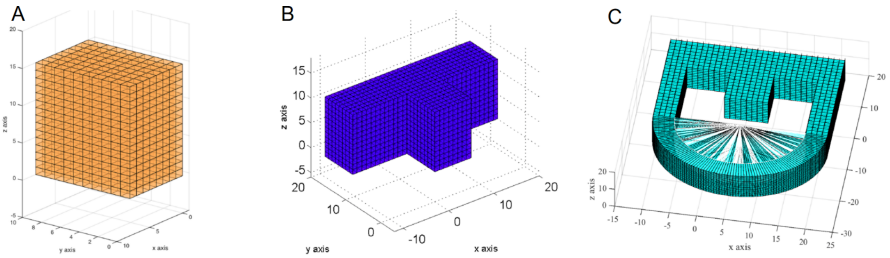

Fig. 3.3 Types of grids used in the tests of our algorithms

optimization problem we use the constraints given in Section 3.2 (formulated in mathematical terms in the following section).

3.4 Optimization Algorithms for Cluster Formation

Actually the problem described in the previous section is very similar to the *balanced graph clustering* problem, which we now define formally.

Input: Graph $G = (V, E)$, node weights $w_n \colon V \mapsto \mathbb{N}_+$, edge weights $w_e \colon E \mapsto \mathbb{N}_+$, number of colours k.

Objective: Find a colouring of nodes $c \colon V \mapsto P$ (where $P = \{1, \ldots, k\}$) such that $T = c_1 \max_{p \in P} V_p + c_2 \max_{p \in P} B_p$ is minimized. Here the volume of a partition is defined as $V_p = \sum_{u \in V} w_n(u) \delta(c(u), p)$ and the boundary of a partition as $B_p = \sum_{(u,v) \in E} w_e\big((u, v)\big) \big[1 - \delta\big(c(u), c(v)\big)\big] \delta(c(u), p)$, where δ denotes the indicator function.

For solving the balanced graph clustering problem, one can

- use software products such as KaHIP and METIS, which are free and available (they have slightly different objective functions),
- use a heuristic based on label propagation, or
- formulate the problem as an integer programming problem and solve it.

Figure 3.4 displays the result of the application of KaHIP to cluster formation in a specific case.

We now describe a heuristic partitioning method applied to structured grids. Algorithm 3.1 contains the description of the heuristic in pseudocode and Figure 3.5 a schematic description of the same heuristic.

Figure 3.6 displays a structured grid and Figure 3.7 the partition obtained for this grid by applying the heuristic algorithm.

Figure 3.8 displays a T-shaped grid and decompositions obtained by our heuristic algorithm (for 5 and 8 processors, respectively).

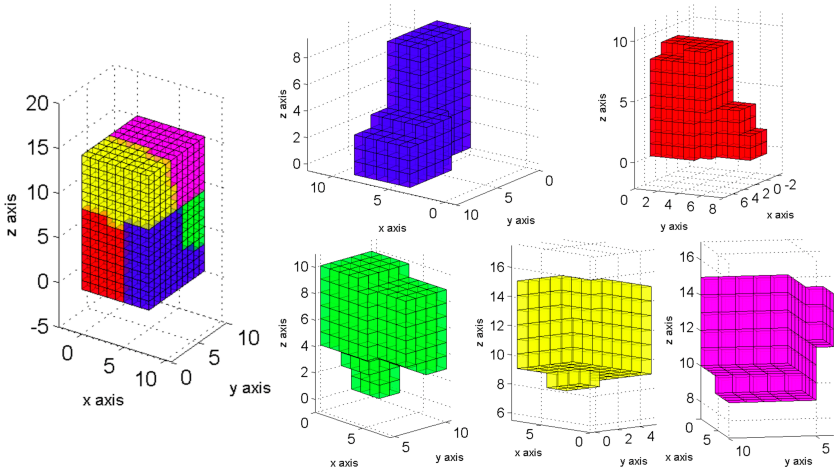

Fig. 3.4 $8 \times 10 \times 15$ structured grid

1 Set nodes to random sets forming a partition. Use the previous partition as a starting point. **foreach** node $u \in V$ (in random order) **do**
2 Find colours in the neighbourhood of u: $C_{\text{candidates}} = \{c(v) : (u, v) \in E\}$. **foreach** $p \in C_{\text{candidates}}$ **do**
3 Find change in the objective function $\Delta_{u,p}T$ if $c(u)$ is set to p. Select p^* that minimizes $\Delta_{u,p}T$. **if** $\Delta_{u,p^*}T < 0$ **then**
4 Set $c(u)$ to p^*.
5 **end if**
6 **end foreach**
7 **end foreach**
8 **if** at least one colour was changed in last run of point 3 **then**
9 Repeat point 3.
10 **end if**

Algorithm 3.1 Description of the heuristic in pseudocode

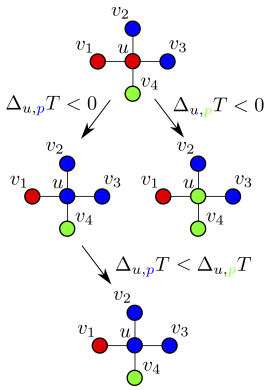

Fig. 3.5 Schematics of the heuristic

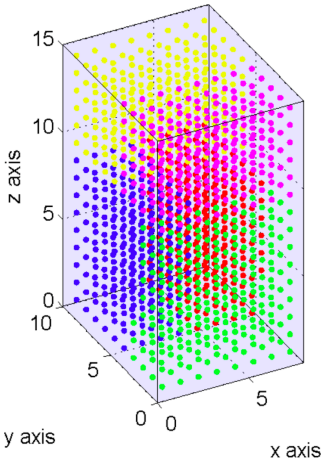

Fig. 3.6 $8 \times 10 \times 15$ Structured grid

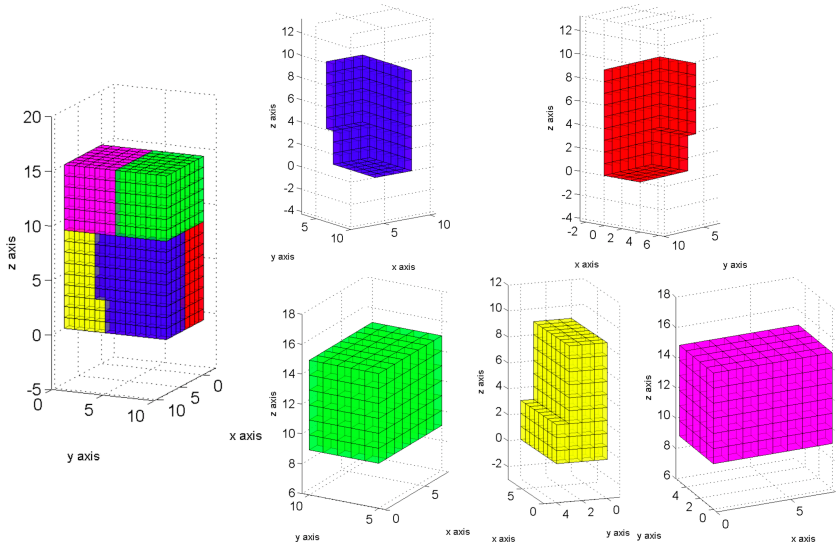

Fig. 3.7 $8 \times 10 \times 15$ Structured grid partitioned by our algorithm

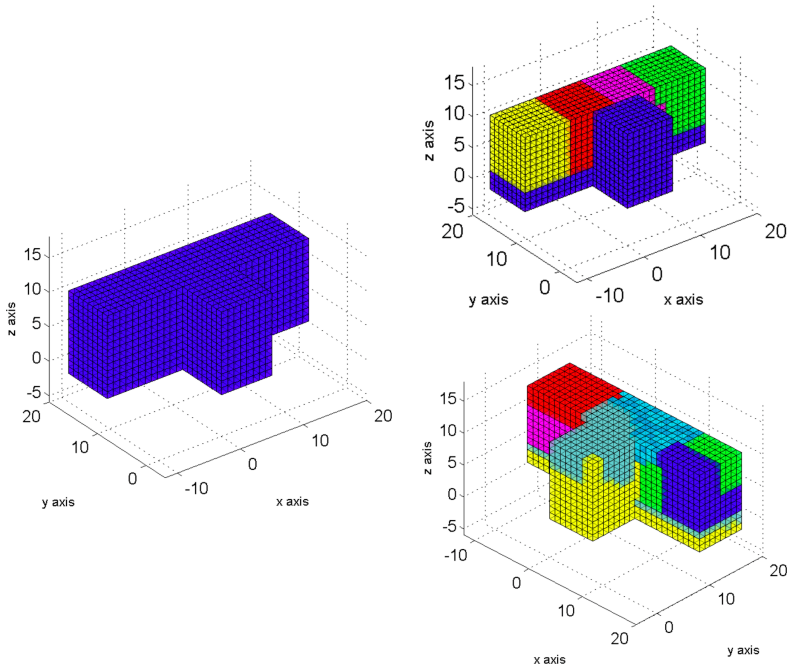

Fig. 3.8 Structure partitioned into 5 processors and into 8 processors

We now turn to a mathematical formulation of our problem. Recall that our goal is to assign the work units (blocks) to processors in order to minimize a weighted sum of the maximum load and the maximum communication cost. Let y_{vp} be a binary variable taking the value 1 if and only if the block v is assigned to the processor p. The objective function can be expressed as

$$\text{minimize } \left(C_1 \max_{p \in P} \left\{ \sum_{v \in V} w_v y_{vp} \right\} + C_2 \max_{p} \left\{ \sum_{p^\star \neq p} \sum_{(v,u) \in E} w_{(v,u)} y_{vp} y_{up^\star} \right\} \right).$$

The constraint that a block is allowed to a single processor can be expressed as

$$\sum_{p \in P} y_{vp} = 1, \quad \forall v \in V.$$

Finally the constraint that y_{vp} is a binary variable can be expressed as

$$y_{vp} \in \{0, 1\}, \quad \forall p \in P, v \in V.$$

We obtain an equivalent formulation by introducing the variables T_1 and T_2.

$$\text{minimize } (C_1 T_1 + C_2 T_2)$$

such that

$$\begin{aligned}
&\sum_{v \in V} w_v y_{vp} \leq T_1, && \forall p \in P \\
&\sum_{p^\star \neq p} \sum_{(v,u) \in E} w_{(v,u)} y_{vp} y_{up^\star} \leq T_2, && \forall p \in P \\
&\sum_{p \in P} y_{vp} = 1, && \forall v \in V \\
&y_{vp} \in \{0, 1\}, && \forall p \in P, v \in V
\end{aligned}$$

In the latter formulation we can linearize the term $y_{vp} y_{up^\star}$. We do this by introducing the binary variables $x_{vu}^{pp^\star}$, where $x_{vu}^{pp^\star}$ equals 1 if and only if block v is assigned to processor p and u to processor $p^\star$. To enforce the definition of $x_{vu}^{pp^\star}$, we include the following constraints into the model, for every arc (v, u) and any pair of distinct processors p and $p^\star$.

$$\begin{aligned}
x_{vu}^{pp^\star} &\geq y_{vp} + y_{up^\star} - 1 \\
x_{vu}^{pp^\star} &\leq y_{vp} \\
x_{vu}^{pp^\star} &\leq y_{up^\star}
\end{aligned}$$

Because the model is a minimization program, the constraints $x_{vu}^{pp^\star} \leq y_{vp}$ and $x_{vu}^{pp^\star} \leq y_{up^\star}$ are redundant and the constraint $x_{vu}^{pp^\star} \geq 0$ can replace the

constraint that $x_{vu}^{pp^\star}$ is a binary variable, since $x_{vu}^{pp^\star} \geq 0$ will always equal 1 or 0.

It is difficult to solve the model including the $x_{vu}^{pp^\star}$ because there are too many of those variables. If we assume that all the w_v (i.e., vertex weights) are equal and all the $w_{(v,u)}$ (i.e., arc weights) are equal, we may introduce instead the variables $x_{(v,u)}^p$, where $x_{(v,u)}^p$ equals 1 if and only if (v,u) is an outgoing arc from the cluster assigned to processor p. We can replace the above constraints by

$$\begin{aligned} & x_{vu}^p \geq y_{vp} + y_{up^\star} - 1, \quad p^\star \neq p \\ & x_{vu}^p \geq 0. \end{aligned}$$

We then obtain the following formulation.

$$\text{minimize } C_1 w_1 T_1 + C_2 w_2 T_2$$

such that

$$\begin{aligned} & \sum_{v \in V} y_{vp} \leq T_1, && \forall p \in P \\ & \sum_{p \in P} y_{vp} = 1, && \forall v \in V \\ & \sum_{(v,u) \in E} x_{(v,u)}^p \leq T_2, && \forall p \in P \\ & x_{(v,u)}^p \geq y_{vp} + y_{up^\star} - 1, && \forall (v,u) \in E, \forall p, p^\star \in P, p^\star \neq p \\ & y_{vp} \in \{0,1\}, && \forall p \in P, v \in V \\ & x_{(v,u)}^p \geq 0, && \forall (v,u) \in E, \forall p \in P. \end{aligned}$$

Solving this model is a very challenging combinatorial problem, because the model is highly degenerate and the linear programming relaxation is very weak: in many cases a node is "split into two halves" and those halves belong to different clusters. Of course the model needs to be solved for huge networks. We experimented with three approaches:

- using Cplex;
- using Benders decomposition; and
- designing an MIP-based heuristic.

Note that if we know the values of the y variables, it is almost trivial to compute the communication cost.

Here is the description of our MIP-based heuristic.

1. Find the degree of each node.
2. For each cluster, execute the following steps.
   a. Find the node with the smallest degree (i.e., v).
   b. While the load limit is not reached or no available neighbours exist, do

i. Find all neighbours of v and add them to the current cluster list;
ii. Update V by selecting one of its neighbours for exploration.

c. Fix the binary variables associated with each node in the cluster list

3. Solve the restricted MIP.

The following table summarizes our computational results (the time limit was set to 30 minutes).

Approach	Instance I	Instance II
Cplex	149,951	1,966,080
Benders	150,480	N/A
Heuristic I	162,240	369,280
Heuristic II	130,880	291,136

Table 3.3 Computational results, in seconds

3.5 Multi-Block Reformation

Let us recapitulate: in Section 3.3 the given blocks were decomposed in small blocks of size $8 \times 8 \times 8$. Then in Section 3.4 each small block was assigned to a processor. The final step consists of merging the small blocks assigned to the same processor into larger blocks, in such a way as to increase the efficiency of the computation. This is the topic of the current section.

Consider a particular processor and the set of small blocks assigned to it. This set of blocks must be partitioned in such a way that the small blocks of each cluster form a larger hexahedral block. The Navier–Stokes flow solver will be applied on each of these larger blocks, with some communications involved on the boundaries. Here we will only consider the communication cost arising from communications within a processor, since communications between different processors have been considered in Section 3.4. In order to maximize the efficiency of the algorithm that solves the Navier–Stokes equations, the partition should have the following properties:

- it should have a small number of blocks;
- each block should be as close as possible to a cube (i.e., it should have roughly the same number of cells along each direction); and
- all blocks should have roughly the same size.

This suggests the modelling of the problem of merging the small blocks into larger blocks as a multi-objective programming problem including the two objective functions below:

- Minimize the number of blocks; and
- Maximize the minimum size over all three directions and over all blocks.

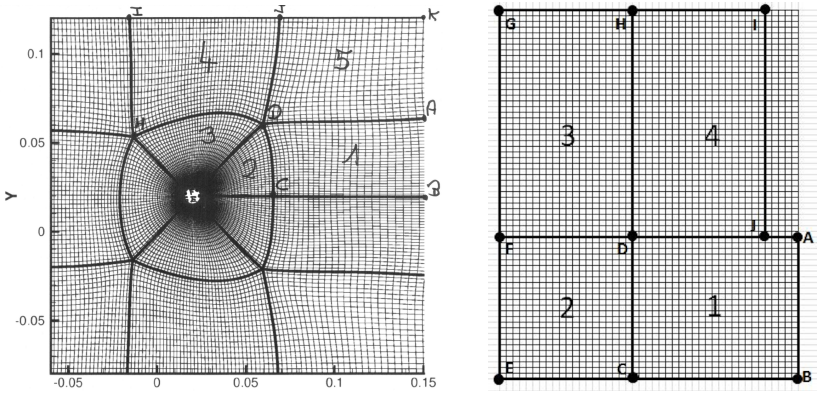

Fig. 3.9 Non-reductibility to a problem on a polyhedron

3.5.1 *A Difficulty: The Volume Defined by a Set of Small Blocks Cannot Be Assumed to be a 3D-Polyhedron*

Whereas each small block can individually be seen as a $8 \times 8 \times 8$ cube, the volume defined by the union of such small blocks cannot be assumed to be a 3D-polyhedron.

To see why, consider the 2D example of Fig. 3.9. Block 1 can be considered as a rectangle because the number of cells on the side AB matches the number of cells on the side CD, and the number of cells on the side AD matches the number of cells on the side BC. Similarly, each of Blocks 2, 3, 4, and 5 can be considered as a rectangle. Considering the adjacency between Blocks 1 and 2, between Blocks 2 and 3, and between Blocks 3 and 4, we are tempted to represent the surface defined by these four blocks as shown in the right-hand part of Fig. 3.9. This representation, however, is not correct, because it suggests that blocks 1 and 4 are adjacent, which is not the case (see the left-hand part of Fig. 3.9).

The case where the volume defined by the small blocks can be considered as a 2D- or a 3D-polyhedron has already been partially addressed in the literature. First notice that as suggested by Fig. 3.9, the resulting polyhedron will be *orthogonal*, in the sense that all the faces of the polyhedron will be parallel or orthogonal to one another.

- The problem of partitioning an orthogonal 2D-polygon into fat rectangles has been considered by O'Rourke and Tewari [6]. This problem consists of finding a partition into rectangles that maximizes the shortest rectangle size over all rectangles. The most general algorithm has a complexity of $O(n^{42})$, where n is the number of vertices of the polygon, while algorithms with lower complexities have been proposed for restricted versions of the problem. We are not aware that this problem has been considered in 3D.

- The problem of partitioning an orthogonal polyhedron into the minimum number of rectangles has been considered both in 2D and in 3D. In 2D the problem can be reduced to finding a maximum matching in a particular bipartite graph. The first polynomial algorithm that runs in $O(n^{5/2})$ has been proposed by Lipski et al. [5] in 1979. Since then, algorithms with lower complexities have been proposed. Currently the best complexity seems to be $O(n^{3/2} \log n)$ (see the survey by Keil [3] for references). In contrast, the 3D version of the problem was shown to be NP-hard by Dielissen and Kaldewaij [1], leading to the proposal of several heuristics (see, e.g., [2] and references therein).
- In the 2D version of the problem, a third objective function has been considered, which is to minimize the total length of the lines used to describe the decomposition. Lingas et al. [4] proposed a $O(n^4)$ algorithm for the case where the polygon has no hole and showed that the problem becomes NP-hard when holes are allowed.

Finally we mention that in his survey, Keil [3] considers the general problem of partitioning/covering a polygon by different kind of geometrical shapes, particularly rectangles and triangles.

3.5.2 *Reduction to the Case Where the Union of Blocks Can Be Considered as a 3D-Polyhedron*

When solving our problem, a possible approach is to test first whether the union of the blocks can be considered as a 3D-polyhedron. If the answer is "yes," we can apply one of the various heuristics developed to partition a 3D-polyhedron. When minimizing the number of resulting blocks, we might however have to slightly adapt those heuristics in order to guarantee that all these blocks have a minimum size of $8 \times 8 \times 8$.

If the union of the given blocks cannot be considered as a 3D-polyhedron, two approaches could be entertained:

- Generalize the aforementioned heuristics to the more general case where the volume cannot be considered a 3D-polyhedron; or
- Partition first the set of blocks into subsets such that for each subset, the union of the blocks in that subset can be considered as a 3D-polyhedron. The aforementioned heuristics would then be applied separately to the blocks of each subset. In this second approach, an important question to address is of course the choice of the algorithm for computing the initial partition into subsets.

In the next section we propose a much simpler heuristic using the "greedy" paradigm.

Input: List L of blocks.

Repeat the following steps as long as the list L contains at least one mergeable block.

Step 1. Choose a mergeable block (let us call it A).
Step 2. Among the blocks that can be merged with A, select a block B.
Step 3. Merge blocks A and B, yielding a new block C.
Step 4. Replace blocks A and B by block C in the list L.

Fig. 3.10 A simple heuristic

3.5.3 *A Simple Greedy Heuristic*

In this section we propose a simple greedy heuristic for the general case, i.e., without assuming that the volume defined by the blocks can be considered as a 3D-polyhedron. This heuristic merges two blocks at each iteration until it is not possible to do so any more.

3.5.3.1 Basic Description

The heuristic takes as input a list L of blocks and outputs a new list of blocks. A block is said to be *mergeable* if there exists another block in the list with which it can be merged in such a way that the result is a (rectangular) block. The heuristic is described in Fig. 3.10.

In order to define completely the heuristic, we have to explain how to select a block in Steps 1 and 2 (section 3.5.3.2), how to detect whether two blocks are mergeable (section 3.5.3.4), and how to merge them (section 3.5.3.5).

3.5.3.2 Selection of Blocks

We considered the following strategies for the selection of a block in Steps 1 and 2 of the heuristic.

- Select the block with the smallest volume.
- Select the block with the smallest size along the three directions.
- Select the block that is the most difficult to merge, in the sense of having the smallest number of blocks with which it can be merged.

3.5.3.3 Representation of a Block and of the Intersection of 2 Blocks

In order to explain how to test whether two blocks are mergeable and how to merge two blocks, we first have to describe the format of the data.

```
=====Block   Imax    Jmax    Kmax      Total surfaces====================
        1     17      17      49               7
  ---BCtype  Ista    Jsta    Ksta    Iend    Jend    Kend    icomp---
      FAR       1       1       1       1      17      49       0
      CON      17       1       1      17      17      25       0
       65       1       1       1       1      17      25       1       2
      CON      17       1      25      17      17      49       0
       67       1       1       1       1      17      25       1       2
      FAR       1       1       1      17       1      49       0
      CON       1      17       1      17      17      49       0
        3       1       1       1      17       1      49       2       0
      SYM       1       1       1      17      17       1     900
      CON       1       1      49      17      17      49       0
        5       1       1       1      17      17       1       0       1
=====Block   Imax    Jmax    Kmax      Total surfaces====================
        2     17      17      49               7
  ---BCtype  Ista    Jsta    Ksta    Iend    Jend    Kend    icomp---
      CON       1       1       1       1      17      25       0
       70      25       1       1      25      17      25       1       2
```

Fig. 3.11 Example of input file

Each block has its own 3-dimensional basis $(O, \vec{\imath}, \vec{\jmath}, \vec{k})$, whose origin corresponds to one of its vertices. The origin has coordinates $(1, 1, 1)$ with respect to this basis while the opposite vertex has coordinates $(i_{\max}, j_{\max}, k_{\max})$. The intersection of the two blocks B_1 and B_2 can be viewed as a subface F_1 of B_1 **and** as a subface F_2 of B_2. Those subfaces are two-dimensional rectangles. Each subface is defined in the basis of its respective block.

For each block, we are given the list of intersecting subfaces. For each subface of the list, the coordinates of two opposite corners defining the rectangle are given first in the basis of the block. If the intersecting subface corresponds to an intersection with another block, the identifier of that block is given, followed by the description of the rectangle in the basis of that block.

Example 1. Figure 3.11 shows part of a data file.

The second line tells us that block 1 has 7 intersecting subfaces. Subfaces that correspond to an intersection with another block are labeled "CON."

Lines 5 and 6 tell us that the second subface results from the intersection of block 1 with block 65. The fifth line gives the definition of the subface in the basis corresponding to block 1. We see in particular that the subface is orthogonal to the vector $\vec{\imath}$ (since $istart = iend = 17$). The rectangle is defined by the two opposite vertices $(17, 1, 1)$ and $(17, 17, 25)$.

Line 6 gives the definition of the subface in the basis corresponding to block 65. In this basis the subface is spanned by the vectors $\vec{\jmath}'$ and $\vec{k}'$ (the vectors defining the basis are noted differently to emphasize that they may be different). The rectangle is defined by the two opposite vertices $(1, 1, 1)$ and $(1, 17, 25)$.

3.5.3.4 Testing if Two Blocks Are Mergeable

In this section we assume that there is at most one intersecting surface for each given pair of blocks (exactly one surface if the blocks intersect, 0 otherwise). Under this assumption two blocks B_1 and B_2 are mergeable if the following conditions are satisfied.

1. There is a surface S_{12} in the list of surfaces of B_1 whose block identifier is B_2 and this surface coincides with the appropriate face of B_1.
2. There is a surface S_{21} in the list of surfaces of B_2 whose block identifier is B_1 and this surface coincides with the appropriate face of B_2.

Example 2. Consider again the example described in Fig. 3.11. Blocks 1 and 65 are not mergeable (merging them would not result in a block) because the size of block 1 along the $\vec{k}$ axis is $k_{\max} = 49$ whereas $k_{\text{end}} = 25$ holds. Blocks 1 and 3, however, can be merged, although the figure does not give all the information necessary to reach this conclusion.

3.5.3.5 Merging Two Blocks

The operation of merging two blocks is too complicated to be described fully here. Basically, the operation involves a change of basis of one of the blocks so that the vectors of the two bases coincide whenever the 2 pairs of opposite corners of the intersecting surface are made to coincide. In particular this forces us to rewrite in the new basis all the intersecting surfaces of the block whose basis has been changed. Once the two bases are made compatible, the origin of one of the two bases does not belong to the intersecting surface, whereas the origin of the other basis does. We choose the basis of the former block as the basis of the merged block. Then it is not difficult to compute the dimensions of the merged block. The last step consists of constructing the intersecting surfaces of the merged block with the other blocks.

3.5.4 *Computational Results*

We considered three instances among those provided by Bombardier and assumed that each instance described the set of blocks allocated to a specific processor. Recall that the heuristic first selects a mergeable block A, then chooses a block B among the blocks that can be merged with A (see Section 3.5.3.1). The four strategies below were considered for the selection of Blocks A and B.

Strategy 1: Each of Blocks A and B is chosen as the block that is the most difficult to merge, i.e., the block that has the smallest number of blocks

Table 3.4 Numerical results for instance dlrf4_wb_eu_74b

Strategy	Cardinality	Size	Volume
-	74	[16; 64]	[4096; 18432]
Strategy 1	28	[16; 192]	[16384; 92160]
Strategy 2	25	[16; 160]	[16384; 92160]
Strategy 3	30	[16; 160]	[16384; 61440]
Strategy 4	34	[16; 160]	[12288; 61440]

Table 3.5 Numerical results for instance fansc_coarse

Strategy	Cardinality	Size	Volume
-	11	[24; 84]	[36864; 276480]
Strategy 1	5	[24; 168]	[36864; 645120]
Strategy 2	5	[24; 168]	[110592; 645120]
Strategy 3	5	[24; 168]	[110592; 645120]
Strategy 4	5	[24; 216]	[55296; 829440]

Table 3.6 Numerical results for instance onera

Strategy	Cardinality	Size	Volume
-	32	[24; 40]	[24576; 30720]
Strategy 1	16	[24; 64]	[49152; 61440]
Strategy 2	7	[24; 256]	[49152; 196608]
Strategy 3	12	[24; 112]	[49152; 86016]
Strategy 4	16	[32; 48]	[49152; 61440]

with which it can be merged. Ties are broken by choosing the block with the smallest volume.

Strategy 2: We choose for Block A a block that is the most difficult to merge, with ties broken by choosing the one with the smallest volume. Among the blocks that can be merged with Block A, we choose for Block B the one with the smallest volume, with ties broken by choosing the one that is the most difficult to merge.

Strategy 3: We transpose the two selection criteria that were considered in Strategy 2: we choose for Block A a block with the smallest volume, with ties broken by choosing the block that is the most difficult to merge. Then among the blocks that can be merged with Block A, we choose for Block B a block that is the most difficult to merge, with ties broken by considering the volume.

Strategy 4: Each of Blocks A and B is chosen as the block with the smallest volume. Ties are broken by choosing the block that is the most difficult to merge.

The results for the 3 instances are given in Tables 3.4, 3.5, and 3.6, respectively. Each line in the table corresponds to a partition, with the first line describing the initial partition. For each partition we give its cardinality, the smallest and largest sizes over the three directions and over all blocks of the partition, and the smallest and largest volumes over all blocks of the partition.

These limited computational experiments are sufficient to demonstrate that different strategies give quite different partitions. Here it seems that the best

strategy is Strategy 2, with respect to the minimization of the cardinality, the maximization of the minimum size, and the maximization of the minimum volume. Recall that Strategy 2 selects Block A as the most difficult to merge, and Block B as the one with the smallest volume.

A lot of work remains to be done. In particular we would like

- to model the partitioning problem in a more formal way, i.e., as a multi-objective programming problem in which the different objective functions are well defined;
- to explore additional strategies for the selection of the blocks to be merged in the greedy heuristic; and
- to embed the greedy heuristic in a meta-heuristic like tabu search, VNS, ...

.

References

[1] V.J. Dielissen and A. Kaldewaij. "Rectangular partition is polynomial in two dimensions but NP-complete in three". *Information Processing Letters* 38:1 (1991), 1–6.

[2] A. Jain, S. Sahni, J. Palta, and J. Dempsey. "Partitioning 3D phantoms into homogeneous cuboids". *International Journal of Foundations of Computer Science* 14:5 (2003), 905–931.

[3] J.M. Keil. "Polygon decomposition". *Handbook of Computational Geometry*. Ed. by J.-R. Sack and J. Urrutia. Amsterdam: Elsevier, 2000. Chap. 11, 491–518.

[4] A. Lingas, R. Pinter, R. Rivest, and A. Shamir. "Minimum edge length partitioning of rectilinear polygons". *Proceedings of the 20th Annual Allerton Conference on Communication, Control and Computing*. Monticello, IL: Allerton House, 1982, 53–63.

[5] W. Lipski, E. Lodi, F. Luccio, C. Mugnai, and L. Pagli. "On two-dimensional data organization. II". *Fundamenta Informaticae* 2 (1979), 245–260.

[6] J. O'Rourke and G. Tewari. "The structure of optimal partitions of orthogonal polygons into fat rectangles". *Computational Geometry* 28:1 (2004), 49–71.

[7] K. Sermeus, É. Laurendeau, and F. Parpia. "Parallelization and performance optimization of Bombardier multiblock structured Navier–Stokes solver on IBM eserver cluster 1600". *45th AIAA Aerospace Sciences Meeting and Exhibit*. (Reno, NV, 2007). Reston, VA: AIAA, 2007, 2007–1109.

Appendix

```
# BLOCKS
      5
=====Block   Imax    Jmax    Kmax    Total surfaces=================
     1       65        65        97         6
  ---BCtype      Ista    Jsta    Ksta    Iend    Jend    Kend    icomp   iiconnect---
     CON           1       65       1      65      65      97       0
       2          97        1       1     161       1      97              2  0
     SYM           1        1       1       1      65      97       0
     SYM          65        1       1      65      65      97       0
     SYM           1        1       1      65       1      97       0
     WAL           1        1       1      65      65       1       100
     FAR           1        1      97      65      65      97       0
=====Block   Imax    Jmax    Kmax    Total surfaces=================
     2      257       65        97        10
  ---BCtype      Ista    Jsta    Ksta    Iend    Jend    Kend    icomp   iiconnect---
     CON          97        1       1     161       1      97       0
       1           1       65       1      65      65      97              2  0
     CON           1        1       1       1      65      97       0
       2         257        1       1     257      65      97              1  2
     CON         257        1       1     257      65      97       0
       2           1        1       1       1      65      97              1  2
     CON           1        1       1      33       1      97       0
       5          33        1       1       1       1      97              2  0
     CON         225        1       1     257       1      97       0
       5          65        1       1      33       1      97              2  0
     CON           1       65       1     257      65      97       0
       3           1        1       1     257       1      97              2  0
     SYM          33        1       1      97       1      97       0
     SYM         161        1       1     225       1      97       0
     WAL           1        1       1     257      65       1       100
     FAR           1        1      97     257      65      97       0
=====Block   Imax    Jmax    Kmax    Total surfaces=================
     3      257       33        97         6
  ---BCtype      Ista    Jsta    Ksta    Iend    Jend    Kend    icomp   iiconnect---
     CON           1        1       1     257       1      97       0
       2           1       65       1     257      65      97              2  0
     CON           1        1       1       1      33      97       0
       3         257        1       1     257      33      97              1  2
     CON         257        1       1     257      33      97       0
       3           1        1       1       1      33      97              1  2
     CON           1       33       1     257      33      97       0
       4           1        1       1     257       1      97              2  0
     WAL           1        1       1     257      33       1       1
     FAR           1        1      97     257      33      97       0
=====Block   Imax    Jmax    Kmax    Total surfaces=================
     4      257       65        97         7
  ---BCtype      Ista    Jsta    Ksta    Iend    Jend    Kend    icomp   iiconnect---
     CON           1        1       1     257       1      97       0
       3           1       33       1     257      33      97              2  0
     CON           1        1       1       1      65      97       0
       4         257        1       1     257      65      97              1  2
     CON         257        1       1     257      65      97       0
       4           1        1       1       1      65      97              1  2
     CON           1       65       1     129      65      97       0
       4         257       65       1     129      65      97              2  0
     CON         129       65       1     257      65      97       0
       4         129       65       1       1      65      97              2  0
     WAL           1        1       1     257      65       1       1
     FAR           1        1      97     257      65      97       0
=====Block   Imax    Jmax    Kmax    Total surfaces=================
     5       65       129        97         7
  ---BCtype      Ista    Jsta    Ksta    Iend    Jend    Kend    icomp   iiconnect---
     CON           1        1       1      33       1      97       0
       2          33        1       1       1       1      97              2  0
     CON          33        1       1      65       1      97       0
       2         257        1       1     225       1      97              2  0
     SYM           1        1       1       1     129      97       0
     SYM          65        1       1      65     129      97       0
     SYM           1      129       1      65     129      97       0
     WAL           1        1       1      65     129       1       100
     FAR           1        1      97      65     129      97       0
```

Code 3.1 Example of connectivity file

4
Visualizing Huge Plots on the Web

Mikola Lysenko and Perouz Taslakian

Abstract We consider a small subset of problems that arise in *Plotly* — an online graphing tool that renders a visualization of a set of data points inside a browser, where computational power is limited and memory is scarce. We show techniques that improve the total number of points that can be visualized in an interactive scatter plot, and give a few possible approaches for solving the same problem in the case of line plots.

4.1 Introduction

The best data visualizations illustrate hidden information and structure contained in a data set. As access to large data sets has grown, so has the need for interactive and scalable solutions connecting computer science, mathematics, and industry.

Plotly is an online data visualization tool designed to render graphs quickly in web browsers. The user base of the tool consists of a wide variety of clients from industry and scientific laboratories. As these clients have gained access to larger data sets in recent years, demand to enable the rendering of huge two-dimensional plots has grown. There are various successful methods for rendering such huge point clouds efficiently. Not many studies, however, have addressed this problem when the rendering needs to be done within a browser (see for example [3]). One of the main challenges of plotting a large number of points within a browser is doing so efficiently while using a limited amount of memory (typically 1 GB). With the current online Plotly interface, the maximum number of points that one can plot is about 100,000 points. The

Mikola Lysenko
Plotly and University of Wisconsin — Madison

Perouz Taslakian
McGill University

primary goal of this report is to find an executable solution that improves the efficiency of point rendering and allows the plotting of data sets of up to 1 million elements, while preserving interactive and exportable features that the clients expect.

At the lowest level, rendering is handled by the GPU (Graphics Processor Unit), which is very efficient in processing a huge number of points, especially when many end up outside the viewport. The bottleneck in GPU processing, however, lies in the rasterization step, i.e., when a large number of points are mapped to the same screen pixel. *WebGL* is a low-level graphics API that uses GPUs directly for extremely fast rendering. It is based on OpenGL but designed for the web. To render an image in WebGL, functions send data to your GPU for processing and images are drawn on top of a canvas element. Shapes are created by scripts that contain vertex and fragment shaders, which assign colours to each pixel contained in the shape. WebGL is fast and offers high-performance rendering, good interactivity, and excellent control.

In this report, we address the problem of rendering (using WebGL) a huge number of points (i.e., at least one million) within a browser (i.e., a context where memory is limited). In addition to drawing considerations, our solution requires thinking about how plots are stored on the server and how stored data is downloaded to the browser. Specifically, we require an efficient way to store and retrieve data on the server side based on a plot viewport. Because of the potential bottleneck at the rasterization step of the GPU, the problem is the following: given a viewport, choose quickly (whether on the server or the client) the points that will be sent to the GPU for rendering so that the plot looks effectively just as it would look if one plotted everything. The client can then query the server on zoom/pan events to update the data.

It is also essential that existing plot features are maintained. In particular, advanced styling features (such as custom glyphs, colors, sizes, borders, etc.) must be preserved, and users must be able to highlight individual points (this is done by hovering the mouse over a point in the plot).

We concentrate on two types of two-dimensional plots: scatter plots and line plots. Our goal is to render over one million data points n by allowing at most $O\big(n \operatorname{polylog}(n)\big)$ steps during preprocessing and using $O(n)$ amount of working memory. In addition, the algorithms we develop must be simple enough to allow an implementation in a reasonable amount of time (i.e., the implementation must be cost-effective).

Given a set of n vectors in 2D and a shape S (which we call the *marker*), a *scatter plot* (see Figure 4.1) is the image formed by the union of n copies of S translated by each vector. A *line plot* (see Figure 4.2) is a piecewise linear curve (i.e., a polyline).

Let $P = \{p_0, p_1, \ldots, p_n\}$ be a set of primitives (i.e., objects represented as sets of points) with $p_i \in \mathbb{R}^2$ for all $i = 0, 1, \ldots, n$. Let $s_0 > s_1 > \cdots > s_k$ be a set of scales (in our case, pixel sizes). The *cover order* of P is the filtration

$$F_0 \subseteq F_1 \subseteq \cdots \subseteq F_k = P,$$

such that $\bigcup P \subseteq \bigcup(F_j \oplus C_{s_j})$ for each $j = 0, 1, \ldots k$, where C_{s_j} is a circle of radius s_j. The *level* of a primitive is defined as follows: $\text{level}(p_i) = \min_{p_i \in F_j} j$. In other words, given a pixel size s_j, we would like to determine a subset of objects $P' \subseteq P$ such that every primitive overlaps with one or more primitives of P' dilated by the circle of radius s_j (ideally, we want the minimum number of objects that would cover all others). This means that for a fixed zoom level, we only need to render the primitives in P' since all the other primitives are "hidden." Therefore, given a set of points/line segments, our problem can be reduced to the problem of finding a cover order of these primitives for a given zoom level (see Figure 4.3).

For pointsets in 2D, a cover order can be easily computed by using a spatial data structure called *quadtree*; computing a cover order is not so evident in the case of line plots, however.

In what follows, we describe a method that will allow us to render and process a scatter plot of 100 million points within a browser. We then discuss the challenges of using the same technique for rendering scatter plots and propose a few alternatives.

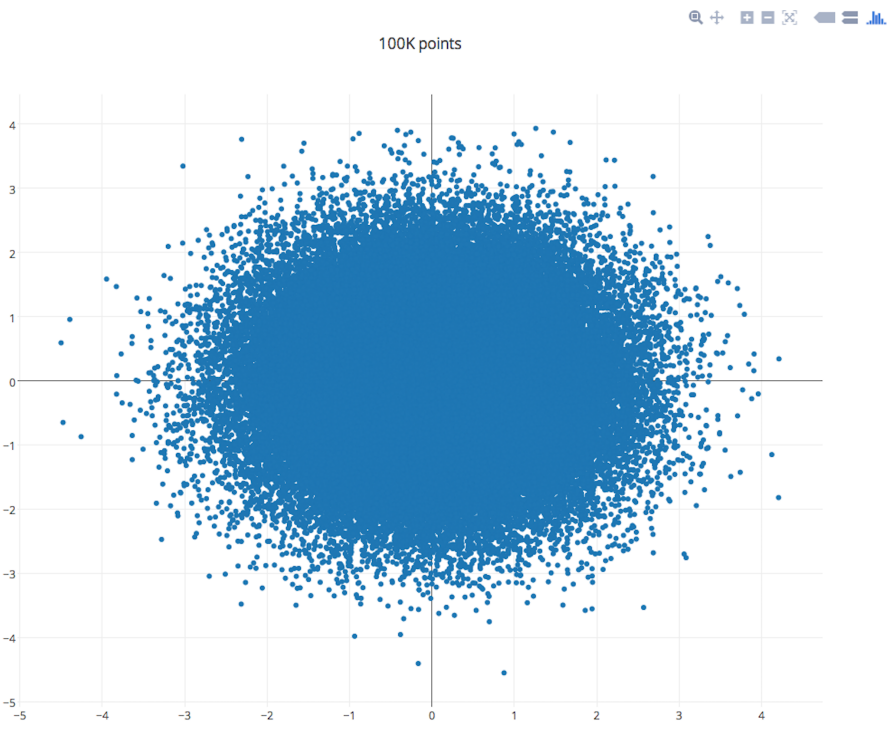

Fig. 4.1 A scatter plot of 100,000 points in 2D.

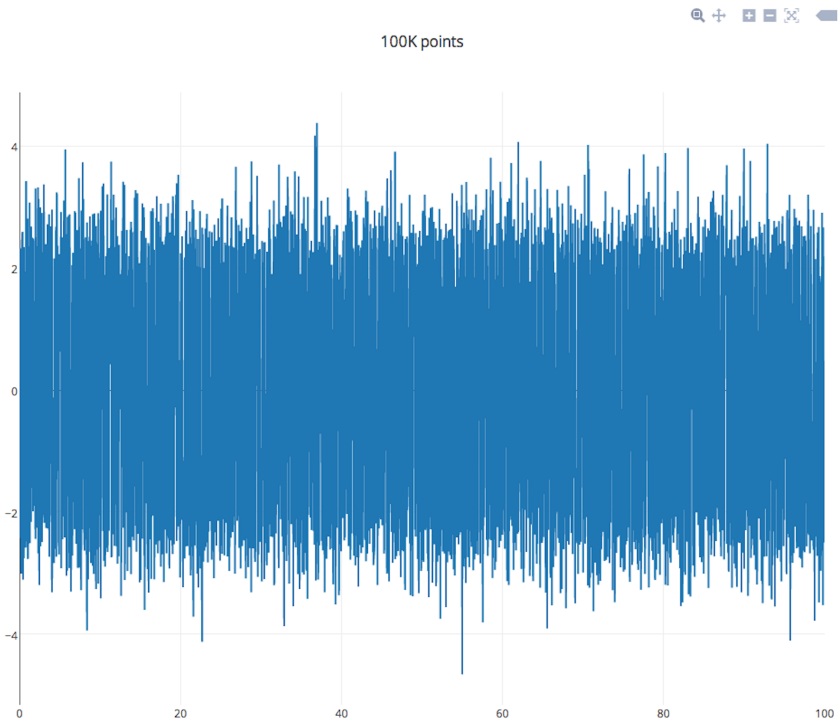

Fig. 4.2 A line plot of 100,000 points in 2D.

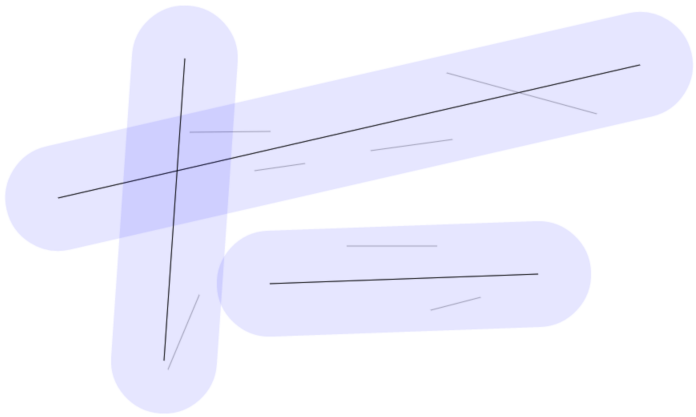

Fig. 4.3 The three thick line segments constitute a cover for all the segments for a given zoom level. Only these three segments need to be rendered by the GPU.

4.2 Progressive Loading of 2D Scatter Plots

In this section we describe an algorithm that allows us to render 100 million points. The key idea behind the algorithm is the use of the quadtree data structure and the building of this data structure in a way that minimizes the use of working memory.

A *quadtree* of a set of points in the plane is a geometric data structure that is constructed by recursively splitting the area (usually the square bounding box of the input pointset) into four equal-sized squares until every square contains at most one point [1].

Let A be a set of n points in the plane, and let z be a zoom level (which is a function of the current pixel size and screen resolution/size).

Preprocessing

Once the points A are loaded in the client (we will assume A is an array of points), we preprocess them as follows.

1. Construct a quadtree of the pointset A using a depth-first traversal of the points and store the level of each point in the quadtree in a new array L. Store the points in array Q. Thus the level of point $Q[i]$ is stored as $L[i]$.
2. Sort the points of Q in increasing order of level and x-coordinate using an in-place sorting algorithm (such as Quicksort). The maximum value stored in array L will be equal to the height of the quadtree.

The ordering of the points by their level in the quadtree gives us a cover order for the points of the scatter plot. For a given zoom level z, this ordering of points by level gives us a way to choose the points that will be rendered (the other points being hidden behind the rendered points). Consider a square area of the quadtree, which corresponds to a node p at some level ℓ in the tree. If the zoom level of the rendering matches with ℓ, then the quadtree structure tells us that all nodes in that single square area will be mapped on top of one another on the screen in the final rendering. Therefore, instead of rendering all the points (p and its descendants), we can instead render only point p.

Rendering

Once array Q is constructed and sorted, we can flush array L and send Q to the GPU. To render for a given level, we request the GPU to draw some superset of the points that are visible on the screen and are not hidden behind others.

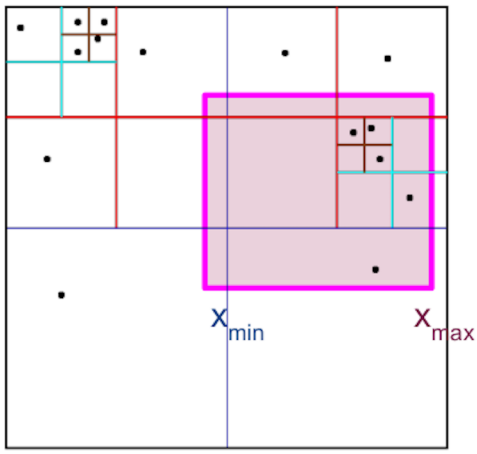

Fig. 4.4 The partitioning of the points into quadtree partitions. The pink/bold rectangle represents the area that will appear on the screen.

1. Compute the size of a pixel in the data coordinates to get the current zoom level z.
2. Compute the $x_{\min}$ and $x_{\max}$ of the screen.
3. Starting at level z, for each level at most z do the following.
   a. Find the predecessor x_p of $x_{\min}$ and successor x_s of $x_{\max}$.
   b. Ask the GPU to draw the points whose x-coordinates are between x_p and x_s.

Conceptually, given a zoom level z, we request the GPU to draw all the points that have level *at most* z, thus ignoring all points that are deeper than z in the quadtree. We further trim the number of rendered points by excluding those that are outside the vertical strip enclosing the sides of the screen (ee Figure 4.4).

Analysis

A quadtree of depth d storing a set of n points has $O\big((d+1)n\big)$ nodes and can be constructed in $O\big((d+1)n\big)$ time. In general, the depth of a quadtree of a set of points in the plane is at most $\log(s/c) + \frac{3}{2}$, where s is the length of one the edges of the bounding box and c is the distance between the closest pair of points. Thus the depth of a quadtree can be arbitrarily bad.

Preprocessing may take a long time if the quadtree ends up having height n. This happens when, for example, the input consists of a big cluster of points that are far from the rest of the points. To avoid such bad cases, we introduce a slight modification to the way we construct the quadtree. After any split of

the current area into four quadrants, we check every quadrant to see whether it contains more than 90% of the points. If such a quadrant is found, then we split the point cluster arbitrarily into two equal sets and construct a quadtree for each one individually. Doing this will preserve the level information for each point, which is all that we need to render effectively, while at the same time decreasing the height of the tree.

The algorithm described in this section has made it possible to render 100 million points within a browser while maintaining reasonable interactivity.

A Note on Transparent Markers of the Same Colour and Shape

So far we have assumed that the markers we are rendering are of solid colour. This assumption made it possible to replace all the points that are stacked on top of one another with a single point on the screen. If the markers are partially transparent, however, replacing overlapping points with a single, transparent marker will not work, because we need to blend the colours of all the markers that are hidden. Blending different colours assumes an ordering of the points, which makes the problem even more complex. Here we propose a simple fix for the case of partially transparent markers that have the same colour and shape. To blend colours properly, we need to know the number of points hidden behind the point we wish to render. We can save this information while creating the quadtree in the preprocessing step — for each node of the quadtree, we keep track of the number of descendants in its subtree and store this number in a separate array. During the rendering process, we can then use this information to blend the colours appropriately.

4.3 Displaying 2D Line Plots

Computing a (reasonable) cover order for line plots is a much more challenging task than for scatter plots. In the line plot setting, the use of quadtrees is not efficient because a segment may hit $O(2^h)$ boxes in level h, resulting in a huge quadtree.

In the line plot setting, we want to build covers inductively, i.e., build $F_j \setminus F_{j-1}$ in order of increasing j. In other words, we want the smallest set of line segments covering $P \setminus F_{j-1}$ at scale s_j.

Ideas

1. *Set Cover Approximation.* The cover ordering problem for line segments can be reduced to the problem of Set Cover (which is NP-complete [2]). This allows us to use a known approximation algorithm for finding set covers. One such algorithm has an approximation factor of $\log n$, with a running time of $O(n^3)$.
2. *Greedy Longest Segment.* In this approach, we sort the segments and process them by decreasing order of length. The running time is $O(n^2)$.
3. *Divide-and-Conquer.* We recursively split the segments (arbitrarily) into two equal groups and process the groups individually. The running time is $O(n \log n)$.

Acknowledgements We would like to thank the participants and organizers of the Sixth Montreal Industrial Problem Solving Workshop (Montréal 2015) where the problem was first posed.

References

[1] R.A. Finkel and J.L. Bentley. "Quad trees a data structure for retrieval on composite keys". *Acta Informatica* 4:1 (1974), 1–9.

[2] M.R. Garey and D.S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. New York: W H. Freeman, 1979.

[3] M. Schütz. "Rendering Large Point Clouds in Web Browsers". *Proceedings of the 19th Central European Seminar on Computer Graphics*. (Smolenice, 2015). Ed. by M. Wimmer, J. Hladůvka, and M. Ilčík. Vienna: Vienna University of Technology, Institute of Computer Graphics and Algorithms, 2015.

5
Lifetime Value in the Bank Industry

Maciej Augustyniak, Wyean Chan, Jean-François Forest-Désaulniers, Maïkel Geagea, Ryan Halabi, Cody Hyndman, Francis Lafontaine, Huimei Li, Kevin Luk, Hervé Mensah, Manuel Morales, Younes Ommane, Raphaël Ramora, Seyedhossein Rezaeilalami, Olivier Trottier, Shohre Zehtabian, and Farshid Zoghalchi

5.1 Introduction

5.1.1 *Context*

In today's hyper-competitive banking market, acquiring and retaining profitable customers is more challenging than ever. Customer demographics, buying behaviour, and needs are evolving rapidly. Banks now need a 360-degree view of each customer in order to target the right products, cross-sell and up-sell, and adapt to customers' changing needs.

Because of these challenges the National Bank of Canada seeks to understand its Customer Lifetime Value (CLV) at an individual customer level,

Maciej Augustyniak · Francis Lafontaine · Huimei Li · Manuel Morales · Raphaël Ramora · Seyedhossein Rezaeilalami
Dép. de mathématiques et de statistique, Université de Montréal

Wyean Chan · Shohre Zehtabian
DIRO, Université de Montréal

Jean-François Forest-Désaulniers
Département de mathématiques, UQAM

Maïkel Geagea GERAD, Université de Montréal · Ryan Halabi
Department of Mathematics, University of California at Davis

Cody Hyndman
Dept. of Mathematics & Statistics, Concordia University

Kevin Luk · Farshid Zoghalchi
Department of Mathematics, University of Toronto

Hervé Mensah · Olivier Trottier
Banque Nationale du Canada

Younes Ommane
Département de mathématiques, Université de Sherbrooke

across all their financial products (deposits, loans, investments, credit cards, etc.). The CLV can be defined as the revenue that a client is expected to generate for the bank throughout his or her involvement with the bank.

The Client Intelligence & Modelization Team of the National Bank of Canada has been assigned the task of developing methods that evaluate precisely the projected cash flows for each client. Although the CLV is a simple concept, it is very difficult to implement in a complex business context.

5.1.2 *Problem*

One of the tasks assigned to this team is to analyze and predict a client's varied behaviours and infer from them a value representing the revenue generated for the bank by the client over his or her involvement with the bank. The team's goal is therefore to propose a method or algorithm for estimating the CLV. The CLV will then help the National Bank define, understand, and predict in a novel way the relationship between a given client and the National Bank. For example, one intended use of the model is to predict which group of clients will respond positively to a marketing offer.

In order to develop a method for computing the CLV that is specific to the banking sector, one must take into account the notions of acquisition and disposal for diverse banking products and services, their volume, usage, and profitability, as well as the clients' characteristics (such as geographical, demographic, or market data). Obviously the profiles, products, or services of the Bank's clients vary greatly; so do their behaviours and expectations.

5.1.3 *Objectives*

The task was divided into three objectives.

1. Identify and segment the variables that are relevant for measuring the CLV. The variables that are available in the data can be separated into three groups.
   (i) Products held by the client: savings (e.g, savings account, mutual funds, deposit certificates), loans (e.g., mortgage, line of credit, credit card), transactions (e.g., checking account);
   (ii) Demographics: age, gender, marital status, income, zip code region;
   (iii) Client profile and behaviour: credit score, average balance in checking account, number of transactions.
2. Develop a model that measures the CLV as a function of these variables.
3. Develop a predictive model to help determine future marketing strategies for customers based on estimated CLVs.

5.2 Description of the Data

To perform data mining or data analysis is it critical to have good quality data. A perfect data set would contain a large quantity of observations (or inputs), contain all relevant variables or attributes (ideally all observable variables), and be clean (i.e., it would contain neither input errors nor missing values). In practice it is hard to get perfect data sets.

This section describes two data sets that the National Bank made available to the team members: a commercial data set and a retail data set.

5.2.1 *Commercial Data Set*

The commercial data set is generated from the monthly statements of the Bank commercial clients between 2013 and 2015. The original data set was used by the bank to produce a preliminary study of the CLV using discrete-time Markov chains. There are three classes of products: savings, loans, and transactions. Each product class consists of several bank products: savings consists of 3 products, loans of 5 products, and transactions of 16 products. Hence there are 24 products in total. A client can possess several products, including several products of the same class. A binary vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_{24})$ is used to represent the products owned by a client at the end of the month, where element x_i equals 1 if that client owns product i and 0 otherwise. Elements x_1 to x_3 correspond to the savings product types, x_4 to x_8 to the loans product types, and $x_9, \ldots, x_{24}$ to the transactions product types. Vector $\boldsymbol{x}$ represents the *state* of a client at the end of the month. For a person who is not yet a client the vector is $(0, 0, \ldots, 0)$.

We received a truncated data set for this workshop. We note that the data does not include any client-specific information (e.g., age, gender, etc.). The following variables are included in the data set:

1. VECT_IN: The set of observed states $\boldsymbol{x}$, say $\mathcal{X}$. This set contains 1,479 states.
2. NB_CLIENTS_VECT_IN: The total number of times state $\boldsymbol{x}$ was observed over the relevant time period for all clients.
3. VALUE_MOD: The average customer value generated within a given month for each state $\boldsymbol{x}$. This value is the sum of owned products averaged over all observations of $\boldsymbol{x}$. We note that the data set does not include the customer value for every combination of product and client.

Table 5.1 shows a sample of the data. For example, the first state represents clients that own(ed) 6 particular products and this state has been observed 27,240 times. A state can be observed multiple times for the same client, and the count will be incremented at each observation. There is a total of 268,293

observations over all $\boldsymbol{x} \in \mathcal{X}$, where 268,293 is the sum of the elements in column NB_CLIENTS_VECT_IN.

To simplify data manipulation, we created an alternative data set where column VECT_IN (or vector $\boldsymbol{x}$) is subdivided into 24 columns with headings from x_1 to x_{24}. Table 5.2 presents a sample of this alternative data set.

The commercial data set has the following limitations. It does not allow us to infer transition probabilities from one state to another and it does not specify the value corresponding to every product owned by the client.

5.2.2 *Retail Data Set*

The retail data set contains information on 4,831 retail clients (individuals) at the end of a particular month and 6,130 attributes are included for each client (for a total of 4,831 rows and 6,130 columns). The data has been partially cleaned up and some attributes have been anonymized (i.e., removed or modified). With the help of experts from the Bank, we proceeded to clean and rearrange the data, which led us to a new data set with only 88 attributes per client. We now describe the retained attributes:

1. Demographic attributes:
   - Segmentations (types 1, 4, and 5) with the following possible modalities: age, region, savings balance, etc. (predefined by the Bank experts).
   - Individual demographic attributes (age, gender, marital status, geographical area, etc.).
2. Channel usage (internet, ATM, phone, etc.).
3. Credit score.
4. Products: the 16 most important products according to the Bank experts. We note that the three principal categories of products are transactional, financial, and investment.
   - Holding: A binary variable (1 if the client owns the product, 0 otherwise).

Table 5.1 Sample of data set 1 (for commercial clients).

VECT_IN	NB_CLIENTS_VECT_IN	VALUE_MOD
000001011010000001100000	27,240	189.2424
000001011010000001110000	21,855	204.9733

Table 5.2 Sample of the alternative table for data set 1.

x_1	x_2	...	x_{24}	NB_CLIENTS_VECT_IN	VALUE_MOD
0	0	...	0	27,240	189.2424
0	0	...	0	21,855	204.9733

- Tenure: The number of consecutive months during which the client owned the product.
- Balance: The ranked score of the balance value over all clients, which is a modified version of the real data.

5. Attributes related to the categories of products and services (also known as spheres) held by the client. These attributes take into account different possibles cases, namely the ones judged to be the most interesting for the Bank experts, such as: monospherical clients (transactional sphere, financial sphere, and investment sphere), and the shiftings from one sphere to another (e.g., mono-transactional to mono-investment, and mono-financial to mono-investment, etc.).
6. The EFT (Electronic Fund Transfers): the total amount for the current month.
7. The AML (Anti-Monetary Laundering): the total amount for the current month.
8. Fees (generated by all products):

   - Total sum of fees in current month.
   - The average amount of fees paid over a period of exactly 6 months (respectively 12 months). If the period is less than 6 months, the value is 0.
   - The maximum monthly fee amount during the last 6 months (respectively 12 months).
   - The maximum monthly fee frequency during the last 6 months (respectively 12 months).

To respect confidentiality, all amounts in EFT, AML, and Fees were replaced by ranked scores.

5.3 Literature Review

The problem proposed by the Client Intelligence & Modelization Team of the National Bank of Canada is focused on an indicator called **Customer Lifetime Value (CLV)**. This is the value to the Bank of the revenue generated by a customer over his period of involvement with the Bank. The team wishes to develop a model or algorithm for estimating and predicting the CLV in order to make better decisions in marketing and customer relationship management. A preliminary review of the academic literature found several studies focusing on CLV in the retail banking industry. The approaches of [1–3] informed our analysis and approach to modelling the CLV.

Working with a German bank, Haenlein et al. [3] proposed a customer valuation model based on a combination of Markov chains and classification and regression trees (CART). The motivation for developing a CLV model

provided by Haenlein et al. [3] is the consolidation of the European banking sector and the resultant need, during mergers and acquisitions, to evaluate the business being acquired and enable the acquiring company to manage the combined business more efficiently and profitably.

The features of the CLV model presented in Haenlein et al. [3] that are relevant for our problem include the ability to handle discrete one-off transactions that occur infrequently and continuous revenue streams. The focus of Haenlein et al. [3] is on providing an easy-to-implement parsimonious model based on homogeneous customer segments. The model depends on four main factors: age, demographics and lifestyle, type and intensity of product usage, and activity level. Each of these factors, or profitability drivers, is measured by multiple indicators that are used as predictor variables of contribution to profit.

In examining the impact of the potential profitability drivers as predictors of profit contribution (or contribution margin) to CLV, Haenlein et al. [3] use CART analysis to cluster the client base into several homogeneous subgroups. In the second step each homogeneous subgroup was considered a state of a certain Markov chain. In that chain customers move from state to state over time with transition probabilities estimated by counting the number of customers moving between two states and dividing by the total number of customers. The calculation of the CLV is accomplished by backward-induction on the customer age group, which was used to segment the customer base, and utilizing the transition probabilities and the sum of the state-dependent contribution margins. The summing over all states of the Markov chain for each age group and the discounting back one period allow the calculation, recursively backward in time, of the CLV of a client in a particular state at the start of the period under consideration.

Glady et al. [2] use the CLV as the basis for the modelling and prediction of a certain type of customer behaviour called "churn," where a customer's CLV is decreasing over time, in order to avoid losing profitable customers. In our project we could not utilize the CLV to drive marketing decisions, but it may be of interest to do so in the future. Classifying the factors having an impact on the CLV in order to attain marketing and profitability objectives, however, is a key step in the eventual implementation of a useful model.

Following earlier work Glady et al. [2] use the variable $x_{i,j,t}$ to represent the usage of the product j, during the time period t, by the customer i. In order to define customer churn the authors consider the model

$$x_{i,j,t+1} = \alpha_{i,j,t} x_{i,j,t},$$

where $\alpha_{i,j,t}$ is the slope of the product usage. A slope value greater than 1 represents growth, a value of 1 represents retention, and a value smaller than 1 represents churn. In the case of a business offering multiple products, a customer might be loyal for certain products or considered a "churner" for other products. Using the above notation, the CLV for customer i from the

period t to $t+h$ is defined as

$$CLV_{i,t} = \sum_{k=1}^{h} \sum_{j=1}^{q} \frac{1}{(1+r)^k} CF_{i,j,t+k}, \tag{5.1}$$

where $CF_{i,j,t}$ is the net cash flow yielded by the transaction on product j and $r > 0$ is a relevant discount rate for the time period $[t, t+1)$ (which may represent a day, a week, a month, or some other unit of time).

The definition of CLV given by equation (5.1) is forward looking and disregards sunk costs. If the average marginal profit per unit of product usage (for the jth product) is denoted by π_j, then the cash flow associated with a customer's utilization of the product is

$$CF_{i,j,t} = \pi_j x_{i,j,t}$$

and the CLV of customer i related to product j becomes

$$CLV_{i,j,t} = \sum_{k=1}^{h} \frac{\prod_{v=0}^{k-1} \alpha_{i,j,t+v}}{(1+r)^k} \pi_j x_{i,j,t}. \tag{5.2}$$

Our analysis was guided by the definitions and notation found in Glady et al. [2]. We did not want, however, to use the slope of the product usage (denoted by $\alpha_{i,j,t}$). Rather we wanted to model $x_{i,j,t}$ (or a vector $x_{i,t}$ representing all product usage for customer i) as a Markov chain.

Glady et al. [2] define a measure based on CLV (given by equation (5.2)) as follows:

$$MAP_{i,j,t} = CLV_{i,j,t}(\text{with action}) - CLV_{i,j,t}(\text{without action}).$$

In so doing they define churn in a novel way, in order to guide certain marketing decisions. Using data from a Belgian financial services company, Glady et al. [2] carry out a classification of customers as non-churners with the goal of avoiding misclassification based on a CLV-sensitive loss function. Various statistical tools and methods (including logistic regression, decision trees, and neural networks) were used and compared for classifying the customers.

Much of the existing research on CLV is focused on companies that offer only one type of product. The aim of Ekinci et al. [1] is to provide a model to guide future marketing decisions via MDP (Markov decision processes), in the case of banks that offer a variety of products. Also, in contrast to existing literature, Ekinci et al. [1] provide real-world applications with their proposed model. In particular they test their model using data from a Turkish bank.

The proposed model of [1] is as follows. First, the decision areas and the variables of CLV are determined via information obtained from both previous literature and in-depth interviews with banking experts. Using the variables obtained in this fashion, the authors proceed to build regression models

(ANN and LSE) to estimate future customer values. For their segmentation technique, they use a customer pyramid approach that clusters customers by different states. After segmentation, transition probabilities between the different states are determined and these form the elements of a Markov chain model. The final (and critical) step of the model of Ekinci et al. [1] is to use MDP to compute the maximum CLV for customers in each state.

The model is applied to the monthly data of 10000 customers at a Turkish bank. The 29 variables used for CLV measurement in Ekinci et al. [1] were determined via interviews with Turkish banking experts. The customers are clustered into four different states and the transitional probabilities for each state are computed. Applying the MDP process, Ekinci et al. [1] are able to determine a number of optimal strategies to move customers into more profitable states.

5.4 Methods Considered

5.4.1 *Self-Organizing Maps (SOMs)*

5.4.1.1 Non-Technical Overview

In this section, we report on our results on the success and applicability of self-organizing maps (SOMs) to customer segmentation in the retail banking context. The general aim of customer segmentation is to identify cohorts of customers that share similar demands, priorities, or characteristics. This will in turn shape the business marketing/development strategies in a more intelligent and targeted way, through a better understanding (by management) of the firm's most influential customer cohorts.

There are different statistical techniques employed for customer segmentation: techniques using a single discrete variable; k-means clustering; finite mixture modelling; and self-organizing maps. The focus of this project is on SOMs (self-organizing maps).

A self-organizing map is a form of unsupervised learning via neural networks. The training input of the algorithm is a set of vectors. Every vector represents a sample (in our context, a bank's customer) and its coordinates represent different attributes of that specific sample (customer's demographic information, products used by the customer, etc). The output will be a 2-dimensional arrangement of these samples having the property that vectors that are close (in terms of the Euclidean norm) correspond to points that are close to one another in the plane.

A simple example is the following: suppose we ask everyone attending a ceremony to stand up and start comparing attributes among themselves (e.g., gender, age, weight, salary, language). They will then move around and try to

stand near the persons that are closest to them (by taking the attributes into account). After some time we ask them to stop. This final arrangement will be an example of SOM output. If one, then, assesses a single attribute in the final arrangement, he can determine how influential that specific attribute is in the final clustering. For example, if we ask people to hold up a card indicating their age, one can see to what extent people within the same age category are close to one another. We have just illustrated the notion of heatmap in a SOM algorithm. In what follows we will focus on the commercial banking context.

5.4.1.2 Introduction

SOM (as a nonlinear principal component analysis method) is an unsupervised data visualization technique that can be used to represent high-dimensional data sets in low-dimensional (typically 2-dimensional) spaces. By using 2-dimensional visualization, SOM finds a pattern within the data by using their topology instead of their distance. For example, if two high-dimensional objects in data space are very similar, then their positions in a 2-dimensional space should be very similar as well.

SOM maps the data objects onto a grid of nodes. To train the data, it starts by assigning a codebook vector to every node. The codebook vector plays the role of a typical pattern corresponding to that node. A subset of the data is usually randomly assigned to all the nodes. Throughout the training process, the objects are randomly presented to the map. The node which is the most similar to the current training object is called the "winning node" and will be updated through the iterations to become more and more similar to the objects presented. The updated value of this winning node, which is a weighted average, is kept as the new object. The weight, called "learning rate,"

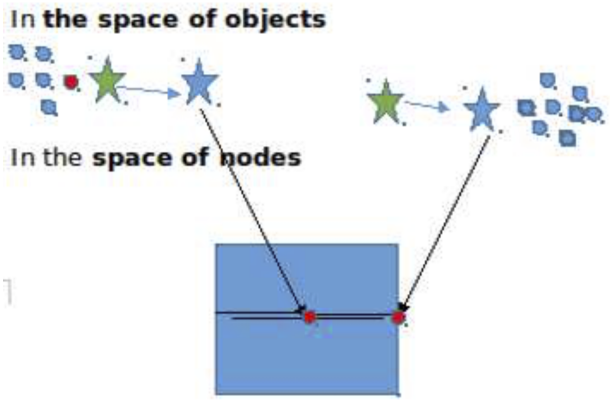

Fig. 5.1 Simple example with 2 nodes

is one of the training parameters of the SOM and has in general a small value (ex. 0.01). Over the training iterations, this value is increased (to a maximum of 0.05) in order for the map to converge. The SOM algorithm updates not only the winning node, but also its neighbours (i.e., nodes having similar codebook vectors). The size of the neighbourhood decreases throughout the algorithm and after a while the winning nodes are the only ones to be updated.

5.4.1.3 Visualization of the Data by SOM

The SOM vizualisation consists of a grid of multiple nodes. Each node vector has:

- a fixed position on the grid;
- a weight vector of the same dimension as the input space. For instance, if the input data, representing the bank's customers, has the variables "age," "sex," "language," and "geographical area," then each node on the grid will also have values for these variables;
- associated objects from the input data. Each object in the input space is "mapped" to a node on the grid. One node can represent several input objects.

As mentioned before, the main feature of SOMs is that the topological features of the original input data are preserved on the map. What this means is that similar input objects (customers) are positioned on the SOM grid at nodes that are close to one another. In the case of bank customers, similarity is defined in terms of the input variables such as age, sex, language, and geographical area. For example, all 55-year-old females who speak English and live in Montréal-Nord will be mapped to nodes in the same area of the grid. The French-speaking females will be mapped elsewhere, taking all variables into account. French-speaking males living in Montréal-Sud will be closer to French-speaking females living in Montréal-Sud than English-speaking males living in Montréal-Nord, and so on.

Typically SOM visualization is helped by "heatmaps." A heatmap shows the value distribution of a specific variable across the SOM. Imagine the SOM as a room full of customers. We ask each customer in the room to hold up a coloured card representing his geographical area. The result is a SOM heatmap. Customers with similar geographical areas would normally be aggregated in the same area. The same process can be repeated for age, sex, and language. Diverse heatmaps can help us explore the relationship between the input variables.

5.4.1.4 CLV Data Analysis by SOM

To analyze the retail data, we focus on 13 variables, whose list is given below.

age: age of the customer, in years
gender: gender of the customer (female/male)
language: preferred language of the customer (English/French)
EFT_Amount_1: the total amount in "Electronic Fund Transfers" during the last month
AML_Amount_1: the amount in NON-EFT transactions during the last month
FEE_Amount_1: the total fee amount during the last month
FEE_Number_1: the total fee number during the last month
FEE_Avg_A_Total_12: the average total amount during the last 12 months
FEE_Avg_A_Total_6: the average total amount during the last 6 months
FEES_Max_A_Total_12: the maximum total amount during the last 12 months
FEES_Max_A_Total_6: the maximum total amount during the last 6 months
FEES_Max_T_Total_12: the maximum total fee number during the last 12 months
FEES_Max_T_Total_6: the maximum total fee number during the last 6 months

It is possible to plot the heatmaps of the variables to find a pattern. In a heatmap the quantitative variables have minimum and maximum values that are displayed on the map by blue and red colours (respectively). In the case of qualitative variables such as gender, women are displayed in red and men in blue. The areas of the maps coloured with colours other than red and blue represent a mix of customers. We note that all of the visualizations have been carried out without the Not Available (NA) data and that the scaling of the heatmaps is a normalized scaling.

5.4.1.5 Gender

In gender heatmaps red and blue colours show female and male customers, respectively. On the maps the completely red nodes represent the groups in which the clients are *predominantly* (*predom.*) female and the completely blue nodes represent the opposite, i.e., groups of clients who are predominantly male. The nodes with colours other than red and blue represent groups in which the customers are mixed and it was not possible to find a similarity between them based on their gender.

By comparing the gender heatmap with other variables we can infer that

- the maximum amount of e-fund transfers and
- the maximum amount of AML transactions

have been associated with a group of *predom.* female customers during the last month. Apart from this group, the rest of female and male customers

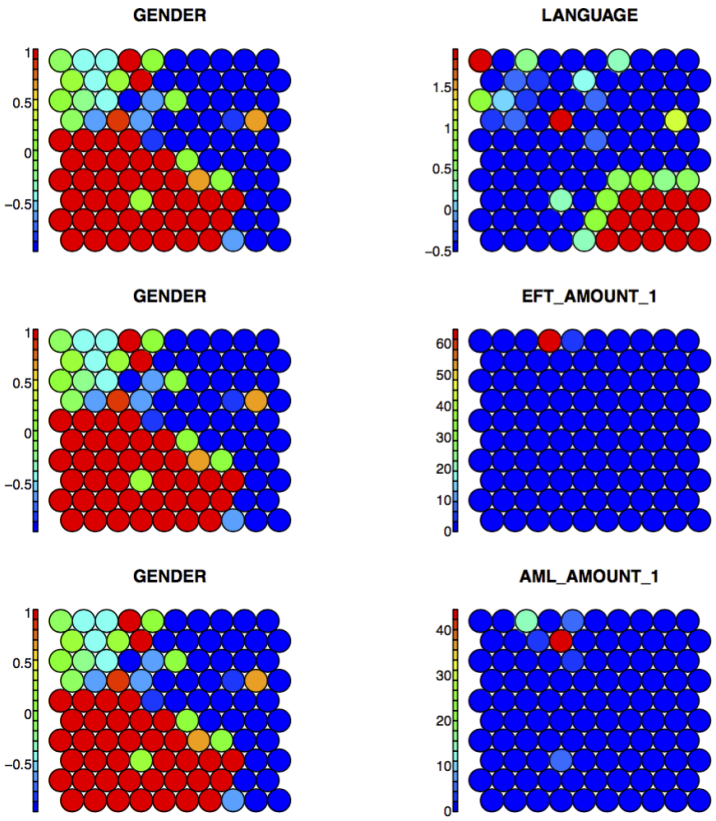

Fig. 5.2 Gender

have *predom.* low EFT and AML amounts and also low amount and number of fees.

5.4.1.6 Language

In language heatmaps red and blue colours represent English-speaking and French-speaking customers, respectively. On the maps the completely red nodes represent the groups in which the customers are *predom.* English-speaking and the completely blue nodes represent the opposite, i.e., groups of customers who are *predom.* mostly French-speaking. In nodes coloured differently, the customers are not grouped in accordance with their preferred language.

The heatmaps comparisons reveal that the majority of customers are French-speaking and

- the maximum amount of e-fund transfers,

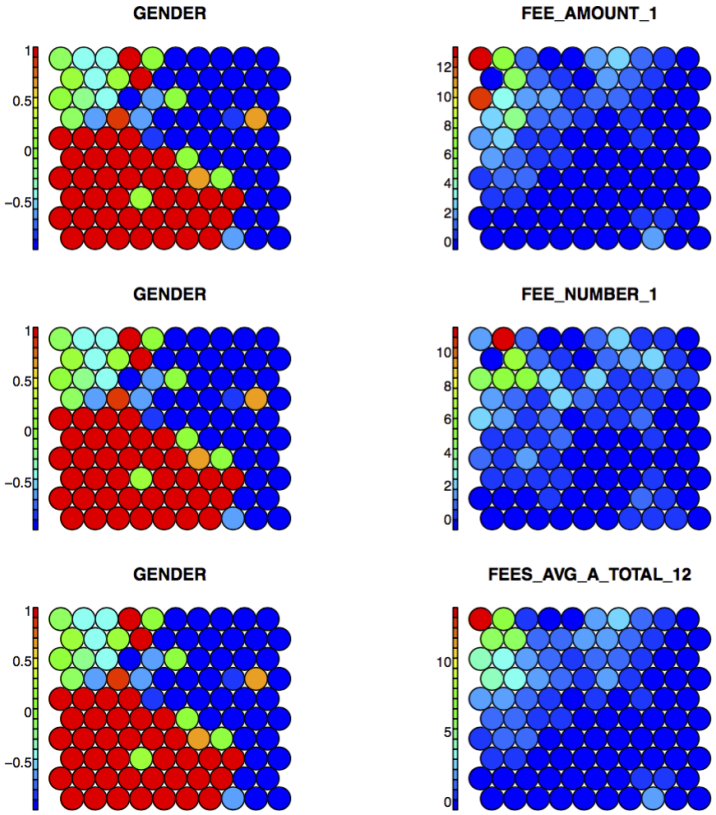

Fig. 5.3 Gender

- the maximum amount of AML transactions, and
- the maximum number of fees

are associated with a group of *predom.* French-speaking customers during the last month. Also

- the highest fee during the last month and
- the highest fee over the last 12 months on average

are associated with a group of *predom.* English-speaking customers.

5.4.1.7 EFT_Amount_1

In EFT heatmaps the red and blue colours represent the maximum and minimum amount (respectively) of e-fund transfers during the last month. The colours in the range between red and blue represent diverse values between maximum and minimum. From these heatmaps we can obviously see that the

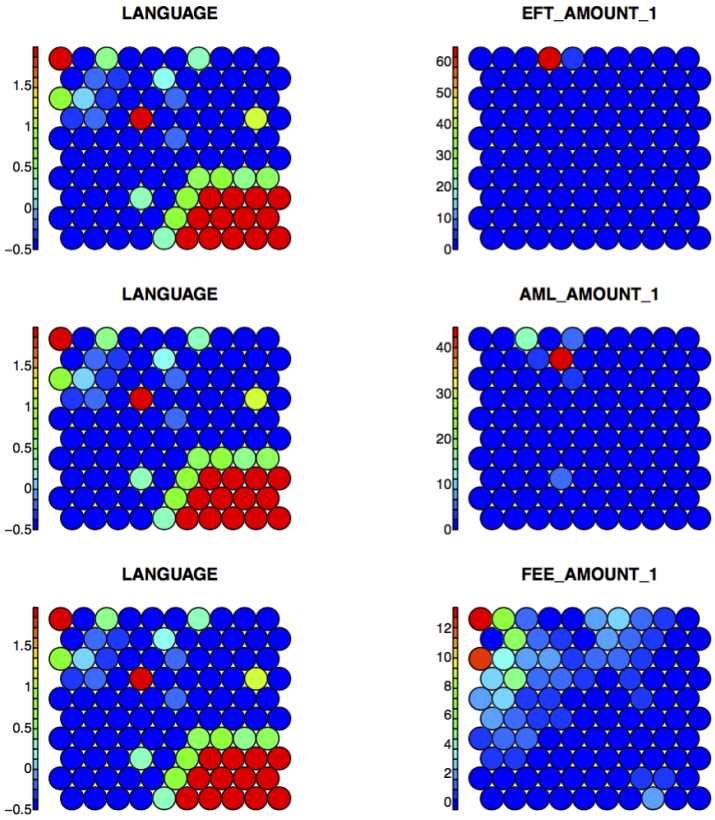

Fig. 5.4 Language

majority of customers have had a very low (minimum or close to minimum) e-fund transfer and there is a single group that has a high EFT.

Comparing EFT_AMOUNT heatmaps with other variables, we see that

- the maximum amount of AML transactions during the last month,
- the highest fee and the maximum fee number during the last month, and
- the highest fee over the last 12 months on average

are associated with a group of customers who have *predom.* the lowest EFT_AMOUNT. On the other hand, the only group with the maximum e-fund transfer has paid *predom.* minimum fees and had the minimum AML (NON-EFT) transaction amount.

5.4.1.8 AML_Amount_1

In AML heatmaps, red and blue colours represent the maximum and minimum amounts (respectively) of AML (NON-EFT) transactions during the last

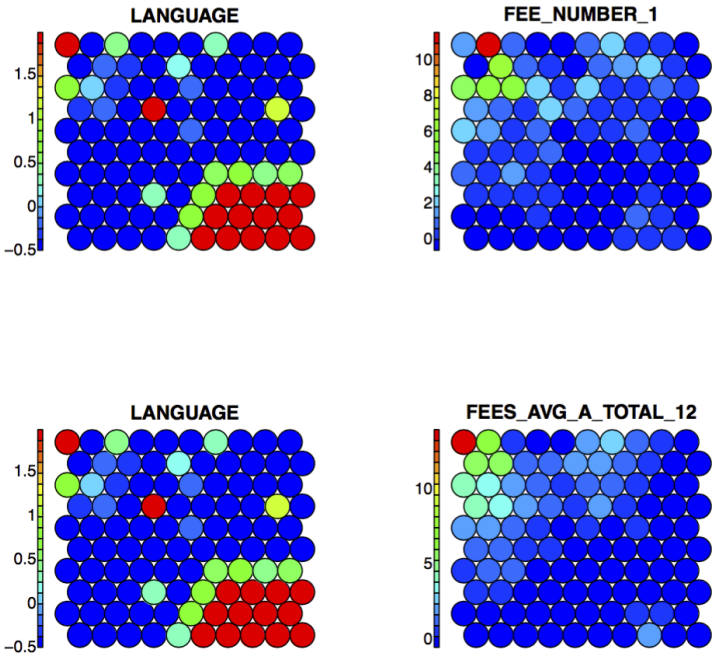

Fig. 5.5 Language

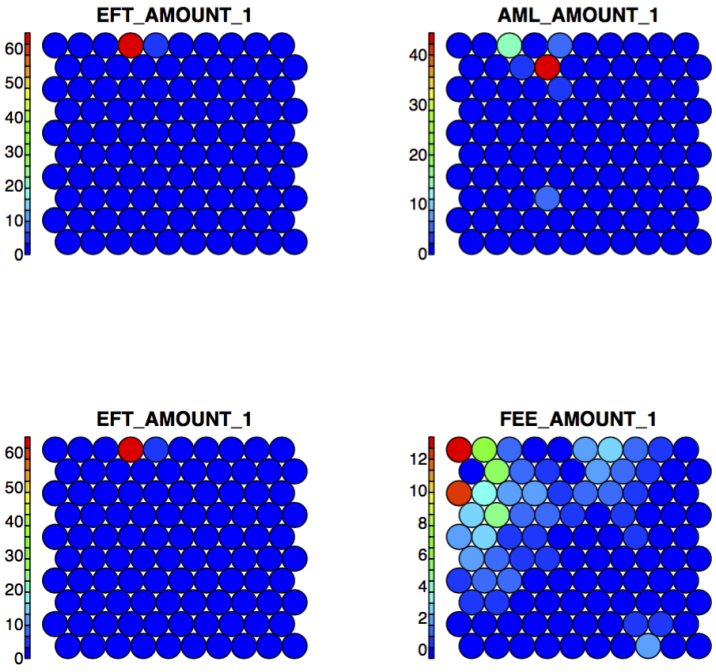

Fig. 5.6 EFT_Amount_1

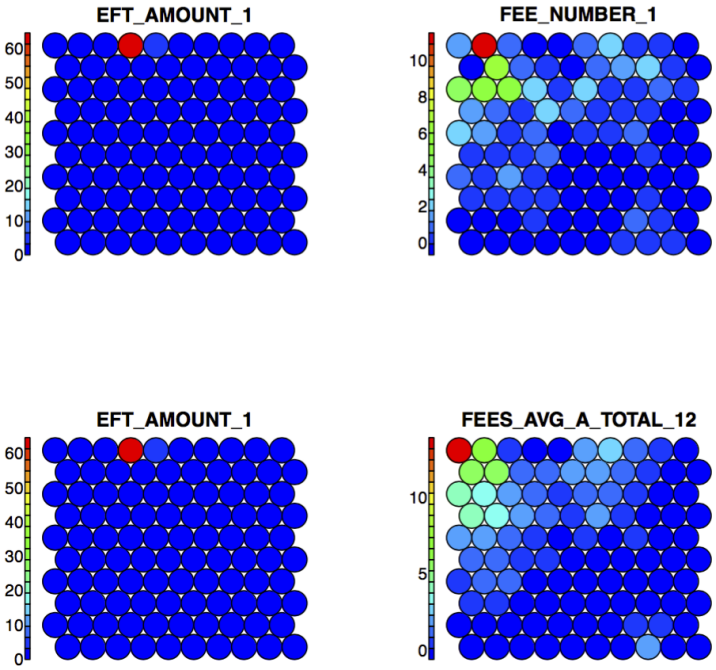

Fig. 5.7 EFT_Amount_1

month. The colours in the range between red and blue represent diverse values between the maximum and minimum values. The heatmaps show that in general the customers have had low AML transactions and there are a few groups of customers with higher AML transactions.

Comparing AML_AMOUNT heatmaps with the rest of variables, we see that

- the highest fee and the maximum fee number during the last month and
- the highest fee over the last 12 months on average

are associated with a group of customers who have *predom.* a low AML amount. The only group with the maximum AML amount has paid *predom.* minimum fees.

5.4.1.9 Clustering

In order to find a suitable number of customer clusters, we first run a k-means algorithm on the average characteristics of nodes (i.e., on the codebooks). In this way we obtain a "within cluster sum-of-squares" plot that displays the sum-of-squares of customers' variables when there are k clusters (where k is at most 100 since there are 100 nodes on the map). The point on the plot at

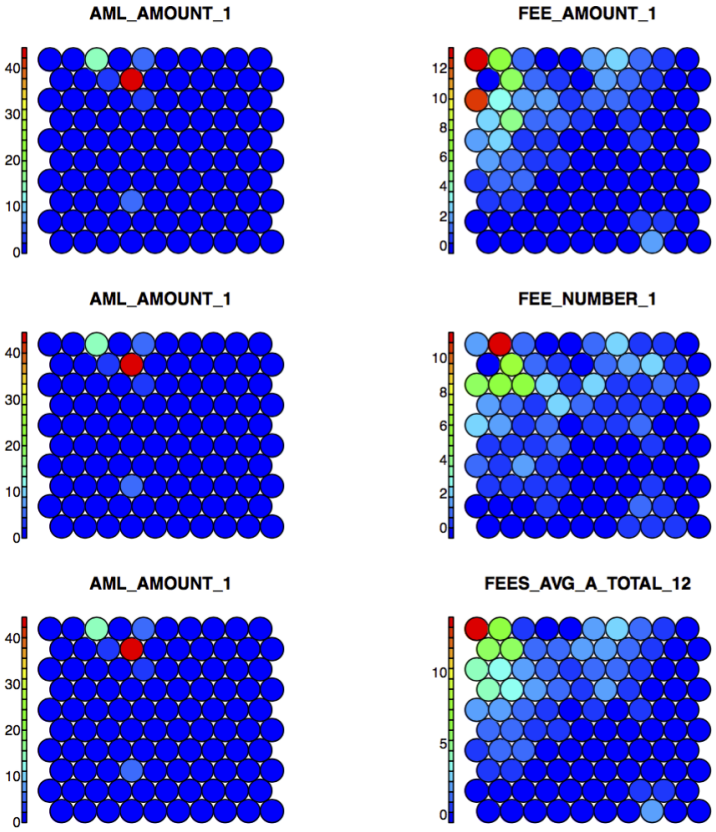

Fig. 5.8 AML_Amount_1

which the sum of squares is minimized gives an indication of how many main clusters should be chosen.

Considering the k-means plot, we recognize that 6 may be a suitable number of clusters. We now apply SOM clustering, which is basically a hierarchical clustering, on the map of trained data. In this process each node is initially considered as a cluster. Then similar nodes are combined if they are neighbours and the process continues until all the nodes have been combined.

This analysis allowed us to find the following clusters (or groups):

- A group in which female customers with maximal amounts of EFT and AML predominate;
- A group in which French-speaking customers with maximal amounts of EFT and AML and a maximal number of fees predominate;
- A group in which English-speaking customers with maximal fees predominate;

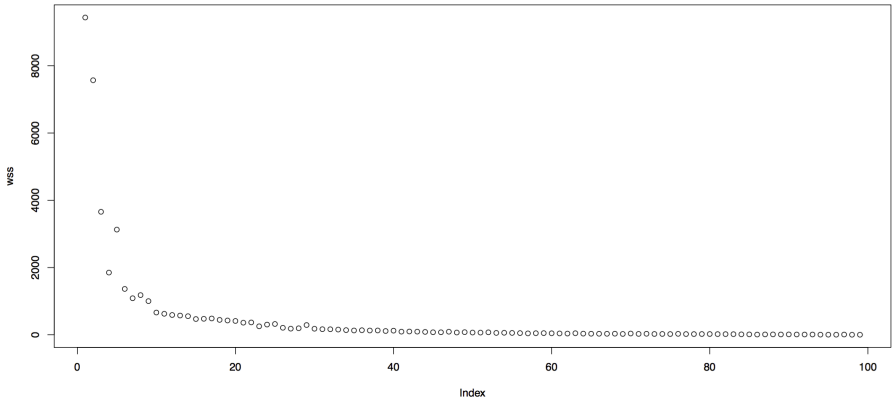

Fig. 5.9 Choosing the number of clusters by applying the k-means algorithm

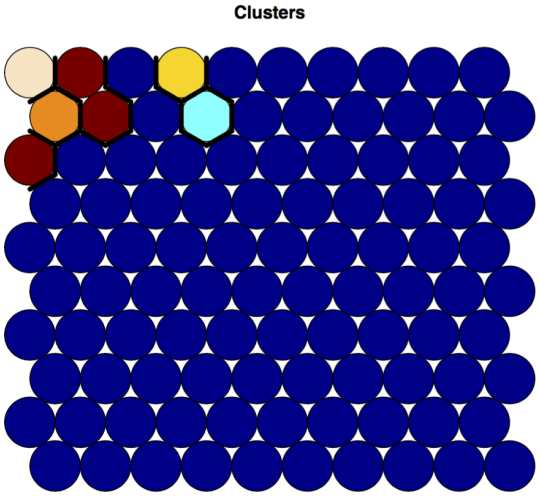

Fig. 5.10 Clustering

- A group in which customers with minimal fees, minimal AML amount, and maximal EFT amount predominate;
- A group in which customers with minimum fees and maximum AML amount predominate.

The rest of the customers can be clustered using some other common characteristics such as: French-speaking, paying low fees, barely using e-fund transfers, etc. The SOM includes all these customers within a large cluster on the map. Therefore we have defined six clusters.

The clusters and heatmaps are summarized in Figures 5.10 and 5.11.

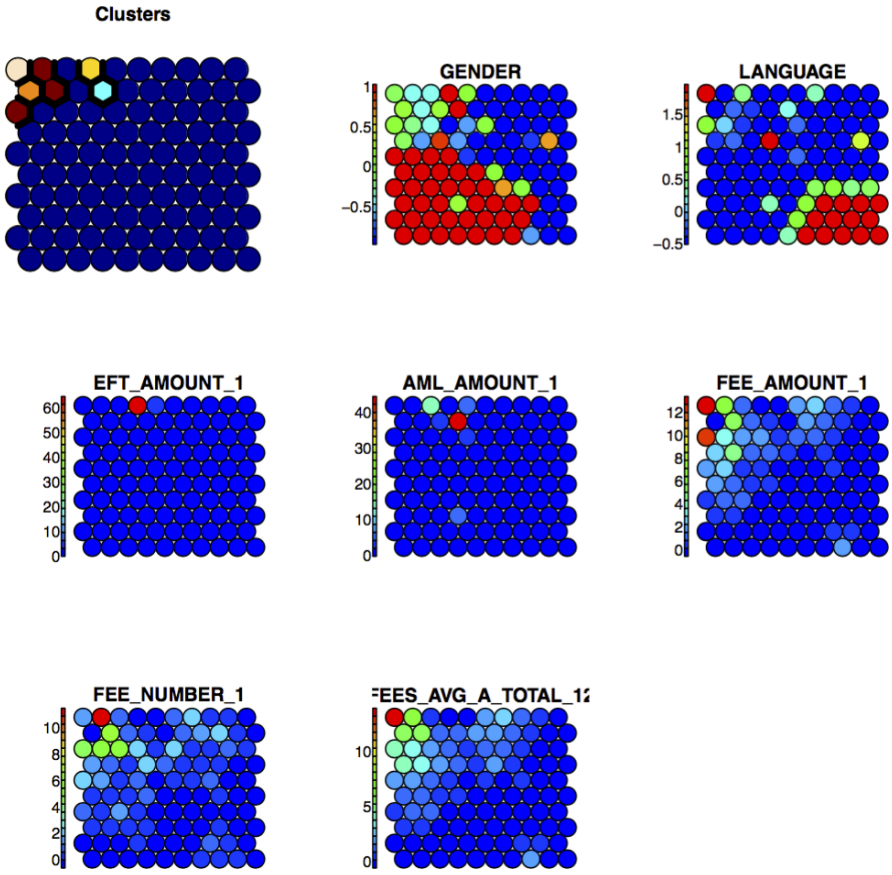

Fig. 5.11 Summary

5.4.1.10 Conclusion and Future Work

- We managed to identify some particular customer groups and determine, by looking at some of the available demographic data, which customers are most likely to use e-fund transfers and AML transactions, which customers pay the maximal fees, etc.
- We have considered only 13 variables (out of 88 variables from the clean retail data), but it is possible to take more variables into account, cluster the clients based on these variables, and find out which group of clients belongs to a given cluster. This will be the subject of future work.
- Provided that more complete and high-quality data sets become available, we could pursue our investigations and try to predict which group of (potential) customers may be of interest to the bank.
- By using SOM, it is possible to analyze each variable or characteristic over the whole set of customers and find out how dominant that variable could

be from the bank's point of view. We could then make suggestions for maximizing the CLV.

5.4.2 *Stepwise Regression*

One of the goals of the present project was to decrease the dimension of the set of available variables. *Stepwise regression* is one of the methods that can be used to achieve this goal. Stepwise regression attempts to determine which variables have a coefficient equal to zero in a multilinear regression (then these variables are not statistically significant). There are three variants of stepwise regression: the *Forward* variant, the *Backward* variant, and the *combined variant*. The Forward variant consists of beginning the multilinear regression without any explanatory variable and adding those variables that improve the model. The process is iterated until the model cannot be improved any more. The Backward variant is similar to the Forward variant, except that it begins the multilinear regression with all the explanatory variables and removes some variables at each iteration instead of adding some variables. Finally the combined variant compares the p-value of a statistic F with a certain threshold and the result of the comparison determines whether the parameter is added to or removed from the model.

Note also that some explanatory variables may be included in the model a priori. In this case the resulting model may be a "local" but not a "global" optimum. We applied this method with the 24 products of our data base; the CLV was the dependent variable. The `stepwisefit` function was used. Since there was no real correlation between the products, the Stepwise regression was not very useful. Indeed the number of products was reduced to 18. We have also reason to believe that this decrease reflects the presence of outliers.

5.4.3 *Principal Component Analysis (PCA)*

Principal Component Analysis (PCA) consists of replacing correlated variables by linearly independent variables. This allows us to determine a set of variables giving the "best explanation" of the other variables. In many cases only 2 or 3 variables are necessary to explain the other variables. The PCA algorithm consists essentially of computing the eigenvalues of the matrix of correlations between the variables for centered and normalized data. Note that the PCA is very sensitive to scaling. We applied PCA to 16 products and observed that the line-of-credit variable explained more than 50% of the variation in the other producs and the personal-loans variable explained more that 45% of this variation.

5.4.4 *Dependent Transition Probabilities through a Logistic Regression*

The objective here is to make transition probabilities dependent on many factors that are client-specific. In this fashion we model separately the value of a state and the probability that a client ends up in that state. In a linear regression the value of the dependent variable (i.e., the variable to be explained) is a real number. Hence we cannot use a linear regression because our dependent variable is a state transition probability, whose value is in $[0, 1]$ by definition. We will use a logistic regression instead.

Definition. The logistic function is defined as:

$$\sigma(t) = \frac{\mathrm{e}^t}{\mathrm{e}^t + 1} = \frac{1}{1 + \mathrm{e}^{-t}}$$

Its value is always between 0 and 1 and the graph of this function is a sigmoid curve.

We want to be able to use several explanatory variables in a regression in order to determine the probability of going from one state to another state. Suppose we have a vector x of dimension $1 \times n$, where n is the number of variables. Let us define the t in the logistic function as:

$$t = \beta^T \boldsymbol{x},$$

where β is the vector of parameters and is also of dimension n. In that case the logistic regression will be defined as

$$\begin{aligned} y &= \sigma(t) \\ &= \sigma(\beta^T \boldsymbol{x}) \\ &= \frac{1}{1 + \mathrm{e}^{-t(\beta^T \boldsymbol{x})}}. \end{aligned}$$

We will now look at *multiclass logistic regression*, in which the number of outputs is greater than 2. The probability of being in class C_k is defined by the following equation:

$$P(C_k \mid \boldsymbol{x}) = \frac{\exp(t_k)}{\sum_{j=1}^{m} \exp(t_i)},$$

where m is the number of different classes and

$$t_i = \beta_i^T \boldsymbol{x}.$$

Hence there are $m \times n$ parameters to estimate.

The objective is to find an accurate model for the state transition probabilities, which we will then insert in a semi-Markov chain. Therefore each and every client will have his own probability distribution model. A classification can also be carried out by using these results. Indeed, after computing the probability of being in each class, we can put the sample in the class with the highest probability value.

5.4.5 *Product Profile Matching*

5.4.5.1 Product Codes

In the commercial banking data set we are given a product code, the number of clients who own it, and the average value of the product held by the clients. Product codes represent the products (Table 5.3) owned by a customer.

For example the product code

$$y = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

represents the class of clients who own Deposit Certificates only.

5.4.5.2 Finding Product Code Values for Underpopulated Data

Since there are 24 products, the set of all possible product profiles

Table 5.3

Savings		**Loans**	
1	Deposit Certificates	5	Personal Loans
2	Savings Accounts	6	Mortgage
3	Mutual Funds C5	7	Letter of Credit
		8	Credit Card
Transactions			
9	Checking Account (CAN $)	17	Conciliation de cheques
10	Checking Account (US $))	18	Client Card (ATM)
11	Package	19	Internet Banking
12	Point of Service	20	Individual Insurance
13	EFT (debit)	21	Collective Insurance
14	EFT (credit)	22	Foreign Exchange
15	Gestion de l'encaisse	23	Affacturuage
16	Serice de Paie	24	Import/Export

$$A = \{y = (y_1, \ldots, y_{24}) : y_i \in \{0, 1\}\}$$

is of size $2^{24} = 16{,}777{,}216$. There are many product profiles in A in which the number of clients is too small for us to have confidence in the accuracy of the given average value.

In Figure 5.12 we see that most products are used by a small number of clients while a few are very popular. A simple approach is to partition the set of clients A into the following sets.

$$B = \{y \in A \mid \#\text{ of customers of } y \geq N\}$$
$$C = \{y \in A \mid \#\text{ of customers of } y < N\}$$

Note that N is a suitable cut-off value: we trust the accuracy of the average value given for product profiles with more than N customers.

To generate a value representing the similarity between y and z, it is natural to use the function $f(y, z)$ defined below.

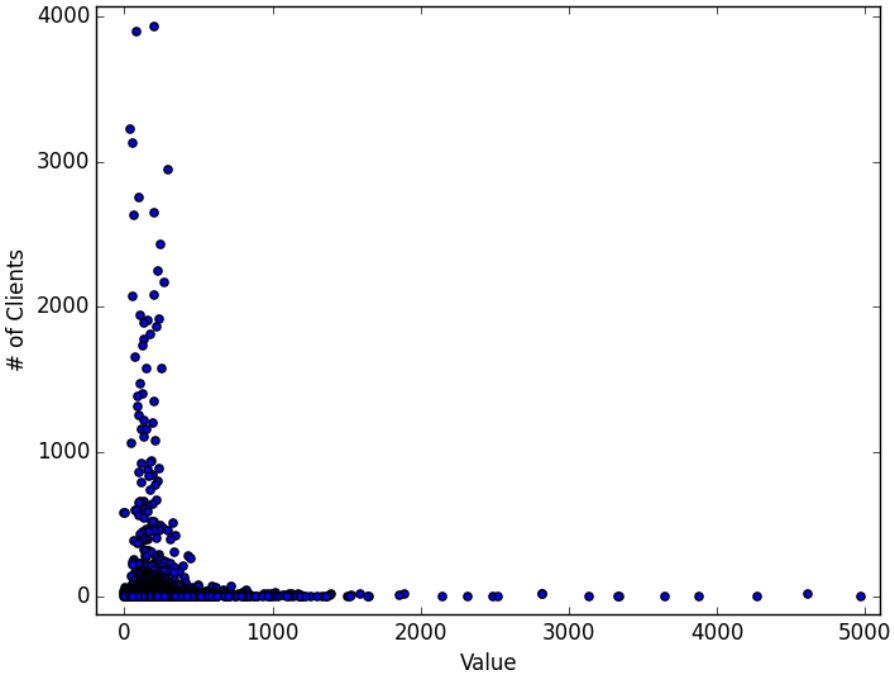

Fig. 5.12 Each point represents a Product Code with the position indicating its value and the number of clients using it.

$$f(y,z) = \begin{pmatrix} v_1 \\ \vdots \\ v_k \end{pmatrix}^t \cdot \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{pmatrix} = v_1\alpha_1 + \cdots v_k\alpha_k \text{ with } \alpha_i = \begin{cases} 1 & \text{if } y_i = z_i = 1 \\ -1 & \text{if } y_i \neq z_i \\ 0 & \text{if } y_i = z_i = 0 \end{cases}$$

When assessing the similarity, some products are more important than others. Thus the values of the v_i should reflect the ranking of products by the bank. At the present time we do not have access to this information and we assumed that the Loans variable (mortgages) is the most important, followed by the Transactions variable (checking accounts) and then Savings Accounts. Therefore we chose the following values.

$$v_1 = \cdots = v_3 = 1, \quad v_4 = \cdots = v_8 = 2, \quad v_9 = \cdots = v_{24} = 3$$

In summary, for an underpopulated product code $y \in C$, we find the $z \in B$ that is closest to it by maximizing the function $f(y,z)$.

5.4.5.3 Finding V

More systematic approaches are readily available for assigning values to the v_i. One such approach would be a numerical method analyzing the effect of the ownership of a product on a customer's value: given the set of populated product profiles contained in B, find the vector V that minimizes the error given by

$$\|V \cdot y - \text{ActualValue}(y)\|.$$

One could use for instance the L^2 norm, defined as

$$\|V \cdot y - \text{ActualValue}(y)\|_{L^2} = \left(\sum_{y \in B} \big(V \cdot y - \text{ActualValue}(y)\big)^2 \right)^{1/2}.$$

The minimal value of V can be found by the Conjugate Gradient method but we still have the problem of verifying the results of these computations. A possible approach (which is also easy to implement) is to partition B into a training set and a test set in order to evaluate the effectiveness of this method.

5.4.5.4 Nonlinear Approach

The function $f(y,z)$ given above is linear. It might be relevant to choose a nonlinear function, for instance the following function (when there are only 3 savings products):

$$f(y,z) = c_1\sigma_1 \mathrm{e}^{-d_{1,2}\sigma_2 - d_{1,3}\sigma_3} + \cdots + c_3\sigma_3 e^{-d_{3,1}\sigma_1 - d_{3,2}\sigma_2}.$$

We note that if y and z differ in 2 products, the value of f will not simply be twice the value it takes when y and z differ in 1 product.

5.4.5.5 Retail Data

While the commercial data includes Products Codes, their associated average values, and the number of customers owning them, the retail data provided is much larger. Data is available over an 18-month period for different users including: asset values, fees, length of ownership, etc. As for the commercial data, the goal is to create a model that could identify a customer's value based on his (or her) readily available information: age, gender, etc. The method used for the commercial data could be used by averaging over the different products to create data similar to the commercial data.

Banks are mostly interested in a customer's future lifetime value. For this reason a temporal method that takes into account a customer's potential would be ideal. For example, banks offer credit cards to students in the hope of building brand loyalty among future High Earners. In the models discussed in this report, we only give a snapshot of a customer's current value, based on data collected over a short time span. Any method for making future predictions will either rely upon data collected over a long period of time, or put more emphasis on customer properties that influence future value (education, for instance).

References

[1] Y. Ekinci, F. Ülengin, N. Uray, and B. Ülengin. "Analysis of customer lifetime value and marketing expenditure decisions through a Markovian-based model". *European Journal of Operational Research* 237 (2014), 278–288.

[2] N. Glady, B. Baesens, and C. Croux. "Modeling churn using customer lifetime value". *European Journal of Operational Research* 197:1 (2009), 402–411.

[3] M. Haenlein, A.M. Kaplin, and A.J. Beeser. "A model to determine customer lifetime value in a retail banking context". *European Management Journal* 25:3 (2007), 221–234.

6
Finite Element Analysis of Ultralight Metallic Lattices

Thomas Briffard, Guillaume Fortier, Alireza Khademi, Yves Martin, Judith Müller, Argyrios Petras, Benoît Pouliot, Serge Prudhomme, Patrick Terriault, José Urquiza, Yingjun Wang, Tyler Wilson, and Huang Xu

6.1 The Problem

6.1.1 *Context*

Pratt & Whitney Canada is considering the possibility of producing ultralight lattice structures through 3D printing (also called additive manufacturing), in order to reduce the weight of some pieces of machinery while preserving or improving their characteristics.

These three-dimensional structures are obtained by assembling (in a periodic or non-periodic fashion) a huge number of unit cells consisting of nodes and ribs. Several specific cell geometries are proposed in the literature in

Thomas Briffard · Benoît Pouliot · José Urquiza
Université Laval

Guillaume Fortier · Yves Martin
Pratt & Whitney Canada

Alireza Khademi · Serge Prudhomme
Polytechnique Montréal

Judith Müller
Burlington, VT

Argyrios Petras
Simon Fraser University

Patrick Terriault
École de technologie supérieure

Yingjun Wang · Huang Xu
McGill University

Tyler Wilson
University of Toronto

order to achieve certain mechanical behaviours that are required for particular applications.

Because the cell size is usually of the order of a millimeter, the number of elements required for the analysis of the structure corresponding to the entire piece is enormous, and it becomes difficult or impossible to use finite element analysis.

6.1.2 *Project Goal and Research Avenues*

In order for a finite element analysis of pieces including lattice structures to be carried out at a reasonable cost and within a reasonable time, it would be highly desirable to define a replacement solid with properties equivalent to those of the original piece (elasticity modulus, shear modulus, Poisson coefficient, in particular).

Here are a few important considerations:

1. The cell geometries proposed in the literature are in general anisotropic. Their orientation and the orientation of the overall structure have a big impact on the characteristics obtained.
2. The cell geometries vary a lot and will change over time, because of the evolution of design and manufacturing technology. The solution we are looking for must be adaptable and take this possible evolution into account.
3. Many applications would benefit from an adaptive meshing, which varies through the piece, in order to allow one to do the following (among other things):
   a. Adjust the properties locally according to the requirements (load restoration, heat transfer, energy absorption, variation in rigidity, etc.); and
   b. Follow a curved or irregular surface.
4. Homogenization is a standard technique applied to periodic lattice structures. It allows one to obtain an approximation (to a given order) of the lattice using a (plain) homogeneous 3D equivalent material (i.e., equivalent from a mechanical point of view). Replacing the lattice geometry with a plain homogeneous body will ensure that there are much fewer degrees of freedom in the computational analysis.

Fig. 6.1 3D homogeneous plate lattice (Url)

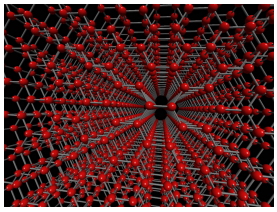

Fig. 6.2 3D lattice (Url)

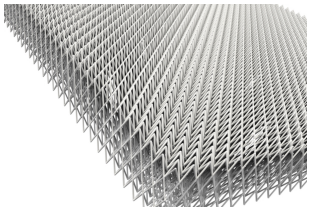

Fig. 6.3 3D plate lattice (Url)

6.2 The Team's Results

6.2.1 *Pratt & Whitney Specific Needs*

First of all we had to ascertain the needs of Pratt & Whitney. There is currently an observation/evaluation process at Pratt & Whitney. They want to study the current methods used to model linear elasticity in lattice-shaped domains (note that "lattice" is translated as (*treillis* in French). A good look at the literature would be interesting for them.

Also they have not yet chosen the macroscopic configuration of their lattices. For example a fully shaped solid with a 3D lattice will not behave in the same way as a shell/plate lattice (*coque de treillis*, in French).

A plate lattice can be obtained by considering an infinite solid lattice in x and y but with only a few layers in z, whereas a full lattice is obtained by considering an infinite solid lattice in x, y and z (see Figures 6.1, 6.2, and 6.3 for some examples).

Let l denote the relative size of a cell in the lattice and L the depth of the plate lattice (the smallest length in x, y or z). The ratio

$$\varepsilon \coloneqq \frac{l}{L}$$

characterizes the type of lattice we are dealing with. If $\varepsilon < \frac{1}{7}$, then we may consider the region as a full lattice. On the other hand, if $\varepsilon > \frac{1}{7}$, then the theory of full lattices cannot be applied to the region (the discriminating value $\frac{1}{7}$ is purely heuristic and comes from our discussions – some of us having already performed computational analysis of lattice structures for biomedical applications).

The third aspect of the problem proposed by Pratt & Whitney is their need for a fast method of resolution. One may consider using a finite element method applied to the entire lattice region. This is not appropriate because the number of elements required increases rapidly. Neither is it possible to

use adapted meshes as a workaround. Figures 6.4 and 6.5 show an adapted mesh in a simulated 2D plate lattice.

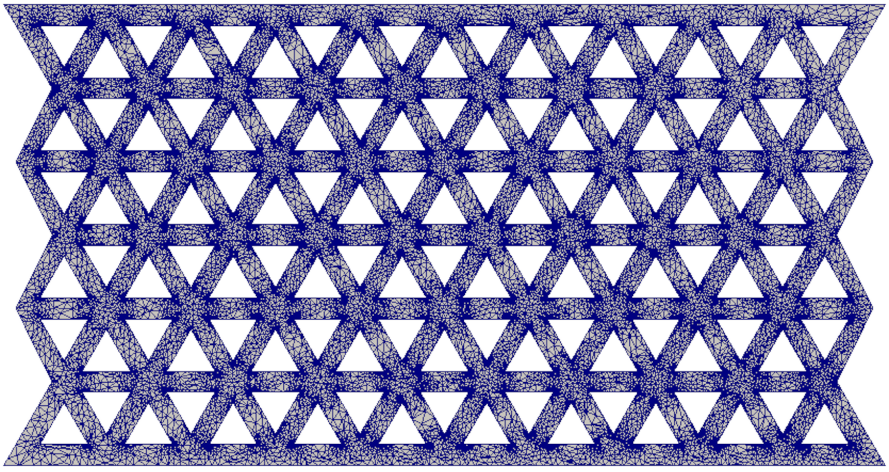

Fig. 6.4 Adapted 2D plate lattice

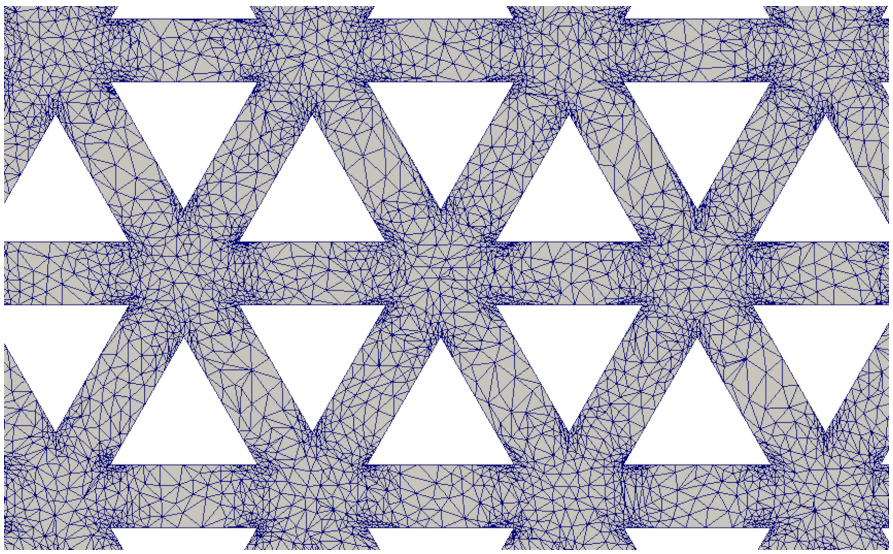

Fig. 6.5 Zoomed adapted 2D plate lattice

6.2.2 *The Literature*

We consider two distinct situations (both in the 3D case): that of a full domain lattice and that of a partial plate lattice.

We can find a substantial literature for those two situations. We only cite some interesting articles and books (which we found during the workshop).

6.2.2.1 Full Lattices

First of all, the Pasini Lab (at McGill University) specializes in lattice structures. Many reports and articles (for instance [2, 7–9]) can be found on their web pages (http://pasini.ca).

Popular ideas for solving the lattice problem are the RVE (representative volume element) decomposition and the homogenization method, which have the same goals. The aim is to approximate the lattice structure by a homogeneous plain domain (see the references [1, 3, 6]).

6.2.2.2 Plate Lattices

Plate lattices are a special case and need specialized processing (see for instance [4, 5]).

6.2.3 *Examples of Computations for the Full Lattice and the Plate Lattice*

We used the two-scale asymptotic expansion (an homogenization technique), which is described in the book of G. Allaire [1] (p. 168, in French).

6.2.3.1 Full Lattices

First of all, we consider a scaled cell consisting of a single lattice element with coordinates $\boldsymbol{y} \in Y = [0,1]^3$. In this way the cell does not have a particular physical unit. We assume that we know the mechanical proprieties $A(\boldsymbol{y})$ of the cell. Let $\boldsymbol{x}$ be the coordinates of the macroscopic domain Ω. Let $\boldsymbol{y} = \boldsymbol{x}/\varepsilon$ (assumed to be ε-periodic), and $A = A(\boldsymbol{x}/\varepsilon)$. We assume that this tensor is coercive and bounded. We recall that ε is a magnification parameter tending to zero at the limit.

Let us consider a diffusion problem on a full lattice.

$$\begin{cases} -\operatorname{div}\big(A(\boldsymbol{x}/\varepsilon)\nabla u_\varepsilon(\boldsymbol{x})\big) = f(\boldsymbol{x}) & \text{in } \Omega, \\ u_\varepsilon = 0 & \text{on } \partial\Omega \end{cases} \tag{6.1}$$

We assume $f(\boldsymbol{x})$ to be macroscopic only (i.e., independent of the microscopic coordinates $\boldsymbol{y}$). It is well known that this equation has a unique solution $u_\varepsilon \in H_0^1(\Omega)$ if $f \in L^2(\Omega)$. Our goal is to find a constant tensor A^* that, when used in (6.1) instead of $A(\boldsymbol{y})$, yields a solution to the equation that is a "good" approximation of the real solution u_ε. This procedure is called homogenization. To proceed, we decompose the solution u_ε as follows:

$$u_\varepsilon = \sum_{k=0}^{\infty} \varepsilon^k u_k.$$

The term u_0 is the homogenized term of u_ε and will represent a smooth solution. We apply the two-scale asymptotic expansion by setting each term u_k on $\Omega \times Y$:

$$u_k(\boldsymbol{x}, \boldsymbol{y}).$$

The $\boldsymbol{x}$ term is macroscopic (slow) and the $\boldsymbol{y}$ term is microscopic (fast). The function $u_k(\boldsymbol{x}, \boldsymbol{y})$ is ε-periodic in $\boldsymbol{y}$. We have $\boldsymbol{y} = \frac{\boldsymbol{x}}{\varepsilon}$ and thus

$$u_\varepsilon = \sum_{k=0}^{\infty} \varepsilon^k u_k\left(x, \frac{x}{\varepsilon}\right).$$

We can then introduce u_ε in the equation (6.1). Only the terms in ε^{-2}, ε^{-1}, and ε^0 are kept. We obtain the following equation.

$$\begin{aligned} f(\boldsymbol{x}) = &-\frac{1}{\varepsilon^2}\big(\operatorname{div}_{\boldsymbol{y}}(A\nabla_{\boldsymbol{y}} u_0)\big)\big|_{(\boldsymbol{x},\boldsymbol{x}/\varepsilon)} \\ &- \frac{1}{\varepsilon}\big(\operatorname{div}_{\boldsymbol{y}}\big(A(\nabla_{\boldsymbol{x}} u_0 + \nabla_{\boldsymbol{y}} u_1)\big) + \operatorname{div}_{\boldsymbol{x}}(A\nabla_{\boldsymbol{y}} u_0)\big)\big|_{(\boldsymbol{x},\boldsymbol{x}/\varepsilon)} \\ &- \Big(\operatorname{div}_{\boldsymbol{x}}\big(A(\nabla_{\boldsymbol{x}} u_0 + \nabla_{\boldsymbol{y}} u_1)\big) + \operatorname{div}_{\boldsymbol{y}}\big(A(\nabla_{\boldsymbol{x}} u_1 + \nabla_{\boldsymbol{y}} u_2)\big)\Big)\Big|_{(\boldsymbol{x},\boldsymbol{x}/\varepsilon)} \\ &+ O(\varepsilon) \end{aligned} \tag{6.2}$$

The first equation (in ε^{-2}) yields

$$\operatorname{div}_{\boldsymbol{y}}\big(A(\boldsymbol{y})\nabla_{\boldsymbol{y}} u_0(\boldsymbol{x}, \boldsymbol{y})\big) = 0.$$

We can conclude directly that there is a function $u(\boldsymbol{x})$ (depending only on $\boldsymbol{x}$) for which

$$u_0(\boldsymbol{x}, \boldsymbol{y}) = u(\boldsymbol{x})$$

holds. Thus u_0 does not change on the microscopic scale. With this consideration, the second equation (in ε^{-1}) yields

$$\operatorname{div}_{\boldsymbol{y}}\big(A(\nabla_{\boldsymbol{x}}u+\nabla_{\boldsymbol{y}}u_1)bilg\big)=0.$$

We use the lemma in [1, Lemma 7.4] and obtain a unique solution for u_1 (up to an additive constant), in terms of u.

$$u_1(\boldsymbol{x},\boldsymbol{y})=\sum_{i=1}^{3}\frac{\partial u(\boldsymbol{x})}{\partial x_i}w_i(\boldsymbol{y})$$

Let $\{e_i\}$ be the canonical base of $\mathbb{R}^3$. We define w_i to be the solution of

$$\begin{cases}\operatorname{div}_{\boldsymbol{y}}\big(A(\boldsymbol{y})\big(e_i+\nabla_{\boldsymbol{y}}w_i(\boldsymbol{y})\big)\big)=0 & \text{in } Y,\\ w_i(\boldsymbol{y}) \text{ periodic} & \text{on } \partial Y.\end{cases}\tag{6.3}$$

We will only need $\nabla_{\boldsymbol{y}}u_1$, so the lack of uniqueness of u_1 is not a problem.

Finally the third line in (6.2) yields

$$-\operatorname{div}_{\boldsymbol{y}}(A\nabla_{\boldsymbol{y}}u_2)=\operatorname{div}_{\boldsymbol{y}}(A\nabla_{\boldsymbol{x}}u_1)+\operatorname{div}_{\boldsymbol{x}}\big(A(\nabla_{\boldsymbol{x}}u+\nabla_{\boldsymbol{y}}u_1)\big)+f.$$

Again, with the help of Lemma 7.4, we have a unique solution (up to a constant) for u_2 in terms of u if the following compatibility condition is satisfied.

$$\int_Y\big[\operatorname{div}_{\boldsymbol{y}}(A(\boldsymbol{y})\nabla_{\boldsymbol{x}}u_1)+\operatorname{div}_{\boldsymbol{x}}\big(A(\boldsymbol{y})(\nabla_{\boldsymbol{x}}u+\nabla_{\boldsymbol{y}}u_1)\big)+f(\boldsymbol{x})\big]\,\mathrm{d}\boldsymbol{y}=0$$

The first term is zero and we obtain

$$-\operatorname{div}_{\boldsymbol{x}}\int_Y[A(\boldsymbol{y})(\nabla_{\boldsymbol{x}}u+\nabla_{\boldsymbol{y}}u_1)]\,\mathrm{d}\boldsymbol{y}=f(\boldsymbol{x})\quad\text{in }\Omega.\tag{6.4}$$

Recall that $Y=[0,1]^3$. Let us define the homogenized tensor A^* as follows.

$$A^*_{ij}:=\int_Y A(\boldsymbol{y})(e_j+\nabla_{\boldsymbol{y}}w_j)\cdot e_i\,\mathrm{d}\boldsymbol{y}$$

A symmetric formula also exists. We then have the following product.

$$A^*\nabla_{\boldsymbol{x}}u=\sum_{j=1}^{3}\frac{\partial u}{\partial x_j}\int_Y A(\boldsymbol{y})(e_j+\nabla_{\boldsymbol{y}}w_j)\,\mathrm{d}\boldsymbol{y}$$

We obtain, using (6.4), the new PDE

$$\begin{cases}-\operatorname{div}_{\boldsymbol{x}}(A^*\nabla_{\boldsymbol{x}}u)=f(\boldsymbol{x}) & \text{in } \Omega,\\ u=0 & \text{on } \partial\Omega.\end{cases}\tag{6.5}$$

We have transformed our original equation (6.1) into this new one. A standard FEM can now be used to solve this PDE with homogeneous parameters over Ω.

For a convergence theorem, we refer to the book by Allaire [1].

6.2.3.2 Plate Lattices

We may now consider the case of a plate: an infinite series of copies of the unit lattice in the x and y directions, and K copies of the unit lattice in the z direction. In this case, we cannot consider only one cell because ε would be too large. To keep ε small, we consider all K layers of the domain in the z direction and only one layer of cells in the x and y directions. This domain is our super-cell domain ($Y = [0,1]^2 \times [0,K]$), in which we will compute the homogenized tensor.

For this case we will assume that we have natural boundary conditions on each side of the plate Γ_P in the z direction.

$$\begin{cases} -\operatorname{div}\left(A\left(\dfrac{\boldsymbol{x}}{\varepsilon}\right)\nabla u_\varepsilon(\boldsymbol{x})\right) = f(\overline{\boldsymbol{x}}) & \text{in } \Omega, \\ A\left(\dfrac{\boldsymbol{x}}{\varepsilon}\right)\nabla u_\varepsilon(\boldsymbol{x})\cdot\vec{n} = 0 & \text{on } \Gamma_P, \\ u_\varepsilon = 0 & \text{on } \partial\Omega \setminus \Gamma_P \end{cases} \tag{6.6}$$

Let $\overline{\boldsymbol{x}} = (x, y)$. The general case for the boundary conditions is more complicated and requires further consideration (beyond the present study).

We believe that the case of a homogeneous lattice in z can be solved by a simple separation-of-variables method in (x, y) and z. In this case the tensor A can be decomposed as follows.

$$A(\boldsymbol{y}) = \begin{bmatrix} a_{11}(\boldsymbol{y}) & a_{12}(\boldsymbol{y}) & \\ a_{21}(\boldsymbol{y}) & a_{22}(\boldsymbol{y}) & \\ & & a_{33}(\boldsymbol{y}) \end{bmatrix}$$

Then we can assume that $u_\varepsilon(\boldsymbol{x}) = v_\varepsilon(x, y)t(z)$ holds. After the decomposition, we must solve a 2D lattice problem for v_ε. We may use the same methods as in the previous section to solve this problem. In the z direction, the model is a standard EDO equation. Further study is required to check the validity of the procedure we have just outlined.

In the other case (the general 3D plate lattice), we take the same route as for the full lattice problem. The difference is that we require that $u_\varepsilon(\boldsymbol{x})$ be constant in the macroscopic scale for z. To proced, we decompose the solution u_ε as follows.

$$u_\varepsilon = \sum_{k=0}^{\infty} \varepsilon^k u_k$$

The term u_0 is the homogenized term of u_ε and will represent a smooth solution. We apply the two-scale asymptotic expansion by setting each term u_k on $\Omega \times Y$

$$u_k(\overline{\boldsymbol{x}}, \boldsymbol{y})$$

The $\overline{\boldsymbol{x}}$ term is macroscopic (slow) and the $\boldsymbol{y}$ term is microscopic (fast). We may say that u_ε is so thin that it does not change in the macroscopic scale in z. The function $u_k(\overline{\boldsymbol{x}}, \boldsymbol{y})$ is ε-periodic in $\boldsymbol{y}$. Also we assume that the $u_k(\overline{\boldsymbol{x}}, \boldsymbol{y})$ have free natural boundary conditions over $\Gamma_P \cap Y$. We have $\boldsymbol{y} = \boldsymbol{x}/\varepsilon$ and thus

$$u_\varepsilon = \sum_{k=0}^{\infty} \varepsilon^k u_k\left(\overline{\boldsymbol{x}}, \frac{\boldsymbol{x}}{\varepsilon}\right).$$

We may then introduce u_ε in the equation (6.6). Only the terms in ε^{-2}, ε^{-1} and ε^0 are kept. We obtain,

(6.7)

$$\begin{aligned} f(\overline{\boldsymbol{x}}) = &- \frac{1}{\varepsilon^2}\big(\operatorname{div}_{\boldsymbol{y}}(A\nabla_{\boldsymbol{y}} u_0)\big)\big|_{(\overline{\boldsymbol{x}}, \boldsymbol{x}/\varepsilon)} \\ &- \frac{1}{\varepsilon}\big(\operatorname{div}_{\boldsymbol{y}}\big(A(\nabla_{\boldsymbol{x}} u_0 + \nabla_{\boldsymbol{y}} u_1)\big) + \operatorname{div}_{\boldsymbol{x}}(A\nabla_{\boldsymbol{y}} u_0)\big)\big|_{(\overline{\boldsymbol{x}}, \boldsymbol{x}/\varepsilon)} \\ &- \Big(\operatorname{div}_{\boldsymbol{x}}\big(A(\nabla_{\boldsymbol{x}} u_0 + \nabla_{\boldsymbol{y}} u_1)\big) + \operatorname{div}_{\boldsymbol{y}}\big(A(\nabla_{\boldsymbol{x}} u_1 + \nabla_{\boldsymbol{y}} u_2)\big)\Big)\Big|_{(\overline{\boldsymbol{x}}, \boldsymbol{x}/\varepsilon)} \\ &+ O(\varepsilon). \end{aligned}$$

We note that the derivative of all u_k with respect to z is 0.

The first equation (in ε^{-2}) yields

$$\operatorname{div}_{\boldsymbol{y}}\big(A(\boldsymbol{y})\nabla_{\boldsymbol{y}} u_0(\overline{\boldsymbol{x}}, \boldsymbol{y})\big) = 0.$$

We can directly conclude that there is a function $u(\overline{\boldsymbol{x}})$ (depending only on $\overline{\boldsymbol{x}}$) for which

$$u_0(\overline{\boldsymbol{x}}, \boldsymbol{y}) = u(\overline{\boldsymbol{x}}).$$

Hence u_0 does not change on the microscopic scale. Given these considerations, the second equation (in ε^{-1}) yields

$$\operatorname{div}_{\boldsymbol{y}}\big(A(\nabla_{\boldsymbol{x}} u + \nabla_{\boldsymbol{y}} u_1)\big) = 0.$$

We use Lemma 7.4 in [1, Lemma 7.4] and obtain a unique solution for u_1 (up to a constant) in terms of u.

$$u_1(\overline{\boldsymbol{x}}, \boldsymbol{y}) = \sum_{i=1}^{2} \frac{\partial u(\overline{\boldsymbol{x}})}{\partial x_i} w_i(\boldsymbol{y}).$$

Let us define $e_1 = (1, 0, 0)^t$ and $e_2 = (0, 1, 0)^t$. Then w_i is the solution of

$$\begin{cases} \operatorname{div}_{\boldsymbol{y}}\big(A(\boldsymbol{y})(e_i + \nabla_{\boldsymbol{y}} w_i(\boldsymbol{y}))\big) = 0 & \text{in } Y, \\ A(\boldsymbol{y})\nabla_{\boldsymbol{y}} w_i \cdot \vec{n} = 0 & \text{on } Y \cap \Gamma_P, \\ w_i(\boldsymbol{y}) \text{ periodic} & \text{on } \partial Y \backslash \Gamma_P. \end{cases} \tag{6.8}$$

We will only need $\nabla_{\boldsymbol{y}} u_1$, so the lack of uniqueness of u_1 is not a problem.

Finally the third line in (6.7) yields

$$-\operatorname{div}_{\boldsymbol{y}}(A\nabla_{\boldsymbol{y}} u_2) = \operatorname{div}_{\boldsymbol{y}}(A\nabla_{\boldsymbol{x}} u_1) + \operatorname{div}_{\boldsymbol{x}}\big(A(\nabla_{\boldsymbol{x}} u + \nabla_{\boldsymbol{y}} u_1)\big) + f.$$

Again, with the help of Lemma 7.4, we have a unique solution (up to a constant) for u_2 in terms of u if the following compatibility condition is satisfied.

$$\int_Y \big[\operatorname{div}_{\boldsymbol{y}}(A(\boldsymbol{y})\nabla_{\boldsymbol{x}} u_1) + \operatorname{div}_{\boldsymbol{x}}\big(A(\boldsymbol{y})(\nabla_{\boldsymbol{x}} u + \nabla_{\boldsymbol{y}} u_1)\big) + f(\overline{\boldsymbol{x}})\big]\,\mathrm{d}\boldsymbol{y} = 0$$

The first term equals zero and we obtain

$$-\operatorname{div}_{\boldsymbol{x}} \int_Y [A(\boldsymbol{y})(\nabla_{\boldsymbol{x}} u + \nabla_{\boldsymbol{y}} u_1)]\,\mathrm{d}\boldsymbol{y} = Kf(\overline{\boldsymbol{x}}) \quad \text{in } \Omega^*. \tag{6.9}$$

Recall that $Y = [0,1]^2 \times [0,K]$. Also Ω^* is the (x,y) part of Ω. Let us construct the homogenized tensor A^* as follows.

$$A^*_{ij} \coloneqq \int_Y A(\boldsymbol{y})(e_j + \nabla_{\boldsymbol{y}} w_j) \cdot e_i \,\mathrm{d}\boldsymbol{y}$$

The dimension of this tensor is only 2×2. A symmetric formula also exists. We then have the product

$$A^* \nabla_{\boldsymbol{x}} u = \sum_{j=1}^{2} \frac{\partial u}{\partial x_j} \int_Y A(\boldsymbol{y})(e_j + \nabla_{\boldsymbol{y}} w_j)\,\mathrm{d}\boldsymbol{y}.$$

Using (6.9), we obtain the new PDE

$$\begin{cases} -\operatorname{div}_{\boldsymbol{x}}(A^*\nabla_{\boldsymbol{x}} u) = Kf(\overline{\boldsymbol{x}}) & \text{in } \Omega^*, \\ u = 0 & \text{on } \partial\Omega^*. \end{cases} \tag{6.10}$$

We have thus transformed our original equation (6.6) into this new equation, which can be solved over Ω^* by using a standard FEM.

References

[1] G. Allaire. *Conception optimale de structure*. Mathématiques et Applications, volume 58. Berlin: Springer, 2007.

[2] S. Arabnejad and D. Pasini. "Mechanical properties of planar lattice materials via asymptotic homogenization and comparison with alternative homogenization methods". *International Journal of Mechanical Sciences* 77 (2013), 249–262.

[3] B. Hassani and E. Hinton. "A review of homogenization and topology optimisation. Homogenization theory for media with periodic structure". I. *Computer and Structures* 69:6 (1997), 707–717.

[4] A. Lebée and K. Sab. "Homogenization of a space frame as a thick plate: Application of the Bending-Gradient theory to a beam lattice". *Computer and Structures* 127 (2013), 88–101.

[5] G.I. Pshenichnov. *A Theory of Latticed Plates and Shells*. Series on Advances in Mathematics for Applied Sciences, volume 5. River Edge, NJ: World Scientific, 1993.

[6] H. Tollenaere and D. Caillerie. "Continuous modeling of lattice structures by homogenization". *Advances in Engineering Software* 29:7-9 (1998), 699–705.

[7] A. Vigliotti, V.S. Deshpande, and D. Pasini. "Non linear constitutive models for lattice materials". *Journal of the Mechanics and Physics of Solids* 64 (2014), 44–60.

[8] A. Vigliotti and D. Pasini. "Linear multiscale analysis and finite element validation of stretching and bending dominated lattice materials". *Mechanics of Materials* 46 (2012), 57–68.

[9] A. Vigliotti and D. Pasini. "Stiffness and strength of tridimensional periodic lattices". *Computer Methods in Applied Mechanics and Engineering* 229–232 (2012), 27–43.