The Pennsylvania State University
The Graduate School

MULTILEVEL AND ADAPTIVE METHODS FOR SOME NONLINEAR OPTIMIZATION PROBLEMS

A Thesis in
Mathematics
by
Maria Emelianenko

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2005

The thesis of Maria Emelianenko was reviewed and approved * by the following:

Qiang Du
Professor of Mathematics
Chair of Committee
Thesis Advisor

Ludmil Zikatanov
Assistant Professor of Mathematics

Wen Shen
Assistant Professor of Mathematics

Zi-Kui Liu
Associate Professor of Materials Science and Engineering

Hongyuan Zha
Professor of Computer Science

Nigel Higson
Professor of Mathematics
Head of the Department of Mathematics

* Signatures are on file in the Graduate School.

Abstract

In this thesis, we propose new multilevel and adaptive methods for solving nonlinear non-convex optimization problems without relying on the linearization. We focus on two particular applications, that come from the fields of quantization and materials science. For the first problem, a multilevel quantization scheme is developed, that possesses a uniform convergence independent of the problem size. This is the first multilevel quantization scheme in the literature with a rigorous proof of uniform convergence with respect to the grid size and the number of grid levels for nonconstant densities. The proposed scheme can be generalized to higher dimensions, and both scalar and vector versions demonstrate significant speedup comparing to the traditional Lloyd method. We also provide some new characterizations for the convergence of the Lloyd iteration and other possible acceleration techniques including Newton-like methods. For the second optimization problem, this thesis presents a novel algorithm aimed at automating phase diagram construction in complex multicomponent systems. The new method utilizes the geometric properties of the energy surfaces together with adaptivity and effective sampling techniques to improve on the starting points for the minimization. It is shown that the new approach overcomes the drawbacks of the previously known algorithms, at the same time giving comparable accuracy to the solution.

Table of Contents

List of Figures vii

Acknowledgments ix

Chapter 1
Introduction 1

Chapter 2
Some existing algorithms for computing CVTs and their convergence 5
2.1 Overview . . . 5
2.2 Lloyd's iteration . . . 8
2.3 Convergence . . . 9
2.3.1 Monotonicity properties of the energy functional . . . 9
2.3.2 Existence of convergent subsequence . . . 12
2.3.3 Global convergence . . . 14
2.3.4 The compactness in the one dimensional case . . . 16
2.3.5 The logarithmic concave density in the one dimensional case 22
2.4 Extensions to constrained CVTs . . . 26
2.5 Numerical examples . . . 30
2.5.1 Constant density . . . 30
2.5.2 Non-constant density . . . 30
2.6 Conclusions . . . 33

Chapter 3
New algorithms for the construction of CVTs 35
3.1 Overview . . . 35

3.2 Newton's method and related results . . . 36
3.2.1 Notations . . . 36
3.2.2 Theoretical results . . . 38
3.2.3 Description of the algorithm . . . 42
3.2.4 Local vs. global minimizers . . . 44
3.2.5 Computational complexity . . . 45
3.2.6 Parallel implementation . . . 46
3.2.7 Numerical Implementation . . . 47
3.2.8 Stopping criterion . . . 48
3.2.9 One-dimensional examples . . . 48
3.2.10 Two-dimensional examples . . . 49
3.3 Newton-based multilevel algorithm . . . 53
3.4 The new energy-based nonlinear multilevel algorithm . . . 57
3.4.1 Space decomposition . . . 58
3.4.2 Description of the algorithm . . . 59
3.4.3 Technical lemmas . . . 61
3.4.4 Uniform convergence theorem . . . 68
3.4.5 Proof of the main result . . . 68
3.4.6 Numerical results . . . 70
3.5 Conclusion . . . 75

Chapter 4
A new algorithm for the automation of phase diagram calculation 77
4.1 Overview . . . 77
4.2 Theoretical aspects of phase diagram calculation . . . 79
4.2.1 Mathematical model . . . 79
4.2.2 Geometrical considerations . . . 85
4.2.3 Existing algorithms and motivation . . . 87
4.3 A new algorithm . . . 89
4.3.1 Description of the algorithm: binary case . . . 89
4.3.2 Description of the algorithm: ternary case . . . 97
4.3.3 Computational complexity estimate for the binary case . . . 100
4.3.4 Generalization to higher dimensions and sampling schemes . 103
4.4 Results for binary and ternary systems . . . 105
4.4.1 Binary examples . . . 106
4.4.2 Ternary examples . . . 107
4.5 Conclusion . . . 108

Chapter 5
Summary and discussion 110

Bibliography **116**

List of Figures

2.1 Voronoi tessellation on a square (left), CVT on a square (middle) and on a sphere (right) for constant density 6
2.2 An example of a CVT-based 3-d mesh 7
2.3 Examples of constrained CVTs (CCVTs) for a circle (dots are for generators and dash lines show the partition of the constrained Voronoi regions) and for a sphere (dots are generators, lines are planar projections of Voronoi edges, only portion in one hemisphere is shown). 28
2.4 Convergence of Lloyd method for constant density. 30
2.5 Convergence factor of Lloyd method for $\rho(x) = e^{-x^2}$ 31
2.6 Convergence factor of Lloyd method for $\rho(x) = 1 + x^4 cos(\pi x)$ 31
2.7 Convergence factor for $k = 16$ and $\rho(x) = 1 + \epsilon \cos^2(\pi x)$ with $\epsilon = 10^{-10} : 10^{10}$ 32
2.8 Convergence factor for $\rho(x) = 1 + 10^3 \cos^2(\pi x)$ and $k = 2 : 40$ . . . 32
2.9 Asymptotic behavior of the convergence factor for $\rho(x) = 1 + 10^3 \cos^2(\pi x)$ 33

3.1 Convergence of Lloyd's method to local and global minimizers . . . 45
3.2 1d convergence rates comparison for $k = 4$ (left) and $k = 64$ (right) with $\rho(x) = 1 + x^4 \cos(\pi(x - 0.5))$. Top curves are for Newton iteration and the bottom ones are for Lloyd. 49
3.3 Iteration history of (a) Lloyd-Newton vs. (b) Lloyd method for $\rho(x) = 1, k = 5$ 50
3.4 Convergence rate of the Lloyd-Newton method (top graph) vs. Lloyd iteration (bottom) for $\Omega = [0, 1]^2$, $\rho(x) = 1$, $k = 5$ . . 50
3.5 Iteration history of (a) Lloyd-Newton vs. (b) Lloyd method for $\rho(x) = 1+x+0.1x^2$, $k = 4$. Here the lines connect the generators 51

3.6 Iteration history of (a) Lloyd-Newton vs. (b) Lloyd method, $\rho(x) = 1 + x^4, k = 4$ . . . 52
3.7 Comparison of convergence factors for Newton-Lloyd iteration (top) vs. Lloyd (bottom) for different densities: (a) $\rho(x) = 1 + x + 0.1\,x^2\,, k = 4\,;$ (b) $\rho(x) = 1 + x^4\,, k = 4$ . . . 53
3.8 Plot of the convergence factor over the number of generators for the multigrid method vs. regular Gauss-Seidel method for $\rho = 1$ and $\rho = 1 + 0.1x$ . . . 71
3.9 (a) Convergence history for $k = 64$ generators (log-normal scale); (b)Energy reduction for $k = 64$ generators (log-normal scale) . . . 73
3.10 (a)Original error distribution; (b) x-component of the error after 50 Lloyd iterations; (c) y-component of the error after 50 Lloyd iterations . . . 73
3.11 (a) Convergence factor for the compatible relaxation; (b) Convergence factor of the smoother . . . 74
3.12 (a) Distribution of basis functions supports on the coarsest level; (b) Corresponding hierarchical basis functions . . . 74
3.13 (a)Convergence history of multigrid for $k = 8, 16, 32, 64$ vs. Lloyd iteration; (b) Log-normal plot of the convergence history for $k = 8, 16, 32, 64$ vs. Lloyd iteration . . . 75

4.1 (a) Correct Ca-Li-Na phase diagram; (b) Incorrect Ca-Li-Na diagram produced by Thermocalc . . . 87
4.2 Affine transformation of the axis . . . 90
4.3 Stability regions . . . 96
4.4 An example of one possible distribution of the starting points . . . 97
4.5 Complexity comparison . . . 103
4.6 Effectiveness of the quasirandom (left) vs. uniform(right) sampling in detecting concavity change. Squares denote the points of negative concavity, while dots are positive concavity regions. 50 sampling points are used for both sampling schemes. . . . 104
4.7 (a) Ca-Na diagram produced by Thermocalc; (b) Ca-Na diagram produced by the new method . . . 106
4.8 (a) Li-Na diagram produced by Thermocalc; (b) Li-Na diagram produced by the new method . . . 106
4.9 (a) Al-Zn diagram produced by Thermocalc; (b) Al-Zn diagram produced by the new method . . . 107
4.10 Gibbs energy of the Ca-Li-Na system at $T = 900K$ . . . 108

Acknowledgments

I'm greatly indebted to Professor Qiang Du for all the guidance, support and inspiration he provided me over the years, for teaching me patience, wisdom, self-confidence and being the best advisor I could ever ask for.

I'm also very grateful to Professor Ludmil Zikatanov for his kindness and help at many stages of my graduate career and for numerous valuable discussions that contributed to this thesis, and to Professor Zi-Kui Liu for proposing an exciting new field of research and for his never fading optimism.

I also wish to thank all the members of the Committee for providing valuable comments and suggestions that improved the quality of the thesis.

Finally, I thank my husband Alexei for his constant support, understanding and for always being there for me.

Chapter 1

Introduction

Optimization plays a very important role in numerous scientific and engineering applications, and is a concept that links together such different fields as mathematics, economics, biology and engineering, among others. The search for efficient methods of solving large-scale optimization problems in various contexts has continued for a long time with variable success and is still far from being complete. It is often a major challenge to require a scheme robust enough to be applicable to a wide class of objective functions yet efficient enough to attain the required accuracy of the solution.

For many problems it is typically true that the computational effort to solve a problem increases with the number of variables. In some cases, this leads to very costly, even unfeasible calculations and limits the application to a small subset of treatable problems. In recent years the development of novel multilevel and adaptive techniques has opened new research avenues in the field of numerical analysis and allowed to obtain dramatic improvement in the efficiency of the computational schemes used for some classes of problems. There have been many attempts to apply these techniques in various optimization contexts, including nonlinear opti-

mization problems, that are most frequently encountered in practical applications. Traditionally, due to their high complexity, these nonlinear problems are treated by means of one or another type of a linearization technique.

In this thesis, we adopt a more direct approach and formulate multilevel and adaptive methods for solving nonlinear minimization problems without relying on the linearization. We focus on two particular applications, that come from the fields of quantization and materials science. Both of the problems under consideration can be formulated in terms of nonlinear optimization, while each of them inherits a different set of numerical characteristics and has different challenges associated to it.

For the first application in quantization, a vector quantizer (also called Voronoi tessellation) maps N-dimensional vectors in the domain $\Omega \subset \mathbb{R}^N$ into a finite set of vectors $\{z_i\}_{i=1}^k$, which are called codewords, or generators. Each generator z_i has a region V_i (also called a Voronoi region) associated with it that consists of all points in the domain Ω that are closer to z_i than to other generators. Centroidal Voronoi tessellation, or CVT (also called optimal quantization), is a special case of a Voronoi tessellation (quantization) for which the generators themselves are the centroids of the respective Voronoi regions. Full description of the Centroidal Voronoi tessellations and associated theory are given in Section 2.1. Efficient construction of CVTs is crucial for a variety of scientific and engineering applications, such as image and data analysis, sensor networks, resource optimization and numerical partial differential equations, and the list of these applications has been growing rapidly in recent years.

Second application comes from the field of materials science, where phase diagrams are used as visual representations of the equilibrium phases in a material as a function of temperature, pressure and concentrations of the constituent compo-

nents. The mechanical, electrical and other properties of engineering materials depend strongly upon microstructure. It is therefore important that engineers should possess a basic understanding of how the microstructure is formed, and how this structure influences the engineering properties. The phase diagrams serve as basic blueprints for materials design and can be used as an aid to understand the microstructure. Similar to the quantization applications described above, there is a need for an efficient yet robust scheme that would be able to handle the complexity of the phase diagram construction without sacrificing too much of the computation time.

While both of the above problems lead to nonlinear nonconvex minimization with some constraints and they share some common features from the computational point of view, different critical issues have to be addressed. In the first problem, we look for an algorithm to accelerate existing techniques for finding optimal quantizers (CVTs). The main contribution of this thesis in this area is the development of a multilevel quantization scheme, presented in Section 3.4, that possesses a uniform convergence independent of the problem size for a large class of densities. This is the first multilevel quantization scheme in the literature with a rigorous proof of uniform convergence with respect to the grid size and the number of grid levels for nonconstant densities. The proposed scheme can be generalized to higher dimensions, and both scalar and vector versions demonstrate significant speedup comparing to the traditional Lloyd's method. To fill in the background on the subject, Chapter 2 presents a systematic study on both the local and the global convergence properties of the Lloyd algorithm. It also contains some new rigorous characterizations of the convergence of Lloyd iteration and the first proof of its global convergence in the one-dimensional case. Detailed discussions on acceleration techniques including multigrid and Newton methods are given in Chapter

3.

For the second optimization problem, this thesis presents a novel algorithm aimed at automating phase diagram construction in complex multicomponent systems. The new method focuses on finding all relevant local minimizers of the Gibbs energy functional and utilizes the geometric properties of the energy surfaces together with adaptivity and effective sampling techniques to improve on the starting points for the minimization. The new approach, presented in Section 4.3, overcomes the drawbacks of the previously known algorithms, at the same time giving comparable accuracy of the solution. When coupled with Thermocalc or used as a standalone application, it has the capabilities to automate the calculation of phase equilibria in complicated multicomponent systems, without any manual parameter adjustments. This will allow us to considerably speed up construction and analysis of phase diagrams and will consequently improve the productivity of materials research. Detailed analysis of the algorithm and numerical results are provided in Chapter 4.

Despite of the fact that considerable improvements were achieved in the numerical solution of the aforementioned nonlinear optimization problems, there are still many questions that remain to be answered. These questions and further algorithmic improvements together with other possible applications of the techniques discussed in this thesis are the subjects of future research in the area and are discussed in more details in Chapter 5.

Chapter 2

Some existing algorithms for computing CVTs and their convergence

2.1 Overview

In order to describe the computational algorithms, we begin with a brief description of the CVT. The widely used concept of the Voronoi tessellation (or Voronoi diagram) refers to a tessellation of a given domain Ω by the Voronoi regions $\{V_i\}_{i=1}^k$ associated with a set of given *generating points* or *generators* $\{\mathbf{z}_i\}_{i=1}^k \subset \Omega$ [33, 49, 62]. For each i, $\{V_i\}_{i=1}^k$ consists of all points in the domain Ω that are closer to $\mathbf{z}_i$ than to all the other generating points. A *centroidal Voronoi tessellation* (CVT) refers to a special type of Voronoi tessellation for which the generators themselves are the mass centers of their respective Voronoi regions [15], with respect to a given density function ρ. Here, the mass center of a region V with

respect to the density function ρ is defined by

$$\mathbf{z}^* = \int_V \mathbf{y}\rho(\mathbf{y})\,d\mathbf{y} \Big/ \int_V \rho(\mathbf{y})\,d\mathbf{y}\,. \tag{2.1.1}$$

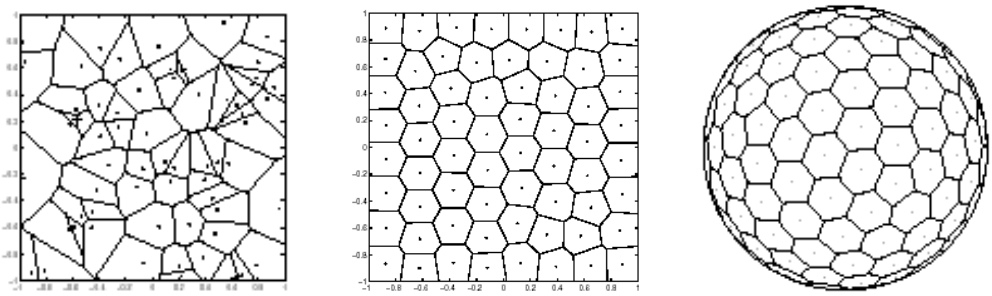


Figure 2.1. Voronoi tessellation on a square (left), CVT on a square (middle) and on a sphere (right) for constant density

CVTs are indeed special as they enjoy very natural optimization properties which make them very popular in diverse scientific and engineering applications that range from art design, astronomy, clustering, geometric modeling, image and data analysis, resource optimization, quadrature design, sensor networks, to numerical solution of partial differential equations [5, 6, 11, 13, 15, 16, 17, 19, 20, 25, 39, 42, 45, 47, 58, 68, 69]. In particular, CVTs have been widely used in the design of optimal vector quantizers in electrical engineering [34, 38, 41, 59, 67]. Figure 2.2 shows an example of the 3-d mesh based on Delaunay triangulation dual to a centroidal Voronoi tessellation (see [25, 26]). We refer to [15] for a recent review of the mathematical theory and diverse applications of CVTs.

CVTs can also be defined in more general cases such as those constrained to a manifold [18, 20, 21], or those corresponding to anisotropic metrics [26, 27], and other more abstract settings [15].

For modern applications of the CVT concept in large scale scientific and engineering problems, it is important to develop robust and efficient algorithms for

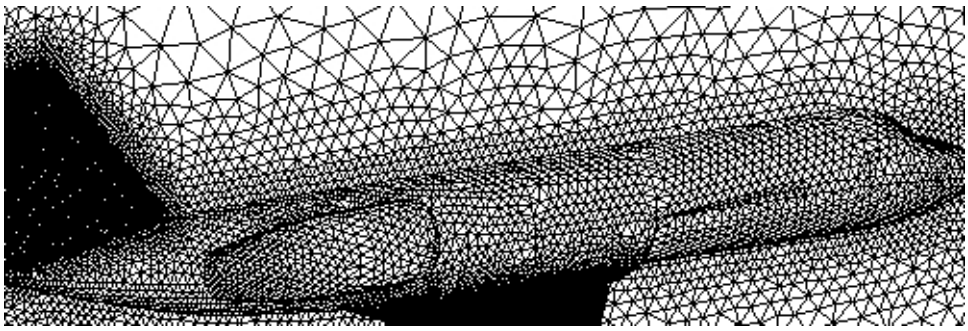


Figure 2.2. An example of a CVT-based 3-d mesh

constructing CVTs in various settings. Historically, a number of algorithms have been studied and widely used [15, 28, 33, 38, 40, 57]. A seminal work is the algorithm first developed in the 1960s at the Bell Lab by S. Lloyd which remains to this day one of the most popular methods due to its effectiveness and simplicity. The algorithm was later officially published in [53], and it is now commonly referred to as the Lloyd algorithm which is the main focus of this Chapter.

The Lloyd algorithm sparked enormous research efforts in later years and their variants have been proposed and studied in many contexts for different applications [35, 37, 38, 41, 46, 48, 52, 54, 59, 67]. A particular extension was made in [45] to combine both the deterministic features of the Lloyd algorithms with the random sampling techniques. Despite of its great success in various applications and extensive studies of its properties over the last few decades, only limited theoretical results on the Lloyd algorithm have been obtained [15] and many fundamental issues concerning its convergence remain open.

The present Chapter presents a systematic study on both the local and the global convergence properties of the Lloyd algorithm. It contains the proofs of a number of global convergence theorems there were first rigorously proved in [22]. These results include the global convergence of subsequences for any density functions, the global convergence of the whole sequence in one space dimension and

the global convergence under some non-degeneracy conditions. We also present some theoretical studies on the local convergence properties of the Lloyd algorithm including estimates on the convergence rates. Some numerical results are also presented to substantiate our theoretical investigation. Many of the techniques employed in this chapter also work for more general settings. As an illustration, we analyze the application of the Lloyd algorithm to the construction of the constrained CVTs on a manifold and present some similar convergence theorems.

The rest of the Chapter is organized as follows. We start with the description of the Lloyd algorithm in 2.2 followed by several major convergence theorems and detailed discussions of local convergence in Section 2.3. Some extensions to more general settings are considered in Section 2.4 and numerical results are given in Section 2.5. Conclusions are drawn in Section 2.6.

2.2 Lloyd's iteration

In the seminal work of Lloyd on the least square quantization [53], one of the proposed algorithms for computing the CVTs (referred to as the optimal quantizers in the particular setting) is an iterative algorithm consisting of the following simple steps: starting from an initial Voronoi tessellation corresponding to an old set of generators, a new set of generators is defined by the mass centers of the Voronoi regions. This process is continued until certain stopping criterion is met. With the notation given above, the Lloyd algorithm for constructing CVTs can be described more precisely by the following procedure

Algorithm 2.2.1. (Lloyd algorithm for computing CVTs)

Input:

Ω , *the domain of interest;* ρ , *a density function defined on* Ω *;*

k , *number of generators;* $\{\mathbf{z}_i\}_{i=1}^k$, *the initial set of generators.*

Output:

$\{V_i\}_{i=1}^k$, *a CVT with* k *generators* $\{\mathbf{z}_i\}_{i=1}^k$ *in* Ω

Iteration:

1. *Construct the Voronoi tessellation* $\{V_i\}_{i=1}^k$ *of* Ω *with generators* $\{\mathbf{z}_i\}_{i=1}^k$.
2. *Take the mass centroids of* $\{V_i\}_{i=1}^k$ *as the new set of generators* $\{\mathbf{z}_i\}_{i=1}^k$.
3. *Repeat the procedure 1 and 2 until some stopping criterion is met.*

Given a set of points $\{\mathbf{z}_i\}_{i=1}^k$ and a tessellation $\{V_i\}_{i=1}^k$ of the domain, we may define the *energy functional* or the *distortion value* for the pair $(\{\mathbf{z}_i\}_{i=1}^k, \{V_i\}_{i=1}^k)$ by:

$$\mathcal{H}\Big(\{\mathbf{z}_i\}_{i=1}^k, \{V_i\}_{i=1}^k\Big) = \sum_{i=1}^k \int_{V_i} \rho(\mathbf{y})|\mathbf{y} - \mathbf{z}_i|^2 \, d\mathbf{y} \, .$$

The minimizer of $\mathcal{H}$ necessarily forms a CVT which illustrates the optimization property of the CVT [15]. Meanwhile, it is easy to see that the Lloyd algorithm is an energy descent iteration, which gives strong indications to its practical convergence, as we show in the next section.

2.3 Convergence

2.3.1 Monotonicity properties of the energy functional

Since Lloyd's pioneering work, many studies have been made on the convergence of the iteration [31, 37, 48, 54]. For example, the local convergence has been proved for

strictly *logarithmically concave* density functions in the one dimensional space [48]. An extension to CVTs defined on a circle is given in [18]. The convergence analysis in multi-dimensional space for general density functions is far from complete. There are very few known conditions that guarantee the global convergence. We now present some new results that have not been previously explored in the literature.

For clarity, since a Voronoi tessellation is defined using a point set with k points $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^k$ as the respective generators, let us re-define the *energy functional* or the *distortion value* as a functional for a pair $(\mathbf{Y}, \mathbf{Z})$ with $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k) \in \mathbb{R}^{kN}$:

$$\mathcal{H}\Big(\mathbf{Y}, \mathbf{Z}\Big) = \sum_{i=1}^{k} \int_{V_i(\mathbf{Y})} \rho(\mathbf{y}) |\mathbf{y} - \mathbf{z}_i|^2 \, d\mathbf{y} \, .$$

where $\{V_i(\mathbf{Y})\}_{i=1}^k$ are the Voronoi regions with respect to $\{\mathbf{y}_i\}_{i=1}^k$. The Lloyd algorithm may be viewed as a fixed point iteration of the so-called Lloyd map [15], a mapping from a set of distinct generators $\{\mathbf{z}_i\}_{i=1}^k \subset \Omega \subset \mathbb{R}^N$ to the corresponding mass centers, defined by $\mathbf{T} = (\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_k)^T : \mathbb{R}^{kN} \to \mathbb{R}^{kN}$ with

$$\mathbf{T}_i(\mathbf{Z}) = \frac{\displaystyle\int_{V_i(\mathbf{Z})} \mathbf{y} \rho(\mathbf{y}) \, d\mathbf{y}}{\displaystyle\int_{V_i(\mathbf{Z})} \rho(\mathbf{y}) \, d\mathbf{y}} \, .$$

A set of generators of a centroidal Voronoi tessellation is obviously a fixed point of $\mathbf{T}$. Moreover, the Lloyd algorithm is equivalent to a fixed point iteration of $\mathbf{T}$:

$$\mathbf{Z}_n = \mathbf{T}(\mathbf{Z}_{n-1}) \, , \qquad \text{for } n \geq 1 \, .$$

Notice that in general, the map $\mathbf{T}$ can only be defined on an open subset of $\Omega^k \subset \mathbb{R}^{kN}$ as we need to ensure that the denominators are non-zero, that is, the corresponding Voronoi regions are non-empty. This, in particular, implies that the

generating points must be distinct. With this being noted, one needs to be cautious in applying general optimization theory concerning the convergence of energy descent algorithms [55] as such abstract theory often requires the compactness of the domain and the closedness of the associated map.

We now first quote some elementary facts, for which one may find more detailed discussions in [15] and [62]:

Lemma 2.3.1. *Let ρ be a positive and smooth density function defined on a smooth bounded domain Ω, then*

1). $\mathcal{H}$ is continuous and differentiable in $\bar{\Omega}^k \times \bar{\Omega}^k$;

2). $\mathcal{H}\Big(\mathbf{Z}, \mathbf{T}(\mathbf{Z})\Big) = \min_{\mathbf{Y}\in\bar{\Omega}^k} \mathcal{H}\Big(\mathbf{Z}, \mathbf{Y}\Big)$;

3). $\mathcal{H}\Big(\mathbf{Z}, \mathbf{Z}\Big) = \min_{\mathbf{Y}\in\bar{\Omega}^k} \mathcal{H}\Big(\mathbf{Y}, \mathbf{Z}\Big)$.

Next, we re-state the strong connections between the map $\mathbf{T}$, the CVTs and the Lloyd algorithm that we alluded to earlier.

Lemma 2.3.2. *Let $\{\mathbf{Z}_n\}_1^\infty$ be the sequence of generating sets produced by the Lloyd algorithm, then:*

1). $\mathbf{Z}_n = \mathbf{T}(\mathbf{Z}_{n-1})$;

2). $\mathcal{H}\Big(\mathbf{Z}_n, \mathbf{Z}_n\Big) \leq \mathcal{H}\Big(\mathbf{Z}_{n-1}, \mathbf{Z}_{n-1}\Big)$.

The first conclusion of the above lemma is obvious while the second one follows from the properties 2) and 3) of Lemma 2.3.1 (for more details, see [15]). The results of Lemma 2.3.2 imply that the distortion (energy) values decrease when they are evaluated at consecutive iterations of the Lloyd algorithm, thus, the energy functional may be viewed as a descent function of the map $\mathbf{T}$, a fact that has been explored in [64], though the notion of a closed algorithm does not readily apply here due to the possible degeneracy of the Lloyd map $\mathbf{T}$ when some of the generating points either coincide or become arbitrarily close.

It is perhaps also interesting to note that the Lloyd algorithm may be viewed as an alternating variable algorithm for minimizing the energy functional, that is, one alternates between minimizing $\mathcal{H}(\mathbf{Y}, \mathbf{Z})$ with respect to $\mathbf{Y}$ and $\mathbf{Z}$. It is well known that there are examples of simple optimization problems with special objective functions for which such an alternating variable algorithm does not always converge. It is thus interesting to see whether the special features of the functional $\mathcal{H}$ can help us to establish the convergence of the Lloyd algorithm.

2.3.2 Existence of convergent subsequence

We now present some new convergence theorems concerning the Lloyd algorithm. It has been shown in [15] that if the density function is positive except on a measure zero set, stationary points of the energy $\mathcal{H}$ are given by fixed points of the Lloyd map $\mathbf{T}$. The result below justifies that fixed points are attainable as a limit of Lloyd iterations.

Theorem 2.3.3. *Any limit point $\mathbf{Z}$ of the Lloyd algorithm is a fixed point of the Lloyd map, and thus, $(\mathbf{Z}, \mathbf{Z})$ is a critical point of $\mathcal{H}$. Moreover, for an iteration started with a given point, all elements in the set of its limit points share the same distortion value.*

Proof. The Lloyd algorithm produces a sequence $\{\mathbf{Z}_n\}$ which is bounded in $\bar{\Omega}^k$ and thus it has a convergent subsequence. Let $\mathbf{Z}$ be a limit point, then there exits a subsequence $\{\mathbf{Z}_{n_j}\}$ such that $\mathbf{Z}_{n_j} \to \mathbf{Z}$ as $n_j \to \infty$. Since the distortion values are monotonically decreasing, it follows that all limiting points must share the same distortion value.

Now, by properties of the iteration, $\mathcal{H}\Big(\mathbf{Z}_n, \mathbf{Z}_n\Big)$ is monotonically decreasing,

so

$$\mathcal{H}\Big(\mathbf{Z},\mathbf{Z}\Big) = \lim \mathcal{H}\Big(\mathbf{Z}_{n_j},\mathbf{Z}_{n_j}\Big) = \inf \mathcal{H}\Big(\mathbf{Z}_n,\mathbf{Z}_n\Big) \ .$$

On the other hand, we know from Lemma 2.3.1 that

$$\mathcal{H}_1\Big(\mathbf{U},\mathbf{Z}_n\Big) \mid_{\mathbf{U}=\mathbf{Z}_n} = 0 \ .$$

Here we use the notation $\mathcal{H}_1$ to denote the partial derivatives with respect to all the components of the first argument (gradient with respect to the first argument $\mathbf{U}$) and $\mathcal{H}_2$ (the gradient) with respect to the second one.

By continuity, we get

$$\mathcal{H}_1\Big(\mathbf{Z},\mathbf{Z}\Big) = 0 \ .$$

Now, if $\mathcal{H}_2\Big(\mathbf{Z},\mathbf{U}\Big) \mid_{\mathbf{U}=\mathbf{Z}} = 0$, $(\mathbf{Z},\mathbf{Z})$ is a critical point of $\mathcal{H}$ and we are done. Otherwise, there exists some $\mathbf{Y}$ such that

$$\mathcal{H}\Big(\mathbf{Z},\mathbf{Y}\Big) < \mathcal{H}\Big(\mathbf{Z},\mathbf{Z}\Big) \ .$$

Thus, for small enough δ, we have for large enough n_j that

$$\begin{aligned}
\mathcal{H}\Big(\mathbf{Z}_{n_j},\mathbf{Y}\Big) &< \mathcal{H}\Big(\mathbf{Z},\mathbf{Y}\Big) + \delta \\
&< \mathcal{H}\Big(\mathbf{Z},\mathbf{Z}\Big) \\
&\leq \mathcal{H}\Big(\mathbf{Z}_{n_j+1},\mathbf{Z}_{n_j+1}\Big) \\
&\leq \mathcal{H}\Big(\mathbf{Z}_{n_j},\mathbf{Z}_{n_j+1}\Big) \ .
\end{aligned}$$

This contradicts to the fact that

$$\mathcal{H}\Big(\mathbf{Z}_{n_j}, \mathbf{Z}_{n_j+1}\Big) = \min_{\mathbf{Y}} \mathcal{H}\Big(\mathbf{Z}_{n_j}, \mathbf{Y}\Big) \, .$$

Thus, the theorem is proved. □

The above theorem may be simply classified as a theorem for the global convergence of subsequences of the Lloyd algorithm. It leads to a more precise characterization of the algorithm and a hint on why it rarely fails, while also motivating the global convergence theorems for the whole sequence with some additional assumptions that we are going to present next.

2.3.3 Global convergence

As an immediate consequence of Theorem 2.3.3, we easily get the following result:

Corollary 2.3.4. *If the fixed point is unique, Lloyd algorithm converges globally.*

The uniqueness of the fixed point has been established in some special cases in the literature. We will come back to this point later in the section. The uniqueness is obviously not a necessary condition, but we may in fact derive the following convergence theorem:

Theorem 2.3.5. *If the set of fixed points with any particular distortion value is finite, the Lloyd algorithm converges globally.*

Proof. Convergence may fail only if the generated sequence possesses infinitely many jumps from a neighborhood of one fixed point to another. Suppose $\mathbf{U}$ and $\mathbf{V}$ are two fixed points with $||\mathbf{U} - \mathbf{V}|| = \delta > 0$. Denote the generated sequence of the Lloyd algorithm as $\mathbf{Z}_n$, i.e. $\mathbf{Z}_{n+1} = \mathbf{T}(\mathbf{Z}_n)$.

Suppose $\mathbf{Z}_{n_r} \to \mathbf{U}$ and $\mathbf{Z}_{n_l} \to \mathbf{V}$. Then for any $\delta > 0$, there exists $M > 0$, such that for all $n_r, n_l > M$ we have $||\mathbf{Z}_{n_r} - \mathbf{U}|| < \delta/3$ and $||\mathbf{Z}_{n_l} - \mathbf{V}|| < \delta/3$. Lloyd map is continuous near the fixed points (see Proposition 3.5, [15]), so M can be chosen to be suitably large to assure

$$||\mathbf{T}(\mathbf{Z}_{n_r}) - \mathbf{Z}_{n_r}|| < \delta/3.$$

Now suppose the sequence makes infinitely many jumps from subsequence $\{n_r\}$ to $\{n_l\}$, i.e. there are infinitely many μ, ν s.t. $n_{l_\mu} = n_{r_\nu} + 1$. Then $||\mathbf{T}(\mathbf{Z}_{n_{r_\nu}}) - \mathbf{V}|| = ||\mathbf{Z}_{n_{r_\nu}+1} - \mathbf{V}|| = ||\mathbf{Z}_{n_{l_\mu}} - \mathbf{V}||$. Hence

$$\delta = ||\mathbf{U} - \mathbf{V}|| \leq ||\mathbf{U} - \mathbf{Z}_{n_{r_\nu}}|| + ||\mathbf{Z}_{n_{r_\nu}} - \mathbf{T}(\mathbf{Z}_{n_{r_\nu}})|| + ||\mathbf{T}(\mathbf{Z}_{n_{r_\nu}}) - \mathbf{V}|| < \delta.$$

We get a contradiction. □

To this end, we have proved the global convergence of the Lloyd method in case the set of fixed points, Γ, does not have an accumulation point. Note that there are situations where Γ contains accumulation points and all points in Γ share the same distortion value. For example, consider the CVTs formed with two generators in a unit disc centered at the origin for the constant density function, simple calculation shows that the critical points fill a circle of radius $4/(3\pi)$. That is, due to the rotation symmetry, any pair of points in the opposite ends of such a circle determines a CVT and all the critical points share the same energy values. Of course, cases like this are very rare, so this fact does not present any difficulties for the convergence of the Lloyd algorithm in most practical applications.

We now present another result which further substantiates the global convergence of Lloyd algorithm in general.

Theorem 2.3.6. *If the iterations in the Lloyd algorithm stay in a compact set where the Lloyd map* $\mathbf{T}$ *is continuous, then the algorithm is globally convergent to a critical point of* $\mathcal{H}$ *.*

Proof. The proposition follows from the Global Convergence Theorem (GCT) [55] and similar arguments have been made in [64]. Indeed, Lloyd algorithm can be regarded as a descent method with the descent function given by $\mathcal{H}(\cdot, \mathbf{T}(\cdot))$. Let $\{\mathbf{Z}_n\}_{n=1}^{\infty}$ be a sequence generated by $\mathbf{Z}_{n+1} = \mathbf{T}(\mathbf{Z}_n)$. All $\mathbf{Z}_n$'s are contained in a compact set. If Γ is the set of solutions, $\mathcal{H}(\mathbf{Y}, \mathbf{T}(\mathbf{Y})) < \mathcal{H}(\mathbf{Z}, \mathbf{T}(\mathbf{Z}))$ for all $\mathbf{Z} \notin \Gamma$, $\mathbf{Y} \in \mathbf{T}(\mathbf{Z})$ and $\mathcal{H}(\mathbf{Y}, \mathbf{T}(\mathbf{Y})) = \mathcal{H}(\mathbf{Z}, \mathbf{T}(\mathbf{Z}))$ for all $\mathbf{Z} \in \Gamma$, $\mathbf{Y} \in \mathbf{T}(\mathbf{Z})$. The continuity implies the closedness of $\mathbf{T}$ in a compact set. Applying the GCT, we get the convergence of the sequence $\mathbf{Z}_n$ and the limit $\mathbf{Z}$ is a fixed point of $\mathbf{T}$, thus, the algorithm converges to a critical point of $\mathcal{H}$. □

We note that the compactness of the iteration seems to be intuitively true but it has not been rigorously justified in the literature. The difficulty is related to showing that during the iteration, the generators of the Voronoi regions do not get arbitrarily close as the Lloyd map is not well defined at degenerating points where some of the generators may coincide.

2.3.4 The compactness in the one dimensional case

Here, we take $\Omega = [a, b]$, a compact interval, let ρ be smooth and positive and assume that $0 < M_1 \leq ||\rho||_{\infty,\Omega} \leq M_2 < \infty$. Let $M_c = M_2/M_1$, obviously, $M_c \geq 1$. We verify that throughout the Lloyd algorithm, the Voronoi regions remain non-degenerate, (i.e., the generating points remain distinct), thus, it will lead to the global convergence.

First, we have the following simple fact:

Lemma 2.3.7. *Given an interval* $V = [z_l, z_r] \in \Omega$ *and let* z^* *be the mass centroid of* V *with respect to the density function* ρ*, then we have*

$$L(V) \leq 2M_c \min(z^* - z_l,\ z_r - z^*) \tag{2.3.2}$$

where $L(V)$ *denotes the length of* V*.*

Proof. Without loss of generality, we suppose that $z^* - z_l \leq z_r - z^*$. By the definition of mass centroid, we have

$$z^* - z_l = \frac{\int_{z_l}^{z_r} (x - z_l)\rho(x)\, dx}{\int_{z_l}^{z_r} \rho(x)\, dx} \geq \frac{M_1}{2M_2}(z_r - z_l),$$

so we get

$$z_r - z_l \leq 2M_c(z^* - z_l).$$

With $z^* - z_l \leq z_r - z^*$, we get the inequality (2.3.2). □

Denote by $\{z_i^{(n)}\}_{i=1}^k$ $(z_1^{(0)} < z_2^{(0)} < \cdots < z_k^{(0)},\ n \geq 0)$ the positions of the generators after n iterations in Lloyd method and by $\{V_i^{(n)} = (y_{i-1}^{(n)}, y_i^{(n)})\}_{i=1}^k$ the corresponding Voronoi regions. Clearly, $y_0^{(n)} = a$ and $y_k^{(n)} = b$. We now present a non-degeneracy result:

Lemma 2.3.8. *For any* $1 < i < k$*, we have*

$$L(V_i^{(n+1)}) < \min\Big(\frac{L(V_i^{(n)}) + L(V_{i+1}^{(n)})}{2} + L(V_{i-1}^{(n+1)}),\ \frac{L(V_i^{(n)}) + L(V_{i-1}^{(n)})}{2} + L(V_{i+1}^{(n+1)})\Big).$$

Proof. First we have

$$L(V_i^{(n+1)}) = \frac{z_{i+1}^{(n+1)} - z_i^{(n+1)}}{2} + \frac{z_i^{(n+1)} - z_{i-1}^{(n+1)}}{2}.$$

Since $z_i^{(n+1)} \in V_i^{(n)}, z_{i+1}^{(n+1)} \in V_{i+1}^{(n)}$, we know

$$\frac{z_{i+1}^{(n+1)} - z_i^{(n+1)}}{2} < \frac{L(V_i^{(n)}) + L(V_{i+1}^{(n)})}{2}.$$

With $L(V_{i-1}^{(n+1)}) > (z_i^{(n+1)} - z_{i-1}^{(n+1)})/2$, we get

$$L(V_i^{(n+1)}) < \frac{L(V_i^{(n)}) + L(V_{i+1}^{(n)})}{2} + L(V_{i-1}^{(n+1)}). \tag{2.3.3}$$

Similarly, we can prove that

$$L(V_i^{(n+1)}) < \frac{L(V_i^{(n)}) + L(V_{i-1}^{(n)})}{2} + L(V_{i+1}^{(n+1)}). \tag{2.3.4}$$

Combining (2.3.3) and (2.3.4), we complete the proof. □

This leads to the following uniform lower bound between the adjacent generators throughout the Lloyd algorithm:

Proposition 2.3.1. *Let* $d_i^{(n)} = z_{i+1}^{(n)} - z_i^{(n)}$ *for* $i = 1, 2, \cdots, k-1$. *then we have*

$$d_i^{(n)} > \frac{b-a}{k4^{2k-1}M_c^k}, \qquad n > k, \tag{2.3.5}$$

and consequently,

$$L(V_i^{(n)}) > \frac{b-a}{k4^{2k-1}M_c^k}, \quad 1 < i < k, \quad n > k \tag{2.3.6}$$

and

$$L(V_i^{(n)}) > \frac{b-a}{2k4^{2k-1}M_c^k}, \quad i = 1 \text{ or } k, \quad n > k \tag{2.3.7}$$

Proof. Let us consider any $d_i^{(n)}$ for $1 \leq i \leq k-1$ and $n > k$. Since $d_i^{(n)} = z_{i+1}^{(n)} - z_i^{(n)}$ and $y_i^{(n-1)} < z_{i+1}^{(n)}$, we have

$$y_i^{(n-1)} - z_i^{(n)} < d_i^{(n)} .$$

Then from Lemma 2.3.7, we have

$$L(V_i^{(n-1)}) < 2M_c d_i^{(n)} . \tag{2.3.8}$$

On the other hand , we know $L(V_i^{(n-1)}) > (z_{i+1}^{(n-1)} - z_i^{(n-1)})/2$ which means

$$d_i^{(n-1)} < 2L(V_i^{(n-1)}) < 4M_c d_i^{(n)} .$$

Again by Lemma 2.3.7, we know

$$L(V_{i-1}^{(n-2)}) < 8M_c^2 d_i^{(n)} .$$

Repeat this process, we have for $j = 1, \cdots, i$,

$$L(V_{i-j+1}^{(n-j)}) < 2^{2j-1}M_c^j d_i^{(n)} .$$

Now let us consider $j = i$. Clearly, $V_1^{(n-i)} = (a, y_1^{(n-i)})$, and we have

$$\begin{aligned} L(V_1^{(n-i+1)}) &< L(V_1^{(n-i)}) + L(V_2^{(n-i+1)}) \\ &< 2^{2i-1} M_c^i d_i^{(n)} + 2^{2i-3} M_c^{i-1} d_i^{(n)} \\ &< 4^i M_c^i d_i^{(n)} . \end{aligned}$$

Furthermore, by Lemma 2.3.8, we get

$$\begin{aligned} L(V_2^{(n-i+2)}) &< \frac{L(V_2^{(n-i+1)}) + L(V_1^{(n-i+1)})}{2} + L(V_3^{(n-i+2)}) \\ &< \frac{2^{2i-3} M_c^{i-1} d_i^{(n)} + 4^i M_c^i d_i^{(n)}}{2} + 2^{2i-5} M_c^{i-2} d_i^{(n)} \\ &< 4^i M_c^i d_i^{(n)} . \end{aligned}$$

Repeat this process, we have for $j = 1, \cdots, i-1$,

$$L(V_j^{(n-i+j)}) < 4^i M_c^i d_i^{(n)} ,$$

which means

$$L(V_{i-1}^{(n-1)}) < 4^i M_c^i d_i^{(n)} .$$

Using the same trick again and again, we finally arrive at

$$L(V_{i-j}^{(n-1)}) < 4^{i+j-1} M_c^i d_i^{(n)}, \quad j = 1, \cdots, i-1.$$

Combining (2.3.8) and and the above equation with $i, j \le k$, we get

$$L(V_j^{(n-1)}) < 4^{2k-1} M_c^k d_i^{(n)}, \qquad j = 1, \cdots, i . \tag{2.3.9}$$

By symmetry, we also have

$$L(V_j^{(n-1)}) < 4^{2k-1} M_c^k d_i^{(n)}, \qquad j = i+1, \cdots, k\ .$$

Then, we get

$$b - a = L(\Omega) = \sum_{j=1}^{k} L(V_j^{(n-1)}) < k 4^{2k-1} M_c^k d_i^{(n)},$$

which implies (2.3.5), (2.3.6) and (2.3.7). □

We then have

Theorem 2.3.9. *For any positive and smooth density function in 1d and a given set of k distinct generators as a starting point, the Lloyd map is continuous at any of the iteration points.*

Proof. In order to show the continuity it is enough to justify the fact that Voronoi cells do not collapse. Indeed, after sufficient number of steps, the latter is the direct consequence of Proposition 2.3.1. For the initial finite number of iterations, the continuity is obvious. □

Finally, using Theorems 2.3.6 and 2.3.9, we get

Theorem 2.3.10. *Lloyd algorithm is globally convergent in 1d for any positive and smooth density function.*

Proof. Using the result of Theorem 2.3.9, we see that we can define a compact set (away from the degenerating points) such that for any initial condition, the Lloyd iteration (the images of the Lloyd maps) will stay in such a compact set after sufficiently many steps. Thus, we may apply the Theorem 2.3.6 to deduce the convergence of the algorithm. □

The above theorem provides an affirmative answer to the global convergence of the Lloyd algorithm for the one dimensional interval case without any restrictive assumptions on the density functions. It remains open to verify the same conclusion in the multidimensional case.

2.3.5 The logarithmic concave density in the one dimensional case

Beyond the study on the global convergence, the characterization of the convergence rate is often also important in practice. For instance, one may inquire if a geometric convergence rate can be established. This is indeed verified in [15] for the constant density function corresponding to the unit interval $[0,1]$, where, via the spectral analysis of $\mathbf{dT}$ at the minimizer, the established geometric convergence rate r is shown to satisfy

$$\sin^2(\frac{\pi}{2(k+1)}) \leq r \leq \sin^2(\frac{\pi}{2(k-1)}) \,, \tag{2.3.10}$$

so that asymptotically for large k (the total number of generators) the convergence rate is on the order of $1-\pi^2/(4k^2)$, as verified by the numerical experiments in the next section.

In general, finding the convergence rate exactly is not possible but estimates may be obtained from the analytical bounds of the $\|\mathbf{dT}\|$.

First, it follows from Theorem 2.3.9 that $\mathbf{T}:\Omega^k \to \Omega^k$ is a continuously differentiable mapping away from the degenerate points where the generating points collapse. If this mapping $\mathbf{T}$ is a contraction, i.e. $||\mathbf{dT}|| < 1$ at all nondegenerate points, the Contraction Mapping Theorem can be used to get a good estimate of

the local convergence rate for the corresponding fixed point iteration, which in our case is the Lloyd algorithm. Moreover, the contraction mapping properties also imply that $\mathbf{T}$ has a unique fixed point $\mathbf{z}^*$ in the set of nondegenerate points upon a consistent ordering. Indeed, if there existed two fixed points $\mathbf{x} = \{x_i\}_{i=1}^k$ and $\mathbf{y} = \{y_i\}_{i=1}^k$, with components corresponding to generating points whose coordinates are ordered from small to large, that is $x_i < x_{i+1}$ and $y_i < y_{i+1}$ for all indices i, then any point along the line segment $(1-t)\mathbf{x} + t\mathbf{y}$ would remain nondegenerate and thus, by uniform continuity, we may assume that

$$\sup_{0\le t\le 1} ||\mathbf{dT}(\mathbf{x} + t(\mathbf{y} - \mathbf{x}))|| \le \alpha(\mathbf{x}, \mathbf{y}) < 1$$

for some constant $\alpha(\mathbf{x}, \mathbf{y})$ independent of t. From the multidimensional form of the mean value theorem, we then get

$$||\mathbf{x} - \mathbf{y}|| = ||\mathbf{Tx} - \mathbf{Ty}|| \le \sup_{0\le t\le 1} ||\mathbf{dT}(\mathbf{x} + t(\mathbf{y} - \mathbf{x}))||\, ||\mathbf{x} - \mathbf{y}|| \le \alpha ||\mathbf{x} - \mathbf{y}||\,,$$

which is only possible if $\mathbf{x} = \mathbf{y}$, thus, we have the uniqueness. We refer to [48] for a similar discussions.

The concept of logarithmic concavity has played an important role in the classification of one dimensional density functions since it is a class of density functions for which the Lloyd maps can be shown to be contractions [15].

Let us take a closer look at the structure of the Jacobian $\mathbf{dT}$. By the notations of the previous section, for the 1d case (i.e., $\Omega = [a, b]$), we have

$$\frac{\partial T_i}{\partial z_i} = \frac{\partial T_i}{\partial z_{i-1}} + \frac{\partial T_i}{\partial z_{i+1}},$$

$$\frac{\partial T_i}{\partial z_{i-1}} = \frac{\rho(z_i^-)(T_i - z_i^-)}{2R_i} \text{ and } \frac{\partial T_i}{\partial z_{i+1}} = \frac{\rho(z_i^+)(z_i^+ - T_i)}{2R_i} \tag{2.3.11}$$

where $R_i = \int_{V_i} \rho(y)dy$ and $V_i = [z_i^-, z_i^+]$.

The following useful relation may be found in [15, 37]:

$$R_i^2\big(1 - \sum_j \frac{\partial T_i}{\partial z_j}\big) = \frac{1}{2}\int_{V_i}\int_{V_i} \rho(t)\rho(s)\Big(\frac{\rho'(s)}{\rho(s)} - \frac{\rho'(t)}{\rho(t)}\Big)(t-s)dt\ ds \tag{2.3.12}$$

at a fixed point $\mathbf{z} = \mathbf{T}(\mathbf{z})$.

Based on this, it can be shown that for the class of logarithmically concave functions (i.e., $(\log\rho)'' < 0$), the spectral radius of the Jacobi map is less than 1 in the neighborhood of a fixed point. In fact, it is easy to show that the same estimate holds for all points as the identity (2.3.12) remains universally true. Hence the fixed point of the Lloyd map is unique when the generators are ordered in increasing manner. The following convergence of the Lloyd algorithm for the logarithmically concave case is easily one of the most popular results studied in the literature.

Proposition 2.3.2. *In 1d, in case of logarithmically concave density, the Lloyd algorithm converges globally to the unique fixed point.*

The class of logarithmically concave functions covers many densities used in practice, for instance, linear densities and normal distributions. Notice that the result quoted in Proposition 2.3.2 does not provide the estimate of the actual distance of the spectral radius from 1. We now focus on getting estimates on $\theta = 1 - ||\mathbf{dT}||$ more accurately. For this, we use a more precise measure of the logarithmic concavity for the density, that is, we assume that:

$$\rho(t)\rho(s)\Big(\frac{\rho'(s)}{\rho(s)} - \frac{\rho'(t)}{\rho(t)}\Big)(t-s) \geq c_0^2(t-s)^2 \tag{2.3.13}$$

for some constant $c_0 > 0$ and any (t, s) except for a set of measure zero. Upon availability of an estimate of this type, the following conclusion can be reached

$$1 - ||\mathbf{dT}|| \geq c_0^2 \min_i \{R_i^{-2} \int_{V_i} \int_{V_i} (t-s)^2 dt ds\} \sim \frac{c_0^2}{12} \min\{\frac{h_i^2}{\rho(\zeta_i)^2}\}$$

for some $\zeta_i \in V_i$ and $h_i = z_i^+ - z_i^-$. Let $h = \min_i h_i$, the smallest Voronoi cell size, and $M = \sup_{x\in[0,1]} \rho(x)$, then we can rewrite the above result as:

Lemma 2.3.11. *For any smooth density ρ satisfying (2.3.13) on the unit interval, the Lloyd algorithm is globally convergent with a geometric convergence rate no larger than*

$$||\mathbf{dT}|| \leq 1 - \frac{c_0^2}{12} \frac{h^2}{M^2} \,. \tag{2.3.14}$$

Convergence estimate obtained here essentially depends on characteristics c_0 and the relative size of a Voronoi cell in comparison with the density distribution. Since the minimizer of the energy gives a non-degenerate Voronoi diagram (Proposition 3.5 in [15]), there is a positive lower bound for the distance h in the neighborhood of the solution in terms of the density and the number of generators. Moreover, for large k, due to the asymptotic equi-partition of energy property in 1d [15], after sufficiently many iterations, one can roughly estimate each cell size as

$$h_i \sim k^{-1} \rho(\zeta_i)^{-1/3} \int_0^1 \rho^{1/3}(x) dx \,.$$

Thus, we have effectively $\theta = 1 - ||\mathbf{dT}|| \geq \left(\frac{c_1}{k}\right)^2$, where for large k,

$$c_1 \sim \frac{c_0}{\sqrt{12} M^{4/3}} \int_0^1 \rho^{1/3}(x) dx \,. \tag{2.3.15}$$

The estimate (2.3.15) in general tends to be rather pessimistic, for instance, for a linear perturbation of the constant density $\rho(x) = 1 - \epsilon x$ for a small ϵ, we have $c_1 \sim \frac{3}{4\sqrt{12}}(1-(1-\epsilon)^{4/3})$ which is significantly different from $\pi/2$ in the limit as $\epsilon \to 0$ (for constant density case, c_1 can be estimated more accurately from the estimate (2.3.10) as $\pi/2$). This is due to the fact that the class of constant densities shares zero value of the parameter c_0. However, it allows us to reach the conclusion that the geometric convergence rate for all densities satisfying (2.3.13) is comparable with that of the constant density in the sense that θ remains to be of the order k^{-2} for large values of k.

We expect that such conclusion holds for even more general density functions, but the rigorous analysis is still not available.

2.4 Extensions to constrained CVTs

We now make a brief illustration that much of our earlier analysis can be extended to more general settings where the concept of CVTs can be defined. The example to be used is the constrained CVTs on general surfaces as defined in [18].

Consider a compact and smooth surface $\mathbf{S} \subset \mathbb{R}^N$. Similar to the definition of conventional CVTs, for a given set of points $\{\mathbf{z}_i\}_{i=1}^k \in \mathbf{S}$, one may define their corresponding Voronoi regions on $\mathbf{S}$ by

$$V_i = \{\, \mathbf{x} \in \mathbf{S} \ : \ |\mathbf{x} - \mathbf{z}_i| < |\mathbf{x} - \mathbf{z}_j| \quad \text{for } j = 1, \ldots, k,\, j \neq i \,\}. \tag{2.4.16}$$

For a density function ρ defined on the surface $\mathbf{S}$ and positive almost everywhere, one may encounter a problem with the original definition when one defines centroidal Voronoi tessellations $\{(\mathbf{z}_i, V_i)\}_{i=1}^k$ of $\mathbf{S}$: the mass centroids $\{\mathbf{z}_i^*\}_{i=1}^k$ of

$\{V_i\}_{i=1}^k$ as defined by (2.1.1) do not in general belong to $\mathbf{S}$. For example, the mass centroid of any region on the surface of a sphere is always located in the interior of the sphere. Therefore, a generalized definition of a mass centroid on surfaces is needed. For each Voronoi region $V_i \subset \mathbf{S}$, we call $\mathbf{z}_i^c$ the *constrained mass centroid* of V_i on $\mathbf{S}$ if $\mathbf{z}_i^c$ is a solution of the following problem:

$$\min_{\mathbf{z}\in\mathbf{S}} F_i(\mathbf{z}), \qquad \text{where} \qquad F_i(\mathbf{z}) = \int_{V_i} \rho(\mathbf{x})|\mathbf{x}-\mathbf{z}|^2\, d\mathbf{x}. \tag{2.4.17}$$

The integral over $\{V_i\}$ is understood as standard surface integration on $\mathbf{S}$. Note that the constrained mass centroid coincides with the conventional mass center if $\mathbf{S}$ is replaced by $\mathbb{R}^N$ and V_i is a convex subset of $\mathbb{R}^N$. Clearly, for each $i = 1, \ldots, k$, $F_i(\cdot)$ is convex. Since $\mathbf{S}$ is compact and $\rho(\cdot)$ is continuous almost everywhere, there exists a constant C such that for any $\mathbf{z}_1, \mathbf{z}_2 \in \mathbf{S}$, we have

$$|F_i(\mathbf{z}_1) - F_i(\mathbf{z}_2)| = \left| \int_{V_i} \rho(\mathbf{x})(|\mathbf{x}-\mathbf{z}_1|^2 - |\mathbf{x}-\mathbf{z}_2|^2)\, d\mathbf{x} \right| \le C|\mathbf{z}_1 - \mathbf{z}_2|.$$

Thus, F_i is continuous and compact, and consequently we have the existence of solutions of (2.4.17), although the solution may not be unique.

We call the tessellation defined by (2.4.16) a *constrained centroidal Voronoi tessellation* (CCVT) if and only if the points $\{\mathbf{z}_i\}_{i=1}^k$ which serve as the generators associated with the Voronoi regions $\{V_i\}_{i=1}^k$ are the constrained mass centroids of those regions [18]. This definition of CCVT conforms with that of CVT for general spaces and clearly the energy $\mathcal{H}$ defined in (2.4.17) for CVTs is still valid for CCVTs. In Figure 2.3, we give two examples of CCVTs, one with six generators constrained to a circle (one dimensional curve), and the other with 162 generators constrained to a sphere (two dimensional surface), both correspond to the constant

density.

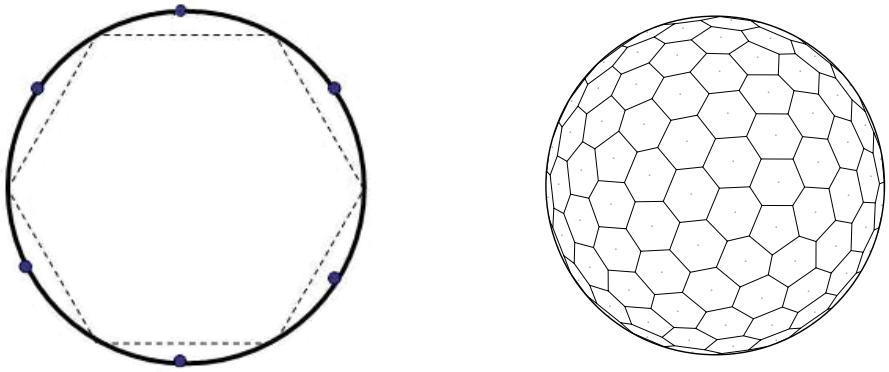


Figure 2.3. Examples of constrained CVTs (CCVTs) for a circle (dots are for generators and dash lines show the partition of the constrained Voronoi regions) and for a sphere (dots are generators, lines are planar projections of Voronoi edges, only portion in one hemisphere is shown).

The generalized Lloyd algorithm for computing CCVTs was proposed in [18]:

Algorithm 2.4.1. (Lloyd algorithm for computing CCVTs)

Input:

$\mathbf{S}$, *the surface of interest;* ρ, *a density function defined on* $\mathbf{S}$;

k, *number of generators;* $\{\mathbf{z}_i\}_{i=1}^k$, *the initial set of generators.*

Output:

$\{V_i\}_{i=1}^k$, *a CCVT with* k *generators* $\{\mathbf{z}_i\}_{i=1}^k$ *in* $\mathbf{S}$.

Iteration:

1. Construct the Voronoi tessellation $\{V_i\}_{i=1}^k$ *of* $\mathbf{S}$ *with generators* $\{\mathbf{z}_i\}_{i=1}^k$.

2. Take the Constrained mass centroids of $\{V_i\}_{i=1}^k$ *as the new set of generators* $\{\mathbf{z}_i\}_{i=1}^k$.

3. Repeat the procedure 1 and 2 until some stopping criterion is met.

It is clear that Algorithm 2.4.1 is almost identical to Algorithm 2.2.1 except the constrained mass centroids are used instead of standard mass centroids in the

step 2 of each iteration. So the Algorithm 2.4.1 again can be regarded as a fixed point iteration of $\mathbf{T}$, the Lloyd map for CCVTs which now is defined to map the current generators to the constrained mass centroids of the corresponding Voronoi regions. It is transparent that the analysis done in Sections 2.1 and 2.2 can all be applied here, so we obtain the following general results similar to Theorems 2.3.3 and 2.3.5:

Theorem 2.4.1. *Any limit point* $\mathbf{Z}$ *of the Lloyd algorithm for computing CCVTs is a fixed point of the Lloyd map for CCVTs, and thus,* $(\mathbf{Z}, \mathbf{Z})$ *is a stationary point of* $\mathcal{H}$*. The set of limit points share the same distortion value* $\mathcal{H}$ *for a given iteration. Furthermore, if the set of fixed points with the same distortion value is finite, the Lloyd iteration for CCVTs converges globally.*

Now suppose that $\mathbf{S}$ is a smooth curve without self-intersection such as $\mathbf{S} = f(\Omega)$ where $\Omega = [a, b]$ for some smooth function f, then using the analysis similar to that provided in Section 2.3, we obtain the following result:

Theorem 2.4.2. *The Lloyd algorithm for computing CCVTs of* $\mathbf{S}$ *is globally convergent for any positive and smooth density function when* $\mathbf{S}$ *is a bounded smooth curve.*

Note that unlike the one dimensional conventional CVT in $\mathbb{R}^1$, we have not given any general estimate here on the convergence rate of the Lloyd algorithm for CCVTs. Even for the case where $\mathbf{S}$ is a bounded smooth curve, the geometric convergence rate has not been carefully derived, though the notion of contraction for the Lloyd map has been studied for density functions which share similar logarithmic concave properties with respect to the angular variable in the case of a perfect disc [18]. There are also natural generalizations of the Lloyd algorithm to the anisotropic CVTs as defined in [26] and also [27]. The details are omitted here.

2.5 Numerical examples

To further substantiate some of our earlier analysis, we now present a few numerical examples. All examples given below correspond to the Lloyd iteration on the interval $[0, 1]$.

2.5.1 Constant density

In Figure 2.4, we show a log-log plot of both the numerical estimates and the analytical estimate $1 - ||\mathbf{dT}|| \sim \pi^2/(4k^2)$ with respect to the constant density, for various values of k, the number of generating points. The two estimates match very well and the results verify that the analytical estimates are very sharp.

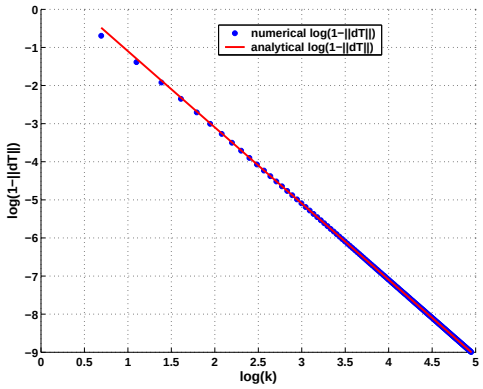


Figure 2.4. Convergence of Lloyd method for constant density.

2.5.2 Non-constant density

Consider the case of $\rho(x) = e^{-x^2}$. Figure 2.5 compares the analytical estimate with the computed norms of the Jacobian for different system sizes. Here, the analytical estimate is based on $c_1^2 k^{-2}$ with the constant c_1 estimated by (2.3.15) with $c_0 = \sqrt{2/e}$, $M = 1$ and $\int_0^1 \rho^{1/3}(x)dx = \sqrt{3\pi} \cdot \mathrm{Erf}(1/\sqrt{3})/2$ which leads to $c_1 = \sqrt{\pi} \cdot \mathrm{Erf}(1/\sqrt{3})/2e \sim 0.19$. The plot is again given in log-log scale, and we

see that although we underestimated the exact value of c_1, the slope was equal to -2 for both estimates, which indicates the good agreement of the asymptotic rates on the order of $1 - O(1/k^2)$.

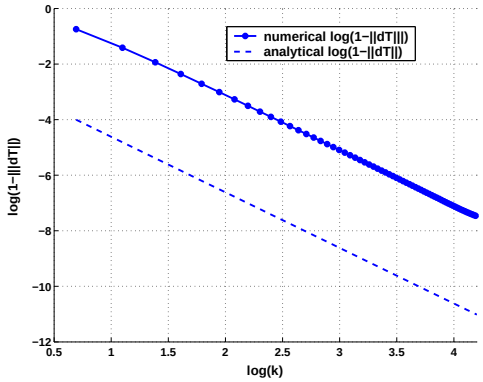


Figure 2.5. Convergence factor of Lloyd method for $\rho(x) = e^{-x^2}$.

Figure 2.6 gives a similar comparison for $\rho(x) = 1 + x^4 cos(\pi x)$. The numerical data in this case was compared to the asymptotic rate of $1 - \pi^2/4k^2$.

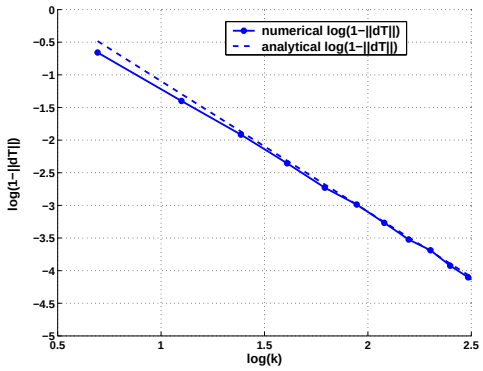


Figure 2.6. Convergence factor of Lloyd method for $\rho(x) = 1 + x^4 cos(\pi x)$.

Figures 2.7-2.9 provide some insight into the dependence of the actual convergence factor on the number of generators and on the density function. The convergence factor in the plot is defined as the ratio of the 2-norm defects between two consecutive iterations after sufficiently many steps. A density function of the form $\rho(x) = 1 + \epsilon \cos^2(\pi x)$ is chosen. In Figure 2.7, we fix the number of generators to be $k = 16$, while letting ϵ vary in the range $[10^{-10}, 10^{10}]$. It is seen that

the actual convergence factor and the theoretical estimate given by $||\mathbf{dT}||$ agree well in general.

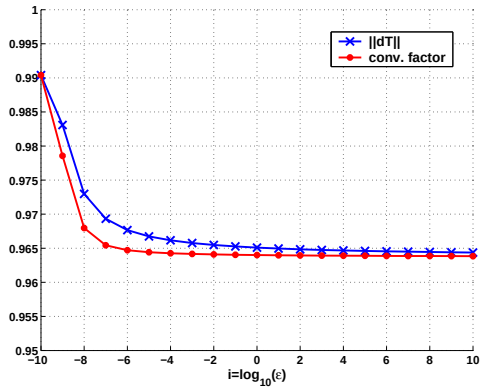


Figure 2.7. Convergence factor for $k = 16$ and $\rho(x) = 1 + \epsilon \cos^2(\pi x)$ with $\epsilon = 10^{-10} : 10^{10}$.

To see the effect of the increasing k, in Figure 2.8, we fix ϵ and let the number of generators vary. The two estimates again compare well with each other.

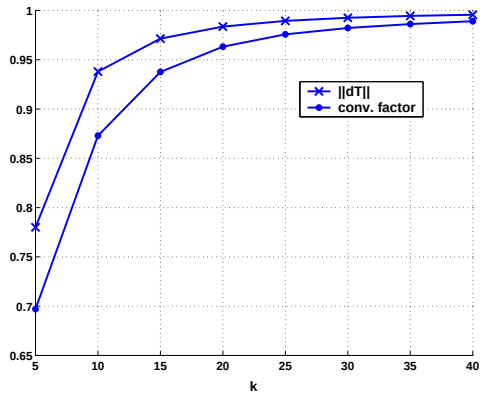


Figure 2.8. Convergence factor for $\rho(x) = 1 + 10^3 \cos^2(\pi x)$ and $k = 2 : 40$.

To see more clearly the dependence of convergence rates on k, we again plot the data in a log-log scale for the density $\rho(x) = 1 + 10^3 \cos^2(\pi x)$ against the number of generators. The slope value of -2 is very evident from the picture, which is consistent with our earlier analysis.

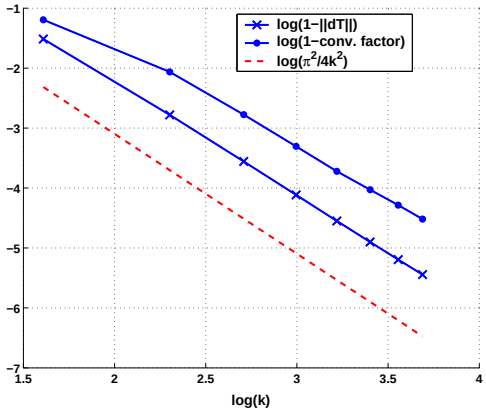


Figure 2.9. Asymptotic behavior of the convergence factor for $\rho(x) = 1+10^3\cos^2(\pi x)$.

2.6 Conclusions

In many practical applications of the centroidal Voronoi tessellations, it is very important to find their reliable and efficient constructions. Lloyd algorithm has been one of the most widely used techniques for such purposes. In this chapter, a systematic study on both the local and the global convergence properties of the Lloyd algorithm was presented. We carried out the proofs of several recent convergence theorems, made further characterization on the properties of the iteration, established geometric convergence rate for a wider class of functions and performed relevant numerical experiments. We also extended our discussion to more general settings such as the construction of the constrained CVTs on a manifold. Still, one important open question remains, that is, the global convergence of the Lloyd algorithm in any dimensions for any smooth density. The non-degeneracy of the Lloyd map should be true in this general case, but its proof has not been produced rigorously except for the one-dimensional case discussed here. We hope that our present study generates further interest along this direction as there are certainly many issues that remain to be considered. In particular, the improvement of the Lloyd method for large number of generators. Even in the one-dimensional case,

both our theoretical estimates and the experiments indicate the possible slow convergence rates. The next chapter presents the results of our recent work toward such improvements. There are two major directions of this work, so the chapter can be roughly divided into two parts. The first one explores the coupling of Lloyd iteration with Newton-like methods, while the second one introduces the ideas of multigrid in the quantization setting. As previously studied in [45], one may also consider parallelization issues for these approaches, that we also briefly discuss in Chapter 3.

Chapter 3

New algorithms for the construction of CVTs

3.1 Overview

The evidence of slow convergence of the Lloyd iteration and its descent properties motivated our search for a Lloyd iteration based numerical scheme with superior convergence properties. One may view finding the mass centers of the CVT as a problem of solving a nonlinear system of equations. Thus, Newton's method is a natural candidate for performing such a task. However, the cost of inverting the Jacobian can be expected to raise the complexity of such a scheme, while the convergence region of the Newton iteration is not guaranteed to be large enough to provide the desired acceleration.

In the first half of this chapter, we design a Newton-based algorithm that represents a coupling of the global descent properties of the Lloyd iteration with the acceleration provided by the Newton iteration within its convergence radius. In Section 3.2.2 we provide several theoretical estimates on the convergence of the

associated quasi-Newton algorithm using the results of Chapter 2. In Section 3.2.3 we provide a detailed description of the proposed algorithm followed by the results of numerical experiments in Section 3.2.7. These results demonstrate how we can achieve the best performance of the proposed method and discuss the improvement of convergence rate it provides for the Lloyd iteration. Computational complexity estimates are provided in Section 3.2.5.

The second part is dedicated to another possible approach to the problem of accelerating convergence of the Lloyd's method. In Section 3.4.2 we give the details of the new algorithm, that is based on a multilevel nonlinear scheme and takes advantage of energy minimizing properties of the Lloyd iteration. In Section 3.4.6 we show that energy minimization possesses sufficient smoothing properties for both scalar and higher dimensional quantization to be used as an effective relaxation in the multigrid cycle and show the results of the numerical experiments. Rigorous proof of the uniform convergence of the proposed method for a large class of densities in 1-dimensional case is provided in Sections 3.4.3, 3.4.4.

3.2 Newton's method and related results

3.2.1 Notations

For a general nonlinear equation $\mathbf{f}(\mathbf{z}) = 0$ with vector argument $\mathbf{z}$, the Newton iteration is given by:

$$\mathbf{z}_n = \mathbf{z}_{n-1} - \mathbf{df}|^{-1}_{\mathbf{z}_{n-1}} \mathbf{f}(\mathbf{z}_{n-1})$$

where $\mathbf{df}$ is the Jacobian matrix of the map $\mathbf{f}$. Applying Lloyd's algorithm to the computation of CVT, we obtain the problem of solving $\mathbf{T}(\mathbf{Z}_{n-1}) = \mathbf{Z}_n$, as

discussed earlier. Newton's method in this setting takes on the form

$$\mathbf{Z}_n = \mathbf{Z}_{n-1} + (\mathbf{dT}|_{\mathbf{Z}_{n-1}} - \mathbf{I})^{-1}(\mathbf{Z}_{n-1} - \mathbf{T}(\mathbf{Z}_{n-1}))$$

Since $\mathbf{T} : \mathbb{R}^{kN} \to \mathbb{R}^{kN}$, corresponding Jacobi matrix $\mathbf{dT} = \{\frac{\partial \mathbf{T}_i}{\partial \mathbf{Z}_j}\}_{i,j}$ has dimensions $kN \times kN$:

$$\mathbf{dT} = \begin{pmatrix} \frac{\partial \mathbf{T}_1^{(1)}}{\partial \mathbf{Z}_1^{(1)}} & \cdots & \frac{\partial \mathbf{T}_1^{(1)}}{\partial \mathbf{Z}_k^{(1)}} & \cdots & \frac{\partial \mathbf{T}_1^{(1)}}{\partial \mathbf{Z}_1^{(N)}} & \cdots & \frac{\partial \mathbf{T}_1^{(1)}}{\partial \mathbf{Z}_k^{(N)}} \\ \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ \frac{\partial \mathbf{T}_1^{(N)}}{\partial \mathbf{Z}_1^{(1)}} & \cdots & \frac{\partial \mathbf{T}_1^{(N)}}{\partial \mathbf{Z}_k^{(1)}} & \cdots & \frac{\partial \mathbf{T}_1^{(N)}}{\partial \mathbf{Z}_1^{(N)}} & \cdots & \frac{\partial \mathbf{T}_1^{(N)}}{\partial \mathbf{Z}_k^{(N)}} \\ \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ \frac{\partial \mathbf{T}_k^{(1)}}{\partial \mathbf{Z}_1^{(1)}} & \cdots & \frac{\partial \mathbf{T}_k^{(1)}}{\partial \mathbf{Z}_k^{(1)}} & \cdots & \frac{\partial \mathbf{T}_k^{(1)}}{\partial \mathbf{Z}_1^{(N)}} & \cdots & \frac{\partial \mathbf{T}_k^{(1)}}{\partial \mathbf{Z}_k^{(N)}} \\ \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ \frac{\partial \mathbf{T}_k^{(N)}}{\partial \mathbf{Z}_1^{(1)}} & \cdots & \frac{\partial \mathbf{T}_k^{(N)}}{\partial \mathbf{Z}_k^{(1)}} & \cdots & \frac{\partial \mathbf{T}_k^{(N)}}{\partial \mathbf{Z}_1^{(N)}} & \cdots & \frac{\partial \mathbf{T}_k^{(N)}}{\partial \mathbf{Z}_k^{(N)}} \end{pmatrix}$$

We arrive at a necessity to calculate partial derivatives of $\mathbf{T}_i(\mathbf{Z})$. The following result (see [15]) is of use:

Lemma 3.2.1. *Let $\Omega = \Omega(\mathbf{U})$ be a region that depends smoothly on $\mathbf{U}$ and that has a well-defined boundary. If $F = \int_{\Omega(\mathbf{U})} f(\mathbf{y}) d\mathbf{y}$, then*

$$\frac{dF}{d\mathbf{U}} = \int_{\partial\Omega(\mathbf{U})} f(\mathbf{y}) \dot{\mathbf{y}} \cdot \mathbf{n} \, d\mathbf{y}$$

where $\mathbf{n}$ is the unit outward normal and $\dot{\mathbf{y}}$ denotes the derivative of the boundary points with respect to changes in $\mathbf{U}$.

Since

$$\mathbf{T}_i(\mathbf{Z}) = \frac{\int_{V_i(\mathbf{Z})} \mathbf{y} \rho(\mathbf{y}) \, d\mathbf{y}}{\int_{V_i(\mathbf{Z})} \rho(\mathbf{y}) \, d\mathbf{y}},$$

we have that

$$\frac{\partial \mathbf{T}_i^{(m)}}{\partial \mathbf{Z}_j^{(n)}} = \Big(\int_{\partial V_i} \rho(\mathbf{y}) \mathbf{y}^{(m)} \frac{\partial \mathbf{y}}{\partial \mathbf{Z}_j^{(n)}} \cdot \mathbf{n} \, d\mathbf{y} \Big) \Big/ \int_{V_i(\mathbf{Z})} \rho(\mathbf{y}) \, d\mathbf{y} -$$
$$- \Big(\int_{\partial V_i} \rho(\mathbf{y}) \frac{\partial \mathbf{y}}{\partial \mathbf{Z}_j^{(n)}} \cdot \mathbf{n} \, d\mathbf{y} \Big) \int_{V_i(\mathbf{Z})} \rho(\mathbf{y}) \mathbf{y}^{(m)} \, d\mathbf{y} \Big/ \Big(\int_{V_i(\mathbf{Z})} \rho(\mathbf{y}) \, d\mathbf{y} \Big)^2$$

Here $1 \leq m \leq N$ and $1 \leq n \leq N$.

Analytic representation of $\frac{\partial \mathbf{y}}{\partial \mathbf{Z}_j^{(n)}}$ can be obtained from the following identity:

Lemma 3.2.2. *If* $\{\mathbf{u}_l\}$ *are the vertices of the common face* Δ_i^j *between adjacent Voronoi regions generated by* $\mathbf{Z}_i$ *and* $\mathbf{Z}_j$, *then for any set of* $\{\lambda_l\}, \lambda_l \geq 0$ *with* $\sum_{l \geq 0} \lambda_l = 1$, *such that* $\sum_{l \geq 0} \lambda_l \mathbf{u}_l \in \Delta_i^j$

$$\Big(\sum_{l \geq 0} \lambda_l \mathbf{u}_l - \frac{\mathbf{Z}_i + \mathbf{Z}_j}{2} \Big) \cdot (\mathbf{Z}_j - \mathbf{Z}_i) = 0$$

Differentiating the above expression with respect to $\mathbf{Z}_i^{(m)}$, for any point $\mathbf{y} \in \Delta_i^j$ we get

$$\begin{aligned} \frac{\partial \mathbf{y}}{\partial \mathbf{Z}_i^{(m)}} \cdot (\mathbf{Z}_j - \mathbf{Z}_i) &= \tfrac{1}{2} \, \mathbf{e}_m \cdot (\mathbf{Z}_j - \mathbf{Z}_i) + \mathbf{e}_m \cdot (\mathbf{y} - \tfrac{\mathbf{Z}_i + \mathbf{Z}_j}{2}) \\ \frac{\partial \mathbf{y}}{\partial \mathbf{Z}_j^{(m)}} \cdot (\mathbf{Z}_j - \mathbf{Z}_i) &= \tfrac{1}{2} \, \mathbf{e}_m \cdot (\mathbf{Z}_j - \mathbf{Z}_i) - \mathbf{e}_m \cdot (\mathbf{y} - \tfrac{\mathbf{Z}_i + \mathbf{Z}_j}{2}) \end{aligned}$$

where $\mathbf{e}_m = (0, \ldots, \overset{m}{1}, \ldots, 0)^T \in \mathbb{R}^N$. Since $\mathbf{n} = \frac{\mathbf{Z}_j - \mathbf{Z}_i}{\|\mathbf{Z}_j - \mathbf{Z}_i\|}$, the necessary expression for $\frac{\partial \mathbf{y}}{\partial \mathbf{Z}_i^{(m)}} \cdot \mathbf{n}$ can be easily obtained and used for integration purposes.

3.2.2 Theoretical results

Classical convergence analysis of the Newton's method adopted in the current context relies on the following lemma (see for example [14]):

Lemma 3.2.3. *Suppose* $\mathbf{F}(\mathbf{z}) = \mathbf{z} - \mathbf{T}(\mathbf{z}) : \mathbb{R}^{kN} \to \mathbb{R}^{kN}$ *is continuously differ-*

entiable in an open convex set $D \subset \mathbb{R}^{kN}$. *Assume that there exists a* $\mathbf{z}^* \in \mathbb{R}^{kN}$, *such that* $\mathbf{F}(\mathbf{z}^*) = 0$ *and there are constants* $\beta, \gamma, r > 0$, *such that*

1) $B(\mathbf{z}^*, r) \subset D$ *is an open ball of radius* r *around* $\mathbf{z}^*$

2) $(\mathbf{I} - \mathbf{dT}(\mathbf{z}^*))^{-1}$ *exists and* $||(\mathbf{I} - \mathbf{dT}(\mathbf{z}^*))^{-1}|| < \beta$

3) $(\mathbf{I} - \mathbf{dT}) \in Lip(\gamma, B(\mathbf{z}^*, r))$

Then there is a radius $\epsilon = \min\{r, 1/2\beta\gamma\}$, *such that for any* $\mathbf{z}_0 \in B(\mathbf{z}^*, \epsilon)$, *the sequence generated by* $\mathbf{z}_n = \mathbf{z}_{n-1} - (I - \mathbf{dT})^{-1}\mathbf{F}(\mathbf{z}_{n-1})$ *converges to* $\mathbf{z}^*$ *and obeys* $||\mathbf{z}_n - \mathbf{z}^*|| \leq \beta\gamma||\mathbf{z}_{n-1} - \mathbf{z}^*||^2$.

It is hard in general to produce criteria for the global convergence of the Newton's method. Here we discuss some of the results that help to further characterize the convergence radius of the Newton scheme in quantization context.

First let us denote $h_i(\mathbf{z}) = diam(V_i)$ for each Voronoi cell V_i corresponding to the generators $\mathbf{z}$, and let D be a compact and convex set in the neighborhood of a solution $\mathbf{z}^*$, such that $\mathbf{dT}$ is continuous in D and there are uniform bounds

$$H = \max_{\mathbf{z} \in D} h_i(\mathbf{z}), \; h = \min_{\mathbf{z} \in D} h_i(\mathbf{z})$$

for all $1 \leq i \leq k$. Moreover, let

$$M = \max_{x \in \Omega} \rho(x), \quad m = \min_{x \in \Omega} \rho(x) \quad \text{and } M' = \max_{x \in \Omega} |\nabla \rho(x)| \, .$$

With these notations, we can claim the following Lipschitz continuity result for the Jacobian:

Lemma 3.2.4. *There is a constant* $\gamma > 0$ *such that* $I - \mathbf{dT} \in Lip(\gamma, D)$. *Moreover, in one space dimension, we can explicitly take* $\gamma = \dfrac{36M^2M'H^4}{m^4h^4}$.

Proof

Using the relation given in [15], we have

$$\Big|\sum_j \Big(\frac{\partial T_i}{\partial x_j} - \frac{\partial T_i}{\partial y_j}\Big)\Big| = \Big|\sum_j \Big(1 - \frac{\partial T_i}{\partial y_j} - (1 - \frac{\partial T_i}{\partial x_j})\Big)\Big| \frac{\Big|R_i^2(x)Q_i(y) - R_i^2(y)Q_i(x)\Big|}{R_i^2(x)R_i^2(y)}$$

where $Q_i(y) = \iint_{V_i(y)\times V_i(y)} \nabla\rho(s)\rho(t)(t-s)dtds$ and $R_i(y) = \int_{V_i(y)} \rho(s)ds$. Hence

$$\Big|\sum_j \Big(\frac{\partial T_i}{\partial x_j} - \frac{\partial T_i}{\partial y_j}\Big)\Big| =$$
$$\frac{\Big|(R_i^2(x) + R_i^2(y))(Q_i(x) - Q_i(y)) - (R_i^2(x) - R_i^2(y))(Q_i(x) + Q_i(y))\Big|}{2R_i^2(x)R_i^2(y)} \leq$$
$$\frac{(R_i^2(x) + R_i^2(y))\Big|Q_i(x) - Q_i(y)\Big| + (Q_i(x) + Q_i(y))\Big|R_i^2(x) - R_i^2(y)\Big|}{2R_i^2(x)R_i^2(y)}.$$

Let $V_i(x-y) = (V_i(x) \setminus V_i(y)) \cup (V_i(y) \setminus V_i(x))$. Notice that there exists a constant such that $|V_i(x-y)| \leq c||x-y||$. For the one dimensional case, we can simply take $c = 2$. We then have the following upper bounds:

$$Q_i(x) \leq MM'H^3, \quad R_i(x) \leq MH \text{ and}$$
$$\Big|Q_i(x) - Q_i(y)\Big| \leq 2^{N+1}MM'H^{N+1}|V_i(x-y)| \leq 2^{N+1}cMM'H^{N+1}||x-y|| ,$$
$$\Big|R_i^2(x) - R_i^2(y)\Big| = \Big|R_i(x) - R_i(y)\Big|\Big(R_i(x) + R_i(y)\Big) \leq 2cM^2H||x-y|| .$$

Hence we end up with the following Lipschitz condition for $\mathbf{dT}$:

$$||\mathbf{dT}(x) - \mathbf{dT}(y)|| \leq \gamma||x-y||$$

where, for the 1-d case, by keeping track of the constants, we have $\gamma = \dfrac{36M^2M'H^4}{m^4h^4}$.

Proposition 3.2.1. *For the computation of one dimensional CVTs in the case of constant or log-concave densities, the Newton's method is quadratically convergent*

in a subset of D *where* $||\mathbf{z}-\mathbf{z}^*|| < \frac{\theta}{2\gamma}$, *with* $(\mathbf{I}-\mathbf{dT}) \in Lip(\gamma, D)$ *and* $\theta = 1 - \max_D ||\mathbf{dT}|| > 0$.

Proof

It was shown in [22] that Lloyd's map is continuous in the neighborhood D for any smooth density in 1d. Notice also, that in the regions where the Lloyd's map is continuous, it is, in fact, continuously differentiable, so it remains to estimate the constants β and γ in this region. As shown in [22], for constant and log-concave densities we have $\theta = 1 - \max_D ||\mathbf{dT}|| > 0$. Notice that $||\mathbf{dT}|| < 1$ implies that $\mathbf{I}-\mathbf{dT}$ is invertible with $||(\mathbf{I}-\mathbf{dT})^{-1}|| \leq \beta$ with $\beta = \dfrac{1}{1-||\mathbf{dT}||} \leq \dfrac{1}{\theta}$ (see [36]). The conclusion then follows from Lemmas 3.2.3, 3.2.4.

Now let us investigate the effect of the round-off and integration errors on the convergence of the algorithm.

Proposition 3.2.2. *If* $\theta = 1 - \max_D ||\mathbf{dT}|| > 0$ *and* $||E|| < \theta$ *for some either constant or log-concave density in 1d, then the quasi-Newton iteration given by*

$$\mathbf{z}_n = \mathbf{z}_{n-1} + (\mathbf{dT}|_{\mathbf{z}_{n-1}} - \mathbf{I} + E)^{-1}(\mathbf{z}_{n-1} - \mathbf{T}(\mathbf{z}_{n-1}))$$

is locally convergent and the convergence is at least superlinear.

Proof

Notice that for superlinear convergence we only need to show that $\mathbf{dT} - \mathbf{I} + E$ is nonsingular and continuous. Continuity in the neighborhood of the solution is guaranteed by the results in [22], while $\mathbf{dT} - \mathbf{I} + E$ is nonsingular whenever $||\mathbf{dT} + E|| < 1$. We know that for this class of densities there is a constant $\theta > 0$, such that $||\mathbf{dT}|| \leq 1 - \theta$. Hence if the perturbation satisfies $||E|| < \theta$, we have

$||\mathbf{dT}+E|| \leq ||\mathbf{dT}||+||E|| \leq 1-\theta+||E|| < 1$, so the iteration remains convergent.

We can regard the application of the approximate evaluations of the Jacobian, e.g. by a quadrature scheme, as a quasi-Newton method, since it amounts to the use of an approximate Jacobi matrix. ¿From a different perspective, we can also consider a sequence of algorithms A_n representing numerical approximations to the Newton algorithm denoted by A, an approach used in [64] for the treatment of the Generalized Lloyd Algorithm. Denote $\{A_n(x)\}^-$ the set consisting of all points y such that $x_n \to x, y_n \in A_n(x_n)$ and y is an accumulation point of y_n. The sequence $\{A_n\}^- : x \to \{A_n(x)\}^-$ is called the sequential accumulation of A_n. It is a generalization of the concept of the closed algorithm in a sense that if $A_n = A$ for each n, $\{A_n\}^- \subset A$ is equivalent to A being closed.

The following proposition can be proved (see [64]):

Proposition 3.2.3. *Suppose* $\{A_n\}^- \subset A$ *and* Γ *is a set of fixed points of* A. *If* x_n^* *is a fixed point of* A_n, *then* $\sigma(x_n^*, \Gamma) \to 0$.

This proposition shows that a fixed point of A_n is nearly equal to some fixed point of A, so the use of A_n is consistent with the use of the original scheme A.

With these ideas in mind, let us now design a new algorithm to accelerate the convergence of the Lloyd iteration. Knowing the issues associated with both Lloyd and Newton approaches, we will try to incorporate their best features into a coupled Lloyd-Newton scheme, as described next.

3.2.3 Description of the algorithm

The results mentioned above do not provide means of identifying the actual region of convergence for an arbitrary density function. In order to use the Newton's

approach to speed up the fixed point Lloyd's iteration, we can deal with this problem by coupling the two algorithms into one hybrid scheme. For example, let us look at the following implementation of this idea.

Algorithm 3.2.1. **Lloyd-Newton iteration**

Input:

Ω , *the domain of interest;* ρ , *a probability distribution on* Ω ;

k , *number of generators;* $\mathbf{Z} = \{z_i\}_1^k$, *the initial set of generators;* ϵ - *tolerance.*

Output:

$\{V_i\}_1^k$, *a CVT with* k *generators* $\{z_i\}_1^k$ *in* Ω

Method:

1. Construct the Voronoi tessellation $\{V_i\}_1^k$ *of* Ω *with generators* $\mathbf{Z} = \{z_i\}_1^k$.

2. Compute the mass centroids $\mathbf{X} = \{x_i\}_1^k$ *of* $\{V_i\}_1^k$.

3. If $\|\mathcal{H}(\mathbf{X}, V_i) - \mathcal{H}(\mathbf{Z}, V_i)\| \geq \epsilon$, *take mass centroids as generators, goto step 1.*

Otherwise fix $\alpha = 1$.

4. Let $\mathbf{T}(\mathbf{X}) = \mathbf{Z}$. *Perform a step of Newton's method:*

$\tilde{\mathbf{Z}} = \mathbf{Z} + \alpha(\mathbf{dT}|_{\mathbf{Z}} - \mathbf{I})^{-1}(\mathbf{Z} - \mathbf{T}(\mathbf{Z}))$

5. Let $I = \{i | 1 \leq i \leq k\,, \tilde{z}_i \notin \Omega\,\}$.

If $|I| = 0$ *take* $\tilde{\mathbf{Z}}$ *as generators and goto step 4.*

If $|I| = 1$ *reduce Newton's step size:* $\alpha = \alpha/2$, *goto step 4.*

Otherwise take $\mathbf{Z}$ *as generators and goto step 1.*

6. Repeat until some stopping criterion is met.

We will show that this approach can be used to accelerate the Lloyd's scheme. As shown in Section 3.2.2, Newton iteration gives superlinear convergence, whenever the convergence region is reached. However, there are possible difficulties

associated to this approach. Namely,

- Numerical error in calculation of elements of Jacobi matrix
- Possible increase of the condition number resulting in instability
- The region of convergence for Newton iteration can be hard to reach

As shown in Proposition 3.2.2, relatively small numerical error does not represent a threat for the general superlinear convergence of the algorithm. However, the associated convergence region might potentially become too small to yield noticeable improvement over the Lloyd iteration. In the examples presented in 3.2.7 we address these issues and show the numerical results that justify the use of the Newton approach as a local accelerator of the Lloyd iteration.

3.2.4 Local vs. global minimizers

Let us note that in general, there is no guarantee for the Lloyd's method to converge to the global minimizer of the energy. Take the example of two points and the square $[-1,1]^2$ given in Figure 3.1. One can show that the partition along the midline corresponds to a local minimum but the one along the diagonal corresponds to a saddle point. In fact, the energy decreases as the diagonal rotates toward the middle vertical line.

The methods discussed in this Chapter do not provide the means of reaching the global minimizer, but instead concentrate on the acceleration of convergence in the local vicinity of any solution. However, it is possible to couple these fast converging schemes with some global minimization methods to achieve the optimal performance. We will return to this discussion later in Chapter 5.

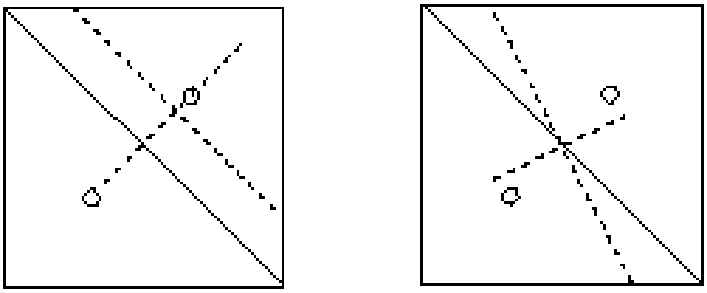


Figure 3.1. Convergence of Lloyd's method to local and global minimizers

3.2.5 Computational complexity

Let us now briefly look at the complexity of the proposed algorithm. Each step of the adaptive Lloyd-Newton algorithm includes:

1. Construction of Voronoi diagram.

   For two dimensions, we use an embedded Matlab routine, which is of order $O(k)$ [2, 32]. In N-dimensions, the average complexity estimate of $O(k)$ is expected when the average number of Voronoi neighbors is bounded [29].

2. Calculation of centroids for each region.

   This involves calculation of two integrals per region, hence a total of $2k$ integrals, again $O(k)$.

3. Calculation of the Jacobi matrix.

   Each element of the matrix involves four new integrations, assuming we store the results of the previous centroidal calculations. There are k elements. Assume each one has on average m neighbors, $m < k$. Then we need a total of $4\,m\,k$ integrations. If we are sufficiently close to the optimal configuration, m does not exceed 8 (see [62]), which makes this step worth $O(k)$.

4. Solving the resulting linear system.

   The complexity of the linear solver highly depends on the structure of the matrix. With a sparse banded matrix structure due to limited number of neighbors for each generator we can adopt fast inversion algorithms that minimize the fill in of the LU decomposition, for instance, the nested dissection methods with complexity on the order $O(k^{3/2})$. Iterative methods with lower complexity can also be considered, we comment on this in the later discussions.

5. Updating procedure for generators.

   This is a simple element-by-element addition, requiring $2k$ operations.

Overall, it is clear that the total complexity depends critically on the linear solver and the algorithm for the construction of the Voronoi tessellations. For well-distributed points, however, it is reasonable to expect an optimistic linear time complexity.

3.2.6 Parallel implementation

Due to a large amount of calculations involved in the realization of Newton scheme, it seems natural to look at possible parallelizability of the algorithm. There are several places where parallelism could be exploited:

- Domain decomposition can bring significant speedup in the construction of Voronoi diagram as well as in the calculation of centroids

- Jacobi matrix computation is the most computationally intensive procedure, which can be parallelized using block matrix structure or by splitting the individual tasks in element calculations

These and other algorithmic improvements will be considered in our upcoming publications on this subject.

3.2.7 Numerical Implementation

We present the results of numerical computations on the square $\Omega = [0,1]^2$ for different types of density functions. All computations were made in Matlab 6.5. We used embedded Matlab functions *voronoi* and *voronoin* for diagram construction.

For the Lloyd's method, once the Voronoi construction is available, the only computational task left is to find the mass centroids of the Voronoi regions. For Newton's method, we also need to compute the entries of the Jacobian matrix, which adds up to the complexity of the problem. Computational properties of the problem heavily depend on the form of the density function used. One always needs to find a compromise between the accuracy and resource consumption for a particular algorithm. Since complexity of the quadrature is tightly bound with the computational cost of the algorithm, quadrature rules have to be tuned up depending on the form of the density function.

In 1-d case, integrals can be computed exactly. In two dimensional case, the Voronoi regions are of polygonal shape, so one may use triangle based integration rules or tensor-product based one dimensional rules. This can be done using a triangulation of any kind.

We tested different integration rules for various types of density functions. For boundary integrals, we used Simpson's rule for polynomial densities of degree less than 3 and k-node quadrature rules otherwise. For area integrals, midpoint Δ rule is used for all densities. This rule was exact for polynomials of degree no greater than 3.

Here the name "midpoint Δ rule" refers to the following triangular based quadrature rule: $\int_\Delta f = \frac{1}{3}|\Delta| \cdot (f(x_{12}) + f(x_{13}) + f(x_{23}))$, where x_{12}, x_{23}, x_{13} are the midpoints of the sides of the triangle Δ.

3.2.8 Stopping criterion

There are several criteria one can adopt in this situation:

1. $\|\mathbf{Z}_n - \mathbf{Z}_{n-1}\| < \epsilon$

   i.e. when the distance between two consecutive configurations becomes small enough

2. $\|E_n - E_{n-1}\| < \epsilon$

   i.e. when changes in energy become sufficiently small

3. Nstep > MaxNumSteps

   i.e. maximum number of steps is reached

Quite often it is a combination of the above conditions that makes a good stopping criterion. In our examples, the algorithm was stopped whenever a failure of either condition 1. or 3. was discovered.

3.2.9 One-dimensional examples

For one-dimensional intervals, since finding the Voronoi regions is trivial, most of the computation is associated with finding centroids. Numerical errors for such tasks are negligible, so the algorithm converges in several steps.

In Figure 3.2 we plot the ratio $\log(e_k) \Big/ \log(e_{k-1})$ for both Newton (top) and Lloyd (bottom) iterations. It can be readily seen that the limit is 2 in the Newton

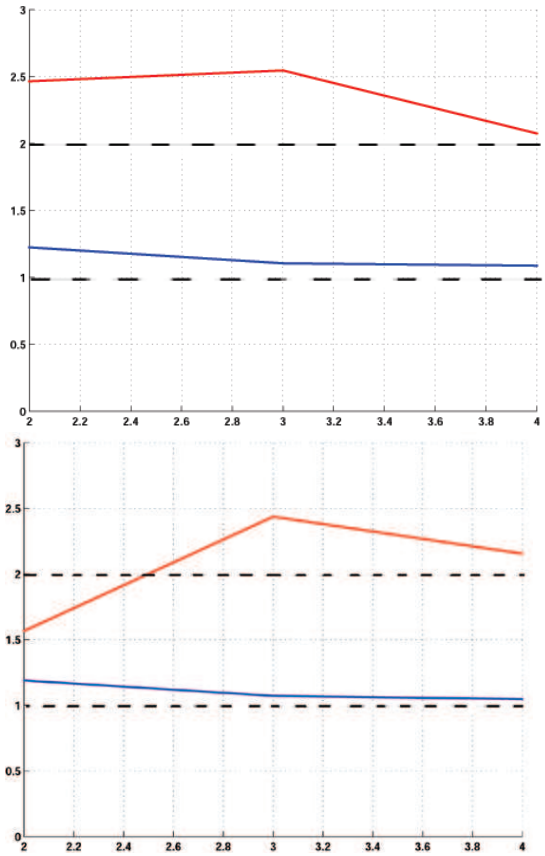


Figure 3.2. `1d convergence rates comparison for` $k = 4$ `(left) and` $k = 64$ `(right) with` $\rho(x) = 1 + x^4 \cos(\pi(x - 0.5))$ `. Top curves are for Newton iteration and the bottom ones are for Lloyd.`

case, which justifies the quadratic convergence. Lloyd's method converges at a linear rate.

3.2.10 Two-dimensional examples

In the two dimensional case, the effect of the roundoff and numerical integration errors becomes more pronounced. In case of a constant density, we are still able to get almost flawless performance. Figure 3.3 shows convergence of both methods for a random 5 generator configuration. Here dots denote positions of the generators at each step of the iteration and lines are used to separate the corresponding Voronoi

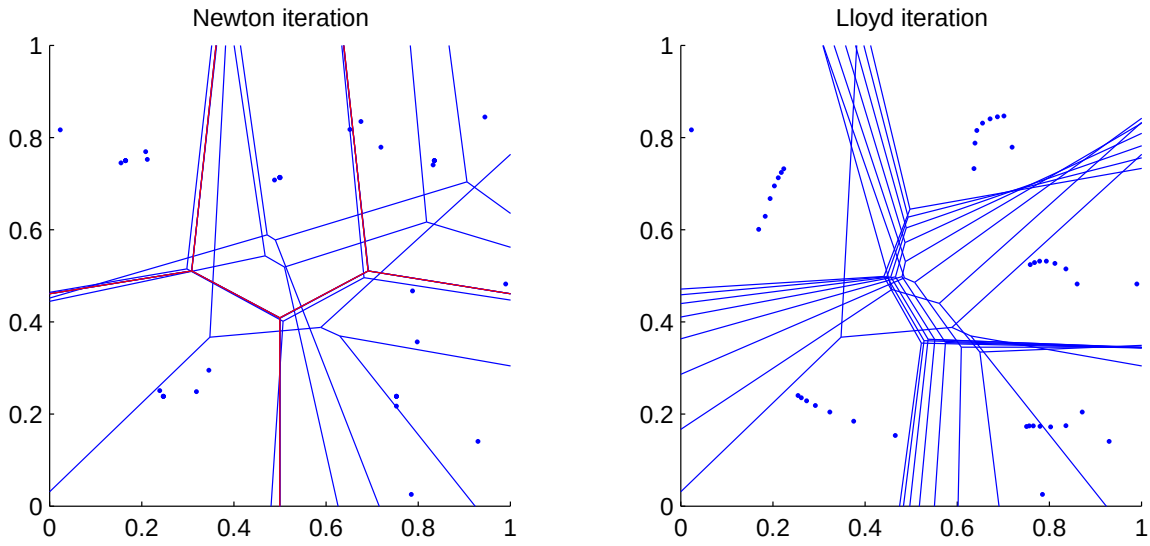


Figure 3.3. `Iteration history of (a) Lloyd-Newton vs. (b) Lloyd method for` $\rho(x) = 1, k = 5$.

regions. Lloyd-Newton iteration converged after 7 Newton steps, and convergence became quadratic as soon as the convergence region was reached, as shown in Figure 3.4.

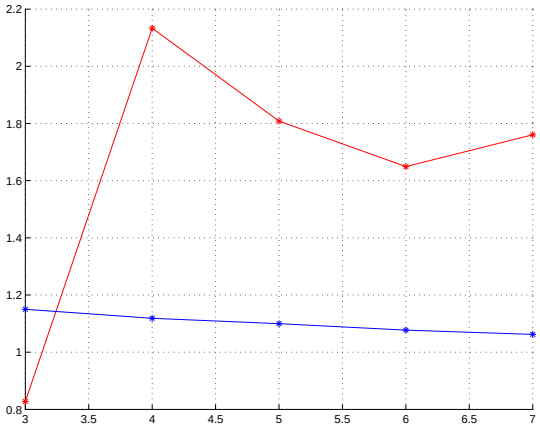


Figure 3.4. `Convergence rate of the Lloyd-Newton method (top graph) vs. Lloyd iteration (bottom) for` $\Omega = [0,1]^2$, $\rho(x) = 1$, $k = 5$

The next two pictures (Figures 3.5 and 3.6) demonstrate the performance of both methods in non-constant density cases. The Lloyd-Newton method converged

after 6 Newton steps for $\rho(x) = 1 + x + 0.1x^2$ and after 9 Newton steps for $\rho(x) = 1 + x^4$.

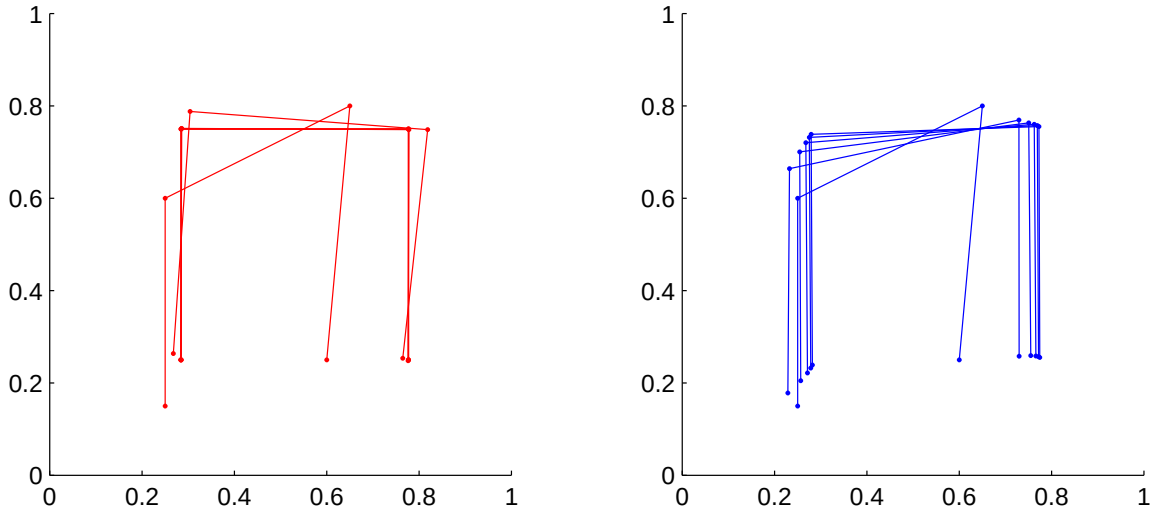


Figure 3.5. Iteration history of (a) Lloyd-Newton vs. (b) Lloyd method for $\rho(x) = 1 + x + 0.1x^2$, $k = 4$. Here the lines connect the generators

For a more precise comparison, Table I below shows the decrease of the error for Lloyd-Newton and Lloyd methods in the case of a quadratic density function $\rho(x) = 1 + x + 0.1x^2$ after 5 consecutive iterations respectively:

Iteration	Lloyd's iteration error	Lloyd-Newton's iteration error
1	0.08641081909378	0.16571484289620
2	0.03313222925306	0.03144575914202
3	0.01849005608503	0.00159901251769
4	0.01041059669286	0.00000571605675
5	0.00599684938138	0.00000000572324

Table I. Error reduction of the Lloyd-Newton and the Lloyd iterations.

Adding higher order terms to the density function introduces numerical error

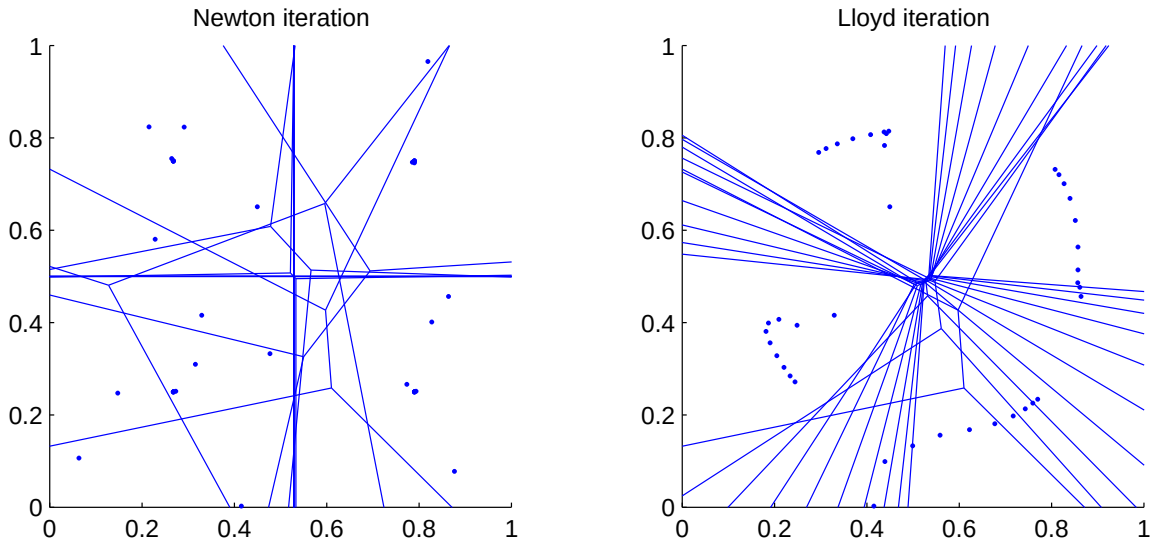


Figure 3.6. `Iteration history of (a) Lloyd-Newton vs. (b) Lloyd method,` $\rho(x) = 1 + x^4, k = 4$

in the calculation of both boundary and area integrals. Here we compare the exact and inexact calculations made using Simpson's rule for line integrals and midpoint triangle rule for the area. Figure 3.7(a) shows results we got for a quadratic function, for which integration is exact, whereas the graph in Figure 3.7(b) shows convergence for a quartic polynomial density function.

Clearly, the integration errors do have an effect on the convergence of the overall scheme, so for the best performance of the algorithm the optimal tradeof of the integration scheme accuracy and overall complexity should be made. As mentioned above, for the density functions up to certain order it is possible to nullify the numerical integration error by picking a more accurate quadrature rule. However, this might not be possible for a large class of functions, e.g. functions with singularities of a much higher order. Despite these natural restrictions, the results shown above justify the fact that for an adequately chosen quadrature Newton-Lloyd method outperforms the Lloyd iteration and allows to reach the

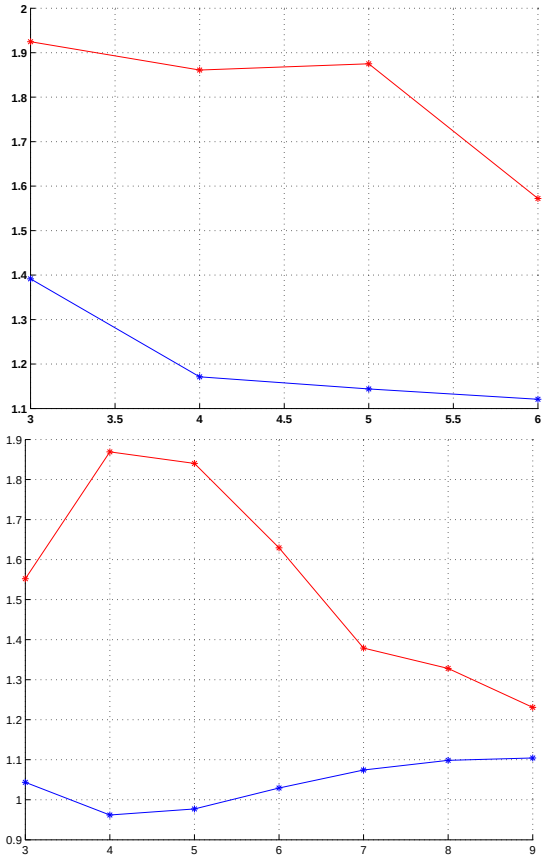


Figure 3.7. Comparison of convergence factors for Newton-Lloyd iteration (top) vs. Lloyd (bottom) for different densities: (a) $\rho(x) = 1 + x + 0.1\,x^2\,, k = 4\,;$ (b) $\rho(x) = 1 + x^4\,, k = 4$

desired solution significantly faster.

Another possible approach, as mentioned above, consists of taking a Newton iteration as part of the relaxation within the *outer* framework provided by some type of nonlinear multigrid procedure. The next section is dedicated to a possible implementation of this type of algorithm.

3.3 Newton-based multilevel algorithm

While the results presented in the previous section show that the Lloyd-Newton scheme generally performs better than the traditional fixed-point iteration, im-

provement can be made, in particular, by reducing the computational cost of solving the linear system for the Newton increment. In this regard, one possibility is to use multilevel techniques to solve the linear system inside the Newton iteration framework. We refer to this approach as the *inner* multigrid scheme. Naturally, the other possibility is to rely on the nonlinear multigrid solver with the Lloyd-Newton scheme being part of the inner relaxation procedure which would give an *outer* multigrid approach. We now first discuss the former case while leave the latter to the next section.

Recall that the nonlinear problem under consideration is to find the fixed points of the Lloyd map $\mathbf{T}(\mathbf{Z}) = \mathbf{Z}$. As shown above, the Newton linearization

$$(\mathbf{I} - \mathbf{dT}|_{\mathbf{Z}_{n-1}})(\mathbf{Z}_n - \mathbf{Z}_{n-1}) = \mathbf{T}(\mathbf{Z}_{n-1}) - \mathbf{Z}_{n-1}$$

gives a fast convergent iterative scheme in the neighborhood of the solution The performance can be greatly enhanced if some fast sparse solvers are used to reduce the computational complexity associated with the solution of linear systems. For instance, let us outline an algorithm that uses algebraic multigrid techniques for these purposes.

Algorithm 3.3.1. **AMG-BGS-Newton iteration**

Input:

Ω , *the domain of interest;* ρ , *a probability distribution on* Ω ;

k , *number of generators;* $\mathbf{Z} = \{z_i\}_1^k$, *the initial set of generators;*

Output:

$\{V_i\}_1^k$, *a CVT with* k *generators* $\mathbf{Z} = \{z_i\}_1^k$ *in* Ω

Method:

1. Given $n-th$ *iterate* $\mathbf{Z}_n$, *calculate* $\mathbf{T}(\mathbf{Z}_n)$, $\mathbf{dT}(\mathbf{Z}_n)$.

2. Put

$$A = \mathbf{I} - \mathbf{dT}(\mathbf{Z}_n) = \begin{pmatrix} \mathbf{I} - \mathbf{T}_{xx} & \mathbf{T}_{xy} \\ \mathbf{T}_{yx} & \mathbf{I} - \mathbf{T}_{yy} \end{pmatrix} \quad , \quad M = \begin{pmatrix} \mathbf{I} - \mathbf{T}_{xx} & 0 \\ \mathbf{T}_{yx} & \mathbf{I} - \mathbf{T}_{yy} \end{pmatrix}$$

and $b = \mathbf{T}(\mathbf{Z}_n) - \mathbf{Z}_n$.

3. Solve $M\mathbf{Z}_{n+1} = b - (A - M)\mathbf{Z}_n$, *where the system for each of the diagonal blocks involving* $(\mathbf{I} - \mathbf{T}_{xx})$ *and* $(\mathbf{I} - \mathbf{T}_{yy})$ *is solved using AMG*

4. Repeat the procedure 1 to 3 until some stopping criterion is met.

Let us now discuss the key elements of the scheme introduced above. First key observation is related to the choice of a triangular iteration matrix

$$M = \begin{pmatrix} \mathbf{I} - \mathbf{T}_{xx} & 0 \\ \mathbf{T}_{yx} & \mathbf{I} - \mathbf{T}_{yy} \end{pmatrix}$$

for solving the linearized system. In making this choice, we relied on the fact that the matrix $A = \mathbf{I} - \mathbf{dT}$ has a block structure with the contribution of the off-

diagonal blocks being relatively small. To solve the corresponding linear system, one can either perform the GMRES iteration with M being a preconditioner or resort to the block Gauss-Seidel (BGS) method taking M to be the corresponding iteration matrix.

The next key feature of this algorithm is the use of the algebraic multigrid method (AMG) [4, 7, 63] to solve the linear systems corresponding to each of the diagonal blocks of M. Indeed, such an approach is justified by the fact that both of the blocks $\mathbf{I} - \mathbf{T}_{xx}$ and $\mathbf{I} - \mathbf{T}_{yy}$ are symmetric and often share diagonal dominance properties. An example of using the *classical* AMG approach based on the standard coarse-grid correction scheme is given as follows:

1. Perform relaxation of the fine grid until the error is smooth: $A^h u^h = b^h$
2. Compute residual $r^h = b^h - A^h b^h$ and transfer to the coarse grid $r^{2h} = I_h^{2h} r^h$
3. Solve the coarse-grid residual equation in terms of the error $A^{2h} e^{2h} = r^{2h}$
4. Interpolate the error to the fine grid and correct the fine-grid solution: $u^h = u^h + I_{2h}^h e^{2h}$.

Here the restriction operator I_h^{2h} is dependent on the solution at the current iteration and represents a coarsening procedure, while the iteration dependent operator I_{2h}^h represents the standard interpolation. Naturally, a setup phase has to be implemented first based on the entries of A so that these operators are suitably defined [63]. Combining these considerations, we can design the AMG-BGS-Newton scheme, as shown in Algorithm 3.3.1.

The efficiency of such an algebraic multigrid implementation relies on the observation that each of the diagonal blocks of the M matrix become diagonally dominant in the vicinity of the solution. Theoretical arguments leading to this

conclusion have been carried out in 1d for the class of strongly logarithmically concave densities in [22]. In fact, in this case the Lloyd map was shown to be a local contraction, implying diagonal dominance for the matrix $\mathbf{I} - \mathbf{dT}$. For these densities, a multilevel scheme designed this way outperforms regular Newton iteration in its convergence.

3.4 The new energy-based nonlinear multilevel algorithm

Another possible approach to the problem of speeding up convergence for Lloyd's method is to use a domain decomposition or multigrid strategies. Since the original concept of centroidal Voronoi tessellations is related to the solution of a nonlinear optimization problem, and the monotone energy descent property is preserved by the Lloyd's fixed point iteration ([15]), we may investigate whether monotone energy reduction can be achieved in a multilevel procedure which would also improve the performance of the simple-minded fixed point iteration.

The problem of constructing a CVT is nonlinear in nature, hence standard linear multigrid theory cannot be directly applied. There are still several ways one could implement a nonlinear multilevel scheme in this context (see [23],[24],[50],[51]). The Newton type acceleration methods described earlier are based on some global linearization as the outer loop, coupled with other fast solvers in the inner loop. Alternatively, we now study an approach that overcomes the difficulties of the nonlinearity by essentially relying on the direct energy minimization without any type of global linearization.

We note that the optimality property implies that at the CVT (or optimal

quantizer), we have $\nabla\mathcal{H} = 0$. This is the key characterization to be used in the later discussion.

3.4.1 Space decomposition

Since the energy functional is in general non-convex, it turns out to be very effective to relate our problem to a convex optimization problem through a technique that mimics the role of a dynamic nonlinear preconditioner. More precisely, denote $R = \mathtt{diag}\{R_i^{-1}\}, i = 1, \ldots, k+1$ where $R_i = \int_{V_i} \rho(\mathbf{y})\, d\mathbf{y}$ are the masses of the corresponding Voronoi cells. We arrive at an equivalent formulation of the minimization problem: $R\nabla\mathcal{H} = 0$, or $\min ||R\nabla\mathcal{H}||^2$. A key observation is that as R varies with respect to the generators, the above transformation or *dynamic preconditioning* makes the modified energy functional convex in a large neighborhood of the minimizer and therefore makes the new formulation more amenable than the original problem. Hence, let us define the set of iteration points $\mathbf{W}$ by

$$\mathbf{W} = \{(w_i)|_{i=0}^{k+1}| \quad 0 = w_0 \leq w_i \leq w_{i+1} \leq w_{k+1} = 1,\ \forall 0 \leq i \leq k\}\ ,$$

and let us design a new multilevel algorithm based on the following nonlinear optimization problem

$$\min_{\mathbf{Z}\in\mathbf{W}} \tilde{\mathcal{H}}(\mathbf{Z}),\ \text{where}\ \tilde{\mathcal{H}}(\mathbf{Z} = \{\mathbf{z}_i\}_{i=0}^{k+1}) = ||R\nabla\mathcal{H}(\{\mathbf{z}_i\}_{i=1}^{k}, \{V_i\}_{i=1}^{k})||^2 \tag{3.4.1}$$

Here $\{V_i\}_{i=1}^{k}$ is the Voronoi tessellation corresponding to the generators $\{\mathbf{z}_i\}_{i=1}^{k}$. Let us take $\mathcal{T} = \mathcal{T}_J$ as a finite element mesh corresponding to $\mathbf{W}$. Consider a sequence of nested quasi-uniform finite element meshes $\mathcal{T}_1 \subset \mathcal{T}_2 \subset \ldots \mathcal{T}_J$, where $\mathcal{T}_i$ consists of all finite element meshes $\{\tau_j^i\}_{j=1}^{n_i}$ with mesh parameter h_i, such that

$\cup_{j=1}^{n_i}\tau_j^i = \Omega$. Corresponding to each finite element partition $\mathcal{T}_i$ there is a finite element space $\mathbf{W}_i$ defined by

$$\mathbf{W}_i = \{v \in H^1(\Omega) \mid \quad v|_\tau \in \mathcal{P}_1(\tau), \forall \tau \in \mathcal{T}_i\}$$

For each $\mathbf{W}_i$ there corresponds a nodal basis $\{\psi_j^i\}_{j=1}^{n_i}$, such that $\psi_j^i(x_k^i) = \delta_{jk}$, where $\{x_k^i\}_{k=1}^{n_i}$ is the set of all nodes of the elements of $\mathcal{T}_i$ and $x_1^J = 0, x_{n_J}^J = 1$. Define the corresponding one-dimensional subspaces $\mathbf{W}_{i,j} = \text{span}\{\psi_j^i\}$. Then the decomposition can be regarded as

$$\mathbf{W}_J = \sum_{i=1}^{J}\sum_{j=1}^{n_i}\mathbf{W}_{i,j} = \bigoplus_{i=1}^{J}\bar{\mathbf{W}}_i$$

where $\bar{\mathbf{W}}_i = \mathbf{W}_i/\mathbf{W}_{i-1}$ for $i > 1$ and $\bar{\mathbf{W}}_1 = \mathbf{W}_1$. Now clearly for every function $\psi_j^i \in \mathbf{W}_i$ we can find a vector $\bar{\psi}_j^i = \{\bar{\psi}_{jm}^i\} \in \mathbb{R}^{n_J}$, such that $\psi_j^i(x) = \sum_{m=1}^{n_J}\bar{\psi}_{jm}^i\psi_m^J(x), \forall x \in \Omega$.

We note that in the 1-dimensional case, the set of basis functions

$$Q_i = [\bar{\psi}_1^i, \ldots, \bar{\psi}_{n_i}^i]^T \in \mathbb{R}^{n_i \times k}$$

used at each iteration can be pre-generated using the recursive procedure: $Q_J = I_{k\times k}$ and $Q_{J-s} = (\Pi_{i=1}^s P_{J-i})Q_J$ where P_i is the basis transformation from space $\mathbf{W}_{i+1}$ to $\mathbf{W}_i$ which plays a role of a restriction operator.

3.4.2 Description of the algorithm

Using the above notations, we design a multilevel successive subspace correction algorithm (Algorithm 3.4.1). Each step of the procedure outlined below involves

solving a system of nonlinear equations which plays the role of relaxation. We can use the Newton iteration to solve this nonlinear system. Solution at current iterate is updated after each nonlinear solve by the Gauss-Seidel type procedure, hence the resulting scheme is successive in nature. The algorithm essentially only depends on the proper space decompositions and the correspondence with the set of generators thus is applicable in any dimension. The more general forms will be discussed in our subsequent works.

Algorithm 3.4.1. Successive correction $V(\nu_1, \nu_2)$ scheme

Input:

Ω , *the domain of interest;* ρ , *a probability distribution on* Ω *;*

k , *number of generators;*

$\mathbf{Z} = \{z_i\}_{i=0}^{k+1} \in \mathbf{W}$, *the ends plus the initial set of generators.*

Output:

$\mathbf{Z} = \{z_i\}_{i=0}^{k+1}$, *the ends plus the set of generators for CVT* $\{V_i\}_{i=1}^{k}$.

Method:

1. For i=J:-1:2

Repeat ν_1 *times: given* $\mathbf{Z}$, *find* $\mathbf{Z} = \mathbf{Z} + \alpha_j^0 \bar{\psi}_j^i \in \mathbf{W}$ *sequentially for* $1 \le j \le n_i$ *such that* $\tilde{\mathcal{H}}(\mathbf{Z} + \alpha_j^0 \bar{\psi}_j^i) = \min_{\alpha_j} \tilde{\mathcal{H}}(\mathbf{Z} + \alpha_j \bar{\psi}_j^i)$

endfor

2. $\mathbf{Z} \leftarrow$ *CoarseGridSolve(*$\mathbf{Z}$ *)*

3. For i=2:1:J

Repeat ν_2 *times: given* $\mathbf{Z}$, *find* $\mathbf{Z} = \mathbf{Z} + \alpha_j^0 \bar{\psi}_j^i \in \mathbf{W}$ *sequentially for* $1 \le j \le n_i$ *such that* $\tilde{\mathcal{H}}(\mathbf{Z} + \alpha_j^0 \bar{\psi}_j^i) = \min_{\alpha_j} \tilde{\mathcal{H}}(\mathbf{Z} + \alpha_j \bar{\psi}_j^i)$

endfor

4. Repeat the procedure 1 to 3 until some stopping criterion is met.

Supply $\mathbf{W}$ with the following norm:

$$||y||^2_{1,\mathbf{W}} = \frac{1}{k}\sum_{i=1}^{k+1}(y_i - y_{i-1})^2$$

This definition of the norm on $\mathbf{W}$ is justified by the fact that if $\sum_{i=1}^{k+1}(y_i - y_{i-1})^2 = 0$, we have $y_i - y_{i-1} = 0 \quad \forall i$, which contradicts the condition that $y_0 = a, \quad y_{k+1} = b$ for all points in $\mathbf{W}$, unless $y_i = a, \forall 0 \leq i \leq k+1$.

3.4.3 Technical lemmas

Before we introduce our main convergence results, let us first establish some important properties of the energy functional defined in 3.4.1

In the discussion that follows we will say that a functional F satisfies the *convexity* and *continuity* properties in V, if there exist constants $K > 0, L > 0, p \geq q > 1$ s.t.

$$(F'(w) - F'(v), w - v) \geq K||w - v||^p_V, \forall w, v \in V \tag{3.4.2}$$

$$(F'(w) - F'(v), w - v) \leq L||w - v||^q_V, \forall w, v \in V \tag{3.4.3}$$

To simplify the presentation, let us introduce the following notations: $u_i^- = \frac{u_i + u_{i-1}}{2}$, $u_i^+ = \frac{u_i + u_{i+1}}{2}, i = 1, \ldots, k$. We also let $a_i = u_i - u_{i-1}, b_i = w_i - w_{i-1}, x_i = u_i - w_i$, with $u_0 = w_0 = a, u_{k+1} = w_{k+1} = b$ being fixed ends of the interval.

This said, let us first turn our attention to the case of constant densities. In this simple case for the preconditioned energy functional we get the following result.

Proposition 3.4.1. *Let $\rho(x) = 1$ be the density function on $[a, b]$. Then the*

following relation holds:

$$(\tilde{\mathcal{H}}'(u) - \tilde{\mathcal{H}}'(w), u - w) = \frac{1}{2}\sum_{i=1}^{k+1}(a_i - b_i)^2$$

where $a_i = u_i - u_{i-1}, b_i = w_i - w_{i-1}, i = 1, \ldots, k\,,\ u_0 = w_0 = a, u_{k+1} = w_{k+1} = b\,.$

Proof.

$$\frac{\partial\tilde{\mathcal{H}}}{\partial u_i} = 2(u_i - T_i) = 2\Big(u_i - \frac{u_i^+ + u_i^-}{2}\Big) = \frac{1}{2}(a_i - a_{i+1})$$

$$(\tilde{\mathcal{H}}'(u) - \tilde{\mathcal{H}}'(w), u - w) = \frac{1}{2}\sum_{i=1}^{k}(u_i - w_i)(a_i - a_{i+1} - b_i + b_{i+1}) =$$

$$= \frac{1}{2}\sum_{i=1}^{k}(u_i - w_i)(a_i - b_i) - \frac{1}{2}\sum_{i=1}^{k}(u_{i-1} - w_{i-1})(a_i - b_i) = \frac{1}{2}\sum_{i=1}^{k+1}(a_i - b_i)^2$$

□

Corollary 3.4.1. *For constant densities the energy functional* $\tilde{\mathcal{H}}$ *satisfies continuity and convexity conditions with* $K = L = k/2$ *for all points in* $\mathbf{W}$ *.*

It is possible to extend this result to a broader class of density functions. First let us prove the following auxiliary lemma.

Lemma 3.4.2. *If* $\rho(x) = 1 + \epsilon g(x)\,,\ g(x) = x^n$ *and*

$$Q_i(u) = \frac{\int (2u - (u_i^+ + u_i^-))g(u)du}{(u_i^+ - u_i^-) + \epsilon\int g(u)du},$$

then $|Q_i(u) - Q_i(w)| \le (1+\epsilon)(2n+7)(|u_i^+ - w_i^+| + |u_i^- - w_i^-|)\,.$

Proof. Since $g(x) = x^n$, we get the following expression for $Q_i = \dfrac{N_i}{D_i}$. For the

numerator we have:

$$N_i(u) = \int (2u - (u_i^+ + u_i^-))g(u)du =$$
$$= 2\int u^{n+1}\,du - (u_i^+ + u_i^-)\int u^n\,du = \frac{2u^{n+2}}{n+2}\Big|_{u_i^-}^{u_i^+} - \frac{(u_i^+ + u_i^-)u^{n+1}\big|_{u_i^-}^{u_i^+}}{n+1} =$$
$$= \Big(u_i^+ - u_i^-\Big)\Big(\frac{2}{n+2}\sum_{k+l=n+1}(u_i^+)^k(u_i^-)^l - \frac{(u_i^+ + u_i^-)}{n+1}\sum_{k+l=n}(u_i^+)^k(u_i^-)^l\Big)$$

while the denominator is equal to

$$D_i(u) = (u_i^+ - u_i^-)\Big(1 + \frac{\epsilon}{n+1}\sum_{k+l=n}(u_i^+)^k(u_i^-)^l\Big)$$

Finally, for the ratio $Q_i = \dfrac{N_i}{D_i}$ we have

$$\frac{N_i(u)}{D_i(u)} = \Big(\frac{2}{n+2}\sum_{k+l=n+1}(u_i^+)^k(u_i^-)^l - \frac{(u_i^+ + u_i^-)}{n+1}\sum_{k+l=n}(u_i^+)^k(u_i^-)^l\Big)/$$
$$\Big(1 + \frac{\epsilon}{n+1}\sum_{k+l=n}(u_i^+)^k(u_i^-)^l\Big) =$$
$$= \Big(\frac{2}{n+2}S_{n+1}(u_i^+, u_i^-) - \frac{(u_i^+ + u_i^-)}{n+1}S_n(u_i^+, u_i^-)\Big)/\Big(1 + \frac{\epsilon}{n+1}S_n(u_i^+, u_i^-)\Big)$$

where

$$S_n(u_i^+, u_i^-) = \sum_{k+l=n}(u_i^+)^k(u_i^-)^l$$

For simplicity let us redefine the modified numerator as $\tilde{N}_i(u) = \dfrac{2}{n+2}S_{n+1}(u_i^+, u_i^-) - \dfrac{(u_i^+ + u_i^-)}{n+1}S_n(u_i^+, u_i^-)$ and denominator as $\tilde{D}_i(u) = 1 + \dfrac{\epsilon}{n+1}S_n(u_i^+, u_i^-) \geq 1$. Then

$$|Q_i(u) - Q_i(w)| \leq \frac{1}{\tilde{D}_i(u)\tilde{D}_i(w)}\Big|\tilde{N}_i(u)\tilde{D}_i(w) - \tilde{N}_i(w)\tilde{D}_i(u)\Big| \leq$$
$$\leq \frac{1}{2}\Big|\tilde{N}_i(u) - \tilde{N}_i(w)\Big|\Big(\tilde{D}_i(u) + \tilde{D}_i(w)\Big) + \frac{1}{2}\Big(\tilde{N}_i(u) + \tilde{N}_i(w)\Big)\Big|\tilde{D}_i(u) - \tilde{D}_i(w)\Big|$$

Notice further that

$$\begin{aligned}\Big|\tilde{D}_i(u)-\tilde{D}_i(w)\Big| &= \frac{\epsilon}{n+1}\Big|S_n(u_i^+,u_i^-)-S_n(w_i^+,w_i^-)\Big| \\ \Big|\tilde{N}_i(u)-\tilde{N}_i(w)\Big| &= \Big|\frac{2}{n+2}\Big(S_{n+1}(u_i^+,u_i^-)-S_{n+1}(w_i^+,w_i^-)\Big)- \\ &- \frac{1}{n+1}\Big((u_i^++u_i^-)S_n(u_i^+,u_i^-)-(w_i^++w_i^-)S_n(w_i^+,w_i^-)\Big)\Big|\end{aligned}$$

In general, if $S_n(a,b)=\sum_{k+l=n}a^kb^l$, then it is not hard to show that $\Big|S_n(a,b)-S_n(c,d)\Big| \le \frac{n(n+3)}{2}\Big(|a-c|+|b-d|\Big)$ and $\Big|(a+b)S_n(a,b)-(c+d)S_n(c,d)\Big| \le (n+4)(n+1)\Big(|a-c|+|b-d|\Big)$. Hence we immediately get the following inequalities:

$$|\tilde{D}_i(u)-\tilde{D}_i(w)| \le \frac{\epsilon(n+3)}{2}(|u_i^+-w_i^+|+|u_i^--w_i^-|)$$
$$|\tilde{N}_i(u)-\tilde{N}_i(w)| \le (n+4)(|u_i^+-w_i^+|+|u_i^--w_i^-|)$$

Finally, since $\tilde{D}_i \le 1+\epsilon$, and $\tilde{N}_i \le 2$,

$$|Q_i(u)-Q_i(w)| \le (1+\epsilon)|\tilde{N}_i(u)-\tilde{N}_i(w)|+2|\tilde{D}_i(u)-\tilde{D}_i(w)| \le$$
$$(1+\epsilon)(2n+7)(|u_i^+-w_i^+|+|u_i^--w_i^-|)$$

so that

$$|Q_i(u)-Q_i(w)| \le (1+\epsilon)(2n+7)(|u_i^+-w_i^+|+|u_i^--w_i^-|) \tag{3.4.4}$$

□

Now let us consider a small enough perturbation of a density in the form $\rho(x)=1+\epsilon g(x)$, where $g(\cdot)$ is any smooth function on $[a,b]$. With the help of Lemma 3.4.2 we can derive the following

Proposition 3.4.2. *For any $\rho(x)=1+\epsilon g(x)$, there exist constants $C_l(\epsilon,k)$ and*

$C_u(\epsilon,k))$ *such that*

$$C_l(\epsilon,k)\sum_{i=1}^{k+1}(a_i-b_i)^2 \le (\tilde{\mathcal{H}}'(u)-\tilde{\mathcal{H}}'(w),u-w) \le C_u(\epsilon,k)\sum_{i=1}^{k+1}(a_i-b_i)^2$$

where $a_i = u_i - u_{i-1}, b_i = w_i - w_{i-1}, i = 1,\dots,k\,,\ u_0 = w_0 = a, u_{k+1} = w_{k+1} = b$ *and*

Proof. Let T_i be the centroid of the i-th cell. Then for any density function of the form $\rho(x) = 1 + \epsilon g(x)$, we have

$$\frac{\partial\tilde{\mathcal{H}}}{\partial u_i} = 2(u_i - T_i) = \frac{2}{M_i(u)}\Big(\int (u_i-u)du + \epsilon\int (u_i-u)g(u)du\Big) =$$

$$= \frac{(u_i^+ - u_i^-)(2u_i - u_i^+ - u_i^-) + 2\epsilon\int (u_i-u)g(u)du}{(u_i^+ - u_i^-) + \epsilon\int g(u)du} =$$

$$= 2u_i - u_i^+ - u_i^- + \frac{2\epsilon\int (u_i-u)g(u)du - \epsilon(u_i^+ - u_i^-)\int g(u)du}{(u_i^+ - u_i^-) + \epsilon\int g(u)du} =$$

$$= \frac{1}{2}\Big(a_i - a_{i+1}\Big) - \epsilon\Big(\frac{\int (2u - (u_i^+ + u_i^-))g(u)du}{(u_i^+ - u_i^-) + \epsilon\int g(u)du}\Big) = \frac{1}{2}\Big(a_i - a_{i+1}\Big) - \epsilon Q_i$$

Then

$$(\tilde{\mathcal{H}}'(u)-\tilde{\mathcal{H}}'(w),u-w) = \frac{1}{2}\sum_{i=1}^{k}(a_i-b_i)^2 - \epsilon(Q(u)-Q(w),u-w) \qquad (3.4.5)$$

The first term in (3.4.5) comes from the constant part of the density and hence complies with the results of the previous theorem. It remains to get a similar estimation for the second term. From Cauchy inequality,

$$|(Q(u)-Q(w),u-w)| \le \sum_{i=1}^{k}|Q_i(u)-Q_i(w)|\cdot|u_i-w_i| \qquad (3.4.6)$$

Combining (3.4.4), (3.4.5) and (3.4.6), we get

$$\begin{aligned} & (\tilde{\mathcal{H}}'(u) - \tilde{\mathcal{H}}'(w), u - w) \leq \\ \leq \ & \tfrac{1}{2} \sum_{i=1}^{N} (a_i - b_i)^2 + 2\epsilon(1+\epsilon)(2n+7) \sum |u_i^+ - w_i^+||u_i - w_i| \\ \leq \ & \tfrac{1}{2} \sum_{i=1}^{N} (a_i - b_i)^2 + 2\epsilon(1+\epsilon)(2n+7) \sum |u_i - w_i|^2 \end{aligned}$$

Since $x_i = x_i - x_0 = \sum_{l=1}^{i} (x_l - x_{l-1})$, L_2-norm can be dominated by H_1-norm as follows: $\sum x_i^2 \leq 2k \sum (x_i - x_{i-1})^2$. Hence

$$(\tilde{\mathcal{H}}'(u) - \tilde{\mathcal{H}}'(w), u - w) \leq (\frac{1}{2} + 4\epsilon(1+\epsilon)k(2n+7)) \sum_{i=1}^{k} (a_i - b_i)^2 = C_u(\epsilon, k, n)||u - w||_1^2$$

Same arguments applied to the lower bound yield

$$(\tilde{\mathcal{H}}'(u) - \tilde{\mathcal{H}}'(w), u - w) \geq (\frac{1}{2} - 4\epsilon(1+\epsilon)k(2n+7)) \sum_{i=1}^{k} (a_i - b_i)^2 = C_l(\epsilon, k, n)||u - w||_1^2$$

By using Taylor expansion, this result can be extended to all smooth functions $g(\cdot)$ on $[a, b]$. The statement of the lemma follows. □

Note that it follows from Proposition 3.4.2 that in order to preserve convexity the perturbation has to be of the order of $\epsilon = O(k^{-1})$.

In addition to showing that the energy functional possesses convexity and continuity properties, we also need the following conditions on the space decomposition to be satisfied:

Condition 3.

$\forall v \in \mathbf{W}, \exists v_i \in \bar{\mathbf{W}}_i s.t. \sum_{i=1}^{J} v_i = v$, and

$$(\sum_{i=1}^{J} ||v_i||_{1,\mathbf{W}}^2)^{1/2} \leq C_1 ||v||_{1,\mathbf{W}}$$

Condition 4. 'Strengthened Cauchy-Schwartz'

$\forall w_{ij} \in \mathbf{W}, u_i \in \bar{\mathbf{W}}_i, v_j \in \bar{\mathbf{W}}_j \Rightarrow$

$$\sum_{i,j=1}^{J} (\tilde{\mathcal{H}}'(w_{ij}+u_i) - \tilde{\mathcal{H}}'(w_{ij}), v_j) \leq C_2 (\sum_{i=1}^{J} ||u_i||^2_{1,\bar{\mathbf{W}}_i})^{1/2} (\sum_{j=1}^{J} ||v_j||^2_{1,\bar{\mathbf{W}}_j})^{1/2}$$

Theorem 3.4.3. *For the nested subspace decomposition with the choice of "'hat"' basis functions,* $(\sum_{i=1}^{J} ||v_i||^2_{1,\mathbf{W}})^{1/2} = ||v||_{1,\mathbf{W}}$ *, so that* $C_1 = 1$ *. Moreover,* C_2 *can be estimated as* $C_2 = L \cdot \max_j (\sum_{l=1}^{J} 2^{-|j-l|}) \leq 2L$ *.*

Proof. Notice that "'hat"' functions form an orthogonal basis, so

$$(\sum_{i=1}^{J} ||v_i||^2_{1,\mathbf{W}})^{1/2} = ||v||_{1,\mathbf{W}}$$

follows easily from calculation. As for the C_2, first notice that for any $w, u, v \in \mathbf{W}$,

$$(F'(w+u) - F'(w), v) \leq L||u||_{1,\mathbf{W},supp(u)\cap supp(v)} ||v||_{1,\mathbf{W},supp(u)\cap supp(v)}$$

Now since $supp(u) \cap supp(v) \subseteq supp(v) \quad \forall u \in \mathbf{W}_j, v \in \mathbf{W}_l$, for the "'hat"' basis we get $||v||_{1,\mathbf{W},supp(v)\cap supp(u)} = (\frac{1}{2})^{|j-l|} ||v||_{1,\mathbf{W}}$. Then

$$\begin{aligned} &\textstyle\sum_{i,j=1}^{J} (F'(w_{ij}+u_i) - F'(w_{ij}), v_j) \leq L \sum_{i,j=1}^{J} (\frac{1}{2})^{|i-j|} ||u_i||_{1,\mathbf{W}} ||v_j||_{1,\mathbf{W}} \leq \\ &\leq L \Big(\max_j \textstyle\sum_{i=1}^{J} (\frac{1}{2})^{|i-j|} \Big) \Big(\textstyle\sum_{i=1}^{J} ||u_i||^2_{1,\bar{\mathbf{W}}_i} \Big)^{1/2} \Big(\textstyle\sum_{j=1}^{J} ||v_j||^2_{1,\bar{\mathbf{W}}_j} \Big)^{1/2} \end{aligned}$$

Henceforth, $C_2 = L \cdot \max_j \sum_{i=1}^{J} (\frac{1}{2})^{|i-j|} \leq 2L$

3.4.4 Uniform convergence theorem

Finally, putting together Conditions 3,4 and using convexity and continuity of $\tilde{\mathcal{H}}$ in $\mathbf{W}$, we're equipped to prove the following uniform convergence result:

Theorem 3.4.4. *Under assumptions 3,4 on space decomposition, Algorithm 3.1 converges uniformly in* $\mathbf{W}$ *for any density of the type* $\rho(x) = 1 + \epsilon g(x)$ *with* $d_n = \tilde{\mathcal{H}}(u_n) - \tilde{\mathcal{H}}(u)$ *satisfying*

$$d_n \leq r d_{n-1}, \quad \rho \in (0,1)$$

for some constant $r = \frac{C}{1+C}$, *where* $C = C_1^2 C_2^2 L/K^3$.

The proof of this result is similar to the one given in [66] and is provided below.

Corollary 3.4.5. *In the case of a "'hat"' basis, the constants* C_1 *and* C_2 *can be estimated as* $C_1 = 1$ *and* $C_2 = 2L$, *so for example when* $\rho(x) = 1$, $C = 4$.

3.4.5 Proof of the main result

In order to prove the main theorem, first consider the following lemma.

Lemma 3.4.6. *Suppose the functional* F *satisfies conditions 1 and 2 in* $\mathbf{W}$. *Then the following statements are true for all points* $v, w \in \mathbf{W}$:

$$F(v) - F(w) \geq (F'(v), w - v) + \tfrac{K}{2}||w - v||^2_{\mathbf{W}}$$
$$F(v) - F(w) \leq (F'(v), w - v) + \tfrac{L}{2}||w - v||^2_{\mathbf{W}}$$

Proof. Let $\phi(\lambda) = F(u + \lambda(v - u))$. Then $\phi'(\lambda) = (v - u, F'(u + \lambda(v - u)))$, $\phi(0) = F(u)$, $\phi(1) = F(v)$. First inequality can be verified using fundamental

theorem of calculus and convexity assumption:

$$F(u) - F(v) = \phi(0) - \phi(1) = -\int_0^1 \phi'(t)dt =$$
$$= -\int_0^1 (v-u, F'(u+t(v-u)))dt = \int_0^1 (u-v, F'(u+t(v-u)))dt =$$
$$= \int_0^1 (u-v, F'(u+t(v-u)) - F'(v))dt + \int_0^1 (u-v, F'(v))dt =$$
$$= (F'(v), u-v) + \int_0^1 (F'(u+t(v-u)) - F'(v), u+t(v-u)-v)\frac{dt}{1-t} \geq$$
$$\geq (F'(v), u-v) + K\int_0^1 ||(1-t)(u-v)||^2 \frac{dt}{1-t} = (F'(v), u-v) + \frac{K}{2}||u-v||^2$$

The proof of the second inequality is analogous and follows from the continuity of functional F. □

Proof of Theorem 3.4.4. Denote u to be the exact solution. First notice that

$$F(u_n) - F(u_{n+1}) = \sum (F(u_{n+\frac{i}{J}}) - F(u_{n+\frac{i-1}{J}})) \geq$$
$$\geq \sum (F'(u_{n+\frac{i-1}{J}}), u_{n+\frac{i}{J}} - u_{n+\frac{i-1}{J}}) + \frac{K}{2}||u_{n+\frac{i}{J}} - u_{n+\frac{i-1}{J}}||^2_{1,\mathbf{W}} = \frac{K}{2}\sum_{i=1}^{m} ||e_n^i||^2_{1,\mathbf{W}}$$

Next, let's use Condition 3 to decompose $u_{n+1} - u = \sum_{i=1}^{J} v_i$. Then

$$(F'(u_{n+1}) - F'(u), u_{n+1} - u) = \sum_{i=1}^{J} (F'(u_{n+1}) - F'(u_{n+\frac{i-1}{J}} + e_i^n), v_i) =$$
$$= \sum_{i=1}^{J}\sum_{j\geq i}^{J} (F'(u_{n+\frac{j}{J}}) - F'(u_{n+\frac{j-1}{J}}), v_i) \leq C_2(\sum_{j=1}^{J} ||e_n^j||^2_{1,\mathbf{W}_j})^{1/2}(\sum_{i=1}^{J} ||v_i||^2_{1,\mathbf{W}_i})^{1/2}$$

Hence

$$(F'(u_{n+1}) - F'(u), u_{n+1} - u) \leq C_1C_2(\sum_{j=1}^{J} ||e_n^j||^2_{1,\mathbf{W}_j})^{1/2}||u_{n+1} - u||_{1,\mathbf{W}} \leq$$
$$\leq C_1C_2(\frac{2}{K}(F(u_n) - F(u_{n+1})))^{1/2}||u_{n+1} - u||_{1,\mathbf{W}}$$

Denote $r_n = F(u_n) - F(u)$, then $F(u_n) - F(u_{n+1}) = r_n - r_{n+1}$ and it follows from the inequality above that

$$
\begin{aligned}
&(\frac{2}{K}(r_n - r_{n+1}))^{1/2} \geq \frac{(F'(u_{n+1}) - F'(u), u_{n+1} - u)}{C_1 C_2 ||u_{n+1} - u||_{1,\mathbf{W}}} \Rightarrow \\
&r_n - r_{n+1} \geq \frac{K}{2}(\frac{(F'(u_{n+1}) - F'(u), u_{n+1} - u)}{C_1 C_2 ||u_{n+1} - u||_{1,\mathbf{W}}})^2 \geq \\
&\frac{K}{2}(C_1 C_2)^{-2} K^2 ||u_{n+1} - u||^2_{1,\mathbf{W}} \geq \frac{K^3}{C_1^2 C_2^2 L} r_{n+1}
\end{aligned}
$$

Last step of the argument uses the result of Lemma 1: $r_{n+1} = F(u_{n+1}) - F(u) \leq \frac{L}{2}||u_{n+1} - u||^2_{1,\mathbf{W}}$. As a consequence, we get

$$
r_{n+1} \leq \frac{C_1^2 C_2^2 L}{K^3}(r_n - r_{n+1}) \Rightarrow r_{n+1} \leq \frac{C}{1+C} r_n, \text{where } C = \frac{C_1^2 C_2^2 L}{K^3}
$$

□

It follows that for the suitable choice of decomposition in 1D the asymptotic factor is independent of the size of the problem and the number of grid levels, which gives a significant speedup comparing to other methods, like the traditional Lloyd iteration. We justify this in the numerical examples that follow.

It follows that for a suitable choice of decomposition in 1D the asymptotic convergence factor of our multilevel algorithm is independent of the size of the problem and the number of grid levels, which gives a significant speedup comparing to other methods, like the traditional Lloyd iteration. This claim can be justified by the following numerical examples, computed using the Matlab 6.5 implementation of the new algorithm on a Pentium IV with 512MB RAM.

3.4.6 Numerical results

- **One-dimensional examples**

Below are the computational results obtained for the V(1,1) multigrid implementation of the new algorithm in comparison with the regular Gauss-Seidel performance. We plot the convergence factor $\rho \approx \dfrac{z_{n+1} - z_n}{z_n - z_{n-1}}$ for each V(1,1) cycle with respect to the total number of generators (grid points) involved.

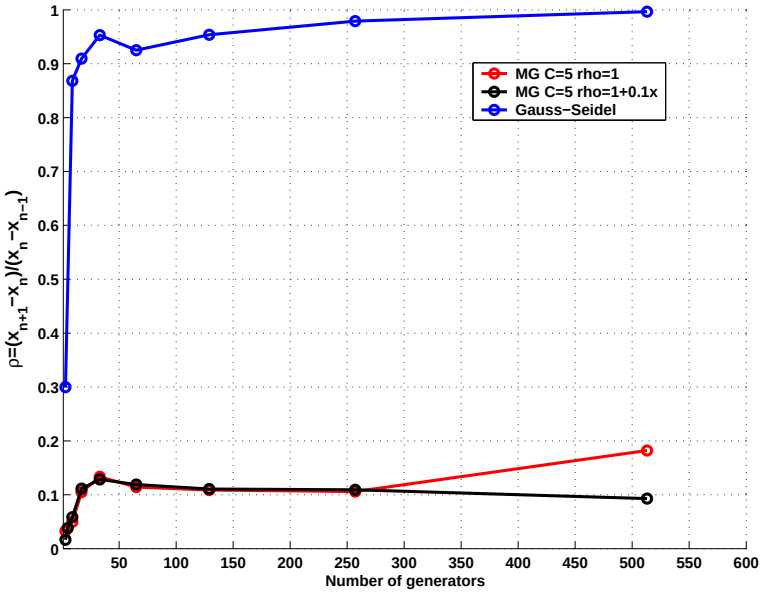


Figure 3.8. Plot of the convergence factor over the number of generators for the multigrid method vs. regular Gauss-Seidel method for $\rho = 1$ and $\rho = 1 + 0.1x$

Table I shows the number of multigrid cycles $V(\nu_1, \nu_2, \mu)$ needed to reduce the error to $\epsilon = 10^{-12}$.

$k/V(\nu_1,\nu_2)$	V(1,0)	V(0,1)	V(1,1)	V(2,0)	V(0,2)	V(2,2)
3	7	8	6	6	7	4
5	11	11	8	8	8	6
9	13	14	9	9	9	7
17	18	18	12	12	12	8
33	21	20	13	12	13	8
65	21	22	12	12	12	8
129	21	21	12	12	12	8
257	20	23	12	12	13	7
513	20	22	12	11	13	7
1025	19	22	11	11	13	7

Table I. Number of $V(\nu_1,\nu_2)$ cycles needed to reduce the error to machine zero vs. number of generators.

Figure 3.8 justifies the fact that the speed of convergence for proposed scheme does not grow with the number of generators.

The geometric rate of energy and error reduction asserted by the Theorem 3.4.4 is confirmed by the experiments. Indeed, Figure 3.9 shows convergence history of a $V(1,1)$-cycle vs. total number of relaxations for the $k = 64$ case.

The results for other nonlinear densities , though not shown here, comply with the theoretical conclusions reached above (see [23]).

- **Two-dimensional examples**

  Let us look at the possibility of extending this framework to higher dimensions. One of the big questions arising with the increase of dimension is

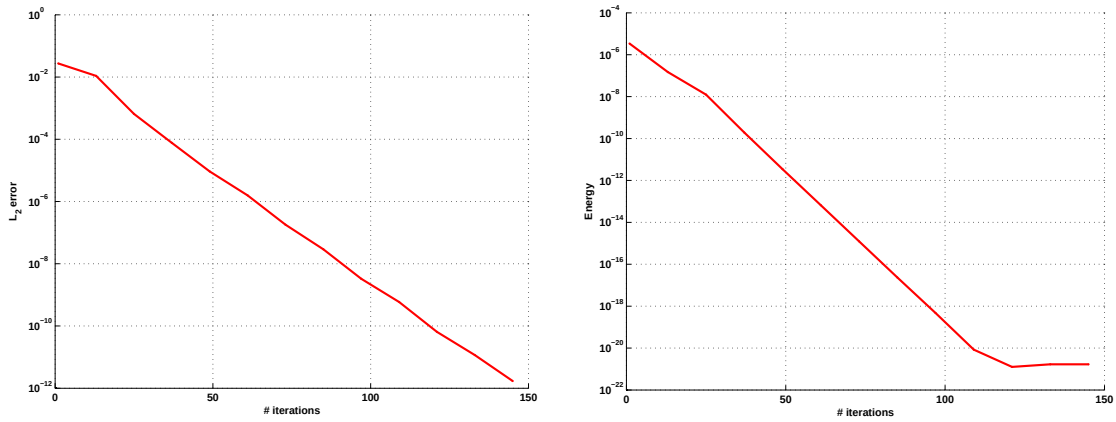


Figure 3.9. (a) Convergence history for $k = 64$ generators (log-normal scale); (b)Energy reduction for $k = 64$ generators (log-normal scale)

whether the multidimensional Lloyd iteration will possess enough smoothing properties to efficiently damp the higher frequency modes of the error. This fact can be readily observed if we plot the components of the error after sufficiently many relaxations compared to the original distribution (see Figure 3.10). More precisely, one can fix a particular coarsening scheme and look

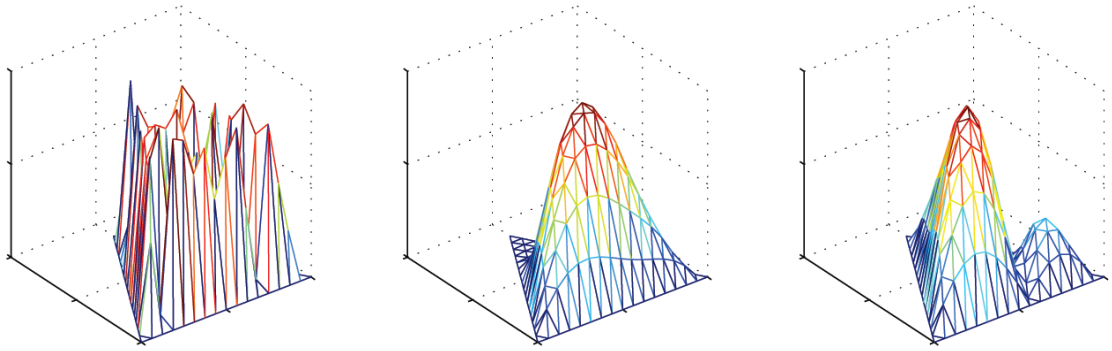


Figure 3.10. (a)Original error distribution; (b) x-component of the error after 50 Lloyd iterations; (c) y-component of the error after 50 Lloyd iterations

at the performance of the relaxation on the fine grid points with the exact solution fixed at the coarse positions. This process of leaving the coarse grid variable invariant was first introduced by Achi Brandt in [3], where he called

it a *compatible relaxation*. In Figure 3.11(left picture), we show an example of the performance of a coarsening scheme for the parallelogram grid and plot the corresponding convergence factor of the Lloyd smoothing applied to the fine grid versus the performance of the full multigrid cycle, shown on the left.

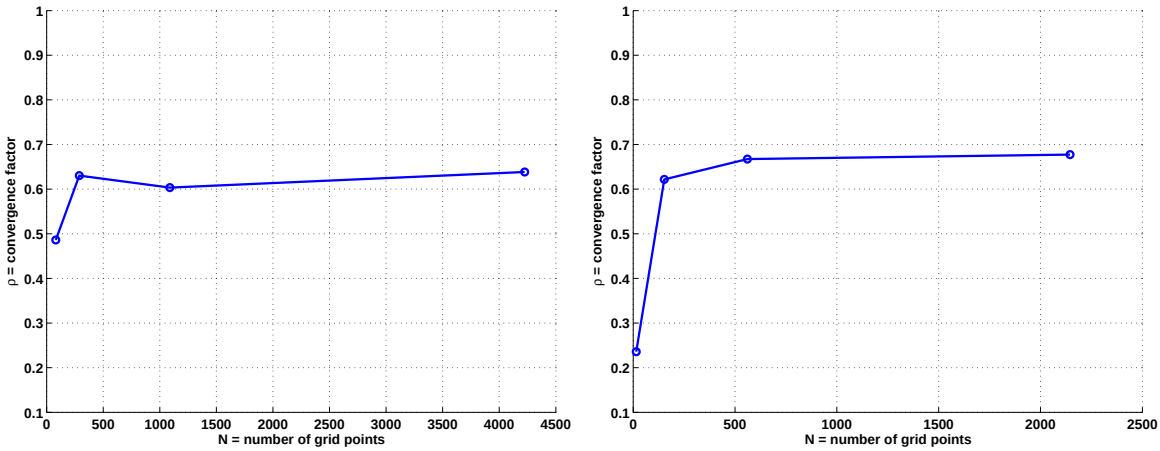


Figure 3.11. (a) Convergence factor for the compatible relaxation; (b) Convergence factor of the smoother

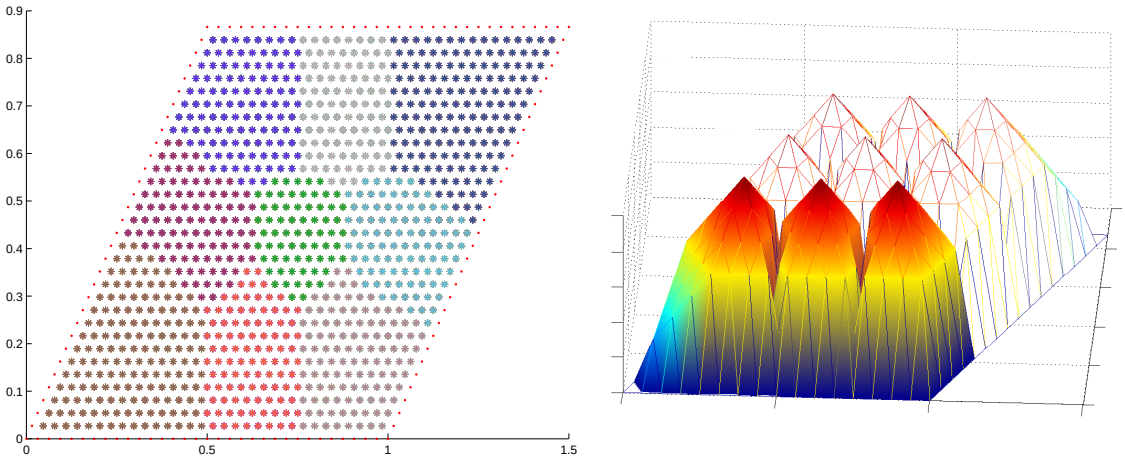


Figure 3.12. (a) Distribution of basis functions supports on the coarsest level; (b) Corresponding hierarchical basis functions

Clearly, convergence factor does not grow with k, justifying the fact that Lloyd iteration is an appropriate smoother for this type of problem.

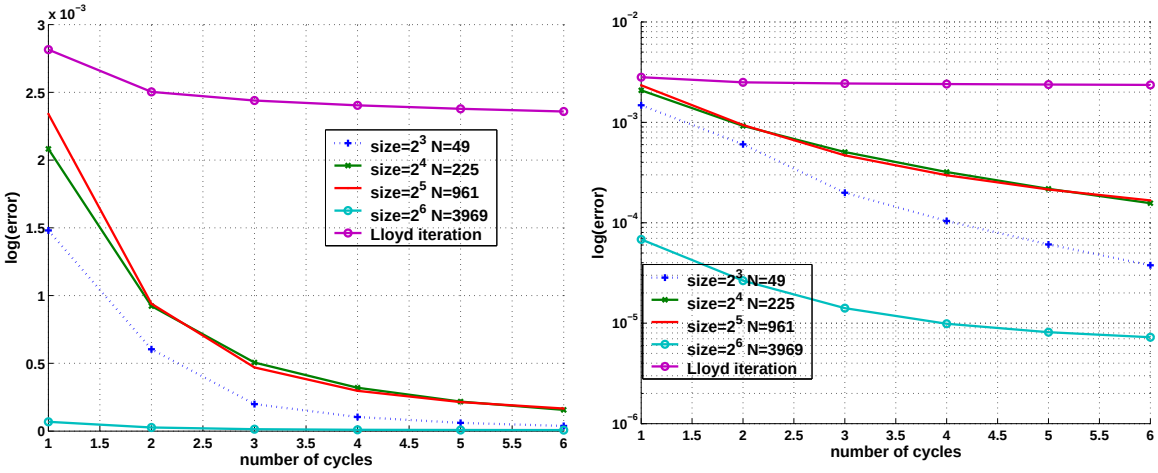


Figure 3.13. (a)Convergence history of multigrid for $k = 8, 16, 32, 64$ vs. Lloyd iteration; (b) Log-normal plot of the convergence history for $k = 8, 16, 32, 64$ vs. Lloyd iteration

The rigorous multidimensional extensions are discussed in more details in [24] .

3.5 Conclusion

In this chapter we introduced two methods for finding optimal quantizers that overcome the drawbacks of the Lloyd algorithm outlined in Chapter 2. The coupling of the fixed point Lloyd iteration with quasi-Newton approach yields faster convergence speeds close to the solution, as demonstrated by the numerical examples. Theoretical results were provided that can help estimate the radius of the convergence region and get the best algorithm performance. Advantages and disadvantages of this scheme were discussed in details.

Another approach presented in this chapter is the new energy-based multigrid method for quantization where a dynamic nonlinear preconditioning helps to take advantage of a nonlinear convex optimization setting. This is the first multigrid quantization scheme in literature with a rigorous proof of uniform convergence with

respect to the grid size and the number of grid levels. The scheme demonstrated significant speedup comparing to the traditional Lloyd's method. More work is under way for the analysis of the multigrid scheme in higher dimensions.

Both of the algorithms presented in this Chapter focus primarily on the efficiency of finding a local minimizer and do not provide any guarantees of global optimality for the solution. As mentioned earlier, the possible ways to combine these schemes with global minimization techniques are part of the ongoing research in the area. In some applications, however, it may be more critical to find an acceptable approximation to the global solution. For these problems, the concepts of adaptivity and efficient data sampling often play important roles. In the next Chapter, we will discuss a problem arising in materials science that requires this type of global optimization techniques.

Chapter 4

A new algorithm for the automation of phase diagram calculation

4.1 Overview

In this Chapter we take a slightly different view of the problems arising in nonlinear optimization context. Along with the issues of stability and fast local convergence discussed in the previous Chapters, other features such as robustness and global optimality may play equally important roles in many practical applications. For the materials science problem we consider next, for example, it is crucial to get a solution sufficiently close to the global minimizer in order to produce the correct phase diagram and even small deviations from the exact solution may have severe manufacturing consequences. There have been several studies of the global optimization problems for nonconvex functionals in various settings, but no universal solution is known to this point. One of the common drawbacks of these numerical methods is the slow adjustability to the geometry and as a consequence a possibly poor quality of the starting point for a general type of the cost functional. The

approach we take is based on the concept of adaptivity and effective sampling techniques that help to improve the initial guess and increase the attainability of the global solution. Apart from the mathematical perspective, the optimization problem presented here has its own materials related specifics that also plays a role in the numerical model.

To give a short overview of the background underlying current study, let us look at the concept of phase diagrams the way they are used in materials science applications. Phase diagrams are visual representations of the equilibrium phases in a material as a function of temperature, pressure and concentrations of the constituent components and are frequently used as basic blueprints for materials research and development. Under typical experimental conditions of constant pressure and temperature and a closed system, calculated phase equilibria are obtained via minimization of the total Gibbs energy of a system by adjusting the compositions and amounts of all individual phases in the system. As in any minimization procedure, the starting values play an important role due to the existence of many possible metastable states.

Many existing software packages lack the ability to automatically determine system properties from initial data and can produce metastable equilibria instead of stable ones or simply diverge if the initial guess is not good enough. Several algorithms were proposed to automate the process of finding suitable starting positions, all of which carry an increased computational cost. In this chapter we make an attempt to improve on the existing strategies for automating phase diagram calculations by introducing a novel reduced complexity algorithm based on adaptive critical point detection approach. In doing so we will mostly rely on the results of [30], where it was first presented. The main advantage of the new scheme lies in its ability to effectively reduce the total number of trial calculations by recognizing

the importance of geometry specific properties of the Gibbs energies.

We start with an introduction to the necessary theoretical background and a short overview of existing techniques in section 4.2, which are succeeded by the detailed description of the new algorithm as well as its generalizations in section 4.3. Section 4.4 contains the results of several numerical calculations for binary and ternary systems. Some concluding remarks are made in section 4.5.

4.2 Theoretical aspects of phase diagram calculation

4.2.1 Mathematical model

Let us fix both the temperature and the pressure as independent system variables with a total of one mole of components. Let f_k be the total content of the k-th component in the system and f_k^i the content of the k-th component in the i-th phase and $f^{(i)}$ the number of moles of the phase i (by our assumption $\sum_{k=1}^{K} f_k^i = f^{(i)}$). We also let $f = (f_k)_{k=1,\ldots,K}$. Since it is easier to work with molar quantities, we use $x^{(i)} = (x_k^i)_{k=1,\ldots,K} = (f_k^i/f^{(i)})_{k=1,\ldots,K}$ to denote the vector consisting of mole fractions of the k-th component in the phase i and $G^{(i)}$ is the corresponding molar Gibbs energy. In this case, the equilibrium analysis of a K-component

system with n phases leads to the following Gibbs energy minimization problem:

$$\begin{cases} \min\limits_{(f,x)} G = \sum\limits_{i=1}^{n} f^{(i)} G^{(i)}(x^{(i)}) \\ \quad \sum\limits_{i=1}^{n} f^{(i)} x^{(i)} = f \\ \sum\limits_{k=1}^{K} x_k^i = 1, i = 1, \ldots, n \\ \quad f^{(i)} \geq 0, x_k^i \geq 0 \end{cases} \tag{4.2.1}$$

The equality constraints given in the above equations arise from the preservation of both component and total mass. The minimization problem is also subject to the above natural inequality constraints that guarantee nonnegativity of contents and mole fractions for each of the phases.

Suppose $x^* = \left(f^{(i)}, x^{(i)}\right)^*$ is an extremum of G. An index set A is defined to include those indices whose corresponding inequality constraint belongs to the active set of constraints at x^*. Recall that a local constrained extremum x^* coincides with an unconstrained critical point of the following Lagrangian, where $\lambda_i, \eta_k, \gamma_i, \tau_k^i$ are the Lagrange multipliers corresponding to the equality and inequality constraints introduced above:

$$\begin{aligned} L = \sum_{i=1}^{n} f^{(i)} G^{(i)}\left(x^{(i)}\right) - \sum_{i=1}^{n} \lambda_i \left(\sum_{k=1}^{K} x_k^i - 1 \right) \quad & - \sum_{k=1}^{K} \eta_k \left(\sum_{i=1}^{n} f^{(i)} x_k^i - f_k \right) - \\ & - \sum_{i \in A} \gamma_i f^{(i)} - \sum_{i,k \in A} \tau_k^i x_k^i \end{aligned} \tag{4.2.2}$$

Here, we may use the theory on the complementarity property of the constraints and the Lagrange multipliers so that only active constraints at the extremum point need to be considered. Notice, however, that if an inequality constraint given in 4.2.1 becomes active, the total content for some phases becomes zero, which implies

that there are phases that do not take part in the equilibrium. If these phases are known at the beginning, we can discard them and therefore reduce the problem dimension. However, it is sometimes difficult to determine which phases would form an equilibrium a priori, so a generalized approach is preferred.

We can, however, make the computation more efficient by monitoring the phase contents in the process of optimization and eliminate those phases whose contents become insignificant. The issue with this approach is that a phase content may accidentally become very small in the course of the numerical procedure, so discarding a phase completely without allowing it to reappear may lead to unwanted consequences. The usual tactic is to assign a small tolerance value $\epsilon > 0$ to all phases whose contents are lower than ϵ during the equilibrium calculation. This assures that all phases have equal chances of contributing to the equilibrium state, at the same time making all inequality constraints inactive. In other words, we arrive at the following problem:

$$L = \sum_{i=1}^{n} f^{(i)} G^{(i)}\left(x^{(i)}\right) - \sum_{i=1}^{n} \lambda_i \left(\sum_{k=1}^{K} x_k^i - 1 \right) - \sum_{k=1}^{K} \eta_k \left(\sum_{i=1}^{n} f^{(i)} x_k^i - f_k \right) \quad (4.2.3)$$

The well known Karush-Kuhn-Tucker theorem for optimization problems (see [61]) asserts that at the critical point, the following set of first order conditions is met:

$$\begin{cases} \dfrac{\partial L}{\partial f^{(i)}}\left(x^*\right) = G^{(i)}\left(x^{(i)}\right) - \displaystyle\sum_{k=1}^{K} \eta_k x_k^i = 0 \\ \dfrac{\partial L}{\partial x_k^i}\left(x^*\right) = f^{(i)} \dfrac{\partial G^{(i)}}{\partial x_k^i} - \lambda_i - f^{(i)} \eta_k = 0 \end{cases} \quad (4.2.4)$$

where $\lambda_i, \eta_k \geq 0$ at x^* and constraint equations in 4.2.1 are satisfied. We can

now solve for η_k from the second equation and substitute into the first one to get:

$$\begin{cases} \eta_k = \dfrac{\partial G^{(i)}}{\partial x_k^i} - \dfrac{\lambda_i}{f^{(i)}} \\ G^{(i)}(x^{(i)}) - \displaystyle\sum_{j=1}^{K} \dfrac{\partial G^{(i)}}{\partial x_j^i} + \dfrac{\lambda_i}{f^{(i)}} = 0 \end{cases}$$

It follows that

$$\eta_k = \frac{\partial G^{(i)}}{\partial x_k^i} + G^{(i)}\left(x^{(i)}\right) - \sum_{j=1}^{K} \frac{\partial G^{(i)}}{\partial x_j^i} x_j^i \tag{4.2.5}$$

for all $i = 1, \ldots, n$. Notice that the expression at the right hand side is the full derivative of the Gibbs energy $\tilde{G}^{(i)}((f_k^i)_{k=1,\ldots,n}) = f^{(i)} G^{(i)}(x^{(i)})$ with respect to contents f_k^i of the k-th component in the i-th phase. In other words, making a change of variables back from molar quantities to f_k^i, we conclude that for given temperature, pressure and overall composition, the minimum of the objective function satisfies the following equations:

$$\begin{cases} \mu_1^1 = \mu_1^2 = \ldots = \mu_1^n = \eta_1 \\ \mu_K^1 = \mu_K^2 = \ldots = \mu_K^n = \eta_K \end{cases} \tag{4.2.6}$$

where $\mu_k^i = \partial \tilde{G} / \partial f_k^i$. These equations are called Gibbs equilibrium conditions, which imply that the value of the chemical potential for each component k is the same in all phases $i = 1, \ldots n$.

Furthermore, from the first equation in 4.2.4, we get that

$$G^{(i)}\left(x^{(i)}\right) - G^{(j)}\left(x^{(j)}\right) = \eta^T \left(x^{(i)} - x^{(j)}\right)$$

Coupled with 4.2.6, this implies the common tangent hyper-plane property in the G-x space with the Lagrange multiplier η being the normal to the plane. Such a

property is well known for the phase diagram calculation.

It should be mentioned that these equations provide only necessary conditions. In order to guarantee that the solution found at this step is indeed a minimizer, one should verify that $w^T(\nabla^2 L)w > 0$ at x^* for all limiting directions w of a feasible sequence. Notice that at x^*,

$$w^T(\nabla^2 L)w = \sum_i w^{iT} \begin{pmatrix} f^{(i)}\left(\dfrac{\partial^2 G^{(i)}}{\partial x_k^i \partial x_j^i}\right) & \left(\dfrac{\partial G^{(i)}}{\partial x_k^i} - \eta_k\right) \\ \left(\dfrac{\partial G^{(i)}}{\partial x_k^i} - \eta_k\right) & 0 \end{pmatrix} w^i = \\ = \sum_i w^{iT} \begin{pmatrix} f^{(i)}\left(\dfrac{\partial^2 G^{(i)}}{\partial x_k^i \partial x_j^i}\right) & \dfrac{\lambda_i}{f^{(i)}} e \\ \dfrac{\lambda_i}{f^{(i)}} e & 0 \end{pmatrix} w^i$$

Here, e denotes the column vector with all components being equal to 1. We may write any feasible direction as $w^i = (w_1^i, w_2^i)^T$ and then make it adhere to the space of constraints (see [61]), which in this case leads to $w_1^{iT} e = 0$. Thus, we get $w^T(\nabla^2 L)w = \sum_i f^{(i)} w_1^{iT} \left(\dfrac{\partial^2 G^{(i)}}{\partial x_k^i \partial x_j^i}\right) w_1^i$ It follows that the solution $x*$ of the unconstrained problem 4.2.3 is a local minimum of the total Gibbs energy of the system provided that the Hessian of the mixing energy is positive definite at $x*$, otherwise it is possible that the point at hand is a maximum or a saddle. This observation explains why the regions of positive concavity are so important for the process of finding good initial guess that we are going to discuss in the following sections.

While the set of conditions provided by the KKT theorem is capable of identifying local solutions of the problem 4.2.3, phase diagrams often require the knowledge of stable equilibria of the system and hence call for a more careful analysis of the minimizers of the total Gibbs energy. As we have already noted, any local solu-

tion should satisfy the common tangent hyper-plane property in the G-x space, whereas stable solutions would have to belong to the lower convex hull determined by these points. In order to eliminate points belonging to the interior of the convex hull at any stage of the algorithm, one can perform two kinds of tests. The first one, referred to as a stability check, consists in determining whether a given phase possesses the minimal energy among all phases considered at this point. The remaining coplanarity check identifies whether there are solutions lying below the plane determined by a set of test points. Clearly any subset of points on the boundary of the convex hull satisfies both of these tests, and we're going to exploit this fact later in designing our method for stable diagram construction.

Going back to equations 4.2.6, notice that they result in a system of nonlinear equations which should be solved numerically. Hence the success of the whole task of calculating phase equilibria depends on the effectiveness of the scheme chosen to solve the nonlinear system.

Technical implementations of the nonlinear solution procedure differ from algorithm to algorithm. The two most popular ones rely on the Newton-Raphson and the simplex methods for iterative solution of the system 4.2.6. All of the CALPHAD-type software tools use methods like the two-step method of Hillert ([1],[44],[43]) or the one step method of Lukas et al. [56] to minimize the Gibbs energy. Typical drawbacks of these strategies include the use of prior knowledge in providing suitable starting points and the possibility of divergence or convergence to metastable minima.

Other methods were proposed that attempt to get a direct solution to the minimization problem 4.2.1 either by constructing phase field boundaries or determining the minimum free energy surface directly [12]. These algorithms do not suffer from stability issues as much as the iterative methods mentioned above, but

face problems with higher computational costs and the possible loss of information due to limited resolution.

We will focus our attention on improving the iterative solution adopted in the packages of the Thermocalc family. In doing so, we are going to follow the line of direct methods by recognizing the importance of geometrical information for the design of an efficient minimization scheme.

4.2.2 Geometrical considerations

As shown above, the procedure of finding solution to the minimization problem described here, from the geometric perspective, is nothing but a common tangent hyper-plane construction for the equilibrium phase surfaces in the $G - x$ space. Indeed, to give a more detailed illustration, let us consider the binary 2-phase case as an example. From the conservation of phase mass condition we have

$$\begin{aligned} x_1^1 = x^{(1)}; \quad x_2^1 = 1 - x^{(1)} \\ x_1^2 = x^{(2)}; \quad x_2^2 = 1 - x^{(2)} \end{aligned}$$

Hence, the minimization problem can be written as

$$\begin{cases} \min\limits_{(f,x)} \{G = f^{(1)}G^{(1)}(x^{(1)}) + f^{(2)}G^{(2)}(x^{(2)})\} \\ f^{(1)}x^{(1)} + f^{(2)}x^{(2)} = f_1 \\ f^{(1)}(1 - x^{(1)}) + f^{(2)}(1 - x^{(2)}) = f_2 \end{cases}$$

In simpler form,

$$\begin{cases} \min\limits_{(f,x)}\{G = f^{(1)}G^{(1)}(x^{(1)}) + f^{(2)}G^{(2)}(x^{(2)})\} \\ f^{(1)}x^{(1)} + f^{(2)}x^{(2)} - f_1 = 0 \\ f^{(1)} + f^{(2)} - (f_1 + f_2) = 0 \end{cases}$$

By means of the Lagrange multipliers μ and η, we can represent the above system in form of the following unconstrained minimization problem:

$$\begin{aligned} L = f^{(1)}G^{(1)}(x^{(1)}) + f^{(2)}G^{(2)}(x^{(2)}) \quad & -\mu\left(f^{(1)}x^{(1)} + f^{(2)}x^{(2)} - f_1\right) \\ & -\eta\left(f^{(1)} + f^{(2)} - (f_1 + f_2)\right) \end{aligned}$$

At an equilibrium, the partial derivatives of the Lagrangian with respect to $(\vec{f}, \vec{x})$ become zero:

$$\begin{gathered} \frac{\partial L}{\partial x^{(1)}} = f^{(1)}\left(\frac{\partial G^{(1)}(x^{(1)})}{\partial x^{(1)}} - \mu\right) = 0; \frac{\partial L}{\partial x^{(2)}} = f^{(2)}\left(\frac{\partial G^{(2)}(x^{(2)})}{\partial x^{(2)}} - \mu\right) = 0 \\ \frac{\partial L}{\partial f^{(1)}} = G^{(1)}(x^{(1)}) - \mu x^{(1)} - \eta = 0; \frac{\partial L}{\partial f^{(2)}} = G^{(2)}(x^{(2)}) - \mu x^{(2)} - \eta = 0; \end{gathered}$$

It follows that $\eta = G^{(1)}(x^{(1)}) - \mu x^{(1)} = G^{(2)}(x^{(2)}) - \mu x^{(2)}$ (similar derivation can be found e.g. in [65]. Hence the solution should satisfy the following equations

$$\begin{cases} \mu = \dfrac{\partial G^{(1)}(x^{(1)})}{\partial x^{(1)}} = \dfrac{\partial G^{(2)}(x^{(2)})}{\partial x^{(2)}} \\ \mu = \dfrac{G^{(1)}(x^{(1)}) - G^{(2)}(x^{(2)})}{x^{(1)} - x^{(2)}} \end{cases}$$

Geometrically, it is the common tangent line of Gibbs energy curves. Similar argument can be carried out in higher dimensions.

With this observation in mind, we are now ready to discuss the problems associated with the construction of phase diagrams using the existing algorithms and

some improved techniques.

4.2.3 Existing algorithms and motivation

As mentioned earlier, the minimization procedures adopted in existing iterative-type software have the following drawbacks:

- (stability) They either fail to converge or perhaps converge to some metastable equilibrium when a starting point is taken too far from the desired minimizer.
- (user-dependence) The computer programs cannot independently determine the existence of a miscibility gap, hence some prior knowledge of system properties is required.

Figure 4.1 demonstrates the problem in producing a correct phase diagram for a system with a miscibility gap. The diagram in Figure 1(a) is the correct phase diagram of the Ca-Li-Na system at $T = 900K$ produced by specifying the "set_miscibility_gap" option, while the wrong diagram in Figure 1(b) is the result of calculations when this option is not provided by the user.

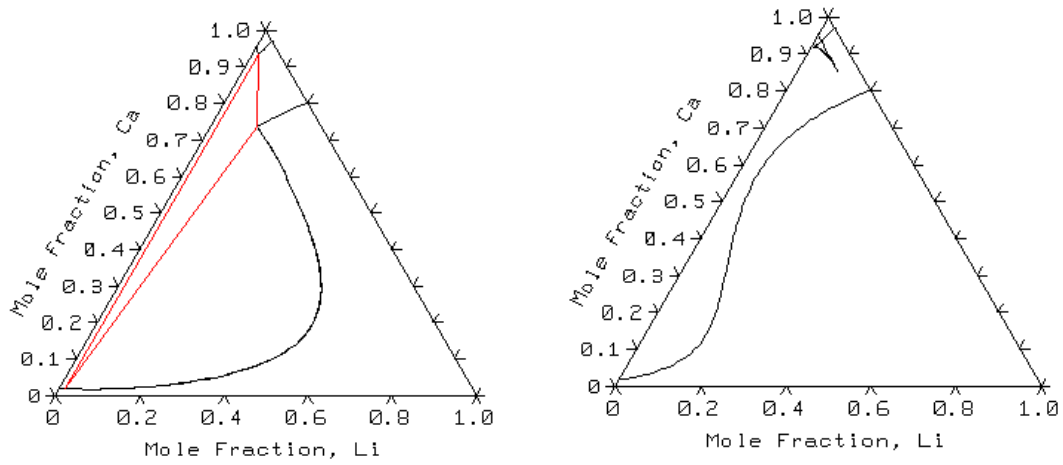


Figure 4.1. (a) Correct Ca-Li-Na phase diagram; (b) Incorrect Ca-Li-Na diagram produced by Thermocalc

Mathematically speaking, a miscibility gap arises when the Gibbs energy of a phase exhibits multiple minima. We need to design an algorithm capable of predicting system properties of this kind from the initial data. Ultimately this algorithm may be used as a basis to automate phase diagram calculation process as a whole.

The problem with miscibility gaps has been addressed before. A solution has been proposed by Chen et al [8], [9], [10]. Their method relies on a discretization of composition axis in order to represent solution phases by a set of stoichiometric compounds. It performs a series of tests to reveal the pairs of points which can coexist in stable equilibrium. These tests include stability checks (discarding points with higher energy values) and "coplanarity" checks, which test a pair for stable 2-phase equilibrium. As soon as candidate pairs are identified, they are taken as initial approximations for a consecutive minimization procedure, which is supposed to lead to the exact solution. In a two-phase case with N stoichiometric compounds the coplanarity (collinearity) condition holds, if

$$\frac{\begin{vmatrix} G_s & G_i & G_j \\ x_{1,s} & x_{1,i} & x_{2,j} \\ x_{2,s} & x_{2,i} & x_{2,j} \end{vmatrix}}{\begin{vmatrix} x_{1,i} & x_{1,j} \\ x_{2,i} & x_{2,j} \end{vmatrix}} > 0$$

for any of the compounds $A_{x_{1,s}}B_{x_{2,s}}$, $s = 1, ...N, s \neq i, j$ (see [8]).

This method generally gives a much better initial point for optimization, but at a higher computational cost. Indeed, even in 2D, the coplanarity check performed for each of the $(N(N-1))/2$ pairs of stoichiometric phases involves $(N-2)$

calculations of the determinants specified above. Since the numerator needs a total of 12 multiplications and 5 additions, while the denominator is calculated after 2 multiplications and 1 addition, the total complexity for the coplanarity check is of the order $O\left(N(N-1)(N-2)\right) \Rightarrow O\left(N^3\right)$ operations, where N is the number of points in the subdivision.

4.3 A new algorithm

Both of the aforementioned drawbacks have to do with the fact that Thermocalc does not possess the ability to recognize and utilize the geometric properties of Gibbs energy curves. The method of Chen et al described above takes into account system geometry, but limits its operation to function values, while it is the derivative information that seems to provide the best insight into the geometric structure of the object under study. Knowledge of the critical and inflection points of a system is a helpful tool in designing an efficient minimization algorithm. This is the key idea behind the new numerical scheme we are about to propose next, which overcomes the drawbacks of the previously mentioned algorithms, at the same time giving comparable accuracy to the solution.

4.3.1 Description of the algorithm: binary case

Since in general the Gibbs energy is represented by a nonlinear functional, the task of finding precise locations of its critical points may be too arduous. It makes sense to either deal with numerical derivatives instead, or to consider a reasonable approximation to the given functional. Although we are going to follow the first approach when presenting the algorithms, it is worth noting that in the binary case it is possible to use a polynomial least-squares fitting. We will return to this

point later in the discussion on the numerical characteristics of the algorithm.

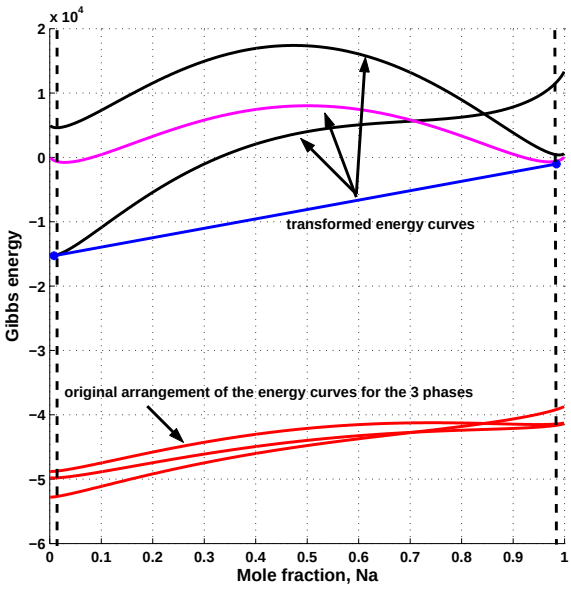


Figure 4.2. Affine transformation of the axis

First, notice that the procedure of finding critical points becomes much more difficult with the decrease of curvature values. Hence it is desirable to deal with functions that are not flat to begin with. In practice, many complex systems exhibit this type of behavior, thus leading to numerous problems and possible failures of computational software. Being aware of this fact, we use the following idea. Since a linear transformation of the carrying axis does not change the relative positions of minima for energy curves and does not significantly change their absolute locations, we can tilt the axis and use the modified geometry to get an initial approximation for the optimization procedure that is carried out later for the original configuration. The transformation suitable for these purposes has the form: $y_{new}(x) = M(y(x) - (y_m(1) - y_m(0))x - y_m(0))$. The constants here are chosen to make sure that the curve having the minimal value at the right end ($y_m(x)$) approaches zero on both sides. A scaling constant M is introduced to increase the curvature (we use $M = 2$ in the examples below). In Figure 4.2 we

display such a transformation for the Ca-Na system.

The transformation described here is performed only once at the initial stage of the construction so it does not increase the scheme complexity.

The main component of the algorithm to be proposed for a binary phase diagram construction is the recursive procedure of finding positions of critical points. This procedure relies on the adaptive refinement strategy and uses first and second order derivatives information to detect possible miscibility gaps and identify local minima with a prescribed accuracy ϵ. The other user defined parameters include the maximum number of refinements and the total number of axis subdivisions at each step. All derivatives in the algorithms described below are computed numerically by some finite difference approximation schemes.

Function *minima* = $AdaptiveSearch(a, b, phase, iter)$

Global parameters: N - the number of axis subdivisions, ϵ - tolerance, $Niter$ - maximum number of allowed refinements
Input parameters: a, b - ends of the interval, *phase* – phase index, *iter* – iteration index
Output parameters: *minima* – approximate position of the minima for the energy of the *phase*
while *(iter<=Niter)*
1. Sample N points $a = x_0 < x_1 < \ldots < x_N < x_{N+1} = b$.
2. For $(iter == 1)$ % finding concavity regions
(a) Calculate $\tilde{G}$ for $j = 1, \ldots, N$.
(b) Locate inflection points by finding indices $s \vert 1 \leq s \leq N$ such that $\tilde{G}$.
(c) Identify interval(s) for refinement by counting inflection points.

If no inflection points found, put k=1, $a(1) = a$, $b(1) = b$, endif.

If one inflection point x_s found and $\tilde{G}$, put k=1, $a(1) = a$, $b(1) = x_s$, endif

If one inflection point x_s found and $\tilde{G}$, put k=1, $a(1) = x_s$, $b(1) = b$, endif

If two inflection points x_{s_1}, x_{s_2} found, put k=2, $a(1)$=a, $b(1) = x_{s_1}$,

$a(2) = x_{s_2}$, $b(2)$=b, endif

(d) Perform recursive search on each of the identified intervals $(a(j), b(j))$:

$minima$(j)= $AdaptiveSearch(a(j), b(j), phase, 2)$

3. For $(iter > 1)$ % recursive search procedure

(a) Calculate G', $j = 1, \ldots, N$.

(b) Find $s = argmin_{j=1,\ldots,N} G'$

(c) If (G') or $(iter == Niter)$ % met stopping criteria

$minima = x_s$, return $minima$

else %recursive refinement

For $\delta = (b - a)/(2N)$ do

$minima = AdaptiveSearch(x_s - \delta, x_s + \delta, phase, iter + 1)$

end if

end while

return $minima$

In simple words, the method attempts to find approximate locations of all possible minima of the energy functional and take them as starting points for the subsequent minimization procedure. Note that since there are at least one and at most two minima for any unordered phase under consideration, the algorithm will refine the grid as long as it cannot detect any of them. As with any discrete numerical approximation, there may still be a chance of missing a minimum. If after a sufficient number of refinements, critical points are still not found, the

algorithm resorts to taking points with the lowest first derivative values as starting points for the later optimization. However, such situations are very rare in practice and are unlikely to cause troubles for most energy functionals due to the adaptive refinement strategy described above. The detection of a miscibility gap is straightforward due to availability of second derivative information.

It has to be noted that, in the 2D case, it is possible to avoid explicit calculations of the derivatives by making use of polynomial (in this case, quadratic) approximation. This approach has the same order of complexity as the method described above, but loses effectiveness when the dimension of the problem is increased. For the sake of generality we will use direct differentiation in all algorithms presented in this chapter.

We now are ready to present the algorithm of calculating the stable binary 2-phase equilibria. First let us introduce a couple of auxiliary structures.

The matrix $A(1:ind, 1:4)$ is used to record stability regions after the first sweep. Its first and second columns represent the coordinates of the left and right ends of the stability regions respectively, while the indices of the phases having lowest energy at those ends are recorded in the third and fourth columns.

Matrix C contains all the points that are obtained as suitable candidates for starting positions after the second sweep. The coordinate(s) of these points is recorded in the first (or first two in the ternary case) column(s) and the index of the corresponding phase goes into the last column of this matrix. The operation of adding a new row to this matrix is denoted everywhere in the text by the arrow sign "$\leftarrow$". With these notations, the stability region calculation procedure is given as follows.

Function $[A, ind] = StabilityRegions(a, b, N, K)$

Input parameters:	*a,b – ends of the interval. N – number of grid points*
	K - number of phases present
Output parameters:	A – array recording stability regions information
	ind – total number of stability regions in array A

1) Subdivide domain $V = [a, b]$ into $N - 1$ subdomains $V_j = [x_j, x_{j+1}]$,

$$a = x_1 < x_2 < \ldots < x_N = b$$

2) Initialize $ind = 1, A(1,1) = a, A(1,2) = a, A(1,3) = 1, A(1,4) = 1$;

For $j = 2, ..., N - 1$ do

For $i = 1, \ldots K$ do

(a) Calculate $G^{(i)}(x_j)$

(b) Find the phase with lowest energy among calculated energy values at x_j and x_{j+1}:

$\sigma_{j,left} = \left\{ s | G^{(s)}(x_j) < G^{(i)}(x_j), \forall i < s \right\}$;

$\sigma_{j,right} = \left\{ s | G^{(s)}(x_{j+1}) < G^{(i)}(x_{j+1}), \forall i < s \right\}$

(c) If $\sigma_{j,left} = \sigma_{j,right}$,

$A(ind, 2) = x_j, A(ind, 4) = \sigma_{j,right}$ % extend old stability region

else % start new stability region

$A(ind, 2) = x_j, A(ind, 4) = \sigma_{j,right}$, $ind = ind + 1$;,

$A(ind, 1) = x_j, A(ind, 3) = \sigma_{j,right}$

end if

end for;

end for

return [A, ind]

The algorithm for constructing binary phase diagram with K phases can be

summarized as follows:

Algorithm 4.3.1. Binary diagram construction

1) Fix N – the number of grid points in major axis subdivision, ϵ *- tolerance,*

Niter - maximum number of allowed refinements

2) Do $[A, ind] =$ *StabilityRegions(0,1,2N,K)* *% Identify stability regions*

3) Calculate starting points for optimization

For $i = 1, ..., ind$ *, do*

$phase_1 = A(i, 3)$ *,* $phase_2 = A(i, 4)$

If $(phase_1 \neq phase_2)$ *%add points at the boundaries of stability regions*

$C \leftarrow (A(i, 1), phase_1)$ *and* $C \leftarrow (A(i, 2), phase_2)$;

else *% find minima inside each stability region*

$minima =$ $AdaptiveSearch(A(i, 1), A(i, 2), phase_1, 1)$

$C \leftarrow$ *(minima,* $phase_1$ *)*

end if

end for

4) Perform coplanarity checks to get the convex hull of points in C

5) Carry out optimization for all remaining pairs of points,

check result for consistency

6) Construct phase diagram using solution obtained in step 5.

Essentially, the method first detects the stability regions of the diagram, i.e. identifies which phase has the lowest energy in each of the intervals formed by the intersection points (see Figure 4.3). Then it proceeds to examine each of the intervals separately, identifying extrema and possibly other points (at most two per each region) that would serve as candidate ends of the common tangents between

curves. As soon as all such points are found, a coplanarity check removes all points that could possibly appear inside the convex hull. At this stage, an exact (or more precisely, a good numerical) solution is obtained by solving the nonlinear optimization problem described earlier. To ensure the most reliable results, the solution is further checked for consistency with one additional coplanarity check.

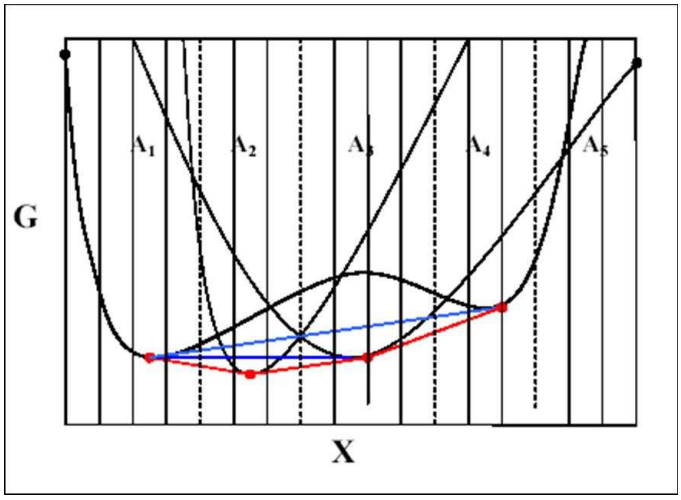


Figure 4.3. Stability regions

Notice that, in general, it is not possible to provide a good initial guess by only considering critical points of the energy curves. An example of such a situation is shown in Figure 4.4.

Although the minima of both parabolas can be easily detected by the adaptive scheme described above, only one of them provides a suitable initial guess for the optimization procedure. In this case it is necessary to pay attention to the appropriate endpoint of the stability region, as explained in the step 3(a) of the algorithm.

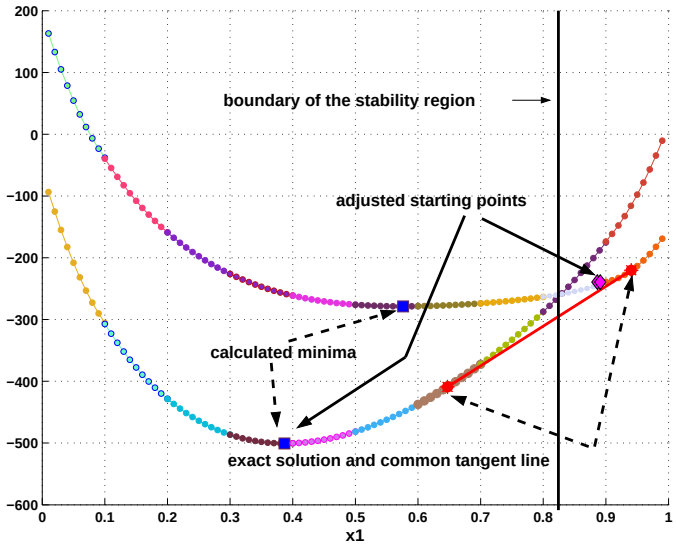


Figure 4.4. An example of one possible distribution of the starting points

4.3.2 Description of the algorithm: ternary case

In two and higher dimensions, due to changes in topological properties comparing to the 1D case, our algorithm needs to be modified accordingly in order to keep the calculation efficient. The first issue is the difficulty of working with curvilinear boundaries. If we want to identify the stability regions like we did in the binary case, we are up for a complicated task of working with unordered sets of data with various possible intersections. Instead of following this approach, we first identify critical point locations for all phases and then perform the coplanarity checks. This reduces the overall complexity, but leaves the necessity of adding boundary points to the set of candidate starting positions.

Another observation that has to be made is that the derivative calculation can hardly be avoided in dimensions higher than two, hence the scheme relies on the numerical differentiation, which calls for a good meshing approach. Some of the possible sampling strategies will be discussed in section 3.3. Here we only mention the importance of data ordering for the successful implementation of the general

Function *minima* = $AdaptiveSearch2D(V, phase, iter)$

Global parameters: N - the number of axis subdivisions, ϵ - tolerance,

$Niter$ - maximum number of allowed refinements

Input parameters: V – given domain, *phase* – phase index,

iter – iteration index

Output parameters: *minima* – positions of minima for the energy of the *phase*

---

while *(iter¡=Niter)*

1. Sample N points $x_j = (x_j(1), x_j(2)), j = 1, ..., N$ on V.

2. For $(iter == 1)$ % finding concavity regions

(a) Calculate $G''^{(phase)}(x_j)$ for $j = 1, \ldots, N$

(b) Find regions of positive concavity by identifying the sets V_i such that

$G''^{(phase)}(x_j)$ for any x in V_i. If there is only one such set, put $V = V_1$

(c) Perform recursive search on each of the identified regions V_i:

minima(i)= $AdaptiveSearch2D(V_i, phase, 2))$

3. For $(iter > 1)$ % recursive search procedure

(a) Calculate $G'^{(phase)}(x_j)$, $j = 1, \ldots, N$

(b) Find $s = \text{argmin}_{j=1,\ldots,N} G'^{(phase)}(x_j)$

(c) If $(G'^{(phase)}(x_s))$ or $(iter == Niter)$ % met stopping criteria

minima= x_s, return *minima*

else for $\delta = diam(V)/(2\sqrt{N})$ and

$V_s = [x_s(1) - \delta, x_s(1) + \delta] \times [x_s(2) - \delta, x_s(2) + \delta]$

$minima = AdaptiveSearch2D(V_s, phase, iter + 1)$%recursive refinement

end if

end while

return *minima*

algorithm to be presented. The recursive procedure of finding the critical points in the ternary case is given above. Below we give the details of the algorithm for computing stable 2-phase equilibia in ternary systems with a total of K phases. ZPF stands for the Zero Phase Fraction method, traditionally used to trace phase boundaries (see for example [10]).

Algorithm 4.3.2. Ternary diagram construction

1) Fix original domain as $V = \{(x, y)|x + y \leq 1\}$,

N – *the number of grid points in major axis subdivision,*

ϵ - *tolerance,* $Niter$ - *maximum number of allowed refinements*

2) For $phase = 1, ..., K$ *do*

(a) $minima = AdaptiveSearch2D(V, phase, 1)$

$C \leftarrow$ *(minima, phase)*

(b) Sample N points bdrypts on the boundary of domain V,

$C \leftarrow (bdrypts, phase)$

end

3) Perform coplanarity checks to get the convex hull of points in C

4) Carry out optimization for all remaining pairs of points,

check result for consistency

5) Use ZPF to track the boundaries and complete the phase

diagram construction.

Similar scheme can be constructed in higher dimensions. Notice that the algorithm is capable of predicting multiple minima using the second order derivative information, which makes it applicable even in the difficult multiphase miscibility gap situations.

The overall performance of this scheme is mostly influenced by the two major

factors: the accuracy in the detection of the critical points and the effectiveness of the chosen sampling scheme. In the later sections, we discuss the theoretical and practical advantages of the new method, in comparison with other existing techniques.

4.3.3 Computational complexity estimate for the binary case

In this section we will compare complexity of our algorithm to other existing schemes from the point of view of required resources and computational workload. First, let h be the smallest mesh size required in order to identify points with the lowest energy on any one of stability regions with a given accuracy ϵ. By measuring the total number of grid points needed to reach this mesh size, we claim that, due to the adaptivity, our proposed scheme requires significantly less subdivisions than other comparable methods.

Indeed, suppose the number of levels required for the adaptive scheme to reach this mesh size is denoted as L. Since after a refinement stage, each interval is either subdivided into N subintervals (there are at most two such intervals) or left unchanged, the mesh size at each level is reduced by a factor of $1/N$ and the total number of intervals is increased by at most $2(N-1)$. Hence $h = 1/N^L$ or $L = \log_N 1/h$. It follows also that the total number of intervals needed to reach a mesh size h is $N_T = N + 2(N-1) \cdot L = N + (2(N-1)\ln 1/h)/\ln N$. Note that N is taken to be a constant independent of h (we use a fixed $N = 10$ in the numerical experiments). A comparable full uniform grid scheme (like the one in the Chen et al algorithm) should have approximately N' grid points to yield the same accuracy as the scheme proposed above. In other words, $N_T = O(\ln N')$,

which implies a significant reduction in the number of axis subdivisions comparing to the uniform scheme.

Second, let us assess the amount of work required by each of the algorithms for finding a starting point for the two-phase equilibrium calculation prior to the optimization based on the same mesh size h. We again can claim an advantage of our scheme in comparison with the approach of Chen et al due to a significant reduction in the number of required coplanarity checks. Below is the step by step analysis of the computational complexity that verifies to our claim.

From the point of view of complexity, Algorithm 1 as given in section 3.1.1 can be divided into the following major stages:

i) $[A, ind] = StabilityRegions(0,1,2N,K)$ – *stability region calculation*

To compare energy values for all K phases at each grid point, we need $2N \cdot K$ operations, which is the total complexity for this stage. Notice that this estimate has no dependence on h.

ii) Starting points calculation

AdaptiveSearch procedure is performed on each of the *ind* stability regions where the total number of stability regions *ind* is a constant independent of h. At the first sweep, we detect inflection points by calculating second derivatives at each grid point: a total of $3N$ function evaluations if a 3-point stencil is used in the derivative calculation. At the second sweep, N first derivatives are calculated on at most 2 subintervals, which adds up to a total of $4N$ function evaluations if a 2-point stencil is used in derivative calculation. If the critical point was detected with predefined accuracy ϵ after $L = (\ln 1/h)/\ln N$ adaptive grid refinements, the first derivative calculation had to be repeated $(\ln 1/h)/\ln N$ times. It follows that the overall complexity of this stage is given by $3N \cdot ind + 4N \cdot ind \cdot (\ln 1/h)/\ln N = O(\ln 1/h)$.

iii) Coplanarity checks

At the last stage of the Algorithm 1, two coplanarity checks are performed for all selected pairs. A coplanarity check for p selected points requires a calculation of a total of $0.5p(p-1)(2+12(p-2))$ operations (determinant calculations for each of the $0.5p(p-1)$ pairs). The total complexity of this stage is thus $P = p(p-1)(2+12(p-2))$. Since the total number of selected pairs is specific to the system configuration and does not depend on the refinements, the final stage does not raise the overall algorithm complexity in terms of $1/h$.

It follows from the above estimation that, for fixed parameter N without further refinement, our scheme has a total complexity of $O(\ln 1/h)$. Likewise we can calculate the number of operations required for the Chen et al algorithm of comparable accuracy ($N' = 1/h$ is the total number of subdivisions required). As we have seen in Section 2.3, it can be roughly estimated as

$$K \cdot N' + \frac{1}{2}N'(N'-1)(2+12(N'-2)) = \frac{K}{h} + \frac{1}{2h}\Big(\frac{1}{h}-1\Big)\Big(2+12\Big(\frac{1}{h}-2\Big)\Big) = O\Big(\frac{1}{h^3}\Big)$$

Here $K \cdot N'$ operations are spent on stability checks, while the coplanarity check for each of the pairs takes up the rest of the complexity.

It is obvious that for small h our complexity $O(\ln 1/h)$ is significantly lower than the $O(1/h^3)$ complexity of the Chen et al algorithm.

As an illustration, let us fix $N = 10$, $ind = 2K$ (the maximum number of stability regions in 2d) and $p = 2 \cdot ind$. The graph in Figure 4.5 illustrates the behavior of calculated complexity estimates for the uniform Chen et al type scheme versus the new algorithm for the $K = 2$ (two phase) case.

It is clear that for a mesh size smaller than a critical value ($h \approx 0.05$ for our example), the adaptive scheme proposed above outperforms the uniform grid

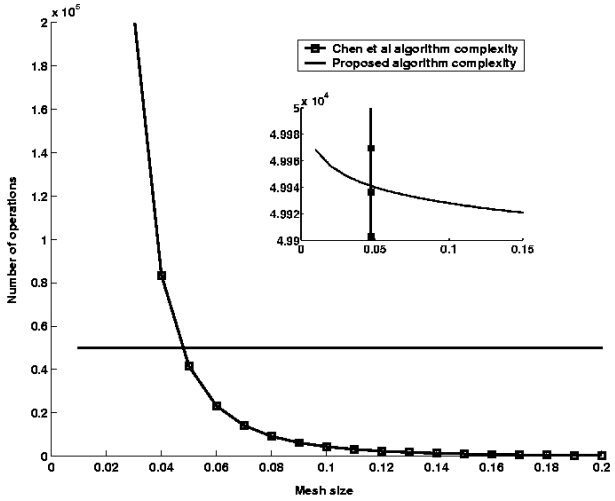


Figure 4.5. Complexity comparison

algorithm, and its advantage becomes even more visible as the mesh size required to detect the lowest energy decreases.

4.3.4 Generalization to higher dimensions and sampling schemes

In light of the results derived in the previous section, the new adaptive technique possesses an advantage over other schemes in that it needs significantly less points and operations to achieve the required accuracy. Still, such a scheme also becomes computationally demanding when the dimension of the system starts to grow. However, there are some ideas that can be entertained in order to reduce the complexity of the method in higher dimensions.

For instance, the Hammersley or Halton quasi-random sequences can help reduce the complexity while allowing for critical point detection with the same accuracy in dimensions up to $s = 8$ (degradation and correlation can occur in higher dimensions). These sequences are low-discrepancy point sets in a sense that the discrepancy (deviation from the uniform distribution) of an N point sequence in

s-dimensional case satisfies $D_N^* = O(N^{-1}(\log N)^{s-1})$ (see [60]). The Halton sequence is superior to that of the Hammersley in that it builds upon the previous sets as the number of points increases. We can hope that the use of these sequences will allow to detect inflection points with the accuracy similar to a uniform approach while reducing the overall computational cost. Figure 4.6 shows an example of detecting concavity regions using both quasirandom and uniform approaches. It is clear that quasirandom sampling helps to identify the miscibility gap earlier than the regular grid. Indeed, by cooling the system from $T = 600$ to $T = 450$, we see an appearance of the miscibility gap much earlier for the quasirandom sequence (at $T = 525$) than for the uniform sampling.

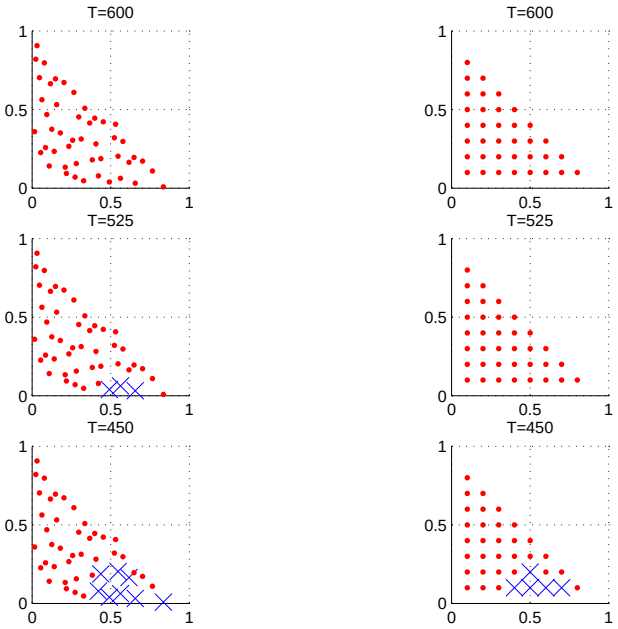


Figure 4.6. Effectiveness of the quasirandom (left) vs. uniform(right) sampling in detecting concavity change. Squares denote the points of negative concavity, while dots are positive concavity regions. 50 sampling points are used for both sampling schemes.

The difficulty of using quasi-random approach with any adaptive refinement strategy is the need for a careful point ordering. One should also think about the possible loss of accuracy in derivative calculations done on such a mesh. The last

obstacle can be avoided by introducing a finer regular grid around each point for finite difference calculations. For sufficiently small grid size h, the error introduced by such types of calculation is at least of the order of $O(h)$ and often $O(h^2)$ for the first and second order derivatives depending on the difference scheme [14]. The overall complexity will not be affected if a fixed number of auxiliary points is used for all grid points. To overcome the ordering difficulty, one can reorder the points independently on each subset after each refinement.

The most attractive property of quasi-random sequences is that they dramatically reduce the error bounds for integration [60]. This gives us reason to believe that the quasi-random construction can potentially work very well in our case, especially if the derivative information is used adaptively in the process of mesh construction.

4.4 Results for binary and ternary systems

All examples given below rely on the following form of the Gibbs energy functional, where the excess Gibbs energy is expressed in the form of Redlich-Kister polynomial:

$$G_m^\Phi = \sum_i x_i^0 G_i^\Phi + RT \sum_i x_i \log x_i + {}^{xs} G_m^\Phi$$
$${}^{xs}G_m^\Phi = \sum_{j>i} x_i x_j \sum_{k=0}^{n} L_{i,j}^\Phi (x_i - x_j)^k$$

Performance estimates have been done with the Matlab 6.5 implementation of the algorithm on a Pentium 4 2.4Ghz machine with 512MB RAM Figures 4.7(b) through 4.9(b) show phase diagrams computed using this implementation of the new method, while Figures 4.7(a) through 4.9(a) are reproduced from [70] and are created using Thermocalc software with the aid of a priori knowledge of the

system.

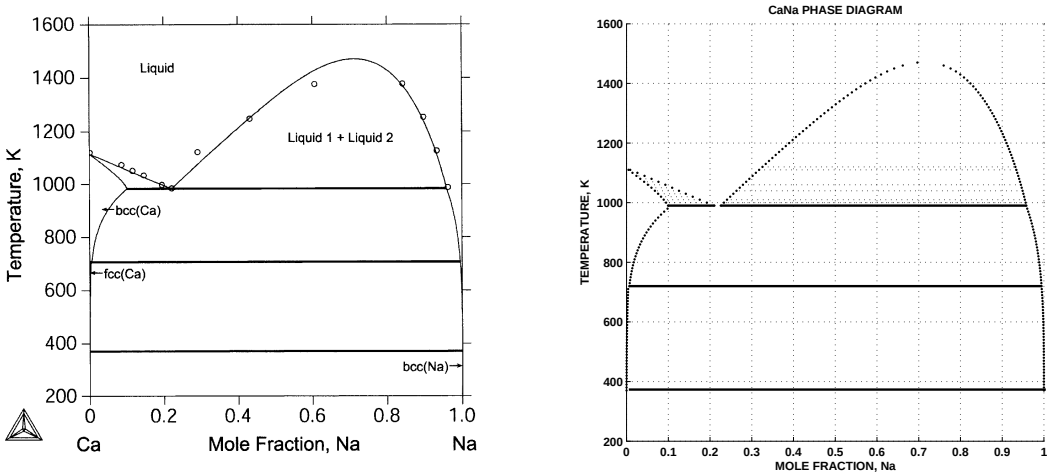


Figure 4.7. (a) Ca-Na diagram produced by Thermocalc; (b) Ca-Na diagram produced by the new method

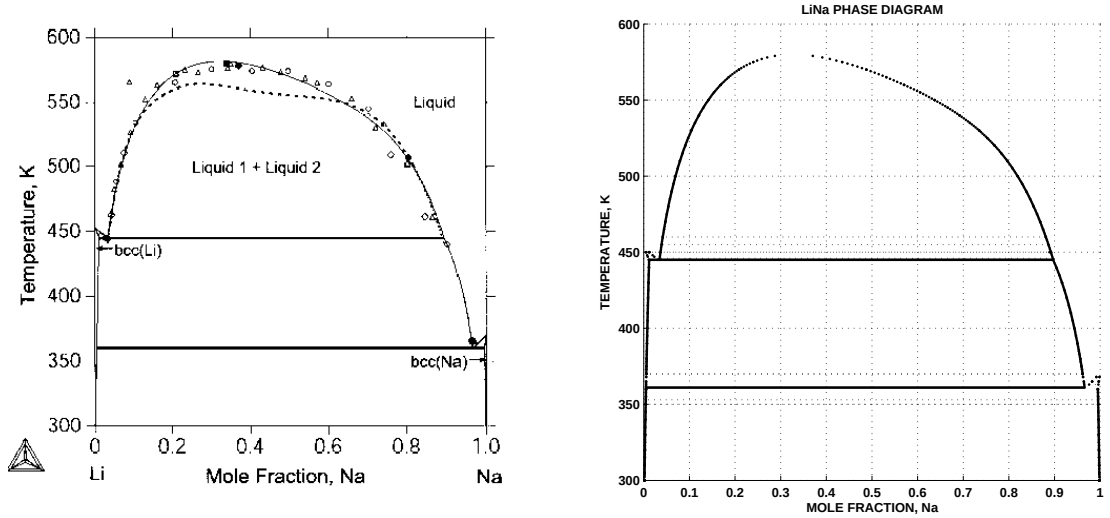


Figure 4.8. (a) Li-Na diagram produced by Thermocalc; (b) Li-Na diagram produced by the new method

4.4.1 Binary examples

First, we consider a Ca-Li-Na system. Figure 4.7(a) represents its binary Li-Na projection for $T = 900K$, where the miscibility gap occurs. As shown in Figure 4.1, the phase diagram calculation done by the Thermocalc software independently

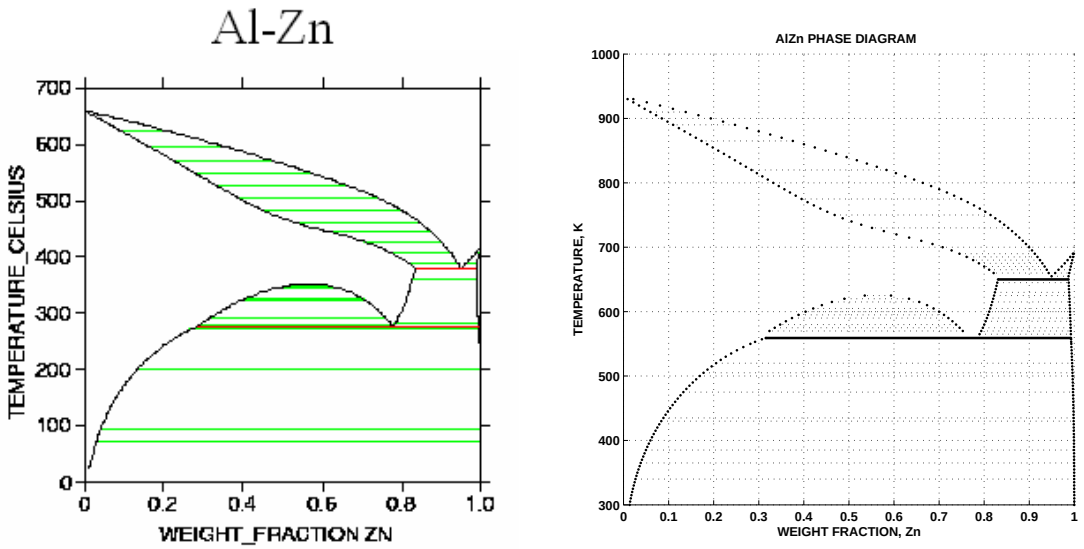


Figure 4.9. (a) Al-Zn diagram produced by Thermocalc; (b) Al-Zn diagram produced by the new method

produces unacceptable results when the miscibility gap is not specified manually. The method described above detects the existing liquid and bcc miscibility gaps and correctly predicts the corresponding phase diagram with absolutely no additional input from the user.

In the Matlab 6.5 implementation, the complete Li-Na phase diagram construction with the temperature step size $dt = 1^oK$ took about 232 sec. Another example is the Ca-Na projection calculated at $T = 900K$. Here all three phases (liquid, bcc, fcc) form stable equilibria at different temperatures and a liquid miscibility gap occurs at temperatures higher than 1000^oK. It took 261 sec to produce the complete diagram which is given in Figure 4.8(a).

4.4.2 Ternary examples

Figure 4.10 shows the Gibbs energy of the ternary Ca-Li-Na system at $T = 900K$. The straight line indicates the common tangent found by the new algorithm for the miscibility gap, which remained undetected during unassisted Thermocalc run

producing incorrect diagram in Figure 4.1(b). The outline of the procedure used to compute ternary diagrams is as follows. The preprocessing module was designed that is capable of handling arbitrary ternary systems from given database specifications and can be integrated directly into the Thermocalc. Steps 1 and 2 of the Algorithm 4.3.2 discussed in section 4.3 are performed in this preprocessing module prior to the optimization. Results of the preprocessing calculation are then automatically recorded in the corresponding macro file that can be further fed into Thermocalc to produce the complete diagram as the one shown in Figure 4.1(a). The timing overhead of the preprocessing routine did not exceed 5 sec for any of the above fixed temperature calculations.

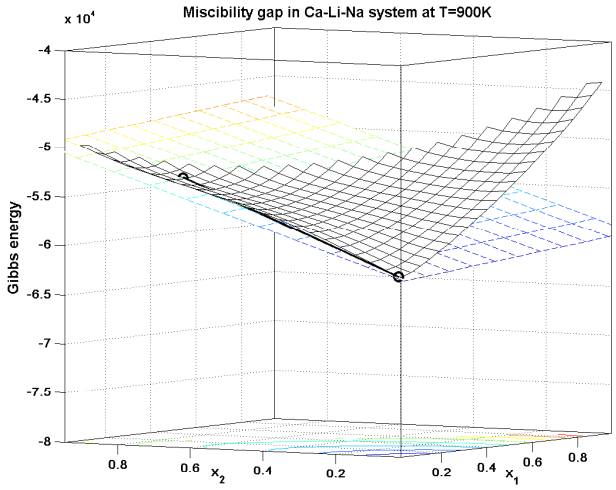


Figure 4.10. Gibbs energy of the Ca-Li-Na system at $T = 900K$.

4.5 Conclusion

In this chapter, we propose a new scheme to optimize the phase diagram construction algorithm adopted in Thermocalc. The new algorithm possesses advantages over existing methods in terms of the convergence speed, the computational complexity and the robustness. It can be used to automate the calculation of phase

equilibria in complicated systems. Numerical results for binary and ternary systems show good agreement of automatic calculations with prior results.

As discussed earlier for the higher space dimensions, the new approach carries an increased computational load, so a tradeoff must be made between the accuracy of the solution and the complexity of the scheme. Possible higher dimensional solutions including better sampling techniques discussed above are the main focus of our current research and will be discussed in future publications.

Chapter 5

Summary and discussion

In this thesis we have successfully developed, analyzed and implemented efficient novel methods for solving some nonlinear optimization problems. Numerical techniques that have been elaborated within the scope of this work have a broad range of applications, including problems not related directly to optimization. A number of important theoretical questions that provide further insight into the characteristics of the applied problems having significant practical value have been answered.

Quantization: results and open questions

In Chapter 2, we have provided some theoretical analysis of the quantization problem and demonstrated the need for a numerical algorithm with superior convergence properties. Two such algorithms have been proposed in Chapter 3 and their advantage over existing algorithms have been shown both theoretically and numerically. One of them explores the coupling of the Lloyd scheme with Newton-like methods. We have numerical and analytical results justifying super-linear convergence of the algorithm within the convergence region, which can be reached after some initial Lloyd iterations.

Another algorithm represents a multilevel scheme in a nonlinear energy-based

optimization setting. Due to the nonlinear nature of the quantization problem it cannot be analyzed using standard linear multigrid approach. Some recent attempts by other groups to construct a multigrid method for quantization problems via conventional full approximation scheme methods have resulted only in limited success for some 1-d problems [50]. We avoided the difficulties associated with the traditional approach essentially by relying on the energy minimization. Since the energy functional is generally non-convex, a dynamic nonlinear preconditioner was proposed to relate our problem to a problem of convex optimization. In the case of one-dimensional problems, we have shown that the nonlinear multilevel algorithms enjoy uniform convergence properties independent of the problem size k, thus a significant speedup comparing to the Lloyd iteration is achieved.

Our theoretical framework can be potentially extended to higher space dimension as we have established proper relationship between CVTs, optimization problems and dynamic nonlinear preconditioning, and can be used for a wider class of nonlinear problems. Numerically, a multilevel routine in the higher space dimension has been implemented with success, but the rigorous proof of convergence is still not available. The analysis of multilevel quantization schemes and the proof of global convergence of the Lloyd iteration in higher dimensions are among the subjects of our current research.There are also a number of other questions that are left to be answered:

- One of them is the geometric convergence rate of the Lloyd iteration for smooth densities. Although it is confirmed by all numerical experiments we have done to this point, this fact still remains to be a conjecture.

- It might be beneficial to explore possible coupling of our algorithm with some existing global optimization techniques to see whether it can serve as

an accelerator in the neighborhood of the global solution. We're currently considering an implementation of this idea based on the multilevel trust region method and other possible extensions.

- We are also constantly looking for new applications that might benefit from the CVT approach. In fact, our short term plans include the application of the techniques developed in this work to a couple of practical problems, such as clustering, grain boundary analysis and different data mining applications. Along these lines, we're currently working toward extending the results of this thesis to the discrete setting, that will make it applicable to a wider array of tessellation contexts. One of the possible discrete analogues of the Lloyd method that can be used to cluster a discrete set of data can be outlined as follows. Given a discrete, finite-dimensional set of points $W = \{y_l\}_{l=1}^{m}$ belonging to $\mathbb{R}^N$, and an initial set of cluster centers $\{z_i\}_{i=1}^{k}$, then for each $y \in W$,

  1. find the z_i that is closest to y; denote the index of that z_i by i^*;
  2. assign y to the cluster corresponding to z_i^* ;
  3. recompute the cluster center z_i^* to be the mean of the points belonging to the corresponding cluster.

  If the centers are confined to W by the application requirements, step 3. can be modified so that the new center is taken to be the point in i-th cluster closest to the corresponding mean value. This version obviously amounts to a combinatorial search and has a finite convergence time, but it suffers from the same slow convergence issues that were established for the continuous version earlier in Chapter 2. In fact, for quadratic problems in $\mathbb{R}^N$ its complexity

rises to $O(m^{kN+1})$, as shown in [15]. It is thus interesting to see whether some speedup can be achieved via the same type of acceleration techniques we used in this work.

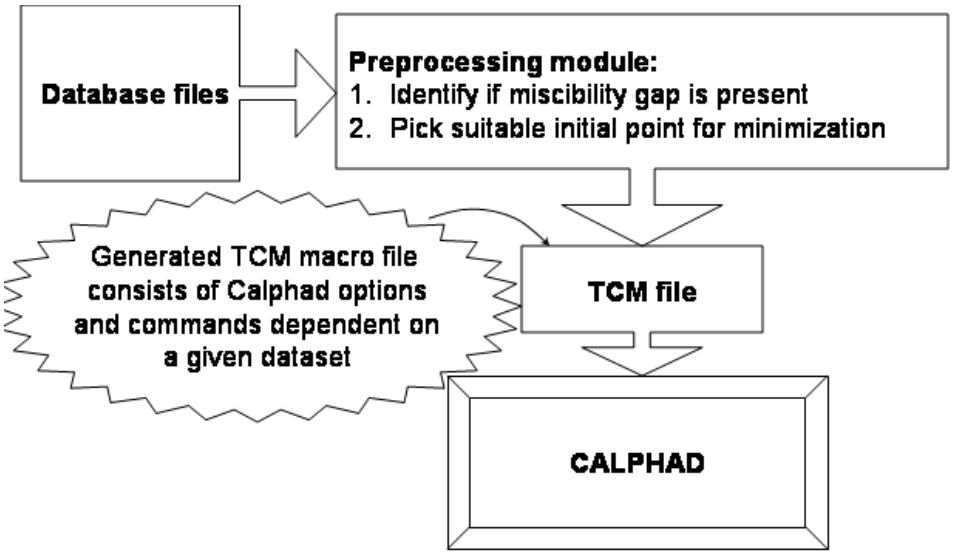
Phase diagrams: results and open questions

In Chapter 4 we introduced a novel algorithm for automating phase diagram construction for materials science applications. The nonlinear constrained optimization problem has been put into geometric context, which allowed to develop a user-independent method of choosing the starting values for the minimization. Adaptive techniques were used to significantly reduce the problem complexity. Analytical and numerical results for two- and higher dimensional calculations have been provided and the advantage of the new algorithm over existing software packages has been confirmed. Many important theoretical and computational questions have been answered within the scope of this work, however, there are still several issues to be discussed. Here's a short summary of the directions of our current work:

- Generalization of the algorithm to higher dimensions without loss of efficiency requires the use of more advanced sampling schemes. Current efforts are devoted to incorporating advanced multidimensional sampling schemes to make phase diagram calculations more computationally effective. Although it has been shown that this approach has a big potential for the problems in moderately high dimensions, it remains to justify its advantages in the case when the number of dimensions is extremely high, where possible deterioration of quasirandom sampling properties may occur.

- Another direction of the ongoing research is the complete integration of the new algorithm with the Thermocalc package. We design a preprocessing

module that would be able to analyze the input data such as Gibbs energy, number of phases and other simple system characteristics and automatically produce a Tharmocalc-compatible macro files with the initial optimization parameters. A user would then be able to use this file to produce the required phase diagram without any additional input. Here's the schematic flow chart visualizing this approach:



After such an integration is complete, we expect to move to the next outstanding challenge. Namely, we want to combine all the developed techniques and design a standalone software package for automatic multiphase and multicomponent phase diagram generation. Such a package will require the development of the user friendly interface as well as good optimization routines and will probably be a cumulative effort of a team of researchers. Once such a package becomes available, it will allow to considerably speed up construction and analysis of phase diagrams and will greatly improve the productivity of research in materials science community.

Overall, this work has demonstrated the efficiency of multigrid and adaptive

techniques applied to some important nonlinear optimization problems and showed the possibility of their rigorous analysis within the chosen framework. Current study raised many interesting and important new questions that will be addressed in the author's future work.

Bibliography

[1] J. O. Andersson, T. Helander, L. H. Hoglund, P. F. Shi and B. Sundman, THERMO-CALC & DICTRA, computational tools for materials science, *CALPHAD*, **Vol.26**, 2002, pp. 273-312.

[2] F. Aurenhammer, Voronoi diagrams. A survey of a fundamental geometric data structure, *ACM Computing Surveys*, **23**, 1990, pp. 345-405.

[3] A. Brandt, General highly accurate algebraic coarsening, *Electronic Transactions on Numerical Analysis*, **10**, 2000, pp. 1-20.

[4] M. Brezina, A. Cleary, R. Falgout, H. Henson, J. Jones, T. Manteuffel, S. McCormick and J. Ruge, Algebraic multigrid based on element interpolation, *SIAM Journal on Scientific Computing*, **22**, 2000, pp. 1570-1592.

[5] J. Burkardt, M Gunzburger and H.-C. Lee, Centroidal Voronoi Tessellation-Based Reduced-Order Modeling of Complex Systems, to appear.

[6] M. Cappellari and Y. Copin, Adaptive spatial binning of integral-field spectroscopic data using Voronoi tessellations, *Monthly Notices Royal Astronomical Soc.*, **342**, 2003, pp.345-354.

[7] Q. Chang and Z. Huang, Efficient algebraic multigrid algorithms and their convergence, *SIAM Journal on Scientific Computing*, **24**, 2002, pp. 597-618.

[8] S.-L. Chen, K.-C. Chou and Y.A. Chang, *CALPHAD*, **17**, 1993, pp. 237-250.

[9] S.-L. Chen, K.-C. Chou and Y.A. Chang, CALPHAD, 17 (1993) 287-302.

[10] S.-L. Chen, S. Daniel, F. Zhang, Y. A. Chang, X.-Y. Yan, F.-Y. Xie, R. Schmid-Fetzer, W. A. Oates, The Pandat Software Package and its Applications, *CALPHAD*, **26**, 2002, pp.175-188

[11] D. Cohen-Steiner, P. Alliez, M. Desbrun, Variational shape approximation, *ACM Transactions on Graphics*, **23**, 2004, pp.905-914.

[12] J.A.D. Connoly, D. M. Kerrick, An algorithm and computer program for calculating composition diagrams, *CALPHAD*, **11**, 1987, pp. 1-54

[13] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, Coverage control for mobile sensing networks *IEEE Tran. Robotics and Automation*, **20**, 2004, pp.243-255.

[14] J.E. Dennis, R. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, 1983

[15] Q. Du, V. Faber and M. Gunzburger, Centroidal Voronoi tessellations: applications and algorithms, *SIAM Review*, **41**, 1999, pp. 637–676.

[16] Q. Du and M. Gunzburger, Grid Generation and Optimization Based on Centroidal Voronoi Tessellations, *Appl. Math. Comp.*, **133**, 2002, pp. 591–607.

[17] Q. Du and M. Gunzburger, Centroidal Voronoi Tessellation Based Proper Orthogonal Decomposition Analysis, *International series of Numerical Mathematics*, **143**, pp.137-150, Birkhauser, 2002

[18] Q. Du, M. Gunzburger, and L. Ju, Constrained centroidal Voronoi tessellations on general surfaces, *SIAM J. Sci. Comput.*, **24**, 2003, pp. 1488–1506.

[19] Q. Du, M. Gunzburger, and L. Ju, Meshfree, probabilistic determination of point sets and support regions for meshless computing, *Comput. Meths. Appl. Mech. Engrg.*, **191**, 2002, pp. 1349–1366.

[20] Q. Du, Max Gunzburger, and L. Ju, Voronoi-based finite volume methods, optimal Voronoi meshes, and PDEs on the sphere, *Comput. Meth. Appl. Mech. Engrg.*, **192**, 2003, pp. 3933–3957.

[21] Q. Du, M. Gunzburger, L. Ju and V. Faber, Finite volume methods on a sphere based on the constrained centroidal Voronoi tessellations, to appear in *Comput. Meth. Appl. Mech. Engrg.*

[22] Q. Du, M. Emelianenko and L. Ju, Convergence Properties of the Lloyd Algorithm for Computing the Centroidal Voronoi Tessellations, submitted to *SIAM J. Num. An.*, 2004.

[23] Q. Du and M. Emelianenko, Uniform Convergence of an Energy-based Multilevel Quantization Scheme, preprint, 2004.

[24] Q. Du, M. Emelianenko and L. Zikatanov, An Energy-based Multigrid Quantization Scheme in Multidimension, in preparation, 2005.

[25] Q. Du and D. Wang, Tetrahedral mesh generation and optimization based on centroidal Voronoi tessellations, *Int. J. Numer. Meth. Eng.*, **56**, No.9, pp.1355-1373, 2002

[26] Q. Du and D. Wang, Anisotropic centroidal Voronoi tessellations and their applications, *SIAM J. Sci. Comput.*, **26**, 2004, pp.737-761.

[27] Q. Du and X. Wang, Centroidal Voronoi tessellation based algorithms for vector fields visualization and segmentation, in *Proceedings of the IEEE Visualization 2004*, Austin, TX, Oct. 2004, IEEE.

[28] Q. Du and T. Wong, Numerical studies of the MacQueen's algorithm for computing the centroidal Voronoi tessellations, *Comp. Math. Appl*, 2001.

[29] R. Dwyer, Higher-dimensional Voronoi diagrams in linear expected time, *Discrete and Computational Geometry*, **6**, 1991, pp.343-367.

[30] M. Emelianenko, Z.-K. Liu and Q. Du, A New Algorithm for the Automation of the Phase Diagram Calculation, to appear in *Computational Materials Science*, 2005.

[31] P. Fleischer, Sufficient conditions for achieving minimum distortion in a quantizer, *IEEE Int. Convention Record*, **I**, 1964, pp.104-111.

[32] S. Fortune, A sweepline algorithm for Voronoi diagrams, *Algorithmica*, **2**, 1987, pp. 153-174.

[33] S. Fortune, *Voronoi diagrams and Delaunay triangulations*, in Computing in Euclidean geometry, World Sci. Publishing, River Edge, NJ, 1992, pp. 193–233,

[34] A. Gersho, Asymptotically optimal block quantization, *IEEE Trans. Inform. Theory*, **25**, 1979, pp. 373–380.

[35] A. Gersho and R. Gray; *Vector Quantization and Signal Compression*, Kluwer, Boston, 1992.

[36] G. Golub, C. Van Loan, *Matrix Computations*, The John Hopkins University Press, 1989

[37] R. Gray, J. Kieffer, and Y. Linde, Locally optimal block quantizer design, *Inform. Control*, **45**, 1980, pp. 178–198.

[38] R. Gray and D. Neuhoff, Quantization, *IEEE Trans. Inform. Theory*, **44**, 1998, pp. 2325–2383.

[39] P. Gruber, Optimum quantization and its applications, *Advances in Mathematics* **186**, 2004, pp.456–497.

[40] J. Hartigan and M. Wong, A k-means clustering algorithm, *Appl. Stat.*, **28**, 1979, pp. 100–108.

[41] P. Heckert, Color image quantization frame buffer display, *ACM Trans. Comp. Graph.*, **16**, 1982, pp. 297–304.

[42] S. Hiller, H. Hellwig, O. Deussen, Beyond stippling - Methods for distributing objects on the plane, *Computer Graphics Forum*, **22**, 2003, p.515–522.

[43] M. Hillert, A discussion of methods of calculating phase diagrams, *Bulletin of Alloy Phase Diagrams*, **2**, pp. 265–268.

[44] Bo Jansson, A General Method for Calculating Phase Equilibria under Different Types of Conditions, *TRITA-MAC-0233*, 1984.

[45] L. Ju, Q. Du, M. Gunzburger; Probablistic methods for centroidal Voronoi tessellations and their parallel implementations, *Parallel Computing*, **28**, 2002, pp.1477–1500

[46] L. Ju, M. Gunzburger and Q. Du, Meshfree, Probabilistic Determination of Points, Support Spheres, and Connectivities for Meshless Computing, *Computer Methods in Applied Mechanics and Engineering*, **191**, 2002, pp.1349–1366

[47] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman and A. Wu, An efficient k-means clustering algorithm: Analysis and implementation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **24**, 2002, p.881–892.

[48] J. Kieffer; Uniqueness of locally optimal quantizer for log-concave density and convex error function, *IEEE Trans. Infor. Theory*, **29**, 1983, pp. 42–47.

[49] R. Klein; *Concrete and Abstract Voronoi Diagrams,* Lecture Notes in Computer Science 400, Springer, Berlin, 1989.

[50] Y. Koren, I. Yavneh and A. Spira, A Multigrid Approach to the 1-D Quantization Problem, May 2003

[51] Y. Koren, I. Yavneh, Adaptive Multiscale Redistribution for Vector Quantization, Jan 2004

[52] Y. Linde, A. Buzo, and R. Gray; An algorithms for vector quantizer design, *IEEE Trans. Comm.*, **28**, 1980, pp. 84–95.

[53] S. Lloyd, Least square quantization in PCM, *IEEE Trans. Infor. Theory*, **28**, 1982, pp. 129–137.

[54] F. Lu and G. Wise; A further investigation of the Lloyd-Max algorithm for quantizer design, *Twenty-First Annual Allerton Conference on Communication, Control, and Computing*, University of Illinois, 1983, pp. 481–490.

[55] D.G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley, 1984

[56] H. L. Lukas, J. Weiss, E-Th. Henig, Strategies for the calculation of phase diagrams, *CALPHAD*, **6**, 1982, pp. 229–251

[57] J. MacQueen; Some methods for classification and analysis of multivariate observations, *Proc. Fifth Berkeley Symposium on Mathematical Statistics and Probability*, **Vol. I**, Ed. by L. Le Cam and J. Neyman, University of California, 1967, pp. 281–297.

[58] A. Mendes and I. Themido, Multi-outlet retail site location assessment. *Inter. Trans. in Operational Research*, **11**, pp1-18, 2004.

[59] U. Moller, M. Galicki, E. Baresova, and H. Witte, An efficient vector quantizer providing globally optimal-solutions, *IEEE Trans. Signal Proc.*, **46** , 1998, pp. 2515–2529.

[60] Harald Niederreiter, Random Number Generation and Quasi-Monte Carlo Methods, *CBMS-NSF regional conference series in applied mathematics*, 1992

[61] J. Nocedal, S. Wright, *Numerical Optimization*, Springer-Verlag, 1999

[62] A. Okabe, B. Boots, and K. Sugihara; *Spatial Tessellations; Concepts and Applications of Voronoi Diagrams,* Wiley, Chichester, 1992.

[63] J. Ruge and K. Stuben, Algebraic multigrid, in *Mutigrid methods. Fronties in applied mathematics. SIAM*; Philadelphia, 1987: 73-130.

[64] J. Sabin and R. Gray, Global Convergence and Empirical Consistency of the Generalized Lloyd Algorithm, *IEEE Trans. on Inform. Theory*, **Vol. IT-32**, no.2, March 1986

[65] S. A. Safran, *Statistical Thermodynamics of Surfaces, Interfaces and Membranes*, Addison-Wesley, 1994

[66] X.-C. Tai and J. Xu, Global and Uniform Convergence of Subspace Correction Methods for Some Convex Optimization Problems, *Math. Comp.*, 1998

[67] A. Trushkin; On the design of an optimal quantizer, *IEEE Trans. Infor. Theory*, **39, 1993, pp. 1180–1194.**

[68] S. Valette and J. Chassery, Approximated Centroidal Voronoi Diagrams for Uniform Polygonal Mesh Coarsening, *Computer Graphics Forum*, 23, 2004, pp.381–390.

[69] C. Wager, B. Coulland N. Lange, Modelling spatial intensity for replicated inhomogeneous point patterns in brain imaging, *J. Royal Stat. Soc. B*, 66, 2004, pp.429–446.

[70] S. J. Zhang, D. W. Shin and Z. K. Liu, Thermodynamic modeling of the Ca-Li-Na system, *CALPHAD*, 27, 2003, pp. 235–241.

Vita

Maria Emelianenko

Maria Emelianenko was born on March 13, 1979 in Dubna, Russia. She received a B.S. and M.S. in Computer Science and Applied Mathematics from Moscow State University (1999 and 2001, respectively) and an M.A. in Mathematics from Pennsylvania State University in 2002. Maria's research is focused on the analysis and development of efficient numerical algorithms. She is a part of the MatCASE project at PSU which is funded by a major NSF-ITR grant to develop computational tools for multicomponent materials design. Recently, she has worked on the design of fast new algorithms for quantization and clustering with the use of concepts like Centroidal Voronoi tessellations and optimization methods for the determination of phase diagrams for multicomponent materials. Her earlier research activities include the analysis of multidimensional birth-death processes and development of efficient pricing schemes in next-generation telecommunication networks, solution of ill-conditioned systems of linear equations, and mathematical models in biology.