USING SYMMETRIES TO SOLVE ASYMMETRIC PROBLEMS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Simon Gravel

January 2009

USING SYMMETRIES TO SOLVE ASYMMETRIC PROBLEMS

Simon Gravel, Ph.D.

Cornell University 2009

This dissertation describes two projects in which the treatment of a difficult and asymmetric problem is simplified by using symmetries of basic building blocks of the problem.

In the first part of this dissertation we address the problem of determining the effective interaction between ions in metallic systems. Our work applies more generally to systems where effective interactions between massive particles can be calculated to take into account, in an average way, the effect of lighter particles present in the system. We find an equality relating the (asymmetric) effective interaction of two massive particles and the (symmetric) effect of a single massive particle on the density of the light particles. We show how this relation can be used to improve upon the precision of effective potentials calculated by perturbative approaches for an assortment of systems including hydrogen in metallic environment.

In the second part of this dissertation we discuss constraint satisfaction problems. We provide multiple examples of constraint satisfaction problems occurring in various scientific areas. In many cases the individual constraints are highly symmetric, while the resulting constraint satisfaction problem is not; there is no symmetry common to all the constraints. We describe divide and concur, a new approach to solve constraint problems, which is based on *projections* to the individual constraint sets. The definition of efficient projection operators are facilitated by symmetries of the constraint sets. We show that this method is competitive with the state-of-the-art on standard benchmark problems, and in the process establish a number of records in finite disk packing prob-

lems. Many applications of the divide and concur approach are still to be explored, and we provide the reader with tools to do so, including promising applications and a list of constraint sets together with efficient projection operators.

BIOGRAPHICAL SKETCH

Simon Gravel was born in Montréal, Québec, Canada, in 1979. In 1982, he started wearing glasses. Big glasses. His childhood was spent in eastern Montréal, where he collected things and played hockey. Simon enjoyed pleasing adults, and cried a lot, which wasn't a very popular combination with the other kids. In 1993, Simon was told that he didn't need glasses any more, and started playing chess, which provided intellectual stimulation and an open-minded community. He entered Collège de Maisonneuve and played lots of badminton and chess. When he started studying physics and mathematics at the Université de Montréal he found both much more fun, useful, and demanding than chess, which he soon gave up. He then spent a magnificent year cycling the alps and studying some more physics and mathematics at the Université Joseph-Fourier in Grenoble, France. He came back to Montréal for a MSc in Physics with Pavel Winternitz, and moved to Ithaca to attend Cornell University, where he worked under the supervision of N.W. Ashcroft and Veit Elser. After having obtained a second MSc in physics, he obtained his PhD in 2009. He will be moving to Germany for a postdoctoral position in the physics department at Cologne University.

To Shira, my Vampire-Samurai.

ACKNOWLEDGEMENTS

I would first like to thank my successive advisors at Cornell, Neil Ashcroft and Veit Elser. I really enjoyed my conversations with Neil Ashcroft, particularly his way of listening and calmly exploring the possible implications of each new finding. He also taught me a new level of meticulousness in scientific writing for which I am grateful. Of Veit Elser, I particularly appreciated the enthusiasm and dedication. Doing research with Veit was a lot of fun and always interesting. Thanks also to Itai Cohen and Erich Mueller for serving on my committee and providing interesting and challenging problems for my A and B exams. I would also like to thank Pavel Winternitz, who gave me excellent projects and guidance during my Master's program in Montréal. Thanks finally to Christiane Rousseau and Mokhtar Rouabhi who supervised my first undergraduate research efforts.

I was blessed with great labmates during my stay at Cornell. Bruno Rousseau, Ivan Rankenburg, Duane Loh, Pierre Thibault, and Yoav Kallus all made my research and my life at Cornell more interesting and more fun. All of them not only contributed to the development of the ideas presented in this thesis, but were also good friends. I would also like to acknowledge useful discussions with Stanimir Bonev, Ji Feng, Lukas Kroc, Jörn Kupferschmidt, James Porter, David Roundy, Ashish Sabharwal, Sarah Shandera, Arend van der Zande, and Brian White.

I will not list all colleagues whose friendship and help I appreciated over the years, as there are too many. Is it appropriate to have a link to a social networking website in a dissertation? Lots of nice times were had with Attila, Brian, Gordon, James, JF, Johannes, Kaden, Mark, Nate, Sophie, Sourish, Steve, Vlad, and the others.

Thanks also to my other friends in Ithaca; Amélie, Angela, Catherine, Catherine, Eric, Ioana, Katharina, Stephanie, and Sylvie.

Les cinq heures de route qui me séparaient de Montréal et de ma famille ont toujours sembé interminables. Merci à ma famille pour m'avoir fait ce que je suis, et pour être ce que vous êtes. Vous êtes la raison pour laquelle, cinq ans plus tard, j'ai toujours aussi hâte de revenir à Montréal.

Merci aussi à mes amis de Montréal; Antoine, Marc-André, JF, JF, Marilène, LP, Véro, Zineb, et à mes amis de Grenoble Uli et Élin.

My housemates and colleagues Arend van der Zande, Jörn Kupferschmidt, Sarah Shandera and house allies Sinja, Kasia, Punita and Louis were great friends and of great support. Thanks Arend for your generosity and tolerance (and your car!), Sarah for your enthusiasm and open-mindedness (and your car!) and Jörn for your integrity and attention (and your furniture?). Thanks to all for general awesomeness.

Shira Adriance was a great friend before becoming a great girlfriend and a great housemate. She managed to stay great on the three fronts for the last four years. Thanks for your continued support, your sincerity, your care for the world and those around you, and for teaching me how pragmatism often works better than ill-conceived idealism.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF PSEUDOCODES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The work presented in this dissertation results from two distinct projects carried at the Laboratory of Atomic and Solid State Physics at Cornell between 2004 and 2008. The first project involved characterizing ionic interactions in materials under high pressure, while the second involved developing numerical methods to solve constraint problems. The combination of the description of these two projects in this single dissertation is due to historical coincidence rather than scientific necessity. However, the two projects have common features. They can be formulated by starting from a spatially uniform system, and successively adding to the system components which, individually, are simple and symmetric, but which collectively result in a difficult and asymmetrical problem. Both projects involved the development of methods that took advantage of this structure, which resulted in the occurrence of similar patterns in both project (compare, for example, Figures 2.10 and 4.10 below).

In the first part of this thesis we address the problem of determining the effective pairwise interactions between massive particles immersed in a system of much lighter particles. The objective is to describe the behavior of hydrogen ions in a metallic environment. In this problem the initial uniform system is the electron gas, and the symmetrical and simple components are the interactions between individual ions and the electron gas. The determination of the effective pair potential between the ions requires evaluating the energy of an electron gas in the presence of the asymmetrical potential generated by a *pair* of ions. The symmetries of the individual potentials can be used to obtain approximate values for the energy of the asymmetrical problem. We introduce in Chapter 2 a method that uses the symmetries of the individual ions to improve the precision of nonlinear perturbation theory, which has been a popular tool to estimate

effective interactions in metals since a series of papers by Brovman, Kagan and others (see [1] and references therein). More specifically, we obtain an identity that includes higher order perturbation terms in a lower order expansion by relating the effective interactions between ions to the spherically symmetric density induced by individual ions. We apply this identity to delta-function potentials in a noninteracting electron gas, the entropic attraction between repulsive particles, and more importantly to the interaction between hydrogen ions in a high-pressure metallic environment. In all cases we find that our identity allows improved precision for the pair potentials, when compared with standard quadratic response results.

In Chapter 3 and subsequent chapters, we concentrate on solving constraint problems. The object of constraint problems, which are frequently encountered in physics, biology, computer science, and mathematics, is the simultaneous satisfaction of multiple constraints. We will usually consider problems where variables are defined on Euclidean spaces, and where each constraint is simple and symmetric. The difficulty in such problems results from the fact that different constraints have different symmetries which are incompatible, so that the resulting problem has few symmetries.

Constraint satisfaction problems of particular interest occur in experimental situations where a sample property can only be measured indirectly, via a number of incomplete observations. Each observation can be interpreted as a constraint on what the configuration of the sample can be. An example of such a situation arises in diffractive imaging (see, e.g., [2, 3]), where only the magnitudes of the diffraction pattern of a sample are measured. The resulting constraint, which has a local gauge symmetry in Fourier space, has to be combined with additional information to obtain the spatial density distribution of the sample. Another interesting example is protein structure determination through nuclear magnetic resonance [4]. In this case, NMR provides constraints on al-

lowed relative position of different atoms in the protein. The information from these translationally and rotationally invariant constraints, combined with knowledge of the protein linear sequence, can be enough to specify a single protein configuration. However, the task of finding this particular configuration in the vast configuration space of the protein is challenging. In general, the problem of assembling independent observations into information about the desired property of the sample requires efficient constraint satisfaction algorithms.

We describe divide and concur, a method we developed to solve efficiently a wide variety of constraint problems. Chapter 3 provides an overview of constraint satisfaction problems and methods, while Chapter 4 describes the philosophy and formalism of the divide and concur approach, with particular attention to the definition of *projections* to individual constraint sets, the basic building blocks of this approach. The efficiency of the projection operators usually depends on the existence of symmetries of the individual constraint sets. Projections can be seen as a tool to probe the geometrical properties of the different constraint sets, while the divide and concur scheme is a way to assemble this information to retrieve a solution to the overall problem.

Chapter 4 also includes empirical observations that have not found place in previously published material but that we believed might benefit a reader interested in developing his own applications of divide and concur. Finally, in Chapter 5, we discuss applications of the method to various problems, including random boolean satisfiability (kSAT), sphere packing, error-correcting codes, and vertex covers.

CHAPTER 2

RESPONSE THEORY AND EFFECTIVE PAIR POTENTIALS

In this chapter we review and reproduce some of the material presented in Reference [5]: S. Gravel and N. Ashcroft, Phys. Rev. B **76**, 144103 (2007) (Copyright 2007 by the American Physical Society). The main topic in this chapter is the applications of response theory to many component systems where massive (M) and light (L) particles share a common volume. A typical example is provided by ions and electrons in a metal. In many cases one is interested in the behavior of the light particles only insofar as they affect the behavior of the massive particles. This might be the case, for example, if one is interested in the atomic structure of a material. Because of the mass difference between the particles, the time scales associated with the light particles is typically shorter than that of the massive particles. If the difference is significant enough, the motion of the massive particles might be slow enough that the change is quasi-adiabatic from the perspective of the light particles, that is, that the light particles are always near equilibrium. Assuming exact equilibrium of the light particles constitutes the adiabatic or Born-Oppenheimer approximation. It can result in very significant simplifications in the treatment of multicomponent systems.

If one can treat the effect of the light particles on the massive particles in an average way, for example, it might not be necessary to keep track of the positions of the light particles. In such a case, one is left with effective interactions between the massive particles, and the degrees of freedom associated with the light particles have been essentially discarded.

Such effective interactions have had their share of success in describing simple metals [6, 7], metals with magnetic impurities [i.e. the Ruderman-Kittel-Kasuya-Yosida (RKKY) interaction [8]] or, in the classical domain, colloidal suspensions [e.g. the

Derjaguin-Landau-Verwey-Overbeek (DLVO) depletion interaction between charged colloidal particles [9, 10] or the depletion interaction [11, 12]].

Effective interactions between the massive particles can be obtained particularly effectively when the interaction between the massive and light particles is weak enough that the effect of the massive particles can be treated as a small perturbation to an otherwise well-understood system of light particles. Linear response in particular has been used extensively to obtain effective pair potentials; its simplicity makes it a very valuable qualitative tool, and in many contexts it also provides sufficient quantitative accuracy. Higher order contributions have been used less often, partly because calculational complexity increases rapidly with the order of perturbation considered. With the advent of density functional theory (DFT) and modern computers, response theory has lost some of its quantitative appeal. DFT, by using an iterative self-consistency scheme proposed by Kohn and Sham [13], provides accurate estimates of the ground state energy of many condensed matter systems. However, the iterative self-consistency process can obscure some of the physics which was made more obvious by the perturbative approach, in particular the dependence of the system behavior on parameters such as electronic density. For this reason, response theory is still being used for a qualitative understanding of general trends in metals [14–16]. Another useful feature of the perturbative estimates of cohesive energies in metals is that the resulting expressions can be decomposed naturally into pairwise interactions, triplet interactions, and so forth. Moreover, many-center interactions result from higher order corrections than few-center interactions, and can therefore often be neglected. This information about the interactions between ions allows for a better understanding of the way ions assemble in crystals, but also in nonperiodic structures such as liquids, glasses, and quasicrystals where the application of DFT is more difficult. Indeed Hafner, in his book *From Hamiltonians to Phase Diagrams* [7], argues:

> The topological disorder in melts and glasses creates serious difficulties in formulating an electronic theory of the cohesive and structural properties. In each case progress will hinge on a knowledge of the interatomic forces in the metal or alloy.

For this reason the so-called simple metals, which are amenable to perturbative approaches, are a good starting point to the study of various phases of matter. We should emphasize that even though the interactions between the ions, the basic building blocks in these systems, can in these cases be described rather simply in terms of few-centers potentials, the resulting structures need not be as simple. The emergence of complex structures from simple building blocks is the topic of emergence, self-assembly, and self-organization studies (see, e.g., [17]). More specifically, the complex behavior of 'simple' metals has also received recent attention, particularly at high pressures [18]. It has also been argued that the pair interaction picture of ions in some simple metals might break down at sufficiently high pressure [19].

In a way, the simplest of all simple metals is to be found in dense hydrogen, as it has only one electron per nucleus, and this nucleus is a simple proton. This statement might be a bit surprising, as hydrogen under ambient pressure is quite far from being a metal. However, dynamic compression experiments have achieved metallization of hydrogen at high temperature [20], and there is an expectation that low-temperature metallization using diamond anvil cells might also be achieved in the relatively near future [21]. Recent work suggested that new metallic phases of matter might be induced in such high pressure environments [22]. High pressure phases of hydrogen-rich systems might also exhibit interesting phase diagrams [23], and might be amenable to a perturbative treatment. Perturbation approaches might be ideally suited to address such high pressure physics problems: as a consequence of the exclusion principle, increased pressure usu-

ally leads to increased electronic kinetic energy, which in turn tends to lead to a better convergence of perturbative approaches. Furthermore, whereas temperature changes often affect only marginally the effective interactions between ions in a metal, a change in pressure can drastically affect this interaction. Understanding the evolution of this effective interaction is a task for which perturbation approaches are well suited.

However, hydrogen has always been difficult to treat by perturbation approaches because the interaction of a proton with the electron gas is not weakened by a repulsive core, as is the case for many elements[24]. It is not even obvious that a perturbation approach converges. Response-based pair potentials were nevertheless used to describe with some success the pressure dependence of the vibron in the hydrogen solid [25], at high compression ratios. We found that the pair potentials obtained from quadratic perturbation theory for dense hydrogen exhibit significant discrepancies when compared with ab initio DFT results [26] for Wigner-Seitz radii as low as 1.3. The Wigner-Seitz radius r_s is related to the unperturbed electronic density ρ_0 by $r_s a_0 = [3/(4\pi\rho_0)]^{1/3}$, where $a_0 = \hbar^2/m_e e^2$ is the Bohr radius.

In order to use the perturbation method as a reliable, analytic alternative approach to DFT for hydrogen systems in this density range, higher-order terms are needed. Going beyond quadratic response using conventional methods is a substantially more difficult task. In order to circumvent this difficulty, we derive below a simple identity that allows us to obtain effective pair potentials beyond quadratic response, which amounts to carrying out a partial sum of perturbation terms. The result is not only very intuitive, expressing the pair potential in terms of physically meaningful expressions, but also very general, since it applies in any dimension and for both classical and quantum systems. Moreover, it does not require explicit knowledge of the nonlinear response functions. It should be noted that if nonlinear effects are to be fully taken into account, many-body

interactions should also be included, either directly or through effective-medium approaches [27]. We focus here on obtaining pair potentials that can be used as a starting point for either approach. Generalizations to many-body interactions are discussed in section 2.6.

In Section 2.1, we outline the response theory formalism and derive the simplest version of our result, equation 2.11. In Section 2.2 and Section 2.5, a generalization is derived that applies to homogeneous, interacting systems. The intuitive nature of this result permits us to explain part of the discrepancies observed between the results of Nagao *et al.* [25] and Bonev and Ashcroft [26], and to improve upon quadratic response. In Section 2.3, we apply our results to obtain pair potentials in various systems, including metallic hydrogen. A simple interpretation in terms of path integrals, and generalizations to many-center interactions and magnetic perturbations are discussed in Section 2.6.

2.1 Effective interactions and response theory

We start with a canonical system composed of a mixture of at least two different types of particles, in contact with a heat bath establishing a temperature T. The ground-state properties of the system can be studied from the limit $T \to 0$.

The particles will be divided in two subsets, labeled L (for Light, typically electrons) and M (for Massive, typically ions). It is assumed that all L particles are identical. On the other hand, the set M can contain various types of particles. The time scales associated with the L particles will be assumed to be much shorter than those associated with the M particles, so that an approximate adiabatic separation of time scales can be assumed. In real systems, this separation is not exact. The assumption of exact

separation of time scales, which will be made throughout this article, is referred to as the adiabatic or Born-Oppenheimer approximation. We will not discuss the accuracy of this approximation here and refer the reader to the existing literature (see, e.g., Ziman [28]).

In the adiabatic approximation, the degrees of freedom associated with the light particles L are treated in an average way, leading to a much simpler *effective* Hamiltonian for the remaining M particles.

The Hamiltonian of the initial system has the form

$$H = \sum_i \frac{\mathbf{p}_i^2}{2m} + \sum_j \frac{\mathbf{P}_j^2}{2M_j} + V^M(\{\mathbf{R}\}) + V^L(\{\mathbf{r}\}) + V^{LM}(\{\mathbf{r}\},\{\mathbf{R}\}). \tag{2.1}$$

The notation $\{\mathbf{r}\}$ designates the set of all coordinates $\{\mathbf{r}_i\}_{i=1,\ldots,N_L}$, and $\mathbf{p}_i$, $\mathbf{r}_i$, and m are the momenta, positions and mass of L particles, while $\mathbf{P}_i$, $\mathbf{R}_i$, and M_i are the corresponding quantities for M particles. We imagine the system to be confined to a macroscopic volume Ω.

In many relevant physical systems the interactions between the particles are largely pairwise. An important example of such a system is a metal, where the L particles could be the valence electrons and the M particles the ionic cores. For pairwise systems we can write

$$V^L(\{\mathbf{r}\}) = \sum_i \phi_L^{(1)}(\mathbf{r}_i) + \sum_{i\neq j} v^L(\mathbf{r}_i,\mathbf{r}_j),$$
$$V^M(\{\mathbf{R}\}) = \sum_i \phi_M^{(1)}(\mathbf{R}_i) + \sum_{i\neq j} v_{ij}^M(\mathbf{R}_i,\mathbf{R}_j),$$

and

$$V^{LM}(\{\mathbf{r}\},\{\mathbf{R}\}) = \sum_{ij} v_j^{LM}(\mathbf{r}_i,\mathbf{R}_j),$$

where $\phi_L^{(1)}$ and $\phi_M^{(1)}$ denote external, one-particle potentials, v^L is the pair potential between L particles, v_{ij}^M is the pair potential between M particles i and j, and v_j^{LM} is the pair potential between the M particle j and the L particles. In the case of neutral systems composed of oppositely-charged particles, such as metals, care should be taken to define the perturbing and perturbed systems in such a way that thermodynamic functions such as the Helmholtz free energy of the separate systems, $F_i(\{\mathbf{R}\}, T)$, $\{i = M, L\}$ are well-defined. This can be achieved by requiring that each physical, charged particle 'carries' a uniform compensating charged background filling the whole available volume, ensuring a cancellation of divergent $\mathbf{k} = 0$ terms. In this case both the perturbed and perturbing systems are neutral and thermodynamically well-defined, and in the combined system the compensating backgrounds cancel out so we recover the physical system. In the case of the electron gas the perturbed system is simply jellium

We will restrict our attention to such pairwise systems, even though our main results require only the slightly weaker assumptions that we can write the interaction between L and M particles as

$$V^{LM}(\{\mathbf{r}\}, \{\mathbf{R}\}) = \sum_i V_1^{LM}(\mathbf{r}_i, \{\mathbf{R}\}).$$

This is to say that the joint effect of all the M particles, if they were held fixed, would amount to an additional one-body potential for the L particles.

2.1.1 Effective interactions and the adiabatic separation of time scales

As mentioned, we want to find an *effective* Hamiltonian for the M particles by tracing out the degrees of freedom associated with the L particles. Since volume (Ω) and

temperature (T) are specified, we begin with the relevant Helmholtz free energy F,

$$F = -kT \ln \mathrm{Tr}_{\{LM\}} e^{-\beta H}. \tag{2.2}$$

We then trace over the degrees of freedom associated with L at fixed configuration of M, assuming an adiabatic separation of time-scales,

$$F = -kT \ln \left(\mathrm{Tr}_M e^{-\beta(T^M + V^M)} \mathrm{Tr}_{L(M)} e^{-\beta(T^L + V^L + V^{LM})} \right),$$

where $\mathrm{Tr}_{L(M)}$ means the trace over states of particles of type L for a fixed configuration of particles M, and $T^{L,M}$ is the kinetic energy associated with the L and M particles, respectively. The free energy of system L for a fixed configuration of M is simply

$$F_L(\{\mathbf{R}\}, T) = -kT \ln \left(\mathrm{Tr}_{L(M)} e^{-\beta(T^L + V^L + V^{LM})} \right). \tag{2.3}$$

The total free energy can then be written as

$$F = -kT \ln \mathrm{Tr}_M e^{-\beta H_M^{eff}}, \tag{2.4}$$

where the effective Hamiltonian has the form

$$H^{eff} = \sum_j \frac{\mathbf{P}_j^2}{2M_j} + V^M(\{\mathbf{R}\}) + V^{eff}(\{\mathbf{R}\})$$

and $V^{eff}(\{\mathbf{R}\}) = F_L(\{\mathbf{R}\}, T)$ is the desired effective interaction between particles of type M. It is clearly state dependent, since it depends on temperature. It also depends on the volume Ω and on the properties of the L particles, including their mass, their number, and the form of their interactions. Note that even if the initial Hamiltonian contains only pairwise interactions, the *effective* Hamiltonian will typically contain many-center interactions as well as volume-dependent but structure-independent terms. Here we will focus on the determination of pair interactions.

In order to obtain the effective interaction, we need to calculate the free energy (2.3) of system L while the M particles are held fixed. To do this we treat the potential

$V_1^{LM}(\mathbf{r}, \{\mathbf{R}\})$ as an external one-body perturbation to the system composed of particles of type L. In the following $V_1^{LM}(\mathbf{r}, \{\mathbf{R}\})$ will therefore be simply written as $V_{ext}(\mathbf{r})$, keeping in mind the dependence of V_{ext} on the $\mathbf{R}_i$. We then proceed to a functional expansion of the free energy in orders of

$$V_{ext}(\mathbf{k}) = \int_\Omega d\mathbf{r} e^{-i\mathbf{k}\cdot\mathbf{r}} V_{ext}(\mathbf{r}),$$

with due regard to the compensating background discussed in section 2.1.

2.1.2 Response theory, and the coupling constant integration method

The change in density induced in a system by a perturbing external potential $V_{ext}(\mathbf{k})$ takes the form

$$\begin{aligned}\delta\rho(\mathbf{k}, T) &= \rho(\mathbf{k}, T) - \rho^{(0)}(\mathbf{k}, T) \\ &= \sum_{\mathbf{k}'} \chi^{(1)}(-\mathbf{k}, \mathbf{k}', T) V_{ext}(\mathbf{k}') + \frac{1}{\Omega} \sum_{\mathbf{k}', \mathbf{k}''} \chi^{(2)}(-\mathbf{k}, \mathbf{k}', \mathbf{k}'', T) V_{ext}(\mathbf{k}') V_{ext}(\mathbf{k}'') + \cdots\end{aligned} \tag{2.5}$$

The functions $\chi^{(n)}$ are, by definition, the response functions of the unperturbed system (which at this point is not necessarily uniform) and carry all the information about this system, including temperature. We have assumed a large but finite volume Ω and a dense but discrete distribution of wavevectors $\mathbf{k}$, since our main interest is the electron gas. Continuous systems can be recovered using the usual prescription [in three dimensions, it reads $1/\Omega \sum_{\mathbf{k}} \to \int d\mathbf{k}/(2\pi)^3$]. Even though the response functions depend on temperature, the methods we present here do not involve this temperature dependence explicitly. In order to simplify the notation, we will not explicitly keep track of the tem-

perature in the following. Unless otherwise specified, $\chi^{(1)}(\mathbf{k},\mathbf{k}')$ will simply stand for $\chi^{(1)}(\mathbf{k},\mathbf{k}',T)$, etc.

Equation (2.5) can be used together with the coupling constant integration method[1] [29] to obtain the variation in the Helmholtz free energy arising from the perturbation V_{ext} : namely,

$$\Delta F = \int_0^1 d\lambda \, \langle V_{ext} \rangle_\lambda \,, \tag{2.6}$$

where $\langle \cdot \rangle_\lambda$ is the statistical average with respect to the states of the Hamiltonian H_λ^{ext}, which is obtained by replacing V_{ext} by λV_{ext} in H^{ext}. In the thermodynamic integration scheme [30], the integrand in equation (2.6) is determined by numerical simulation. In the perturbation approach, it is instead expanded in powers of the external perturbation, yielding

$$\begin{aligned} \Delta F &= \frac{1}{\Omega} \sum_{\mathbf{k},n} \frac{\rho^{(n)}(\mathbf{k}) V_{ext}(-\mathbf{k})}{n+1} \\ &= \sum_{n=0}^{\infty} \frac{1}{(n+1)\Omega^n} \sum_{\mathbf{k}_1,\ldots,\mathbf{k}_{n+1}} \chi^{(n)}(\mathbf{k}_1,\ldots,\mathbf{k}_{n+1}) V_{ext}(\mathbf{k}_1)\ldots V_{ext}(\mathbf{k}_{n+1}), \end{aligned} \tag{2.7}$$

where $\rho^{(n)}$ is the part of (2.5) that is of order n in V_{ext}. The "zeroth order" response function is related to the density of the unperturbed system, $\chi^{(0)}(\mathbf{k}) = \rho^{(0)}(-\mathbf{k})/\Omega$.

Note that terms of n^{th} order in the external potential in expression (2.5) yield terms of order $n+1$ in (2.7). In the following, "n^{th} order response" refers to terms of order n in (2.5), unless otherwise specified.

In order to study pair potentials, let us now assume that the perturbation originates with *two* external sources, located at positions $\mathbf{R}_a$ and $\mathbf{R}_b$, so that

$$V_{ext}(\mathbf{k}) = V_a(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{R}_a} + V_b(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{R}_b}.$$

The free energy now depends on $\mathbf{R}_a$ and $\mathbf{R}_b$ (and would depend only on $\mathbf{R}_a - \mathbf{R}_b$ if

[1] It is assumed here that the free energy varies continuously while λ goes from 0 to 1.

we had restricted ourselves to initially homogeneous systems). This yields an induced effective pair potential $\phi^i(\mathbf{R}_a, \mathbf{R}_b)$ between the two sources, defined as the sum of all terms in (2.7) that depend on both V_a and V_b. Here the standard approach is to keep only terms up to a given order in V_{ext}. Linear response yields the *induced* pair potential

$$\phi^i_{lin}(\mathbf{R}_a, \mathbf{R}_b) = \frac{1}{\Omega} \sum_{\mathbf{k},\mathbf{k}'} \chi^{(1)}(\mathbf{k}, \mathbf{k}') V_a(\mathbf{k}) V_b(\mathbf{k}') e^{i(\mathbf{k}\cdot\mathbf{R}_a + \mathbf{k}'\cdot\mathbf{R}_b)}. \tag{2.8}$$

In order to emphasize the role of the coupling constant integration and to hint at an upcoming result [equation (2.11)], this can be written as

$$\begin{aligned} \phi^i_{lin}(\mathbf{R}_a, \mathbf{R}_b) = & \frac{1}{\Omega} \sum_{\mathbf{k}} \left(V_a(\mathbf{k}) \delta\rho^{lin}_b(-\mathbf{k}, \mathbf{R}_b) e^{i\mathbf{k}\cdot\mathbf{R}_a} + V_b(\mathbf{k}) \delta\rho^{lin}_a(-\mathbf{k}, \mathbf{R}_a) e^{i\mathbf{k}\cdot\mathbf{R}_b} \right) \\ & - \frac{1}{\Omega} \sum_{\mathbf{k}_1,\mathbf{k}_2} \chi^{(1)}(\mathbf{k}_1, \mathbf{k}_2) V_a(\mathbf{k}_1) V_b(\mathbf{k}_2) e^{i(\mathbf{k}_1\cdot\mathbf{R}_a + \mathbf{k}_2\cdot\mathbf{R}_b)}, \end{aligned} \tag{2.9}$$

where $\delta\rho^{lin}_i(-\mathbf{k}, \mathbf{R}_i)$ is the density induced, at linear order, by particle i located at position $\mathbf{R}_i$.

The first two terms in this expression correspond to the Coulombic energy of the system at the level of linear response. The third term, which arises from the coupling constant integration, incorporates the variation in kinetic energy and entropy caused by the perturbation.

Instead of keeping terms linear in V_{ext}, we can choose to keep all terms that are linear in V_a, for all orders in V_b. In this case we find, using equation (2.5), that the pair potential can be written as

$$\phi^i(\mathbf{R}_a, \mathbf{R}_b) \simeq \phi^i_{SPb}(\mathbf{R}_a, \mathbf{R}_b) = \frac{1}{\Omega} \sum_{\mathbf{k}} V_a(\mathbf{k}) \delta\rho_b(-\mathbf{k}, \mathbf{R}_b) e^{i\mathbf{k}\cdot\mathbf{R}_a}. \tag{2.10}$$

Here, $\delta\rho_b(\mathbf{k}, \mathbf{R}_b)$ is the total density that would be induced if the perturbation potential was caused by source b alone, i.e., $V_{ext}(\mathbf{k}) = V_b(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{R}_b}$. Notice that the terms arising from the interaction between $\delta\rho_a$ and V_b and the change in kinetic energy and entropy

arising from the perturbation (taken into account by the coupling constant integration) cancel each other out exactly in this case. One can also obtain a different estimate to the potential by inverting the roles of *a* and *b* in the previous discussion, leading to ϕ^i_{SPa}.

This asymmetric approach will be referred to as the successive perturbation method, or SPM. Indeed, one interpretation of this result is that instead of perturbing the initial system with *a* and *b* simultaneously, we initially add only *b*, determine the properties of the intermediate system, and then, subsequently, add particle *a*. Since our pair potentials result from the calculation of a free energy, the Gibbs-Bogoliubov inequality applies and equation (2.10) is related to a rigorous bound on the pair potential: namely,

$$\phi^i(\mathbf{R}_a, \mathbf{R}_b) \leq \phi^i_{SPb}(\mathbf{R}_a, \mathbf{R}_b) + \langle V_a \rangle_0 - \Delta F_a,$$

where $\langle V_a \rangle_0$ is the statistical average of the operator $V_a(\mathbf{k})e^{i\mathbf{k}\cdot\mathbf{R}_a}$ with respect to states of the initial, unperturbed system, and ΔF_a is the change in free energy induced by adding *a* to the unperturbed system, which can be obtained by setting V_b=0 in (2.7).

Equation (2.10) is the first term in the expansion of the energy if *b* is treated exactly and *a* is treated as a perturbation. Note that higher-order terms in the expansion would simply involve higher-order functional derivatives of the free energy of the intermediate system including *b*, which are related to density-density correlation functions of that system.

The SPM is especially useful if V_a is weak and V_b is strong [31], but is not as useful when both sources of perturbation are strong and require nonlinear treatment. On the other hand, we can easily obtain from this relation another expression for the induced pair potential that incorporates all contributions that are linear in *either* source. This expression therefore includes all contributions to the pair potential up to third order in

V_a and V_b. It reads

$$\begin{aligned}\phi^i_{SSP}(\mathbf{R}_a,\mathbf{R}_b) =& \frac{1}{\Omega}\sum_{\mathbf{k}}\left(V_a(\mathbf{k})\delta\rho_b(-\mathbf{k},\mathbf{R}_b)e^{i\mathbf{k}\cdot\mathbf{R}_a} + V_b(\mathbf{k})\delta\rho_a(-\mathbf{k},\mathbf{R}_a)e^{i\mathbf{k}\cdot\mathbf{R}_b}\right)\\ &- \frac{1}{\Omega}\sum_{\mathbf{k}_1,\mathbf{k}_2}\chi^{(1)}(\mathbf{k}_1,\mathbf{k}_2)V_a(\mathbf{k}_1)V_b(\mathbf{k}_2)e^{i(\mathbf{k}_1\cdot\mathbf{R}_a+\mathbf{k}_2\cdot\mathbf{R}_b)}.\end{aligned} \tag{2.11}$$

This result, which emerges from what we will refer to as the symmetrized successive perturbation method (SSPM), can also be seen to be a natural generalization of linear response (2.9). An alternate derivation in terms of a path integral in coupling parameter space is provided in section 2.6.1 Equation (2.11) is our main result for noninteracting systems. It also applies to interacting systems, but in that case it can be improved upon. We will do this in Section 2.2.

Equation (2.11), despite its simplicity, has many interesting features:

(a) It is very general; it can be used in any dimension, for classical and quantum, homogeneous and inhomogeneous, noninteracting and interacting systems (although, as mentioned, it can be improved for interacting systems; see Section 2.2). (b) It is intuitive: the first two terms are the interaction of the potential energy associated with each perturbation with the density induced by the other. The last term, which is equal to the negative of the linear response potential, accounts for the change in kinetic energy and entropy of the perturbed system and the contributions to the density that are not additive in V_a and V_b. (c) It includes all contributing terms up to quadratic response [yielding third-order terms in equation (2.7)], plus 8 out of 14 contributing terms of third-order response: it includes more terms than quadratic response. (d) It does not require the explicit knowledge of the second-order response function, but only that of the linear response function of the initial, unperturbed system. (e) It expresses the pair potentials in terms of quantities that are simple, symmetric, and can in principle be measured. Finally (f), since it takes the effect of a single, isolated perturbation as an external input,

equation (2.11) allows us to treat the effect of stronger, localized perturbations which are often difficult to treat with purely perturbative methods.

We use this result in two test cases in section 2.3. The first example is the effective interaction between particles perturbing, via delta-function potentials, a noninteracting quantum one-dimensional electron gas. The second is a version of the classical Asakura-Oosawa model [11, 12] of the depletion interaction, with finite square wells replacing hard-sphere potentials. We then apply it to the more realistic calculation of the pair potential between protons in a metallic environment.

Before we do this, we use the intuitive form of equation (2.11) to suggest the existence of a higher-order correction that applies to interacting systems, which will be derived in Section 2.5.

2.2 Corrections specific to interacting systems

As mentioned, equation (2.11) is valid for the interacting electron gas as well as for the noninteracting one: the interactions simply modify the response functions $\chi^{(n)}$. But upon further inspection of this equation, one might wonder about the absence of coupling between the induced densities themselves. The part of this interaction that is linear in V_a or V_b is included in (2.11), but the part that is nonlinear in both V_a and V_b is not. It turns out that some of these contributions can also be expressed intuitively in terms of $\delta\rho_a$ and $\delta\rho_b$.

For *homogeneous* interacting systems we obtain, by summing up higher-order terms that correspond to reducible diagrams (see Section 2.5), an extra term of the form

$$\phi_{red}(\mathbf{R}_{ab}) = \frac{1}{\Omega}\sum_{\mathbf{k}} \delta\rho_a^{NL}(\mathbf{k})\tilde{v}(k)\delta\rho_b^{NL}(-\mathbf{k})e^{i\mathbf{k}\cdot\mathbf{R}_{ab}}, \tag{2.12}$$

where $\mathbf{R}_{ab}$ is the separation between a and b, $\delta\rho_a^{NL} = \delta\rho_a - \delta\rho_a^{(1)}$, and $\tilde{v}$ is the effective interaction between the induced densities. It reads

$$\tilde{v}(k) = \epsilon(k)c_0^{(1)}(k), \tag{2.13}$$

where

$$\epsilon(k) = 1 - c_0^{(1)}(k)\chi_0(k)$$

involves the *noninteracting* response function $\chi_0^{(1)}$. In classical statistical mechanics $c_0^{(1)}(k)$ is the Ornstein-Zernike function for the unperturbed system, multiplied by k_BT. In the case of quantum mechanical particles of charge e, we have

$$c_0^{(1)}(k) = v_c(k) + \mu_1(k),$$

where $v_c(k) = 4\pi e^2/k^2$ is the Coulomb interaction and μ_1 is the first functional derivative with respect to density of the exchange-correlation potential. It is related to the local-field correction G by $\mu_1(k) = -G(k)v_c(k)$.

With some reorganization, we can now write the total induced pair potential in a remarkably simple form: namely,

$$\begin{aligned}\phi^i_{SSP}(\mathbf{k}) = \epsilon(k)\left[V_a(\mathbf{k})\delta\rho_b(-\mathbf{k}) + \delta\rho_a(\mathbf{k})V_b(-\mathbf{k})\right] \\ + \epsilon(k)\left[\delta\rho_a(\mathbf{k})c_0^{(1)}(k)\delta\rho_b(\mathbf{k}) - V_a(\mathbf{k})\chi_0^{(1)}(k)V_b(-\mathbf{k})\right].\end{aligned} \tag{2.14}$$

In real space,

$$\phi^i(\mathbf{R}_{ab}) = \frac{1}{\Omega}\sum_{\mathbf{k}}\phi^i(\mathbf{k})e^{i\mathbf{k}\cdot\mathbf{R}_{ab}}.$$

This result brings up an important point. The energy associated with the interaction between potential V_b and the n^{th} order contribution to the density induced by a, which we write as $\delta\rho_a^{(n)}$, arises at order $n+1$ in the energy expansion. On the other hand, the "electron-electron" interaction between $\delta\rho_a^{(n)}$ and $\delta\rho_b^{(n)}$ appears only at order $2n$ in

the energy. Except for the special case of linear response, the electron-electron interaction terms originate with a higher order in response than the corresponding electron-perturbation terms. Therefore termination of the series (2.7) at any order beyond linear response will typically result in pair potentials that are overly attractive.

Indeed, if we compare the effective pair potentials for hydrogen atoms in jellium obtained from quadratic response [25] to those obtained from ab initio methods [26], we observe exactly such a discrepancy. We calculate in section 2.3.3 the lowest-order contribution arising from (2.12), and find that it indeed improves the agreement between ab initio methods and response theory.

We finally draw the reader's attention to the similarity (for protons in an electron gas) between the ab initio equation (2.14) and the variational Heitler-London evaluation of the isolated hydrogen molecule energy. This will be further discussed in section 2.3.3.

2.3 Examples of applications

2.3.1 Delta functions in a noninteracting electron gas

As a simple instructive example, we first consider a one-dimensional noninteracting electron gas of unperturbed linear density ρ_0 confined to a large length L, with periodic boundary conditions. The external perturbations have the form

$$V_\alpha(r) = \frac{\hbar^2 u}{2m}\delta(r) \qquad (\alpha = a, b).$$

From (2.11) we obtain immediately

$$\phi_{SSP}(R_{ab}) \simeq \frac{\hbar^2 u}{2m}(2\delta\rho_a(R_{ab}) - \delta\rho^{lin}(R_{ab})). \tag{2.15}$$

Here

$$\delta\rho^{lin}(r) = \frac{\hbar^2 u}{2m} \sum_k \chi^{(1)}(k,-k)e^{ikr} \tag{2.16}$$

and $\chi^{(1)}$ is the linear response function of the one-dimensional noninteracting electron gas [8, 32]:

$$\chi^{(1)}(k_1,k_2) = \frac{2m}{\hbar^2 \pi k_1} \ln\left|\frac{k_1+2k_F}{k_1-2k_F}\right| \delta_{k_1,-k_2}.$$

The Fermi wave-number k_F is related to the unperturbed linear density by $k_F = \pi\rho_0$.

Converting the sum into an integral, we find (see Kittel [8], and also Yafet [32] and Giuliani *et al.* [33])

$$\delta\rho^{lin}(r) = \frac{u}{\pi}\mathrm{Si}(2k_F r) - \frac{u}{2}, \tag{2.17}$$

where Si is the sine integral function [34]. The nonlinear induced density $\delta\rho_a(r)$ can also be calculated exactly as the sum of the bound and scattering state densities: namely,

$$\delta\rho_a = \delta\rho_{bound} + \delta\rho_s,$$

with

$$\delta\rho_{bound}(r) = -\Theta(-u)ue^{u|r|} = \begin{cases} -ue^{u|r|}, & \text{if } u<0, \\ 0, & \text{otherwise.} \end{cases} \tag{2.18}$$

Here Θ is the Heaviside step function.

The scattering state density is, in the limit of large L [33],

$$\delta\rho_s(r) = \frac{2}{\pi}\int_0^{k_F} dk \left(\frac{2uk\sin(2k|r|)}{4k^2+u^2} - \frac{u^2\cos(2k|r|)}{4k^2+u^2}\right). \tag{2.19}$$

Factors of 2 are included for spin degeneracy. This integral can be evaluated in terms of the exponential integral E_1 using, for example, relations 5.1.41 and 5.1.42 from Abramowitz and Stegun [34]. We find

$$\delta\rho_s(r) = \frac{e^{u|r|}u}{\pi}\left(\Im m\left[\mathrm{E}_1\left(\left(u+2ik_f\right)|r|\right)\right] + \pi\Theta(-u)\right). \tag{2.20}$$

The total density induced by a single delta function potential in a noninteracting electron gas therefore takes the very simple form

$$\delta\rho_a(r) = \frac{ue^{u|r|}}{\pi}\Im m\left[\mathrm{E}_1\left(\left(u + 2ik_f\right)|r|\right)\right]. \tag{2.21}$$

Note that for $k_f \to 0$ we find that $\delta\rho = \delta\rho_{bound}$, as expected. The appearance of a bound state at $u = 0$ corresponds to the branch cut of E_1 along the negative real axis.

If we use (2.17) and (2.21) in (2.15), we find an expression for the pair potential as a function of R_{ab} which reads

$$\begin{aligned}\phi_{SSP}(R_{ab}) = &\frac{\hbar^2u^2}{\pi m}e^{uR_{ab}}\left(\Im m\left[\mathrm{E}_1\left(\left(u + 2ik_f\right)R_{ab}\right)\right]\right)\\ &-\frac{\hbar^2u^2}{2\pi m}\left(\mathrm{Si}(2k_FR_{ab}) - \frac{\pi}{2}\right).\end{aligned} \tag{2.22}$$

Alternatively, the *exact* pair potential can be calculated by solving the Schrödinger's equation directly for different values of R_{ab}. The result is expressed as a sum over the eigenvalues of the Hamiltonian, which are obtained by numerically solving a set of transcendental equations.

We compare in Figure 2.1 these numerical pair potentials with the analytical ones obtained from the SPM and SSPM [equation (2.22)], and those obtained from linear response, for various interaction strengths, including attractive and repulsive cases. We observe that the SSPM improves upon linear response and the SPM, especially quantitatively, at low u, but also qualitatively (at higher u.)

As a side remark, notice that the system has no bound state for $u > 0$, one even bound state for $u < 0$, and one extra odd bound state when $R_{ab}u < -2$. It has been suggested [35] that the appearance of a bound state might cause the failure of response theory, since such states are qualitatively different from the initial, unperturbed free electron system. We see that this is not the case in this particular example and that the

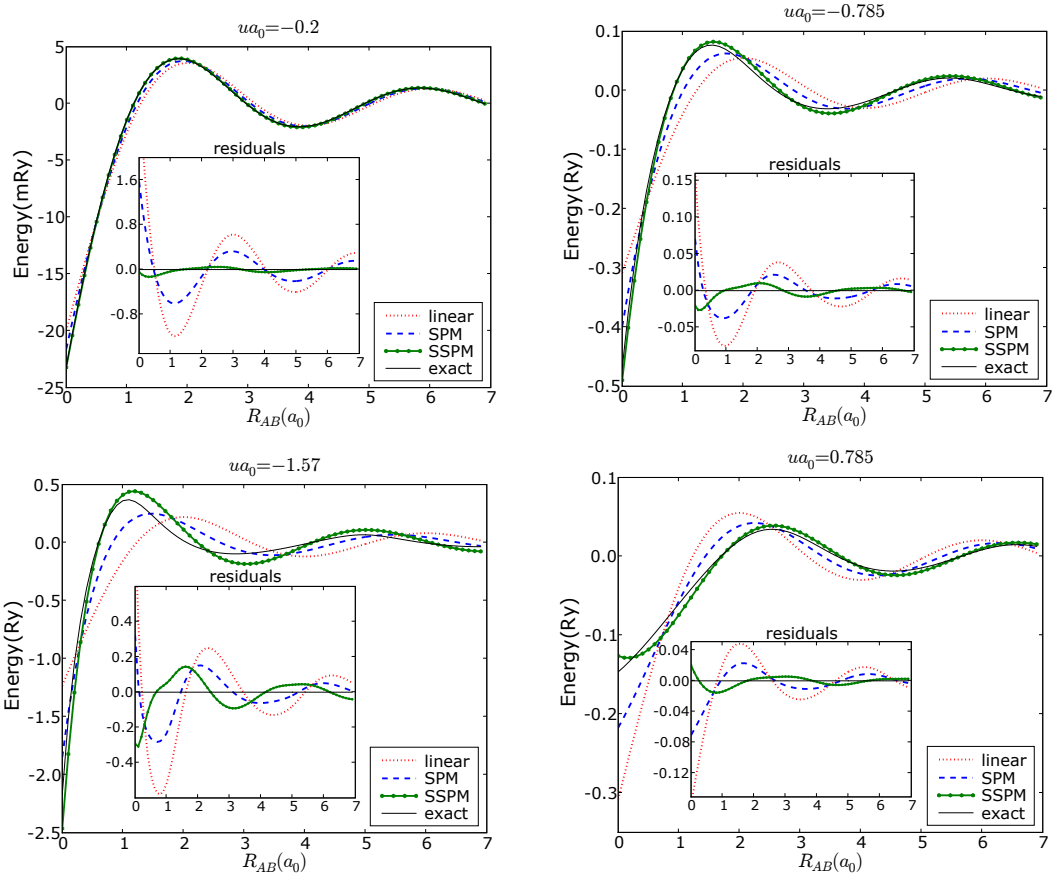

Figure 2.1: Effective pair interaction between delta function potentials in one dimensional noninteracting electron gas: exact, linear response, and the SPM and SSPM results. The Fermi wavevector is set to $k_f = 0.785/a_0$. The residuals (difference between estimated and exact pair potentials) are shown in the inset. Note the difference in behavior of the SSPM near the origin between attractive (first three figures) and repulsive (last figure) delta function potentials.

pair potentials can be accurately described by response theory despite the presence of a bound state. This is consistent with the observation that no discontinuity in the density of charge occurs upon the formation of a localized, bound state (see, e.g., Galindo *et al* [36]).

2.3.2 Classical depletion interaction

The depletion interaction (or entropic attraction) plays a major role in classical colloidal systems, where it is used to tune the interaction between colloidal particles immersed in a solvent. The addition of polymers to the solvent indeed causes an effective attraction between the colloidal particles which can be adjusted by modifying the polymer concentration.

The Asakura-Oosawa model [11, 12] describes this effect by representing colloidal particles as hard spheres of radius D and (folded) polymers as hard spheres of radius δ. The interactions between the polymers are neglected. The polymers are therefore treated as an ideal gas that is excluded from spheres of radius $D + \delta$ surrounding each colloid. Here we will replace the hard-sphere potential with a finite repulsion of magnitude V_0. Within this model the effective attractive potential between the colloids can be calculated exactly and compared to the response theory results, providing a useful benchmark for perturbative approaches in the classical regime.

Suppose we have only two colloidal particles (at positions $\mathbf{R}_{a,b}$) in a bath of polymers of unperturbed density $\rho_0 = N/\Omega$. Since the polymers are taken not to interact, their density is given by

$$\rho(\mathbf{r}) = \frac{Ne^{-\beta V_e(\mathbf{r})}}{\int_\Omega d\mathbf{r}' e^{-\beta V_e(\mathbf{r}')}}, \tag{2.23}$$

where $V_e(\mathbf{r}) = V_a(\mathbf{r}) + V_b(\mathbf{r})$ is the total potential and

$$V_{a,b} = \begin{cases} V_0, & \text{if } |\mathbf{r} - \mathbf{R}_{a,b}| < D + \delta, \\ 0, & \text{otherwise.} \end{cases}$$

Since $V_e(\mathbf{r})$ is zero except within a bounded region, $\rho(\mathbf{r})$ simplifies, in the thermodynamic limit, to

$$\rho(\mathbf{r}) = \rho_0 e^{-\beta V_e(\mathbf{r})}. \tag{2.24}$$

Now we can use the coupling constant integration method to obtain the Helmholtz free energy:

$$\begin{aligned} F - F_0 &= \int_0^1 d\lambda \int d\mathbf{r} V_{ext}(\mathbf{r}) \rho_\lambda(\mathbf{r}) \\ &= -A_o(\mathbf{R}_a, \mathbf{R}_b) \frac{\rho_0}{\beta} \left(e^{-2\beta V_0} - 1 \right) \\ &\quad - A_r(\mathbf{R}_a, \mathbf{R}_b) \frac{\rho_0}{\beta} \left(e^{-\beta V_0} - 1 \right), \end{aligned} \tag{2.25}$$

where A_o is the overlap volume between the two spheres of radius $D + \delta$ centered at $\mathbf{R}_{a,b}$, $A_r = 8\pi(D+\delta)^3/3 - 2A_o$ is the non-overlapping volume of the spheres, and β is the inverse temperature. All the $\mathbf{R}_{a,b}$ dependence is contained in $A_o(\mathbf{R}_a, \mathbf{R}_b) = A_o(\mathbf{R}_a - \mathbf{R}_b)$. Accordingly we write the pair potential as

$$\phi^i(\mathbf{R}_a - \mathbf{R}_b, V_0) = -\frac{\rho_0 A_0(\mathbf{R}_a - \mathbf{R}_b)}{\beta} \left(e^{-\beta V_0} - 1 \right)^2 . \tag{2.26}$$

Notice the obvious limit $V_0 \to \infty$, yielding the familiar Asakura-Oosawa result [11, 12]

$$\phi^i(\mathbf{R}_a - \mathbf{R}_b, \infty) = -\frac{\rho_0 A_0(\mathbf{R}_a - \mathbf{R}_b)}{\beta}. \tag{2.27}$$

This problem can also be treated at various orders of response theory, using, e.g., $\chi^{(1)}(\mathbf{k}, \mathbf{k}') = -\beta\rho_0\delta_{\mathbf{k},-\mathbf{k}'}$. The general form for the pair potential, which can be obtained by carrying the perturbation to any order, can be expressed as

$$\phi_n^i(\mathbf{R}_a - \mathbf{R}_b, V_0) = -f(\beta V_0) \frac{\rho_0 A_0(\mathbf{R}_a - \mathbf{R}_b)}{\beta}.$$

Only the position-independent function f is modified in the various approximations. The exact (nonperturbative) form for the function f is

$$f(x) = (e^{-x} - 1)^2 .$$

We evaluate the performance of the various orders in response and of the SPM and SSPM by comparing the estimates obtained from each method to the exact result for

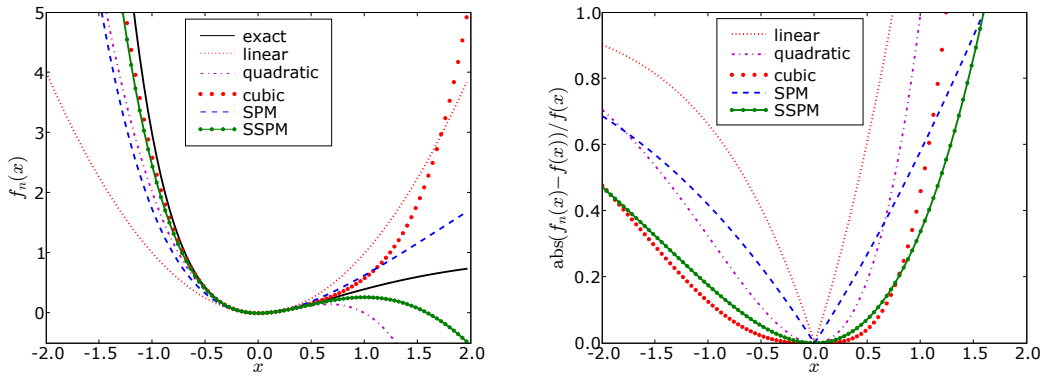

Figure 2.2: (upper figure) Dependence of the effective pair potential between two repulsive colloidal particles on the interaction strength parameter $x = \beta V_0$: exact result versus the SPM and SSPM, and perturbations at various orders. (lower figure) Relative errors $|f_n(x) - f(x)|/f(x)$.

$f(x)$. Since we know the exact result analytically, we can obtain the n^{th} order response estimate to f, and hence to the pair potential, by a simple Taylor expansion of $f(x)$, keeping terms up to $(n+1)^{th}$ order in $x = \beta V_0$, i.e.,

$$f(x) = x^2 - x^3 + \frac{7}{12}x^4 + \cdots$$

Expansions up to third order are shown on Figure 2.2, together with the SPM result [equation (2.10)]

$$f_{SPM}(x) = (1 - e^{-x})\, x = x^2 - \frac{x^3}{2} + \frac{x^4}{6} + \cdots ,$$

and the SSPM result [equation (2.11)]

$$f_{SSPM}(x) = 2\left(x(1 - e^{-x}) - \frac{x^2}{2}\right) = x^2 - x^3 + \frac{x^4}{3} + \cdots$$

Negative values of x are included in Figure 2.2 for illustrative purposes. One can again observe that because of the additional terms it includes, the SSPM is more accurate than linear and quadratic response for all V_0 and more accurate than the SPM for $x < 1.59$. Third-order response, on the other hand, is closer to the exact result than the SSPM

for $-1.93 < x < 0.78$, which is expected since the SSPM does not include all third-order terms. For larger $|x|$, though, the higher-order terms play a more important role and the SSPM is more accurate than third-order response.

This example is also instructive in that it allows us to study directly the convergence of response theory. Note that for $V_0 < 0$ the convergence is monotonous, while for $V_0 > 0$ there is a more complex, alternating approach to the exact result. This can be traced back to the fact that changing the overall sign of the perturbing potential results in changing the sign of odd orders in response, without affecting the even orders. Thus, if a potential exhibits monotonous convergence, its additive inverse exhibits alternating convergence. We might therefore expect, for example, that effective interactions with protons and antiprotons in an electron gas will have completely different convergence behaviors. If this classical example is to be representative, protons would then exhibit uniform convergence, while antiprotons should exhibit alternating convergence.

Note, finally, that in the problem at hand, response theory converges even for arbitrarily large V_0 and $D + \delta$. Since for repulsive spheres the functions $f(\beta V)$ approaches a constant value exponentially, it is even possible, in this case, to obtain a quantitative value for the limit $V_0 \to \infty$ by keeping only a finite number of perturbation terms.

2.3.3 The hydrogen molecule and connections with the Heitler-London approach

We move on to the more realistic system composed of two initially bound proton-electron systems immersed in a uniform, neutral, and *interacting* jellium [37–39] at a temperature much lower than the Fermi temperature. This hydrogen-in-jellium prob-

lem can be linked to real systems in two ways. First, it can be seen as a first step in a formal expansion including three- and many-center terms which, if carried to all orders, should yield the exact total energy of the system, within the adiabatic approximation. In can also be used within an effective-medium approach [27] in an attempt to take into account the many-ion effects in an approximate way. In both cases the pair potentials can be used to derive phonon spectra. In particular the pair potentials obtained from quadratic response were used to predict the infrared and Raman vibron frequencies [25].

To establish a basis of comparison, we first obtain an estimate of the importance of nonlinear corrections to the one-atom density, $\delta\rho_a$, using the ab initio DFT program VASP [40, 41]. The use of self-consistent Kohn-Sham DFT allows us to estimate the effect on the pair potentials of the higher-order terms in the determination of $\delta\rho_a$.

To obtain this, we used a cubic cell of side $13.5a_0$ containing 74 electrons (for $r_s = 2$), together with a $6 \times 6 \times 6$ k-space grid and a standard projector augmented wave (PAW) pseudopotential for hydrogen [42, 43] with cutoffs up to 450 eV. The generalized gradient approximation (GGA) to the exchange-correlation potential, as parametrized by Perdew and Wang [44], was used, together with the Methfessel-Paxton smearing [45], with a smearing temperature of $\sigma = 0.2$ eV. Since it was shown [26] that the proton-proton pair potential does not depend strongly on the choice of a pseudopotential, cutoff energy, and exchange-correlation functional, our choice of parameters should yield sufficient precision for our purposes, even considering possible short-range distortions due to core overlap.

The SSPM pair potentials thus obtained are shown in Figure 2.3, and are compared with the VASP results for the pair potentials from Bonev and Ashcroft [26], the quadratic response results, and the SSPM pair potentials corrected by the inclusion of equation (2.12) to lowest order in the perturbing potential. Note that the calculation of the VASP

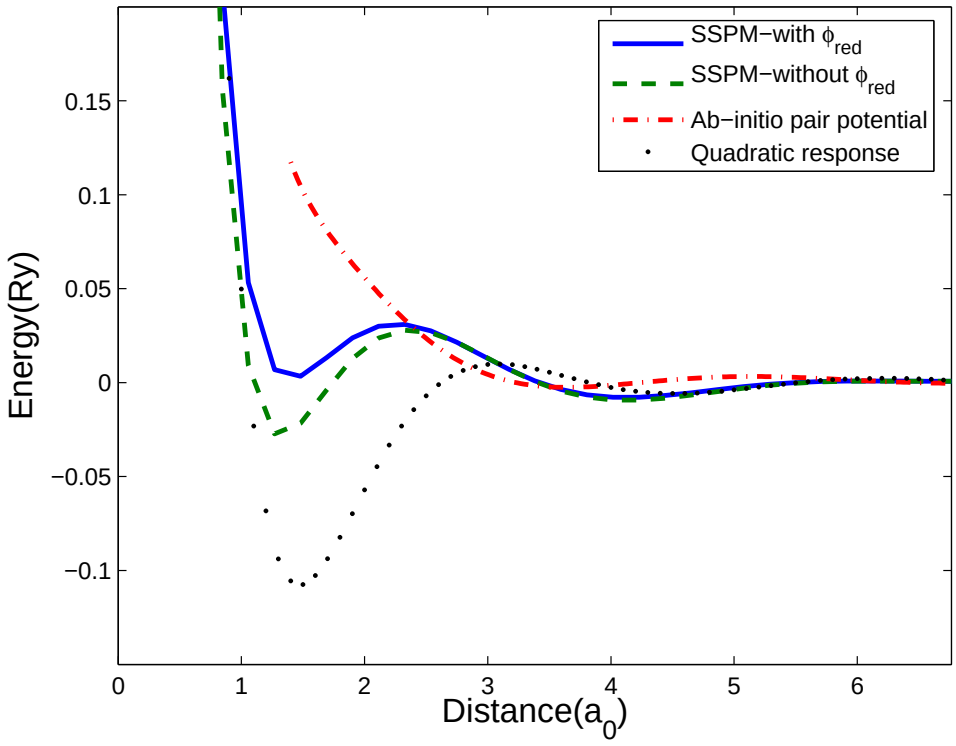

Figure 2.3: Comparison of the SSPM results with and without contributions from equation (2.12) to quadratic response and ab initio results (from Bonev and Ashcroft [26]) for proton-proton pair potentials at $r_s = 2$.

pair potentials used pseudopotentials to circumvent the difficulty caused by the singularity of the electronic wavefunction at the position of the proton, which raises the question of the transferability of the pseudopotential to such high-pressure regimes. In particular, one might be concerned if the core radius of the pseudopotential was of the same order as the interproton separation. However, Bonev and Ashcroft [26] report that the choice of pseudopotential has little impact on the pair potentials, and that tests with the actual Coulombic potential yielded similar results.

To obtain the correction resulting from equation (2.12), we use

$$\delta\rho_a^{NL}(\mathbf{k}) \simeq \frac{1}{\Omega}\sum_{\mathbf{k}',\mathbf{k}''}\chi^{(2)}(-\mathbf{k},\mathbf{k}',\mathbf{k}'')V_{ext}(\mathbf{k}')V_{ext}(\mathbf{k}'').$$

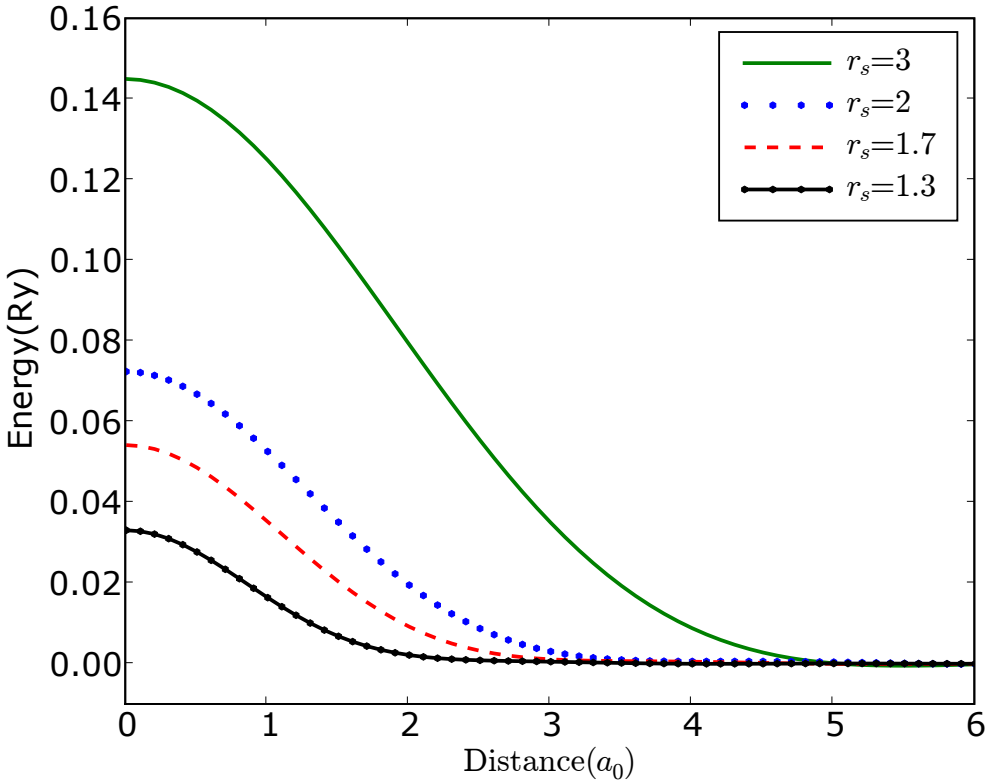

Figure 2.4: Contribution of the diagram from Figure 2.5(a) to the proton-proton pair potential for various values of r_s.

This corresponds to the energy diagram shown as Figure 2.5(a).

We use the second order noninteracting response function of the homogeneous electron gas, $\chi_0^{(2)}(\mathbf{k},\mathbf{k}',\mathbf{k}')$, in the form given by Milchev and Pickenhain [46]. We use the random phase approximation value for the quadratic response function:

$$\chi^{(2)}(\mathbf{k},\mathbf{k}',\mathbf{k}') = \frac{\chi_0^{(2)}(\mathbf{k},\mathbf{k}',\mathbf{k}')}{\epsilon(\mathbf{k})\epsilon(\mathbf{k}')\epsilon(\mathbf{k}'')},$$

This approximation is expected to improve as the density is increased (see, e.g., Pines and Nozières [29]), and should be sufficient to give us information on the general behavior of the potential. Finally we choose the Vashishta-Singwi form [47] for the local field correction $G(\mathbf{k})$.

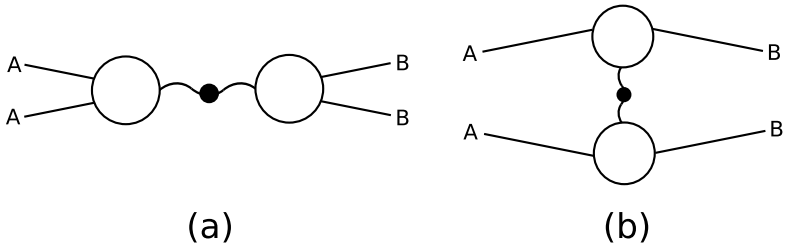

Figure 2.5: (a) Lowest order diagram contributing to the interaction (2.12). (b) Nonreducible diagram equal to diagram (a) in the limit $R_{ab} \to 0$, for identical ions, here protons (diagrammatic conventions are explained in Section 2.5 and Figure 2.7).

Under these assumptions we obtain the contribution to the pair potentials displayed in Figures 2.3 and 2.4.

We see that the inclusion of the potential ϕ_{red} from equation (2.12), and therefore of the density-density interaction $\tilde{v}(q) = \epsilon(q)\left[v_c(q) + \mu_1(q)\right]$, leads to a contribution to the pair potential that is repulsive at typical proton-proton separation. This behavior largely arises from the Coulomb repulsion term v_c. This contribution therefore explains part, but not all, of the discrepancy observed between the pair potentials obtained from quadratic response [25] and ab initio methods [26]. In particular, all perturbation-based methods yield a local minimum around $R = 1.4a_0$, whereas no such minimum is observed in the VASP pair potential at the considered density; such a minimum appears for VASP pair potentials only at lower densities, corresponding to $r_s \gtrsim 3$ (see Bonev and Ashcroft [26]). The consequences of the disappearance of this local minimum at higher density for the stability of the hydrogen molecule (or crystal) has also been discussed in Nagao *et al.* [25] and Díez Muiño and Salin [39].

Note that for each diagram of the form shown in Figure 2.5(a) there are two similar diagrams of the form shown in Figure 2.5(b) which are not reducible and hence not contained in (2.12). In the limiting case where the interparticle distance tends to zero, though, diagrams 2.5(a) and 2.5(b) should provide the same contribution. Therefore

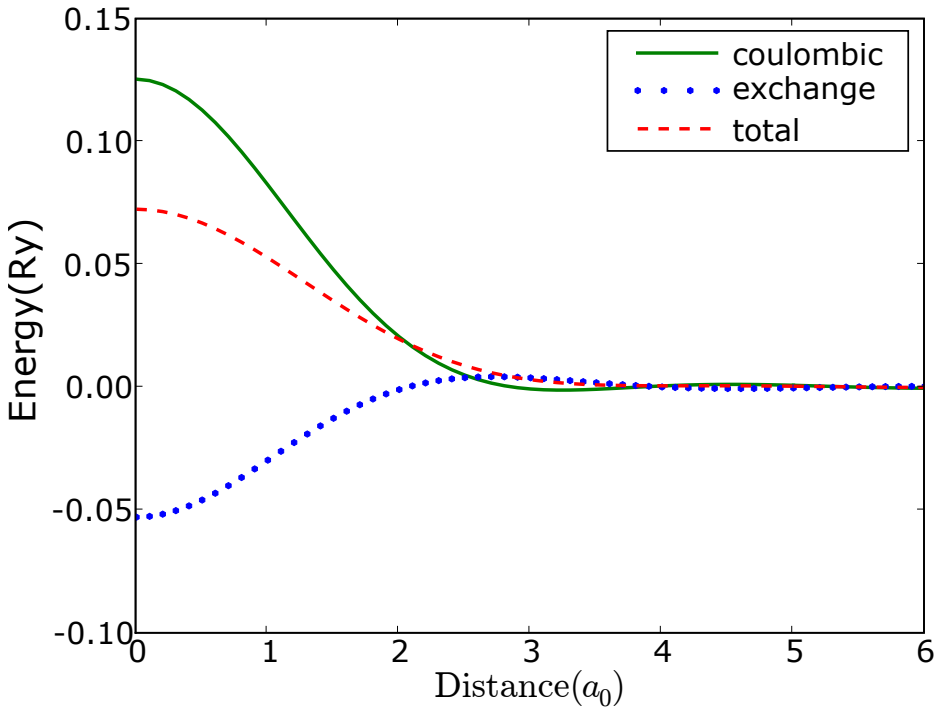

Figure 2.6: Contributions of the Coulombic and exchange parts to the effective proton-proton potential $\tilde{v}$, for $r_s = 2$.

diagrams of the form 2.5(b) should also have an overall repulsive behavior, contributing to further reduce the discrepancy between perturbative and VASP results.

Note also that if we consider separately the contributions of $\tilde{v}_c(q) = \epsilon(q)v_c(q)$ and $\tilde{\mu}_1 = \epsilon(q)\mu_1(q)$, we find that the contribution of $\tilde{\mu}_1$ is mostly attractive, as can be seen in Figure 2.6 for $r_s = 2$. We also observe that terms arising from equation (2.12) exhibit very weak Friedel oscillations. This distinguishes these terms from the other contributions calculated here or ab initio pair potentials [26] (compare Figures 2.4 and 2.3), and we conjecture that this arises from the reducibility of the energy diagram.

We draw the reader's attention to the close similarity of the pair potential of equation (2.14) with corresponding terms in the Heitler-London (HL) picture of the isolated hydrogen molecule. We can identify in both methods (i) the ion-ion repulsions, (ii)

the attractive interactions between ion b (a, respectively) with the density induced by a (b, respectively), (iii) the Coulombic repulsion between the one-atom electronic densities, and (iv) an attractive exchange contribution from the electrons. The fourth term in (2.14), which has no equivalent in the HL picture, goes to zero in the low density, free molecule case. It is interesting that two such different approaches, one being variational in essence and the other one perturbative, yield such similar results.

Some features of the HL pictures cannot be observed in the SSPM, though, because of the nonzero average electronic density considered in the SSPM. For example, while the pair potential between protons immersed in spin-polarized or spin-unpolarized electrons differed only by the exchange term in HL, this is no longer the case in the SSPM approach. The densities $\delta\rho_i$ induced by single protons are indeed quite different for spin-polarized and spin-unpolarized electrons.

2.4 Implications of SSPM results

To obtain effective interactions between particles immersed in a well understood system, such as ions immersed in a uniform jellium, the traditional perturbation approach treats all immersed particles as a single perturbation. One finds the free energy of the perturbed system as a function of the total perturbation potential. The total potential is then separated in the sum of its constituents, which correspond to pair-, triplet-, and many-body effective potentials.

Here we have discussed two alternative approaches. In the first approach, we start by immersing a single particle in the well understood system and then calculate the response functions of the new, but perturbed system. Since these response functions are directly related to the electronic density or density-density correlation functions, they

can be obtained by a variety of techniques, including perturbation theory, simulations, and density functional theory, and even deduced from experiment. The remaining particles are then treated as a further perturbation of this already perturbed system.

Such a two-step process is ideal when only one perturbation is strong [31], but it is intrinsically asymmetrical and not ideal when both perturbations are large enough to induce nonlinear effects. We have suggested a way to improve and symmetrize this procedure which allows one to treat an increased number of perturbation terms with little additional effort. More specifically, by using results of the asymmetrical approach that are exact to linear order in the perturbation potential, it is possible to construct a symmetrized result exact to quadratic order and including additional higher order terms as well. We applied this method to two simple noninteracting test systems and found that it improves upon standard linear and quadratic response, as expected.

More importantly, it was shown that this simple method could be naturally refined by the inclusion of higher order terms describing, in particular, electron-electron interactions. These higher order terms also have intuitive physical meaning, and we used this to argue that the standard termination of the perturbation series at a given order is not the best strategy. The inclusion of the higher order terms was indeed shown to improve considerably the agreement between the perturbation and density functional theory approaches in the problem of proton-proton pair potentials. It is not clear which approximation yields the best estimate for the pair potential between hydrogen ions in jellium; whereas SSP neglects higher-order corrections, the DFT approach relies on the use of pseudopotentials which might not be transferable to this pressure range. The inclusion of higher-order terms in perturbation theory and the application of DFT using pseudopotentials designed for this density range would be in order to clarify this issue.

Even though we considered here effective interactions between identical particles

only, the SSP method is particularly well suited to the description of systems with multiple (say N) types of particles. It indeed reduces the computational difficulty from the determination of $N^2/2$ pair potentials to that of finding N (typically symmetric) induced densities, from which the pair potentials can be obtained in a straightforward manner. It can also be generalized to many-center interactions and to magnetic perturbations (see Section 2.6).

Finally, a similarity is observed between terms in the pair potentials arising from this approach and from the Heitler-London variational approach for diatomic molecules, leading to a possible natural generalization of the Heitler-London approach to metallic systems. This similarity could be explored further by comparing the pair potential for a pair of atoms in a jellium, as derived here, to a pair of atoms in a Wigner-Seitz spherical cell ensuring the same average density, which could be treated within a Heitler-London-like approach.

A more detailed comparison of the SSPM and ab initio pair potentials for hydrogen and other materials, especially at densities higher than those considered here, would be a logical next step to this work, as would be a more detailed treatment of the many-body interactions, both from the ab initio and SSPM perspectives.

2.5 Derivation of equation (2.12)

We first use the Hohenberg-Kohn-Sham approach [13, 48] (and the finite-temperature extension due to Mermin [49]) to write the induced density in terms of the response functions of a noninteracting system in a modified external potential (see, e.g., Lundqvist and

March [50]). We write the density as

$$\begin{aligned}\delta\rho(\mathbf{k}) = &\sum_{\mathbf{k}'} \chi_0^{(1)}(-\mathbf{k},\mathbf{k}')\Gamma(\mathbf{k}') \\ &+ \frac{1}{\Omega}\sum_{\mathbf{k}',\mathbf{k}''} \chi_0^{(2)}(-\mathbf{k},\mathbf{k}',\mathbf{k}'')\Gamma(\mathbf{k}')\Gamma(\mathbf{k}'') + \cdots ,\end{aligned} \tag{2.28}$$

where $\chi_0^{(n)}$ are the response functions of the noninteracting system and Γ is the *effective* perturbation potential. We write Γ as

$$\Gamma(\mathbf{k},\lambda) = \lambda V(\mathbf{k}) + c(\mathbf{k},[\rho]) - c(\mathbf{k},[\rho_0]), \tag{2.29}$$

where $c(\mathbf{k},[\rho])$ is the first order direct correlation function of the unperturbed system, multiplied by k_BT.

The parameter λ is introduced to keep track of the order of the expansion. For example, we can write

$$\Gamma(\mathbf{k}) = \lambda\Gamma^{(1)}(\mathbf{k}) + \lambda^2\Gamma^{(2)}(\mathbf{k}) + \cdots \tag{2.30}$$

The variation $\delta c(\mathbf{k},[\rho]) = c(\mathbf{k},[\rho]) - c(\mathbf{k},[\rho_0])$ can in turn be written as a functional expansion:

$$\begin{aligned}\delta c(\mathbf{k}) = &\sum_{\mathbf{k}'} c_0^{(1)}(-\mathbf{k},\mathbf{k}')\delta\rho(\mathbf{k}') \\ &+ \frac{1}{\Omega}\sum_{\mathbf{k}',\mathbf{k}''} c_0^{(2)}(-\mathbf{k},\mathbf{k}',\mathbf{k}'')\delta\rho(\mathbf{k}')\delta\rho(\mathbf{k}'') + \cdots\end{aligned} \tag{2.31}$$

Note that here $c_0^{(i)}$ is the $(i+1)^{th}$ direct correlation function of the unperturbed system, multiplied by k_BT. This slightly unusual notation is chosen to emphasize the similarity between equations (2.28) and (2.31).

Finally the induced density can also be expressed in powers of the external potential:

$$\delta\rho(\mathbf{k}) = \lambda\delta\rho^{(1)}(\mathbf{k}) + \lambda^2\delta\rho^{(2)}(\mathbf{k}) + \cdots \tag{2.32}$$

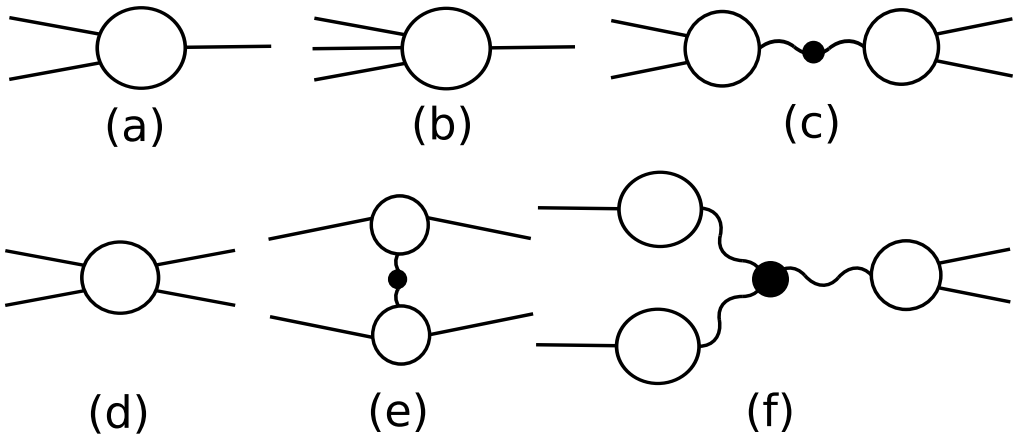

Figure 2.7: Some typical diagrams from the free energy expansion contributing to the pair potential. Loops with $n+1$ legs represent response function of order n. The correlation function $c_0^{(n)}$ is represented by a solid circle surrounded by $n+1$ wavy lines and the external potential by solid lines.. Diagram (a) is a quadratic response contribution. Diagrams (b) and (c) represent some third order contributions taken into account by equations (2.11) and (2.12), respectively. Diagrams (d), (e), and (f) are examples of third-response contributions which are *not* taken into account in the approach presented here.

The variation in the free energy is obtained as before through equation (2.7), which we now write as

$$\begin{aligned} \Delta F &= \frac{1}{\Omega}\sum_{\mathbf{k},n}\frac{\rho^{(n)}(\mathbf{k})V_{ext}(-\mathbf{k})}{n+1} \\ &= \Delta F_1 + \frac{1}{\Omega}\sum_{\mathbf{k},n}\frac{\delta\rho^{(n)}(\mathbf{k})V_{ext}(-\mathbf{k})}{n+1}, \end{aligned} \tag{2.33}$$

where ΔF_1 does not depend on the relative positions of the perturbation sources and will therefore not contribute to the pair potentials.

Using equations (2.28), (2.29), and (2.31), we can represent each term in (2.33) by a diagram using three types of building blocks: namely $\chi_0^{(n)}$ and $c_0^{(n)}$ for $n \geq 1$, and V_{ext}. A few examples are shown in Figure 2.7.

Each V_{ext} is connected to a single $\chi_0^{(n)}$. Each $c_0^{(n)}$ is connected to $n+1$ different $\chi_0^{(n)}$. Each $\chi_0^{(n)}$ is connected to $n+1$ blocks that can be either V_{ext}, or a $c_0^{(n)}$. Conversely, every

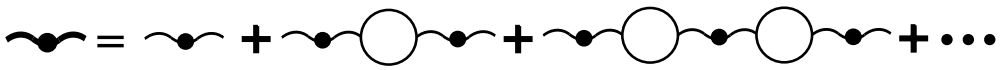

Figure 2.8: Diagrammatic expansion for the dressed $\tilde{c}_0^{(1)}$ in terms of the bare $c_0^{(1)}$ and linear response functions.

treelike diagram obeying these rules corresponds to a term in equation (2.33). Note that since $\delta\rho(\mathbf{q})$ obeys the same rules, it can be represented by the same diagrams, with only one V_{ext} removed.

We call reducible those diagrams in (2.33) that can be separated, by the cutting of a single $c_0^{(1)}$ line, into two diagrams that depend exclusively on V_a or on V_b, respectively, and that are linear in neither V_a nor V_b. In Figure 2.7, diagram (*c*) is reducible, while the others are not.

We want to show that the set of such diagrams is equivalent to those described by equation (2.12). First we will show that there is a one-to-one correspondence between reducible diagrams in (2.33) and diagrams in (2.12). Then we will show that the prefactors also agree.

2.5.1 Diagrammatic equivalence

Consider a reducible diagram *D* contributing to the pair potential. Then consider the set *S* of all diagrams that are different from *D* only by the number of separating interaction lines. The separating interaction lines can only be connected by $\chi_0^{(1)}$ loops. By summing over the different numbers of $\chi_0^{(1)}$ loops and the momenta associated with these loops, we can therefore obtain a dressed propagator in terms of a dielectric function,

$$\tilde{c}_0^{(1)}(\mathbf{k},\mathbf{k}') = \frac{c_0^{(1)}(\mathbf{k},\mathbf{k}')}{\epsilon(\mathbf{k},\mathbf{k}')}, \tag{2.34}$$

which takes a simple form in the diagrammatic language (see Figure 2.8).

All diagrams in S are therefore included in $\sum_{\{\mathbf{k}\}} D_1(\{\mathbf{k}\})/\epsilon(\mathbf{k}_i, \mathbf{k}_j)$ where $D_1(\{\mathbf{k}\})$ is the value, *before* the summation over the momenta $\{\mathbf{k}\}$, of the diagram in S with a single separating $c_0^{(1)}$ propagator. Here $\mathbf{k}_i$ and $\mathbf{k}_j$ are the momenta of the separating propagator.

Since we can construct diagrams contributing to $\delta\rho(\mathbf{k})$ with the same building blocks and the same construction rules as for diagrams contributing to the pair potential, it is easy to find an expression involving only $\delta\rho_i$ and $c_0^{(1)}$ that includes all reducible diagrams contributing to the pair potential. An example of such an expression is

$$\frac{1}{\Omega}\sum_{\mathbf{k},\mathbf{k}'} \delta\rho_a^{NL}(\mathbf{k}, \mathbf{R}_a) c_0^{(1)}(\mathbf{k}, \mathbf{k}') \delta\rho_b^{NL}(\mathbf{k}', \mathbf{R}_b).$$

Since both $\delta\rho_a(\mathbf{k})$ and $\delta\rho_b(\mathbf{k})$ have an arbitrary number of $\chi_0^{(1)}$ loops on the leg with momenta $\mathbf{k}$, though, this expression amounts to screening the $c_0^{(1)}$ interaction twice. To take care of this, we can write instead

$$\phi_{red}(\mathbf{R}_a, \mathbf{R}_b) = \frac{1}{\Omega}\sum_{\mathbf{k},\mathbf{k}'} \delta\rho_a^{NLU}(\mathbf{k}, \mathbf{R}_a) \frac{c_0^{(1)}(\mathbf{k}, \mathbf{k}')}{\epsilon(\mathbf{k}, \mathbf{k}')} \delta\rho_b^{NLU}(\mathbf{k}', \mathbf{R}_b), \tag{2.35}$$

where $\delta\rho_a^{NLU}(\mathbf{k}, \mathbf{R}_a)$ is defined as the sum of all diagrams in $\delta\rho_a^{NL}(\mathbf{k}, \mathbf{R}_a)$ without $\chi^{(1)}$ loops on the leg with the $\mathbf{k}$ momentum (U stands for unscreened). To each reducible diagram contributing to the pair potential corresponds a diagram in (2.35), and vice versa.

For homogeneous systems, we have

$$\begin{aligned}
\chi^{(1)}(\mathbf{k}, \mathbf{k}') &= \chi^{(1)}(k')\delta_{\mathbf{k},-\mathbf{k}'}, \\
c_0^{(1)}(\mathbf{k}, \mathbf{k}') &= c_0^{(1)}(k')\delta_{\mathbf{k},-\mathbf{k}'}, \\
\frac{1}{\epsilon(\mathbf{k}, \mathbf{k}')} = \frac{1}{\epsilon(k)} &= \sum_i (c_0^{(1)}(k)\chi_0^{(1)}(k))^i \\
&= \frac{1}{(1 - c_0^{(1)}(k)\chi_0^{(1)}(k))},
\end{aligned}$$

and

$$\delta\rho_{a,b}(\mathbf{k}, \mathbf{R}_{a,b}) = \rho_{a,b}(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{R}_{a,b}},$$

so that (2.35) simplifies to

$$\phi_{red}(\mathbf{R}_{ab}) = \frac{1}{\Omega}\sum_{\mathbf{k}} \delta\rho_a^{NL}(\mathbf{k})c_0^{(1)}(k)\epsilon(k)\delta\rho_b^{NL}(-\mathbf{k})e^{i\mathbf{k}\cdot\mathbf{R}_{ab}},$$

which is precisely equation (2.12); we have shown that equation (2.12) contains exactly the reducible diagrams from equation (2.33). Now we need to show that the prefactors of these diagrams also agree.

2.5.2 Diagrammatic prefactors and symmetries

A diagram D is said to possess a symmetry of order m if it is possible to cut m legs from a given χ_0 or c_0 and obtain m identical cut-down parts. Each such symmetry contributes a factor $1/m!$ to the total prefactor of the diagram, as compared to an equivalent asymmetrical diagram.

In order to obtain this "equivalent asymmetrical diagram", we replace V_{ext} by $\tilde{V}_{ext} = \sum_{j=1}^{n+1} v_j$ in the original problem, where $n+1$ is the number of external legs of diagram D. We then consider the diagram $\tilde{D}$ that is identical to D, apart from its external legs which are all connected to different v_j. By construction, diagram $\tilde{D}$ can have no symmetry.

Contributions to diagram $\tilde{D}$ can be obtained, in equation (2.33), by replacing V_{ext} by any of the v_j and by calculating the prefactor P_j of the density diagram obtained by removing v_j from $\tilde{D}$. The prefactor $P_{\tilde{D}}$ of $\tilde{D}$ is therefore

$$P_{\tilde{D}} = \frac{1}{n+1}\sum_{j:\substack{\text{external}\\ \text{leg}}} P_j, \tag{2.36}$$

Since there is no symmetry in the diagram, it is relatively straightforward to find its prefactor: each $\chi_0^{(i)}$ or $c_0^{(i)}$ contributes $i!$ to the global prefactor. Therefore all the P_j are

equal, the $(n+1)$ factors cancel out, and we simply get $P_{\tilde{D}} = P_j$. By a similar argument, we obtain

$$P_{\tilde{D}} = P_j = P_{\tilde{D}_a} P_{\tilde{D}_b},$$

where $\tilde{D}_a$ and $\tilde{D}_b$ are the separated diagrams. Asymmetric reducible diagrams therefore have the same prefactor in (2.33) and (2.12).

This result can be extended to symmetrical diagrams in a straightforward manner: if the two identical diagrams obtained by cutting two legs from a given χ or μ in a symmetrical reducible diagram, they cannot contain the separating interaction line. Therefore all symmetries are contained within the separated diagrams. Since each symmetry contributes a factor $1/m!$ to the global prefactor, m being the number of identical branches involved in this symmetry, the symmetry contributions to (2.12) and (2.33) are the same.

We have therefore derived the result that there is a one-to-one correspondence between reducible diagrams contributing to the pair potential [in equation (2.33)], and diagrams contained in equation (2.12). Since the prefactors of these diagrams in each expression also agree, equation (2.12) allows the calculation of the pair potential associated with all reducible diagrams.

2.6 Generalizations of the partial sum approach: an interpretation in terms of parameter path integrals

In this section we discuss a simple interpretation of equation (2.11) in terms of path integrals in parameter space. This interpretation illustrates the simplicity and generality of the SSPM. We use this interpretation to derive two generalizations of the methods introduced above: namely, the case of magnetic perturbations and that of many-center

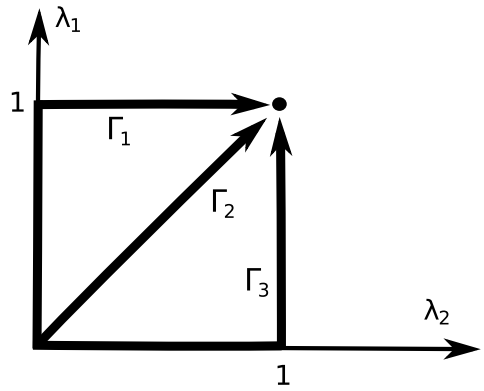

Figure 2.9: Three simple trajectories leading from $(0, 0)$ to $(1, 1)$. Each trajectory corresponds to a different order in which two perturbations can be turned on. Trajectories Γ_1 and Γ_3 have segments following an axis, where only one of the perturbations is turned on.

interactions.

2.6.1 Parameter path integral

Consider the Hamiltonian $H = H_0 + H_1 + H_2$, where H_0 is assumed to be a well-understood Hamiltonian (such as that of the electron gas), and H_1 and H_2 are simple perturbations to this Hamiltonian (for example, the potentials due to the presence of hydrogen ions in the electron gas). Suppose for definiteness that we are interested in the ground state energy of H. We can continuously transform H_0 into the Hamiltonian $H = H_0 + V_1 + V_2$ by introducing a two parameter family of Hamiltonians $H_{\lambda_1,\lambda_2} = H_0 + \lambda_1 V_1 + \lambda_2 V_2$. Typical perturbative approaches use $\lambda_1 = \lambda_2$. However, there is an infinite number of trajectories in the (λ_1, λ_2) plane which lead from H_0 to H. Three particularly simple trajectories are illustrated in Figure 2.9.

In the coupling constant integration scheme (see Section 2.1.2), the ground state energy of H can be recovered by an integral, along either path, of the expectation value of the perturbing Hamiltonian.

Trajectories Γ_1 and Γ_3 have segments following the axis. On the axis only one of the perturbations is turned on, and the corresponding Hamiltonian has higher symmetry than the general Hamiltonian H. We suppose here that the intermediate Hamiltonians $H_{1,0}$ and $H_{0,1}$ are therefore easier to solve than the full Hamiltonian H. The three Γ_i correspond to three ways to reach the final Hamiltonian H: by perturbing $H_{0,0}$ with $H_1 + H_2$, by perturbing $H_{1,0}$ with H_2, or finally by perturbing $H_{0,1}$ with H_1. This provides us with three different estimates for the ground state energy of H. If each perturbation is treated exactly, these three estimates are equal. However, if each perturbation is treated approximately, each Γ_i results in a different estimate E_i for the ground state energy.

The ground state energy of the Hamiltonian H can be expressed as a functional expansion in powers of H_1 and H_2. Suppose we are able to solve the Hamiltonian $H_{1,0}$ exactly, and that we treat perturbation H_2 to n^{th} order in perturbation, following trajectory Γ_3. The resulting expression (E_3) would contain all functional expansion terms which are of order at most n in H_2. Similarly, trajectory Γ_1 could be used to obtain an expression (E_1) containing all terms which are of order at most n in H_1. Adding the result from these two estimates would result in a double counting of the terms which are of order at most n in both perturbations. If one treats the third expression (E_2) to include terms of order at most n in both constraints, we can resolve the double counting problem by choosing a particular combination of the three estimates: $E_{SSP} = E_1 + E_3 - E_2$. This estimate takes into account all terms which are of order at most n in *either one* of the parameters.

This linear combination actually corresponds to a trajectory in parameter space, which is shown in Figure 2.10. We will refer to this trajectory as a Z-Box.

This trajectory, with each asymmetric branch treated to linear order, corresponds to the SSP method described above. From the description presented here, generaliza-

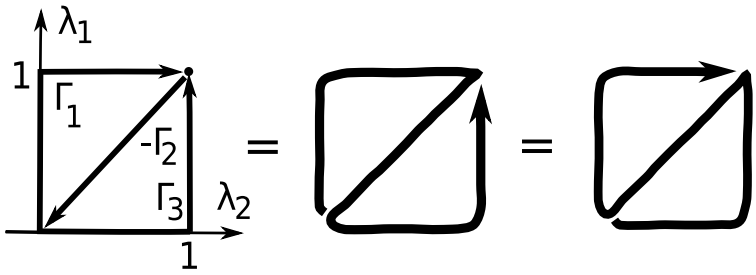

Figure 2.10: A combination of the paths Γ_1, Γ_2, and Γ_3 which ensures a high degree of error cancellation. The path $Z = \Gamma_1 - \Gamma_2 + \Gamma_3$ is symmetric with respect to permutation of the perturbations and can therefore be followed in two alternate, apparently asymmetric ways.

tions are rather straightforward: higher-order corrections can for example be obtained by treating each branch to higher order in response, always keeping the terms which are of order at most n in *either one* of the parameters along Γ_2. We provide below detailed generalization to a magnetic perturbation in an electron gas, and to many-center interactions.

2.6.2 Magnetic perturbations and the RKKY interaction

The interaction between magnetic perturbations (e.g., nuclear spins or magnetic impurity atoms) and an electron gas can be modeled by the Hamiltonian [51]

$$H_{int,\lambda} = \lambda J \sum_{i,j} f(\mathbf{r}_i - \mathbf{R}_j)\mathbf{S}^j \cdot \vec{\sigma}^i,$$

where $\mathbf{r}_i$ and $\vec{\sigma}^i$ are the position and spin operators for electron i, while $\mathbf{R}_j$ and $\mathbf{S}^j$ are the position and spin of the j^{th} magnetic perturbation source.

The Hellman-Feynman theorem reads

$$\begin{aligned} F_1 - F_0 &= \int_0^1 d\lambda \langle H_{int,1} \rangle_\lambda \\ &= J \int_0^1 d\lambda \sum_j \int d\mathbf{r}\, f(\mathbf{r} - \mathbf{R}_j)\mathbf{S}^i \cdot \mathbf{m}(\mathbf{r})_\lambda, \end{aligned} \tag{2.37}$$

where the local magnetization $\mathbf{m}$ is given by

$$\mathbf{m}(\mathbf{r})_\lambda = \left\langle \sum_{\mathbf{i}} \delta(\mathbf{r_i} - \mathbf{r}) \overrightarrow{\sigma}^{\mathbf{i}} \right\rangle_\lambda .$$

By analogy with the induced density case, we can describe the perturbation using

$$V_j(\mathbf{r}) = J \sum_i f(\mathbf{r} - \mathbf{R}_i) S^i_j.$$

Assuming a nonmagnetic unperturbed state,

$$\begin{aligned} m_i(\mathbf{r})_\lambda = & \int d\mathbf{r} \chi_{ij}(\mathbf{r}, \mathbf{r}') \lambda V_j(\mathbf{r}') \\ & + \int d\mathbf{r}' d\mathbf{r}'' \chi^{(2)}_{ijk}(\mathbf{r}, \mathbf{r}', \mathbf{r}'') \lambda^2 V_j(\mathbf{r}') V_k(\mathbf{r}'') + \cdots \end{aligned}$$

If we have two magnetic perturbing sources (a and b), we find an effective interaction of the form

$$\begin{aligned} \phi(\mathbf{R}_a, \mathbf{R}_b, \mathbf{S}^a, \mathbf{S}^b) = & \int d\mathbf{r} V^a(\mathbf{r}) \cdot \mathbf{m}^b(\mathbf{r}, \mathbf{R}_b, \mathbf{S}^b) \\ + V^b(\mathbf{r}) \cdot \mathbf{m}^a(\mathbf{r}, \mathbf{R}_a, \mathbf{S}^a) - & \int d\mathbf{r} d\mathbf{r}' \chi_{ij}(\mathbf{r}, \mathbf{r}') V^a_i(\mathbf{r}) V^b_j(\mathbf{r}'). \end{aligned} \tag{2.38}$$

Here, as before, V^i describes the potential associated with the perturbing source i and $\mathbf{m}^i(\mathbf{r}, \mathbf{R}_i, \mathbf{S}^a)$ is the magnetization induced at $\mathbf{r}$ by the presence of a single perturbation of spin S^i at R_i.

In the particular case of pointlike magnetic perturbations [$f(\mathbf{r}) = \delta(\mathbf{r})$], the effective interaction between the magnetic perturbations takes the simple form

$$\begin{aligned} \phi(\mathbf{R}_a, \mathbf{R}_b, \mathbf{S}^a, \mathbf{S}^b) = & JS^a \cdot \mathbf{m}^b(\mathbf{R}_a, \mathbf{R}_b, \mathbf{S}^b) \\ + JS^b \cdot \mathbf{m}^a(\mathbf{R}_b, \mathbf{R}_a, \mathbf{S}^a) - & J^2 \chi_{ij}(\mathbf{R}_a, \mathbf{R}_b) S^a_i S^b_j. \end{aligned} \tag{2.39}$$

This expression, which goes beyond quadratic response [but does not include corrections of the form (2.12)], could be used to improve upon the standard RKKY potential , which takes into account only the linear order in response [8].

2.6.3 Many-center potentials

Since the sources of the perturbations in equations (2.11) and (2.12) have not been specified, a straightforward way of treating many-particle interactions in this formalism is to take either or both sources to be an ensemble of particles. This might be especially appropriate for problems involving the diffusion of well formed molecules. In the hydrogen problem, this could also be used to study the molecule-molecule interactions near or beyond the onset of metallization.

Many-center interactions can be treated in a more symmetric way through conventional response theory (see, e.g. [52]), but also through a SSPM approach. If many perturbations are to be considered, many more simple paths can lead from the unperturbed to the final Hamiltonian. If we suppose that we can solve only the Hamiltonians with a single perturbation 'turned on', a simple generalization of path Z from figure 2.10 comes to mind: on each trajectory, one first turns on a single perturbation, solves the resulting Hamiltonian, then treats the remaining perturbation at fixed order in response. If we have N perturbations, this will result in N distinct trajectories. As in the pair potential case, different paths lead to different estimates, and include different perturbation terms. A simple strategy is to add the estimates arising from integration along all the distinct paths. To avoid double counting, one then needs to add $N-1$ times the result from the perturbation along the diagonal (corresponding to Γ_2 in the pair potential case). The trajectory in parameter space is therefore a simple generalization of trajectory Z in Figure 2.10. The corresponding expression for the many-body potential is, at lowest order,

$$\phi^{(N)}(\{\mathbf{R}\}) \simeq \frac{(N-2)!}{\Omega^{N-2}} \sum_{i=1}^{N} \sum_{\substack{\{\mathbf{k}_j\} \\ j\neq i}} \chi_{[i]}^{(N-2)}(\mathbf{k}_1, \ldots, \mathbf{k}_{i-1}, \mathbf{k}_{i+1}, \ldots, \mathbf{k}_N) \prod_{j\neq i} V_j(\mathbf{k}_j) e^{i\mathbf{k}\cdot\mathbf{R}_j}$$
$$- \frac{(N-1)(N-1)!}{\Omega^{N-1}} \sum_{\{\mathbf{k}_i\}} \chi^{(N-1)}(\mathbf{k}_1, \ldots, \mathbf{k}_N) \prod_{j=1}^{N} V_j(\mathbf{k}_j) e^{i\mathbf{k}_j\cdot\mathbf{R}_j}. \tag{2.40}$$

Here $\chi_{[i]}^{(N-2)}$ is the $(N-2)^{\text{th}}$ response function the system $H + H_i$. Together with the relation $\chi^0(\mathbf{k}) = \rho^{(0)}(-\mathbf{k})/\Omega$, this equation reduces to the SSP result (equation 2.11) for $N = 2$. Equation (2.40) provides a way to obtain an expression that is exact up to N^{th} order in perturbation, requiring only the explicit knowledge of response functions up to $(N-1)^{th}$ order. It requires the same quantities $\chi_{[i]}^{N-2}$ as for the pair potential calculations at the same order in the perturbations. Therefore, once the $\chi_{[i]}^{N-2}$ have been calculated for all the components in the system, arbitrary N-center potentials can be calculated from Equation (2.40). For this reason, SSP could be used as an effective bookkeeping technique even if the response functions $\chi_{[i]}^{N-2}$ are calculated perturbatively, without using the simplicity of the intermediate problems.

This is to be contrasted with the alternate generalization to many-center interactions presented in [5]. This alternate generalization avoids higher-order response functions of the perturbed systems and expresses the many-body potential in terms of densities induced by subsets of particles:

$$\phi^{(N)}(\{\mathbf{R}\}) \simeq \frac{1}{(N-1)\Omega} \sum_{j=1}^{N} \sum_{\mathbf{k}} V_j(\mathbf{k}) \delta\tilde{\rho}_{[j]}(-\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{R}_j}$$
$$- \frac{1}{(N-1)\Omega^{N-1}} \sum_{\{\mathbf{k}_i\}} \chi^{(N-1)}(\mathbf{k}_1, \ldots, \mathbf{k}_N) \prod_{j=1}^{N} V_j(\mathbf{k}_j) e^{i\mathbf{k}_j\cdot\mathbf{R}_j}. \tag{2.41}$$

Here $\delta\rho_{[j]}(\mathbf{k})$ is the density induced by all involved particles *except* j and $\delta\tilde{\rho}_{[j]}(\mathbf{k})$ is the component of this density that depends on the position of all $N-1$ particles involved.

The latter definition is required simply to avoid double counting the potentials involving less than N particles.

Equation (2.41) also provides a way to obtain an expression that is exact up to N^{th} order, requiring only the explicit knowledge of the $(N-1)^{th}$ order response function and reduces to equation (2.11) for $N = 2$. On the other hand, it requires knowledge of the density induced by N groups of $N-1$ particles, which rapidly becomes more difficult when $N > 2$ and does not have particular symmetry. The bookkeeping required to keep track of "the component of this density that depends on the position of all $N-1$ particles involved" is more involved than that required to calculate equation (2.40), which simply involves induced densities and response functions of perturbed systems.

For these reasons equation (2.40) appears to be a much more promising generalization of (2.11) than equation (2.41).

CHAPTER 3

IDENTIFYING AND SOLVING CONSTRAINT PROBLEMS

3.1 Constraint problems

Constraint problems involve the simultaneous satisfaction of a number of conditions, or *constraints*, on the values of a set of variables. Remarkably, the simplicity of the constraints offers no guarantee of the simplicity of the resulting constraint satisfaction problem. Boolean satisfaction, the phase problem in diffraction imaging, and the determination of protein geometry and ground states of glasses, provide many examples of difficult problems involving the satisfaction of simple constraints. Consider the problem of finding the ground state energy E_0 of an Ising spin glass, where the system state is described by the vector $\mathbf{s} = \{s_i\}_{i=1,\ldots,N}$ taking values in $\{-1, 1\}$. The Hamiltonian is $H = \sum_{i \neq j} J_{ij} s_i s_j$, and the coupling constants $J_{ij} = J_{ji}$ take fixed values in the set $\{-1, 0, 1\}$. The lowest possible value for the ground state energy of this Hamiltonian is $E_b = -\sum_{i,j} \left| J_{ij} \right|$. This energy is reached only if all pair energies are as low as possible, i.e., if $J_{ij} s_i s_j = -1$ for every pair (i, j) with nonzero coupling. Even though it is easy to find a state with a particular pair (i, j) 'properly aligned' with $J_{ij} s_i s_j = -1$, the problem of determining whether all pair energies can be minimized *simultaneously* can be very challenging. Determining whether $E_b = E_0$ is a typical difficult constraint satisfaction problem.

Because constraint problems occur in many different contexts, it is not surprising that many different approaches have been developed to study and solve them. Ideas from information theory, statistical physics, computer science, and optimization theory all contributed to our understanding and solving of constraint problems. The work presented here follows instead from a *geometrical* interpretation of constraint problems.

Before we proceed, we need to establish a few definitions. A constraint on a set X can be interpreted as a function $f_C : X \to \{\text{True}, \text{False}\}$. For a particular $x \in X$, a constraint is *satisfied* if $f_C(x)$ is True. Otherwise, it is *unsatisfied* or *violated*. The constraint set C can also be defined by the subset of X where the constraint is satisfied: $C = f^{-1}(\text{True})$. In this work we identify a constraint with its constraint set C. A constraint is *satisfiable* if $C \neq \emptyset$. In this case, the constraint satisfaction problem is to find an element in C. These definitions also apply when multiple constraints $\{C_i\}_{i=1,\ldots,n}$ are to be satisfied simultaneously, since the simultaneous satisfaction of the $\{C_i\}_{i=1,\ldots,n}$ is in itself a constraint, with constraint set $\cap C_i$.

In the case of the Ising spin glass example mentioned above, the set X of all possible configurations could be taken to be $\{-1, 1\}^N$. Each nonzero coupling J_{ij} can be interpreted as a constraint C_{ij}, with function

$$f_{C_{ij}}(\mathbf{s}) = \begin{cases} 1, & \text{if } J_{ij} s_i s_j = -1 \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

This constraint is *satisfied* if spins s_i and s_j are properly aligned to minimize their pair energy. The *constraint set* is simply is the set of vectors $\mathbf{s}$ in which spins i and j are properly aligned. The problem of determining whether E_b is equal to E_0 is *satisfiable* if and only if there is a configuration where all the pairs are properly aligned.

This is not the only way to express this problem as a constraint problem. In the following we will use methods which require that the space X is a vector space. To achieve this, on can for example allow vectors $\mathbf{s}$ to take values in $\mathbf{R}^N$, and modify the constraints to read

$$f_{C_{ij}}(\mathbf{s}) = \begin{cases} 1, & \text{if } J_{ij} s_i s_j = -1 \text{ and } |s_i| = |s_j| = 1 \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

Both formulations of the spin glass problem lead to identical solutions.

In physics one is often interested in solving the more general problem of finding a low energy state $\mathbf{x}$ of a Hamiltonian $H(\mathbf{x}) = \sum_i \phi_i(\mathbf{x})$, consisting of a sum of many simple terms ϕ_i. This problem is not strictly speaking a constraint problem, as one is interested in the lowest energy solution even when it is impossible minimize individually all terms $\phi_i(\mathbf{x})$, and the ϕ_i can take a continuous range of values. Such problems are often called *restraint* problems, where the ϕ_i are thought of as *restraints*, favoring certain configurations but not strictly constraining $\mathbf{x}$ to a particular set of configurations.

Restraints problems are a generalization of constraint problems, and in practice many constraint problems are treated as restraint problems, as in simulated annealing (see Section 3.3.2, below). We will provide examples below (see, e.g., Section 5.5) of how restraint problems (and, more generally, optimization problems) can also sometimes be expressed as constraint problems.

A convenient and useful way of representing constraint (or restraint) problems is through *factor graphs*. A factor graph has nodes representing both variables and constraints. Edges connect constraints to the variables they constrain. An example is provided in Figure 3.1, where circles represent variables and squares represent constraints. Properties of the resulting graph, in particular the existence and typical length of loops and planarity, are often relevant to the applicability and performance of different algorithms.

As mentioned above, we will be particularly interested in problems where the satisfaction problem for $\cap C_i$ is challenging even though each of the C_i is comparatively easy to satisfy. Many methods for constraint satisfaction attempt to take advantage of the particular structure of such problems.

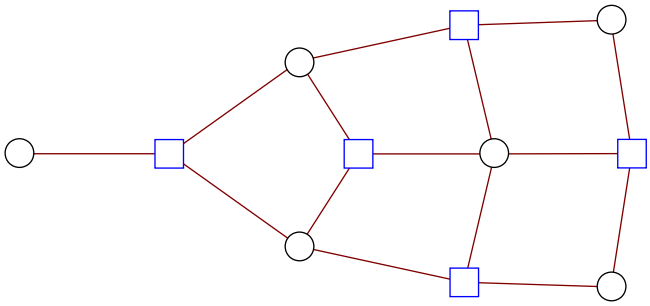

Figure 3.1: An example of a factor graph for a constraint satisfaction problem. Here variables are represented by circles and constraints by squares. A link between a constraint and a variable indicates that the satisfaction of the constraint may depend on the value of the variable. The properties of the graph, such as the connectivity of the nodes, planarity, or the existence and size of loops, can have a strong influence on the difficulty of a problem.

3.2 An overview of some of Nature's constraint satisfaction methods

Many natural systems can find solutions to complicated problems. Such systems act both as a source of motivation and inspiration for the constraint satisfaction community. In physics, finding the state of lowest (free) energy of a collection of interacting particles can be interpreted as a constraint satisfaction problem. The capacity of many materials to spontaneously adopt a crystalline structure is a sign that such low free energy states can often be found simply through the action of particle interactions and thermal fluctuations. The thermal noise plays an important role in this process: systems that are cooled down rapidly from a hot liquid state often end up in a glassy or amorphous state. Systems that are cooled more slowly tend to be able to find lower free energy states. In the case of crystals, a slower cooling schedule typically results in fewer defects in the crystal. Warming up and slowly cooling down crystalline samples to reduce defect

concentration is referred to as annealing. Simulated annealing, described below, is a numerical method inspired from this procedure.

The field of biology provides many more examples of efficient searching processes, among which evolution might be the most spectacular. Various organisms have also developed (through evolution!) their own searching strategy, from fish hunting using Lévi walks [53] and ants looking for food [54], to bacteria following chemical gradients and proteins 'searching for' binding sites on DNA [55]. Cognitive processes provide many additional examples of searching strategies, many of which are still not understood.

3.3 An overview of computational constraint satisfaction methods

3.3.1 Complete methods, and DPLL

The simplest approach to many discrete constraint problem is the so called *brute force* approach, in which one enumerates all variable assignments and verifies each assignment is a solution to the problem at hand. The program terminates when a solution is found or if all configurations have been tested. This approach has the advantage of being *complete*: it will always find a solution if there is one, if it is given enough time to complete its execution. An efficient bookkeeping strategy to achieve this is backtracking. Backtracking uses a search tree whose root corresponds to a state in which no variable is assigned. A variable is chosen and a node is created and attached to the root for each possible value this variable can take. Now each of the nodes thus created can serve as a root to a subtree describing the configurations of the remaining variables. If this procedure is repeated N times, where N is the number of variables, the lower level of the tree corresponds to configurations where all variables have been assigned a value.

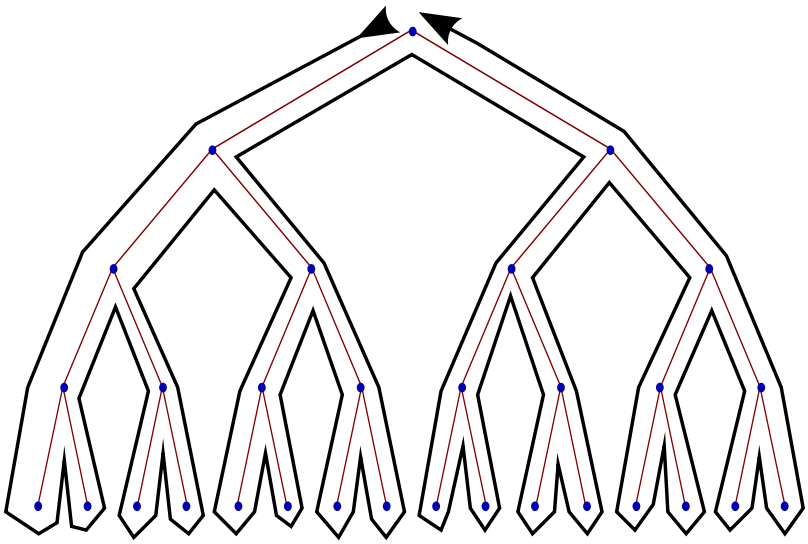

Figure 3.2: Illustration of the order in which the nodes of a search tree are accessed in backtracking, for a binary problem with 4 variables and no solution. The $2^4 = 16$ variable configurations lie on the lowest level of the graph, and are each visited once. The intermediate nodes correspond to partial variable assignments, and are each visited three times.

The brute force backtracking procedure works by assigning values to the variables in turn, starting from the ones closest to the root. When all the variables have been assigned, the resulting configuration is tested for satisfaction. If the configuration solves the problem, the search can be interrupted. If the configuration does not solve the problem, the variable assignment needs to be changed. In backtracking, the change is always performed on the last variable that was assigned and whose alternate configurations have not been explored. The order in which different nodes of the search tree are explored is illustrated in Figure 3.2.

The backtracking approach is simply a bookkeeping strategy and does not result in performance improvement. However, backtracking is commonly used in much more efficient algorithms, which do not explore all configurations yet are complete.

One of the most successful complete algorithms based on backtracking is the Davis-Putnam-Logemann-Loveland (DPLL) algorithm. DPLL is designed to solve Boolean constraint satisfaction problems in the conjunctive normal form (CNF) [56, 57], a particularly important form of constraint problem, which we will further discuss in section 5.1.

DPLL improves upon the brute force by performing a few additional checks each time a variable is assigned. These checks are used to identify branches of the search tree that can be discarded, either because they are guaranteed to contain no solution, or because the existence of a solution in the discarded branch guarantees the existence of a solution in a remaining tree.

More specifically, DPLL checks for constraints that force a variable to take a specific value (in which case branches corresponding to alternate values are discarded), and for constraints that are guaranteed to be violated, no matter what further assignments are made (in which case the algorithm discards the subtree starting at the current node).

DPLL also checks for 'pure' variables. A pure variable has a preferred value, which is at least as good as all other values in solving all constraints in which the variable occurs, for any configuration of the other variables. That is to say that if a constraint is satisfied with the pure variable set to a non preferred value, it will also be solved if the pure variable is changed to its preferred value. In such cases it is safe to set the pure variable to its preferred value and discard all branches corresponding to non-preferred values. Even though solutions might be discarded along with non-preferred branches, the existence of a solution is still guaranteed in the remaining tree.

The checks performed by DPLL are designed to detect variables that can be assigned safely. The order in which variables should be assigned therefore depends on the result

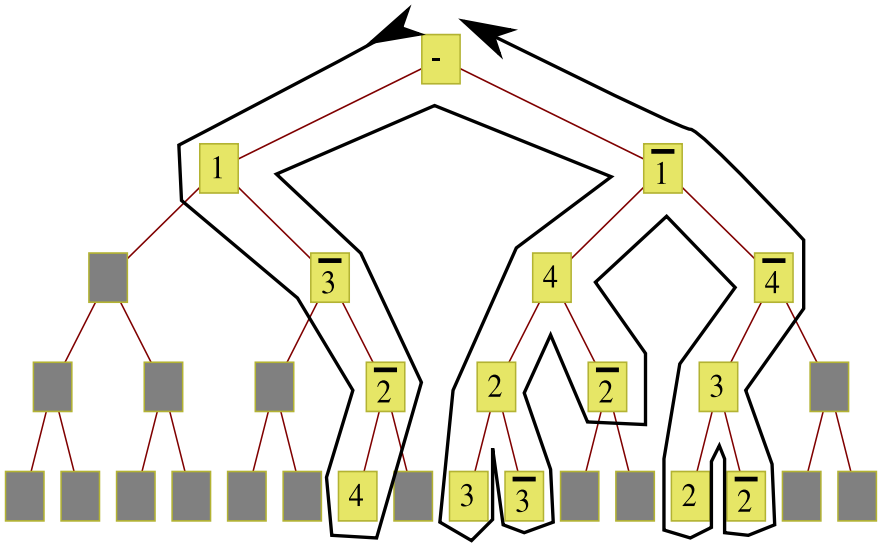

Figure 3.3: Illustration of a possible order in which the nodes of a search tree are accessed in the DPLL algorithm, for a binary problem with 4 variables and no solution. The $2^4 = 16$ variable configurations lie on the lowest level of the graphs, and the search starts at the root of the tree (here on top). Gray nodes are never visited. The number labeling of a node correspond to the variable who was assigned at this node, and an overline represents a 'False' assignment. Notice that the order in which variables are assigned is different in each subtree.

of these tests, and can vary from branch to branch in the tree. Fortunately this can be achieved easily in the backtracking procedure (see Figure 3.3).

Finally, the algorithm checks for constraints that will be satisfied no matter what further variable assignments are made. These constraints do not have to be considered again and can be temporarily 'removed' from the problem until the algorithm backtracks past the point where their satisfaction was guaranteed,.

The DPLL algorithm, which was introduced more than 40 years ago, is still the basis of most complete solvers for Boolean satisfaction, one of the most well-studied constraint satisfaction problem we will discuss in more detail in section 5. However, the

combined requirements of keeping track of the configurations explored and of systematically exploring a large number of configurations can be computationally prohibitive. In some cases using a heuristic search strategy can be much more efficient than complete methods. Incomplete heuristic methods cannot ascertain the unsatisfiability of a problem. However, when the existence of a solution is guaranteed or almost certain, such incomplete methods are often appropriate.

3.3.2 Simulated annealing and cost function optimization approaches

A popular incomplete approach to constraint satisfaction problems involves reformulating the constraint problem as an optimization problem through the definition of a cost function $E : X \rightarrow \mathbf{R}$. The cost function can be simply the number of unsatisfied constraints for a given configuration. It can also be refined to take into account the amount by which each constraint is violated, particularly for continuous problems. When a solution to the constraint problem exists, the minimum of the cost function is 0 and configurations with minimum cost are solutions to the constraint problem. The constraint problem can therefore be solved by minimizing E. Many general-purpose optimizers, such as simulated annealing, can be used for this purpose.

In simulated annealing, an initial random configuration is selected, and random updates are accepted or rejected stochastically depending on their effect on the cost function. Because the cost function is usually easy to calculate, and the optimizer can be used out-of-the-box, this approach is quite simple, and may be an excellent choice for problems that are not too large or hard.

However, many approaches outperform simulated annealing, particularly for the solution of large or difficult constraint problems. A reason for this is apparent in the way information is used in this approach.

The optimizer only relies on single number E as a figure of merit for updating the state of the system. This number averages over the satisfaction status of all constraints. Even though information about the satisfaction status of each constraint is available, this information is not passed on to the optimizer. The optimizer is therefore lacking important information which could be used to determine an appropriate update.

A simple way to modify the cost function approach to include this information is to concentrate on the constraints that are violated, and make updates based on the particular structure of the constraint problem at hand. This is the approach used by many heuristic approaches to the Boolean satisfaction problem, which we will discuss in section 5.1.

3.3.3 WALKSAT and the focused algorithms

Using the particular constraint structure of a problem requires problem-specific development. The efficiency of tailored algorithms often depends quite sensitively on the particular choice of the update mechanism. For well-studied problems, such as Boolean satisfaction, update mechanisms were found that resulted in orders-of-magnitude performance improvement upon pure cost function approaches such as simulated annealing. In many cases, such approaches also use a stochastic element to ensure ergodicity of the search process, that is, to ensure that the search process, given enough time, will visit all configurations in the search space. Ergodicity in turn guarantees that the search will eventually reach a solution, if a solution exists. However, it does not provide a bound on the number of steps necessary to find a solution. WALKSAT [58], and more generally

focused algorithms [59], update only variables involved in constraints that are not satisfied. These methods balance greedy updates with random updates: greedy updates never increase the number of unsatisfied constraints, whereas random updates allow this number to increase.

Various other algorithms are based on a similar balance between greediness and randomness [60]. Interestingly, the efficiency of the search in most of these approaches depend sensitively on the balance between greediness and stochasticity [59]. We should finally mention recent results using the chainsat algorithm [61], which updates single variables in unsatisfied constraints much like local searches, but succeeds in finding solutions without requiring random uphill moves.

3.3.4 Statistical approaches, and message-passing algorithms

In the recent years much effort has been devoted to the understanding of constraint problems from a statistical perspective. If constraints are drawn randomly from an ensemble of constraints, it is often possible to obtain rigorous results on the typical behavior of problems in the limit where the number of constraints is large. Typical quantities of interest are the existence and number of solutions and the existence of one or many clusters of solutions in a typical problem.

Significant progress in the resolution of constraint satisfaction problems came from the application of methods inspired by statistical mechanics to individual problems. The idea is to consider the number of unsatisfied constraints as an effective Hamiltonian. The ground states of this Hamiltonian are the solutions to the constraint problem. If one assumes a Boltzmann distribution of configuration probability, one can use statistical mechanics tools to evaluate expectation values for individual variables. These expectation

values are referred to as marginal probabilities, or simply *marginals*. Low temperature expectation values then correspond to averages of the value of a variable over all solutions of a problem.

If the goal is simply to find a single solution to a Boolean problem (where variables take values in the set $\{1, -1\}$, corresponding to {True,False}), a possibility is to look for variables whose expectation value is very close to either 1 or -1. These *most polarized* variables take the same value in a large fraction of solutions. The *decimation* heuristic works by evaluating the expectation values, assigning the most probable value to the most polarized variables, and simplifying the problem accordingly. This method can be applied recursively on smaller and smaller decimated problems until the statistical approach fails to provide reliable, very polarized expectation values. The resulting decimated problem is then typically solved using a local search method such as Walksat.

This decimation approach relies on an efficient evaluation of the expectation values of individual variables. This is where message-passing algorithms such as belief propagation (BP) (see, e.g., [62]) and survey propagation (SP) [63, 64] come in.

The expectation value of a variable i over the solutions of a problem P can be determined by the probability distribution of its neighboring variables, that is, variables appearing in constraints involving i. However, this probability distribution is usually not available until the problem is solved. In the *cavity method*, one relates the expectation value of a variable to expectation values calculated in *cavity problems*, which we will take here to mean problems obtained from P by removing single constraints. In a tree-like problem (that is, a problem whose factor graph is a tree), removing a constraint breaks down the problem into independent tree-like problems whose solution spaces can be sampled independently, greatly simplifying the problem. In a tree-like graph, the probability that variable i has value x_i is given by

$$p_i(x_i) = C_i \prod_{\alpha \in N(i)} \mu_{\alpha \to i}(x_i). \tag{3.3}$$

In this expression $N(i)$ is the set of constraints in which variables i is involved, that is, the neighbors of variable node i in the factor graph. The quantities $\mu_{\alpha \to i}(x_i)$ are the expectation values of the constraint f_α in the cavity problem with α removed and variable i fixed to the value x_i. The normalization constant C_i is chosen so that $\sum_{x_i} p_i(x_i) = 1$. Note that in the remainder of this section we will use Latin indices i, j to refer to variables, and Greek indices α, β to refer to constraints.

The expectation value $\mu_{\alpha \to i}(x_i)$, can be obtained from cavity expectation values of the other variables j involved in constraint α. Note that the satisfaction of constraint α often depends only on a subset of variables $N(\alpha)$. If we take $\mathbf{x}_\alpha = \{x_j\}_{j \in N(\alpha)}$ to be the values taken by variables in $N(\alpha)$, the domain of f_α can be restricted to $\mathbf{x}_\alpha$, so that $f_\alpha(\mathbf{x}_\alpha)$ is simply the satisfaction of f in configuration $\mathbf{x}$. We these definitions in mind and still assuming a tree-like problem, we can write

$$\mu_{\alpha \to i}(x_i) = \sum_{\substack{x_j, \\ j \in N(\alpha) \setminus i}} f_\alpha(\mathbf{x}_\alpha) \prod_{j \in N(\alpha) \setminus i} \mu_{j \to \alpha}(x_j). \tag{3.4}$$

Here $\mu_{j \to \alpha}(x_j)$ is the probability that variable j has value x_j in the cavity problem where constraint α was removed. This can be calculated using equation (3.3), applied to the cavity graph:

$$\mu_{i \to \alpha}(x_i) = C_{i \to \alpha} \prod_{\beta \in N(i) \setminus \alpha} \mu_{\beta \to i}(x_i). \tag{3.5}$$

Here again $C_{i \to \alpha}$ is a normalization constant ensuring that probabilities add up to one.

The $\mu_{\alpha \to i}$ and $\mu_{i \to \alpha}$ can be interpreted as messages from constraints to variables and

variables to constraints, respectively, summarizing information about the state of the problem 'upstream' of $\alpha \to i$ or $i \to \alpha$, respectively, the arrow determining the direction of the flow of information.

The success of BP is partly due to the fact that equations (3.4) and (3.5) can be solved very efficiently on a tree: by starting from the leafs of the tree and updating a message only if all the upstream messages have been calculated, messages need to be passed only twice on each edge, with only one message sent in each direction. This strategy is similar to the transfer matrix method in physics and the dynamical programming approach of computer science: the order in which computations are carried ensures an optimal use of information.

Equations (3.4) and (3.5) are exact only on tree-like diagrams, as they neglect correlations between the different neighbors of constraint α in the cavity problem where α has been removed. The assumption that this factorization holds approximately in non-tree problems leads to the BP heuristic.

The point of the cavity approach becomes clear here: the variables in $N(\alpha)$ are involved in the same constraint, and their correlations cannot be neglected. However, if one removes the constraint α from the problem, the variables in $N(\alpha)$ are no more likely to be correlated than any two variables taken at random in the problem. If the average correlations are expected to be small, the variables in the cavity graph can be taken to be approximately uncorrelated, as if the problem was a tree.

Equations (3.4) and (3.5) can be solved iteratively and self-consistently, starting from a randomized initial distribution of the $\mu_{i\to\alpha}$ and $\mu_{\alpha\to i}$. This self-consistent BP heuristic has been quite successful at calculating marginals in a variety of contexts, even though there is no guarantee that the BP heuristic converges or has a unique solution. In fact,

in cases where the solution landscape is highly clustered, that is, where multiple islands of solutions are separated by oceans of unsatisfied assignments, BP often has multiple stable fixed points, which indicates that the approximations used have failed (since the marginals have a well-defined value, at least one of the fixed points must be wrong!). SP is based on a similar self-consistent, message-passing idea as BP, but uses statistical mechanics tools to take into account explicitly the clustered nature of the solution space. The expectation values calculated by SP are taken over a different ensemble, giving equal weight to clusters independent of their size. One way to interpret the relation between SP and BP is that SP effectively calculates the expectation values of marginals over the ensemble of solutions to the BP equations: in short, SP is BP applied to BP. More detailed information on BP and SP can be found in References [62] and [63], respectively.

When it was introduced, SP solved problems such as random Boolean k-satisfiability (see Chapter 5) to unprecedented number of variables near the critical region, that is, where the ratio of number of constraints to variables is such that about half the problems have a solution. Even though it was since discovered that local search algorithms such as WALKSAT could exhibit comparable performance if their parameters were optimized carefully (see, e.g, [59]), SP still appears to be the most efficient solver for many randomly generated problems.

3.3.5 Branch and bound

The branch and bound approach is designed for optimization problems and provides a way to relax the completeness requirement while maintaining a rigorous approach. Given a function f to minimize over a domain D, the branch and bound approach sub-

divides the domain D into smaller subdomains D_i over which the function f can be bounded below. A typical approach is to separate domains by hyperplanes and find linear functions $\tilde{f}_i$ that are lower bounds to f over the subdomains D_i. The bounding functions $\tilde{f}_i$ can then be minimized using linear programming to obtain a lower bound to f over the D_i.

The branch and bound approach also requires an upper bound for the global minimum, which is typically the lowest value of f found at a given time over the whole domain D.

If the lower bound for f on a subdomain D_i is higher than the upper bound for the minimum on the whole domain, the global minimum is guaranteed not to lie in D_i. One can therefore continue the search on the reduced domain $D \setminus D_i$. A significant advantage of the branch and bound approach is that the search can be interrupted at any time and still provide rigorous upper and lower bounds on the global optimum. Another advantage is that this approach can be used in conjunction with local searches, which can be very efficient in obtaining a global upper bound and can benefit from information about excluded subdomains.

3.3.6 Divide and conquer

Finally, a complete method worth mentioning is the divide and conquer approach, which can be extremely efficient when it is applicable. It is applicable to problems that can be broken down into a collection of subproblems similar to the initial problem, and where the solution of the global problem can be recovered from the solutions to the individual subproblems.

A typical example of problem that can be solved using divide and conquer is the sorting of a list **L** of n real numbers. Sorting is equivalent to finding an array **s** whose i^{th} element gives the order, in a sorted list, of the ith element of the input list. Constraints in this problem specify that **s** is a permutation of the integers $(1, 2, \ldots, n)$, and that for each index pair (i, j), $s_i > s_j$ if $L_i > L_j$. It is faster to sort two short lists and merge them into a larger sorted list than to sort the composite list using straightforward methods. By subdividing a long list in shorter and shorter sublists and merging the sorted lists one can gain considerable computing time. In this example it is clear that sorting the sublists is providing useful information about the final solution: the sorting of each sublists has a unique solution, and the elements in a sorted sublist will appear in the same order in the final sorted list.

However, many difficult problems require the simultaneous satisfaction of multiple weak constraints, that is, constraints that have a large number of satisfying assignments. It is in general impossible to determine which of these assignments will lead to a global solution without information about the requirements of all the other constraints. In such cases, divide and conquer is not applicable.

CHAPTER 4

SOLVING CONSTRAINT PROBLEMS WITH PROJECTIONS: THE DIFFERENCE MAP, AND DIVIDE AND CONCUR

4.1 Using projections to solve constraint problems

4.1.1 Projecting to constraint sets

The approach to constraint satisfaction we present here is in a way similar to the local approaches discussed in section 3.3: it also proceeds by repeatedly updating a tentative variable assignment, using a simple set of update rules, until a solution is found. The form of the update rules, however, is quite different. One crucial difference is that the updates are generated deterministically, rather than stochastically. Another difference is that the variables are updated synchronously, rather than one at time. Finally, the update rules do not depend only on the local properties of the problem, as is the case in local searches.

The critical conceptual difference between this approach and local search is the use of *projection operators* in the definition of the updates. We have emphasized in section 3.1 that a constraint on a set X can be identified by a constraint set $C \subseteq X$. When X has a Euclidean structure, we define the *projection operator* $P_C : X \to C \subseteq X$ by requiring that $P_C(x) = y$ is the element of C that minimizes the distance $\|x - y\|$ (see Figure 4.1).

This specifies the action of P_C, except for those values of x where many $y \in C$ minimize the distance to x. In practice, such values form a set of measure zero, and the particular choice of a distance-minimizing $y \in C$ was not observed to have any influence for our current purpose. It is therefore not necessary to pay particular attention to

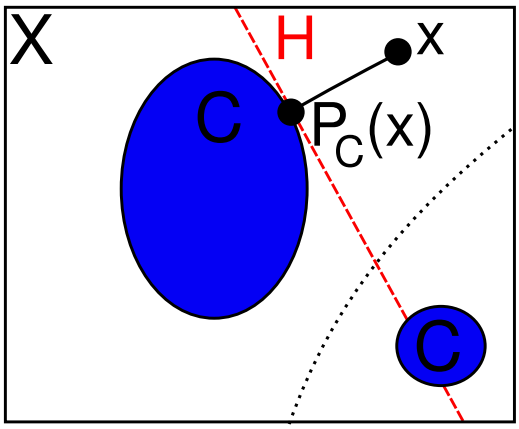

Figure 4.1: The projection of a point x to constraint C. Even though C is neither convex nor connected, the operator P_C is well-defined and continuous everywhere except on a set of measure zero, indicated by a dotted line. The red dashed line indicates H, the hyperplane orthogonal to $x - P_C(x)$.

these ambiguous cases, and the projections $y = P_C(x)$ can simply be defined whichever distance-minimizing y is the most convenient.

Projection operators are commonly used to solve problems involving *convex* constraints [65]. In such problems, finding the solution is not necessarily challenging: the challenge is to obtain fast convergence towards this solution. For that purpose, one can use the fact that x and $y = P_C(x)$ provides considerable information about the constraint C. In particular, one can identify the hyperplane H orthogonal to $x-y$ and going through y. The convex constraint set C is confined to the side of H opposite to x. If C is smooth, H can be seen as a linear local approximation to C, even when $\dim(H) \neq \dim(C)$ and C is not convex. In this work, we will be particularly interested in difficult problems involving the simultaneous satisfaction of *non*convex problems.

In a metric space, the minimal distance of a point to the constraint can be used as a cost function describing how much a constraint is violated. Using a steepest descent approach to the constraint set using this cost function would lead to the same result as

Figure 4.2: An illustration of the difference between the distance minimizing projection y and steepest descent approach z in the case where the constraint set comprises points whose altitude is below sea level, and the starting point is x. The projection y is the point that minimizes the distance between x and the shoreline. The Mississippi, which follows the steepest slope of the landscape, ends up at point z. The projection y does not depend on the landscape outside the constraint surface. (satellite image of the Mississippi river modified from NASA).

projecting, but would be possibly more costly since many gradients would need to be calculated.

However, cost functions can also be quite different from the distance cost function. A natural example is provided by a constraint on the energy in an arbitrary potential $V(\mathbf{r})$. The cost function $c(\mathbf{r}) = V(\mathbf{r}) - E$ can be quite different from the distance cost function, and a steepest descent in $c(\mathbf{r})$ can be quite different from a projection operator, as is illustrated on Figure 4.2.

The projection operator is blind to the energy landscape outside the constraint, and depends only on the boundary of the constraint set. This means the projection operator is insensitive to local minima in the cost function. In the example from Figure 4.2, the projection provides much more information about the constraint space than the gradient

at x.

A drawback of projection approaches, compared with gradient-based approaches, is that they depend on the existence of efficient projection operators, which are not always available. To speak figuratively: if one is lacking any information about the sea shore location, following a river downhill might be an appropriate strategy to reach the ocean.

When available, efficient projection operators provide useful information about constraint sets $\{C_i\}_{i=1,\dots,n}$. They can therefore be used to obtain efficient update rules for finding a point in the intersection $\cap C_i$, provided such an intersection exists. The simplest example of such update rules is the *alternating projection* scheme, defined by the map $x_{i+1} = P_1 \circ P_2 \circ \cdots \circ P_n(x_i)$. This approach can be shown to converge to a point in the intersection when such a point exists and the constraints are convex. However, it is easy to see that this map is prone to stagnation (fixed points that do not correspond to solutions) when the constraints are nonconvex.

Finding an iterated projection scheme that uses the efficient convergence properties of projection-based maps while achieving an efficient search for difficult nonconvex problems is more challenging. The *difference map* [66] was introduced to solve such problems in cases where there are only two constraints to be satisfied.

4.2 The difference map

Given *two* constraints C_1 and C_2, their associated projections P_1 and P_2, and an initial tentative assignment x_0, we are looking for an iterated map $x_{n+1} = f(x_n)$ that will allow us to find points in $C_1 \cap C_2$. More specifically, we will require that the knowledge of any *fixed point* of f allows to retrieve a solutions to the problem. Reasonable requirements

on the map f, if it is to be useful at finding solutions, are that:

1. f is computationally simple
2. The displacement $f(x) - x$ does not depend on the choice of the origin
3. Fixed points exist if there is a solution
4. All attractive fixed points correspond to solutions of the problem
5. 'Near misses' of the constraint sets are not locally attractive
6. There are at most very few attractive limit cycles that do not correspond to solutions
7. The search process explores a significant fraction of the relevant search space, and ideally the whole search space, without requiring restarts,
8. The dynamics of the iterated map rapidly 'finds' its fixed points

Remarkably, given two constraints C_1 and C_2 with their associated projection operators P_1 and P_2 (which are assumed to be simple), the difference map [66] always satisfies the first four conditions and, in many cases, satisfies all eight. It is defined by the update rules

$$\begin{aligned} \mathbf{x}_{n+1} = f(\mathbf{x}_n) &= \mathbf{x}_n + \beta\left(P_1\left(f_2(\mathbf{x}_n)\right) - P_2\left(f_1(\mathbf{x}_n)\right)\right) \\ f_i(\mathbf{y}_n) &= (1+\gamma_i)P_i(\mathbf{x}_n) - \gamma_i \mathbf{x}_n \qquad i = 1, 2, \end{aligned} \tag{4.1}$$

with $\gamma_1 = -1/\beta$ and $\gamma_2 = 1/\beta$. This particular choice of parameters is explained in section 4.8.1 below and in reference [66]. The parameter β is usually chosen to be a real number in $[-2, -.5] \cup [.5, 2]$. However, the difference map would satisfy the fist four conditions for arbitrary real or complex values of β. More discussion on the choice of β can be found in section 4.8.2

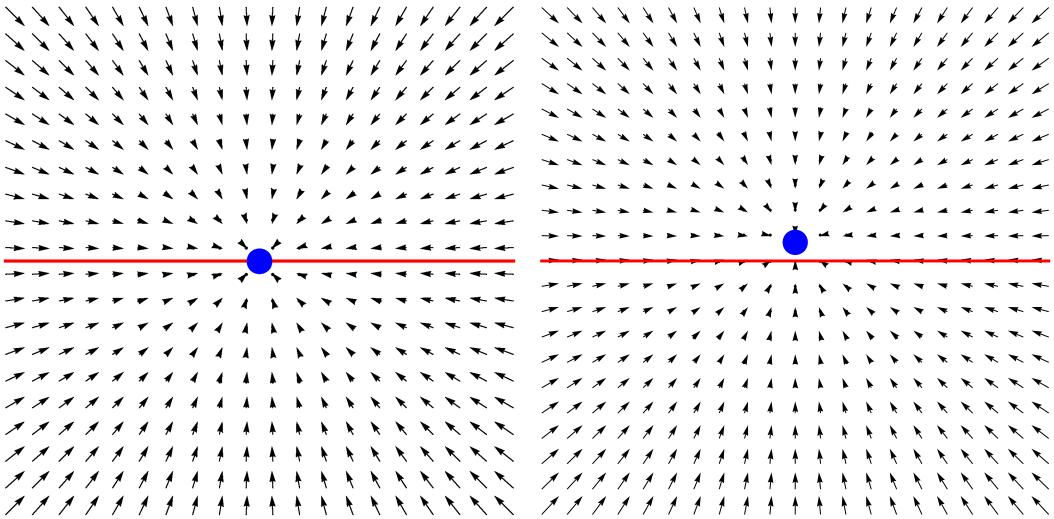

Figure 4.3: The vector field associated with the alternating projection scheme for an isolated solution, and an isolated near solution. Under a small perturbation, the area surrounding an isolated attractive fixed point remains attractive; therefore the near solution is also attractive.

The solution itself is always a fixed point of this map. It should be emphasized that the difference map can have many additional fixed points that are not a solution to the problem, but it is straightforward to verify that for any fixed point $\mathbf{x}^*$, $P_1\left(f_2(\mathbf{x}^*)\right) = P_2\left(f_1(\mathbf{x}^*)\right)$ is a solution that can easily be retrieved from the fixed point. The difference map therefore satisfies condition number 4. The fact that there are typically many fixed points for each solution is actually crucial to the success of this projection approach to constraint satisfaction.

If a solution is associated with an isolated, attractive fixed point of the map, an arbitrary small perturbation of the constraint sets destroying this solution creates an attractive near solution, violating condition 5 (see Figure 4.3). On the other hand, if there exists a continuous variety of fixed points associated with a solution, an infinitesimal perturbation of the constraint sets can turn this fixed points variety into an unstable variety (see Figure 4.4).

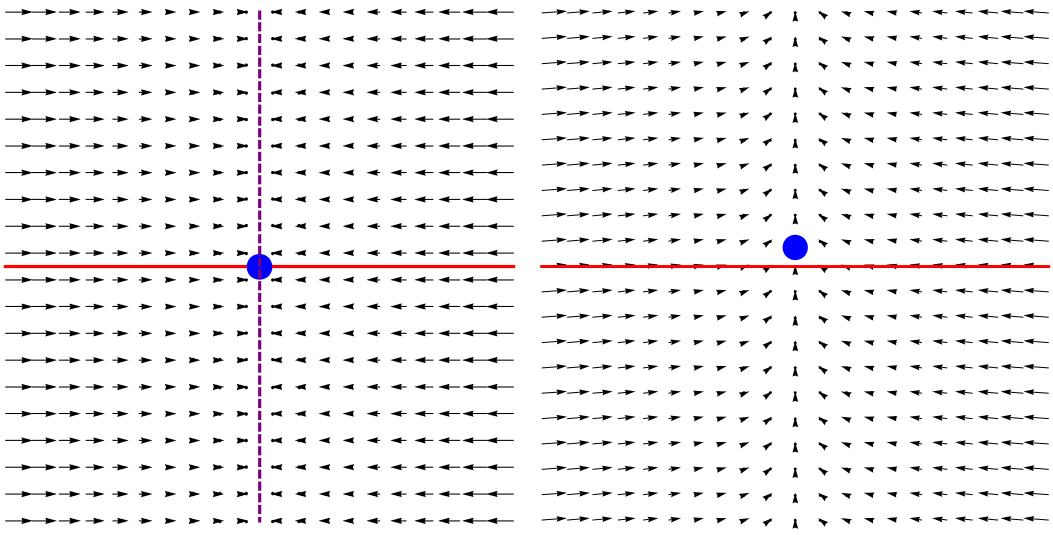

Figure 4.4: The vector field associated with the difference map for an isolated solution, and an isolated near solution. The purple dashed line represents fixed points. A small perturbation of a system with a fixed point variety can destroy all fixed points, a desirable feature for search algorithms.

The generalization of the difference map to an arbitrary number of constraints is not straightforward. Despite this limitation, the difference map was used to solve a variety of problems. In many cases, difficult problems (such as phase retrieval in diffractive imaging) are naturally expressed as two constraint problems [66, 67]. Many more problems involving more than two constraints were reformulated as two constraint problems through the introduction of auxiliary variables and solved using the difference map [68]. In the following section, we describe an approach to reformulate arbitrary constraint problems in terms of only two constraints, whose intersection can then be sought using the difference map.

4.3 Divide and concur

The formulation of a problem in terms of constraints is not unique.

The *divide and concur* approach uses a simple scheme to reformulate many-constraint problems as problems involving only two constraints in a different search space. It will therefore be useful to distinguish between different formulations of the same problem. When applying divide and concur to a particular constraint problem, we will refer to the initial constraints as *primary* constraints, and to the constraints in the reformulated problem as *secondary* constraints.

Broadly speaking, the idea is to allow each variable to take multiple values, and enforce *separately* that all constraints must be satisfied, and that all variables must have a single definite value.

More precisely, given N *primary* constraints expressed as subsets of K, we first define the product space K^N, consisting of N copies (or *replicas*) of K. In this extended space it is therefore possible to assign a copy of the search space to each primary constraint. If the different replicas are allowed to take different values, each constraint can be satisfied independently of the requirements from other constraints; the problem has been divided into individual, simple subproblems. Imposing the simultaneous, independent satisfaction of all primary constraints in the enlarged space constitutes a (secondary) constraint, which we call the *divide* constraint. Satisfying the divide constraint is not enough to solve the initial problem, since this constraint allows variables to take multiple values. The role of the other secondary constraint, or *concurrence* constraint, is simply to impose that all copies of K must be equal. Simultaneous satisfaction of the 'divide' and 'concur' constraints yields a solution to the problem. Yet, given projections to the primary constraints, the projections to the divide and concur projections take a particularly simple form. Given an element $\mathbf{y} = \mathbf{x}^{(1)} \oplus \mathbf{x}^{(2)} \oplus \cdots \oplus \mathbf{x}^{(N)}$ of K^N, the projections are

$$P_D(\mathbf{y}) = P_1(\mathbf{x}^{(1)}) \oplus P_2(\mathbf{x}^{(2)}) \oplus \cdots \oplus P_N(\mathbf{x}^{(N)}), \tag{4.2}$$

for the divide constraint, and

$$P_C(\mathbf{y}) = \bar{\mathbf{x}} \oplus \bar{\mathbf{x}} \oplus \cdots \oplus \bar{\mathbf{x}}, \tag{4.3}$$

for the concurrence constraint. This projection replaces the value of each replica by the average value $\bar{\mathbf{x}}$ of all the replicas. In defining the concurrence projection, different weights λ_i may be assigned to different constraints, i.e., $\bar{\mathbf{x}} = \sum_i(\lambda_i \mathbf{x}^{(i)})/\sum_i \lambda_i$.

Note that the extended search space, as defined above, can be unnecessarily and prohibitively large if the problem involves a large number of constraints. In practice, the satisfaction of a constraint i often depends only on variables spanning a subspace K_i of K. In such a case, the divide and concur search space may be reduced to $K_1 \oplus K_2 \oplus \cdots \oplus K_N$, with important memory and performance gains. If a variable is involved in N_i constraints, there will be N_i copies, or *variable replicas*, of this variable in the reduced search space. This is illustrated using the factor graph representation in Figure 4.5. In Figure 4.5(b), each variable from the factor graph (a) was replaced by a number of replicas, all linked by a concurrence constraint. Figures 4.5(c) and (d) are simple rearrangements of the rendering of the graph from (b). They illustrate how the individual primary constraints decouple under the divide constraint, and how the individual concurrence constraints decouple under the (global) concurrence constraint.

4.3.1 Outline of a divide and concur solver

When solving a problems with the $D - C$ approach, the key conceptual steps are

1. Expressing the problem as a constraint satisfaction problem
2. Defining efficient projection operators to each constraint.

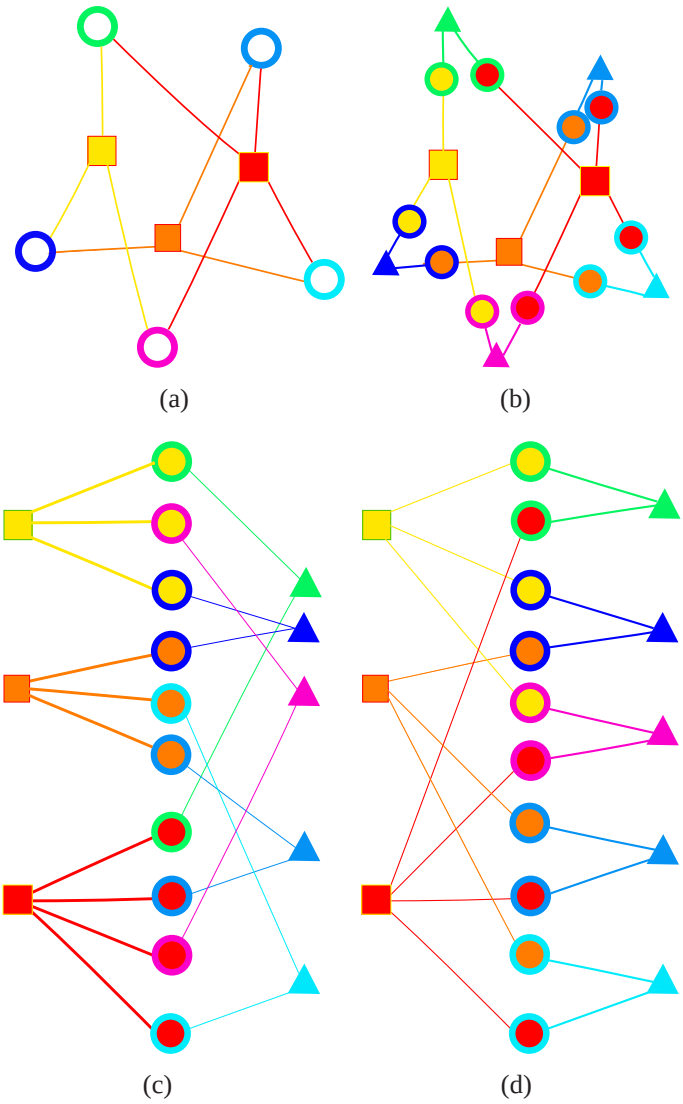

Figure 4.5: Four equivalent ways of representing the same problem. In these factor graphs circles represent variables, while squares represent primary constraints. Triangles represent (secondary) concurrence constraints for individual variables. A link between a constraint and a variable indicates that a the satisfaction of the constraint may depend on the value of the variable. (a) is the factor graph of a problem with three constraints and five variables. (b), (c) and (d) are different arrangements of the factor graph in the enlarged search space, illustrating: (b) the relation of the extended space factor graph with the initial factor graph, (c) the simplicity of the 'divide' constraint and (d) the simplicity of the 'concur constraint

In an implementation of the $D - C$ approach, the following functions need to be defined:

1. an initializer, which reads the input, and initializes data structures
2. an iterator, which updates the value of the tentative solution (or iterate) using the divide and concur projection
3. a concurrence projection operator
4. a divide projection operator, which depends on projection operations to individual constraints
5. projection operators to the individual constraints.

These projection operators to individual constraints are the only elements which vary significantly from problem to problem. Implementation notes and/or pseudocode for the individual projection operators Proj_i to various constraints are provided in section 4.4. We provide in the remainder of this section pseudocode for the other three steps, which are common to all constraint problems.

The choice of the order in which variables are stored in memory may affect the efficiency of the algorithm, as variables are accessed in specific order by each projection operator. The choice made in this section is convenient, but not necessarily optimal. Variables are simply laid out in a one dimensional array in the order in which they appear in the problem. If the problem involves N_v variables, and a total number of variable occurrences N_l, the array has dimension N_l. It is necessary to define two data structures to keep track of the position of the different occurrences of each variables, and of the beginning point of each constraint. This is shown in Pseudocode 4.1.

x =random vector of length N_l;
$B[i]$=beginning position of constraint i. $B[i] = B[i-1]$+number of variables involved in constraint i-1;
$P[i][j]$ = position at which variable i appears for the j^{th} time in the problem;

Pseudocode 4.1: Initialization of $D - C$

The next step is to define the two fundamental projections in divide and concur. With our current choice of data structure, the divide constraint accesses the x array in order. It is shown in Pseudocode 4.2.

input : I
output: O = projection of I to the divide constraint
given : number of constraints N_C, positions in x of beginning of constraints $B[i]$
while $i = 1, \ldots, N_C$ **do**
  $O[B[i] : B[i+1]] = \text{Proj}_i(I[B[i] : B[i+1]])$
endw

Pseudocode 4.2: Divide(I)

In the concurrence constraint, variables are accessed in a different order. In large problems this can be a cause of slowing down due to cache size limitations. Pseudocode 4.3 illustrates the concurrence constraint.

input : I
output: O = projection of I to the concurrence constraint
given : $P[i][j]$, positions of occurrences of variable i in I
for i *in* $1, \ldots, N_V$ **do**
  a = average of values taken in I by all occurrences of variable i;
  for *each position* $p = P[i][j]$ *where variable* i *occurs* **do**
    $O[p] = a$
  endfor
endfor

Pseudocode 4.3: Concur(I)

Once all projection operators are defined the iterations can be simply performed using the difference map update rules. The process is illustrated by Pseudocode 4.4.

The number of iterations needed to find a solution, in the applications we considered in this dissertation, varied from one (in the case of linear orthogonal constraints) to

```
input  : initial vector x
output: search result
given  : parameter β, maximum iteration number I, error threshold T
while The number of iterations is smaller than I do
    t = (1 + 1/β) Divide(x) − 1/β x;
    x_C = Concur(t);
    t = (1 − 1/β) Concur(x) + 1/β x;
    x_D = Divide(t);
    t = x_C − x_D;
    e = norm(t);
    if e < T then
        return success: solution is x_sol = x_D = x_C
    endif
    x = x + βt;
endw
restart, or;
return no solution found after I iterations
```

Pseudocode 4.4: Iterating in $D - C$

many millions. The appropriate choice of a cutoff I therefore depends on the expected difficulty of the problem, as is the case for most heuristic search strategies.

4.3.2 Simple variations on divide and concur

Up until now we described the most straightforward application of the divide and concur strategy, when the problem is to find the intersection of N independent primary constraints, which are all enforced independently through the divide constraints, and where the concur constraint is enforcing the concurrence of the different replicas of the variables. However, both the divide and concur constraints can be given modified roles to accommodate or simplify the treatment of different problems.

For example, the concurrence constraint can be strengthened to enforce additional constraints in addition to the variable concurrence. The additional constraint can involve individual variables, for example by requiring that all replicas concur to a binary

value. The ‘constrained concurrence’ constraint from Section 4.4 has this form. It is also possible to enforce a more global constraint on the values of the variables, and this is illustrated in the vertex cover problem in section 5.3.

Many constrained problems in physics are energy minimization problems, where the energy can be broken down into pair potentials, or few-center potentials. For definiteness we will consider only pair potentials here. Finding the minimum energy does not reduce to the independent minimization of the energy functions of individual pairs as it is usually not possible to minimize simultaneously all pair energies. We can nevertheless proceed similarly as in divide and concur and introduce replicas for each variable for each pair potential in which it is involved. The concurrence constraint in this case ensures as usual that in the solution each variable has a definite position. The divide constraint cannot be carried in a parallel manner: the energy minimization couples the different pair terms. However, the use of replicas in this case separates energetics (dealt with in the divide constraint) from geometrics (dealt with in the concurrence constraint). We illustrate this by the example of particles interacting via pairwise step potentials in section 5.5.

4.4 A projection repertoire

In our discussion of the difference map and the divide and concur approaches to constraint problems, we treated the projection operators as black boxes taking in a value and returning the projection to the appropriate constraint. However, it is not always straightforward or even possible to define efficient projection operators. Since the efficiency of the projection operators play a crucial role in the success of the difference map and divide and concur, it is fortunate that many constraints that are frequently involved in

Table 4.1: A list of discrete constraint sets in continuous space with efficient projection operators. Details on the projection operators and their scaling behavior are presented below.

Name	Parameters/Given	Input	Constraint set
Arbitrary points	S: a set of m points in $\mathbf{R}^n$	$\mathbf{x} \in \mathbf{R}^n$	$\mathbf{x} \in S$
Integers		$x \in \mathbf{R}$	$x \in \mathbb{Z}$
Binary		$x \in \mathbf{R}$	$x \in \{0, 1\}$
Primes		$x \in \mathbf{R}$	x is prime
Simplex		$\mathbf{x} \in \mathbf{R}^n$	$\mathbf{x} \in \{0, 1\}^n$, $\sum x_i = 1$
Select d	$d \in \mathbf{N}$	$\mathbf{x} \in \mathbf{R}^n$	$\mathbf{x} \in \{0, 1\}^n$, $\sum x_i = d$
Select at least d	$d \in \mathbf{N}$	$\mathbf{x} \in \mathbf{R}^n$	$\mathbf{x} \in \{0, 1\}^n$, $\sum x_i > d$
Logical OR	$\{s_i\}_{i=1,\ldots,n} \in \{\pm 1\}^n$	$\mathbf{x} \in \mathbf{R}^n$	$\mathbf{x} \in \{0, 1\}^n$, $\sum s_i x_i > 0$
Logical XOR	$\{s_i\}_{i=1,\ldots,n} \in \{\pm 1\}^n$	$\mathbf{x} \in \mathbf{R}^n$	$x \in \{0, 1\}^n$, $\sum s_i x_i \equiv 1 \mod 2$
Fixed distribution/ permutation	$\{\mathbf{z}_i\}_{i=1,\ldots,n} \in \mathbf{R}^{mn}$	$\{\mathbf{x}_i\}_{i=1,\ldots,n} \in \mathbf{R}^{mn}$	$\{\mathbf{x}_i\}$ is a permutation of $\{\mathbf{z}_i\}$

constraint problems have simple projection operators. In this section, we provide multiple examples of such constraints together with implementation notes for the projection operators. Tables 4.1 and 4.2 provides an overview of the different constraints, Figure 4.6 provides a geometrical illustration of some discrete constraint spaces.

4.4.1 Discrete constraints

Arbitrary points

This is the most general finite discrete constraint set. The constraint set S comprises m arbitrary points in $\mathbf{R}^n$. Given an input $\mathbf{x} \in \mathbf{R}^n$, it is always possible to calculate the distance from $\mathbf{x}$ to each of the points in S, and identify $\mathbf{y}$, the point in S with minimal distance to $\mathbf{x}$. The projection of $\mathbf{x}$ onto S is simply $P(\mathbf{x}) = \mathbf{y}$.

This brute force approach to projection is quite general and scales as $O(mn)$, but

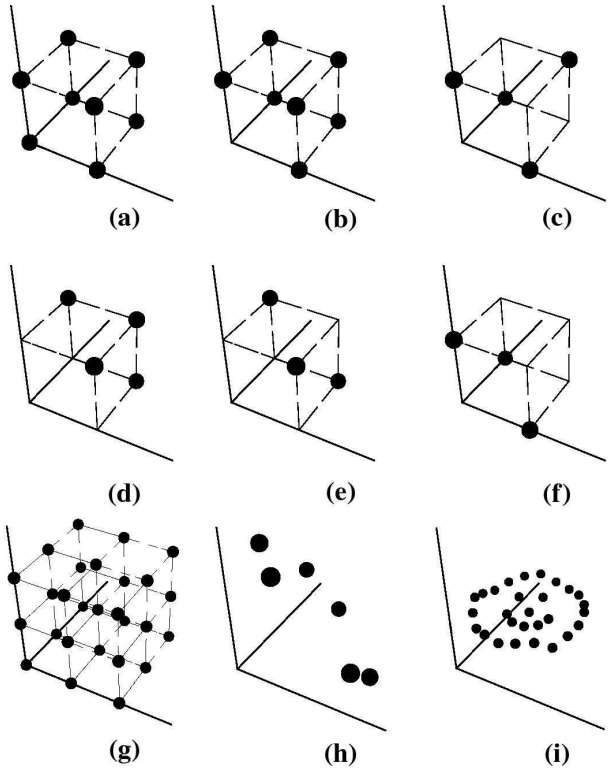

Figure 4.6: A geometrical representation of discrete constraints on variables triplets. Dots indicate points in the constraint set. The constraint sets illustrated are (a) binary (b) logical OR (c) logical XOR (d) select-at-least-2 (e) select-2 (f) simplex (g) integers (h) fixed distribution (i) arbitrary positions ($d = 3$)

not always the most efficient. If the set S has structure, it is often possible to exploit the structure to achieve a more efficient projection; in most of the constraints discussed below the implementation of efficient projection operators makes use of the symmetries of the constraint set.

Even in cases where S does not possess structure it is often possible to improve a lot on the brute force approach, by partitioning the space $\mathbf{R}^N$ in multiple regions. Identifying the region in which $\mathbf{x}$ lies reduces the list of candidate candidate nearest points (see Figure 4.7). An efficient way of partitioning the space is by using k-dimensional (KD)

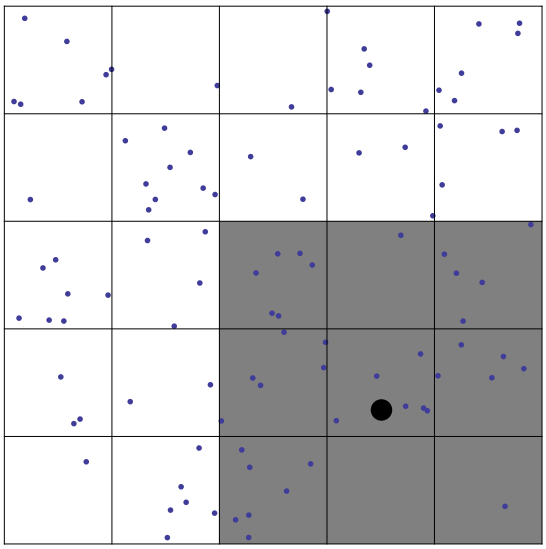

Figure 4.7: An example of a partition of space for projecting to a set of m randomly generated points in the plane. The dark dot represents the starting point for the projection, and the blue points represent the constraint set. Separating the space into boxes narrows down the candidate nearest points in the constraint space. Areas outside the shaded area, for example, can be excluded at once without inspection of individual points, since they are further than one square diagonal from the starting point, and the square in which the initial point lies is not empty. More elaborate partition methods such as KD trees [69] reduce the cost of individual projections to $O(\log m)$ for fixed dimension, once the initial partition has been set up.

trees [69]. For fixed number of dimensions, building the tree structure costs $O(m \log m)$. Once the structure is built, every additional projection costs only $O(\log m)$.

Integer constraint

Rounding, the projection to the set of integers $\mathbb{Z}$, is usually simple to implement. Since c99, the 1999 revision of the iso c programming language standard, it is implemented in the math.h library as y=round(x) for doubles, or y=roundf(x) for floats. Previous to c99 rounding of x could be implemented by using floor (x+.5).

Binary constraint

Projecting to a set of two real numbers (typically {0, 1} or {±1}) is useful in many discrete problems. It is also straightforward to implement. In c, for example, y=(x>.5) ? 1:0.

Prime number constraint

Finding the nearest prime to a real number x exactly takes at most $O(x^{2/3}\log^{\alpha} x)$ computational time and $O(x^{1/3}\log^{\beta} x \log\log x)$ memory space, where α and β are constants. This can be achieved for example by using the prime counting function $\pi(x)$ [70]. However, it seems likely that methods relying on probabilistic primality checking [71] could speed this up considerably.

Simplex constraint

The simplex constraint space contains all n-vectors whose coordinates are permutations of $(1, 0, 0, \cdots, 0)$. It is used in problems where a choice has to be made amongst n options, such as colors in graph coloring problems or numbers in sudoku or Latin squares.

The implementation, which scales as O(n), requires finding the index i of the maximum value in the input **x**, then defining the output **y** using

$$y_j = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise.} \end{cases} \tag{4.4}$$

Select-*d* and select-at-least-*d*

The select-*d* and select-at-least-*d* constraints are a generalization of the simplex constraint where more than one choice amongst n are allowed. Their constraint sets are formed of binary n-vectors who contain exactly d or at least d ones, respectively. They are used in graph problems such as minimal vertex covers (where d is the number of selected nodes) and error-correcting codes (where d is the expected number of error bits).

The projection to these constraints requires finding s, the value of the d^{th} largest element in a list of length n. This can be performed, in the typical case, in time $O(n)$. The optimal implementation depends on the ratio of d and n and of the requirements of the particular application (see chapter 8 in Numerical Recipes[69]). Once s is located, one can simply loop through the input array **x** and compare each element x_i to s. Depending on the constraint and on the result of the comparison, y_i is then set to 0, or 1.

An implementation for select-*d* is shown as Pseudocode 4.5.

```
input  : {x_i}_{i,1,...,n}
output: {y_i}_{i,1,...,n}
S:=d^th largest element of {x_i};
C := 0;
for i = 1, ..., n do
    if (x_i ≥ S) then
        y_i = 1;
        C = C + 1;
    else
        y_i = 0;
    endif
endfor
while C > d do            /* If there were many i for which x_i = S */
    Select an i for which x_i = S, and set y_i = 0;
    C = C − 1.;
endw
```

Pseudocode 4.5: Projection to select-*d* constraint

In this example we had to pay attention to the possibility of ties (i.e., multiple elements in the constraint set having the same distance to the starting point) since without the final loop such a tie would have resulted in an erroneous 'projection' to a point outside the constraint set.

Logical OR constraint

The logical OR constraint on n variables requires that each variable takes a value in $\{0, 1\}$ (or in ± 1), with *at least* one 1. It is a particular case of the select-at-least-d and is used particularly in Boolean satisfaction problems and vertex cover problems.

The implementation is straightforward and scales as $O(n)$. One first projects each element to $\{0, 1\}$. If all rounded elements are zero, one then finds the index j of the largest element of $\mathbf{x}$ and sets $y_j = 1$.

Logical XOR constraint

The logical XOR constraint requires that n variables take values in $\{0, 1\}$ (or $\{\pm 1\}$), with an odd number of elements taking value 1. It is the basic constraint in XORSAT (the problem of satisfying simultaneously a number of XOR constraints), and is also used in decoding low density parity check (LDPC) codes which we will discuss in chapter 5.

The implementation is similar to logical OR. Set the output equal to the projection of the input on $\{0, 1\}^d$, and find the index j of the input element closest to 0.5. If the constraint is not satisfied, change the value of y_j. This scales as $O(n)$.

Fixed distribution or permutation

The fixed distribution or permutation constraint space contains all arrays that are a permutation of a given set of variable values. In one dimension the projection to this constraint simply requires sorting the input. Assuming the set of given values z_i is also sorted, the output y_j is set to $z_{\sigma(j)}$, where $\sigma(j)$ is the order of the input value x_j in the sorted list. This can be performed in $O(n \log n)$. In arbitrary dimension finding the optimal projection is equivalent to a bipartite matching problem and can be solved using the Hungarian algorithm [72, 73] in time $O(n^3)$.

4.4.2 Continuous constraints

Inequality

The inequality constraint requires a real variable to be larger than a fixed value. It can be simply coded in *c* as y=(x>a) ? x:a. More generally, constraints involving inequalities can usually be implemented by verifying whether they are satisfied and, if not, projecting to the corresponding equality constraint.

Amplitudes

The amplitude constraint requires that vectors or complex numbers have a specified amplitude *a*, with no regard to orientation or complex phase. It is used extensively in diffraction imaging, where the amplitudes of the diffraction pattern are measured, and the phases are unknown. It can also be used in the kissing number problem, where the center of a number of spheres is constrained to the surface of a central sphere.

Table 4.2: A list of continuous constraint sets with efficient projection operators. Details on the projection operators and their scaling behavior are presented below.

	Given	input	Constraint
Inequality	$a \in \mathbf{R}$	$x \in \mathbf{R}$	$x > a$
Amplitudes	$a \in \mathbf{R}$	$x \in \mathbf{C}^n$ or $x \in \mathbf{R}^n$	$\|\|\mathbf{x}\|\| = a$
Pair distance	$I \subset \mathbf{R}$	$\mathbf{x}_1, \mathbf{x}_2 \in \mathbf{R}^n$	$\|\|\mathbf{x}_1 - \mathbf{x}_2\|\| \in I$
Concurrence		$\mathbf{x} \in \mathbf{R}^n$	$x_i = x_j \quad \forall i, j$
Constrained concurrence	$I \subset \mathbf{R}$	$\mathbf{x} \in \mathbf{R}^n$	$x_i = x_j \quad \forall i, j,$ $x_i \in I$
Concurrence by clusters	number d of values	$\mathbf{x} \in \mathbf{R}^n$	The x_i can take at most d different values
Dot product/ Weighted sum	n vector $\mathbf{a}$ real value b	$\mathbf{x} \in \mathbf{R}^n$	$\mathbf{a} \cdot \mathbf{x} = b$
Linear equation	$m \times n$ matrix $\mathbf{A}$, $m \times 1$ vector $\mathbf{b}$	$\mathbf{x} \in \mathbf{R}^n$	$\mathbf{Ax} = \mathbf{b}$
Linear relation	$m \times n$ matrix $\mathbf{A}$	$\mathbf{x} \in \mathbf{R}^n$ $\mathbf{y} \in \mathbf{R}^m$	$\mathbf{Ax} = \mathbf{y}$
Hyperplane	dimension d	$\{\mathbf{x}_i\}_{i=1,\ldots,m} \in \mathbf{R}^{m \times n}$	The $\mathbf{x}_i$ lie on a d dimensional hyperplane
Handedness	specified handedness	$\{\mathbf{x}_i\}_{i=1,\ldots,4} \in \mathbf{R}^{3 \times 4}$	The $\mathbf{x}_i$ have the specified handedness
Fixed distribution/ histogram	Disjoint bins $I_1, \ldots, I_m \subset \mathbf{R}$ Occupations $d_1, \ldots, d_m$	$X = \{x_i\}_{i=1,\ldots,n} \in \mathbf{R}^n$ $n = \sum d_i$	bin occupation numbers are satisfied: $\#(X \cap I_i) = d_i$
Fourier amplitudes	measured Fourier amplitudes $\mathbf{f} \in \mathbf{R}^n$	$\mathbf{x} \in \mathbb{C}^n$	$\|\|\mathcal{F}(\mathbf{x})\|\| = \mathbf{f}$
Area	$d \in \mathbf{N}$	$\mathbf{x} \in \mathbf{R}^n$	$x_i \neq 0$ for d indices
Support	indices $S = \{i_k\}_{k=1,\ldots,d}$	$\mathbf{x} \in \mathbf{R}^n$	$x_i = 0$ if $i \notin S$
Energy in step potential	step potentials $\{\phi_i(r)\}_{i=1,\ldots,m}$, Energy E	$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1,\ldots,m} \in \mathbf{R}^{2nm}$	$\sum \phi_i(\|\|\mathbf{x}_i - \mathbf{y}_i\|\|) < E$

The implementation simply requires the calculation of and normalization by the norm of the vector or complex number:

$$y_i = \frac{a x_i}{||x_i||}$$

The computational complexity of this projection scales as the dimension of **x**.

Pair relative distance/exclusion/contact

For a pair of points, this constraint specifies allowed values for their relative distance. It comes up in sphere packing problems (section 5.2) and also in protein structure prediction from from NMR data.

The implementation is simple. One first finds the initial distance $r = ||\mathbf{x}_1 - \mathbf{x}_2||$. Then one calculates $\tilde{r}$, the projection of r onto the set of allowed distances. If necessary, the two points are moved along the line joining them to ensure an appropriate distance:

$$\begin{aligned} \mathbf{y}_1 &= \mathbf{x}_1 + (\mathbf{x}_1 - \mathbf{x}_2)\frac{\tilde{r} - r}{2r}, \\ \mathbf{y}_2 &= \mathbf{x}_2 - (\mathbf{x}_1 - \mathbf{x}_2)\frac{\tilde{r} - r}{2r}. \end{aligned} \tag{4.5}$$

This scales simply as the number of dimensions in the problem.

Concurrence

This constraint, imposing that a number m of variables in n dimensions have identical values, is central to the divide and concur scheme. It is also one of the simplest constraints to implement:

$$y = \frac{\sum_i \lambda_i x_i}{\sum_i \lambda_i},$$

where the λ_i are the relative weights of the different variables. This scales as $O(mn)$

Constrained concurrence

As mentioned above, it can be desirable to enforce additional constraints on the concurrence value of a variable. For example, one might require that a set of variables concur to a *discrete* value, or to within an interval I. It is straightforward to show that to achieve this constrained concurrence projection it suffices to first calculate the concurrence constraint as above, then project the resulting value to the additional constraint. This also scales as $O(mn)$.

Concurrence by clusters

The concurrence constraint imposes that n variables take a single value. The concurrence by clusters imposes that the n variables take a specified number m of different values, assuming $m < n$. The problem of projecting to this constraint set has received considerable attention as it is equivalent to the k-means problem in data clustering applications. It is usually solved approximately using a variant of Lloyd's heuristic, which is usually referred to, somewhat misleadingly, as the k-means method (see, e.g., reference [74] for a detailed description of Lloyd's heuristic). In arbitrary dimensions the use of a heuristic rather than a complete method is usually justified since an exhaustive search would require the testing of a prohibitively large number ($\simeq n^m/m!$) of cluster assignments.

However, in one dimension, the fact that variables can be ordered reduces considerably the number of configurations to be explored. An approach using this fact was proposed by Fisher (see, e.g., reference [74]), and uses dynamic programming to explore efficiently the remaining configurations. The main steps in this approach are:

- Sort the variables to obtain $\{x_0, \ldots, x_n\}$, which requires $O(n \log n)$ operations.
- Calculate the cost for all possible clusters. That is, for each contiguous interval of the sorted variables, determine the total squared distance of the points within the interval to their mean. There are $n(n+1)/2$ clusters, and the calculation of the interval distances can be calculated in $O(n^2)$ and stored for later use.
- Consider the n intervals containing x_0. For each such interval I, find the optimal bisection point. If I is to be split into two intervals, this is the splitting point that would lead to the smallest cost for the resulting two intervals. This requires a total of $O(n^2)$ steps.
- Since trisection of an interval can be reduced to two bisections, finding the optimal trisection of each interval containing x_0 now requires only $O(n^2)$ additional steps. One can now proceed to calculate optimal divisions into $4, 5, \ldots, m$ intervals, for a total of $O(n^2 m)$ steps.
- Once the optimal division into m intervals has been found, the projection can be performed by calculating the average of the points lying in each interval and replacing all values by the average of the corresponding interval.

The complete process therefore takes $O(n^2 m)$ time. This is much more efficient than a brute force approach but can still be more demanding than a heuristic approach such as Lloyd's approach.

Dot product or weighted sum

The dot product or weighted sum constraint is a simple particular case of the linear equation constraint, below. Given a vector $\mathbf{a}$ and a target b, this constraint states that $\mathbf{a} \cdot \mathbf{x} = \sum_i a_i x_i = b$. The projection of $\mathbf{x}$ onto this constraint is $P(\mathbf{x}) = \mathbf{x} + \lambda \mathbf{a}$, with

$\lambda = (b - \mathbf{a} \cdot \mathbf{x})/\mathbf{a} \cdot \mathbf{a})$. In the case of a sum with uniform weight, this projection displaces each variable by the same amount.

Linear equation

The optimal implementation of the projection to the solution of a linear equation $A\mathbf{x} = \mathbf{b}$ depends on the matrix dimension ($m \times n$) and sparseness.

In the general case, one can calculate the Moore-Penrose pseudoinverse A^+ of the matrix A. The projection $\mathbf{y}$ is then given by

$$\mathbf{y} = (1 - (A^+A))\mathbf{x} + A^+\mathbf{b}.$$

Since the pseudoinverse (and A^+A) only has to be calculated once and can then be reused for each iteration, this approach requires $O(mn)$ operations per iteration, in the limit of a large number of iterations.

When the matrix A is sparse, it can be preferable to use conjugate gradient minimization [69] rather than the pseudoinverse approach, since the pseudoinverse of a sparse matrix is not necessarily sparse. If A is not square, as will be the case for typical underdetermined systems, one can for example apply the conjugate gradient method on the normal equation $A^TA\mathbf{x} = A^T\mathbf{b}$. If A has full rank, a solution to the normal equation is also a solution to $A\mathbf{x} = \mathbf{b}$. Another option is to use the biconjugate gradients (see also reference [69]).

The solution found by the conjugate gradient method with starting point $\mathbf{x}_0$ is indeed a projection of $\mathbf{x}_0$ to the constraint space defined by $A\mathbf{x} = \mathbf{b}$. This can be verified by checking that for each iteration of the conjugate gradient method, the displacement

vector is orthogonal to vectors in the nullspace of A, and therefore to the constraint set.

Linear relation

A particular case of a linear equation constraint, used extensively in Elser, Rankenburg, and Thibault [68], occurs when two sets of variables $\mathbf{x}$ and $\mathbf{y}$ are related by a linear equation $A\mathbf{x} = \mathbf{y}$, with A an $m \times n$ matrix. This could be treated as a linear equation of the form $B\mathbf{z} = 0$ by making a composite vector of $\mathbf{x}$ and $\mathbf{y}$, and solving the linear equation

$$(A, -I)\begin{pmatrix} \mathbf{x} \\ \\ \mathbf{y} \end{pmatrix} = 0, \tag{4.6}$$

using the methods described above for the linear equation constraint. This can be simplified by the observation that the distance between the initial points $\mathbf{x}$ and $\mathbf{y}$ and points $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ satisfying the constraint is

$$d^2 = (\mathbf{x}-\tilde{\mathbf{x}})^2+(\mathbf{y}-\tilde{\mathbf{y}})^2 = (\mathbf{x}-\tilde{\mathbf{x}})^2+(\mathbf{y}-A\tilde{\mathbf{x}})^2 = \tilde{\mathbf{x}}^T\left(A^TA + I\right)\tilde{\mathbf{x}}-2(x^T-y^TA)\tilde{\mathbf{x}}+f(\mathbf{x},\mathbf{y}). \tag{4.7}$$

Minimizing this quadratic form with respect to $\tilde{\mathbf{x}}$ through conjugate gradients yields the projection to the constraint set. The scaling behavior of this projection depends on the sparsity of the A matrix. The number of iterations will be at most $O(n)$, and the number of steps per iteration scales as the number of nonzero elements of A. For this projection, it is usually *not* necessary to enforce strict convergence of the conjugate gradient search, as each iteration of the conjugate gradient provides a point on the constraint set. A lack of convergence results in a map to the constraint set that is not precisely a projection. This is to be contrasted with the linear constraint projection, where convergence is necessary to ensure that the final point lies on the constraint set.

Hyperplane

This constraint applies to m vectors $\{\mathbf{x}_j\}_{j=1,\ldots,m}$ in n dimension and requires that all vectors must lie in a d dimensional hyperplane. The projection to this constraint can be separated in three parts: determining the centroid of the cloud of points formed by the $\{\mathbf{x}_i\}$, determining the orientation of the distance-minimizing hyperplane, and projecting the $\{\mathbf{x}_i\}$ onto this plane.

The hyperplane must go through the centroid $\langle \mathbf{x}_j \rangle$ of the $\{\mathbf{x}_j\}$: since the constraint set is invariant with respect to uniform translations, the centroid of the projected vectors $\langle \mathbf{y}_j \rangle$, is equal to the centroid of the input vector $\mathbf{x}$. This being established, we can translate all vectors so that $\langle \mathbf{x}_j \rangle = 0$ and proceed to determine the directions along which of the cloud formed by the $\{\mathbf{x}_i\}$ is 'thinnest'. For convenience we will parametrize the hyperplane by $n - d$ vectors $\{\mathbf{v}_i\}_{i,1,\ldots,n-d}$ orthogonal to the hyperplane and to each other, and minimize the projection distance with respect to these directions. The total squared distance D^2 between $\{\mathbf{x}_j\}$ and $\{\mathbf{y}_j\}$ is therefore simply

$$D^2 = \sum_{j=1}^{m} \sum_{i=1}^{n-d} (\mathbf{x}_j \cdot \mathbf{v}_i)^2 = \sum_{j=1}^{m} \sum_{i=1}^{n-d} \sum_{k=1}^{n} (\mathbf{x}_{j,k} \mathbf{v}_{i,k})^2 = \sum_{j=1}^{m} \sum_{i=1}^{n-d} \sum_{k,\ell=1}^{n} (\mathbf{v}_{i,\ell} \mathbf{x}_{j,\ell})(\mathbf{x}_{j,k} \mathbf{v}_{i,k}). \qquad (4.8)$$

Now taking $\mathbf{M}_{k\ell} = \sum_{j=1}^{M} \mathbf{x}_{j,k} \mathbf{x}_{j,\ell}$ and switching to matrix notation, equation (4.8) is simply $D^2 = \sum_{i=1}^{N-d} \mathbf{v}_i^T \mathbf{M} \mathbf{v}_i$. Since the $\mathbf{v}_i$ have unit norm and $\mathbf{M}$ is a real symmetric matrix, the $\mathbf{v}_i$ can be taken to be the eigenvectors corresponding to the $N - d$ smallest eigenvalues of $\mathbf{M}$. Degeneracies would correspond to a tie between distance-minimizing points, and are expected to happen infrequently as they form a set of measure zero. As discussed above, the particular choice of a tie-breaking mechanism does not seem to affect algorithmic performance and can be done in whichever manner is most convenient.

Once the $\mathbf{v}_i$ have been determined, the projection can be carried straightforwardly by taking $\mathbf{y}_j = \mathbf{x}_j - \sum_{i=1}^{N-d} \mathbf{x}_j \cdot \mathbf{v}_i$.

A very straightforward implementation of this projection would therefore require $O(N^2M)$ to build **M**. Finding *all* eigenvalues and eigenvectors of **M** can be achieved in $O(N^3)$. In many cases only a subset of the eigenvectors is needed and the computation can be made faster (see, e.g., reference [69].

Handedness

When trying to reconstruct the structure of a molecule, information about relative distances between atoms can be complemented by information about the handedness of segments of the molecule. This information is necessary in distinguishing a molecular configuration from its mirror image, and also helps constrain the search at the scale of the considered segments.

The handedness of a set of four points $\{\mathbf{x}_i\}_{i=0,\ldots,3}$ can be calculated from the sign of the determinant of the vectors $\{\mathbf{x}_i - \mathbf{x}_0\}_{i=1,\ldots,3}$. The transition between right- and left-handedness occurs when this determinant is zero, that is, when the $\{\mathbf{x}_i\}$ are coplanar. To implement a projection to (say) right-handedness, one can first check whether the initial points are right-handed, otherwise project the points to the nearest hyperplane, as described above.

If one is given information about the magnitude of the determinant, one might have to choose between projecting exactly to the weaker constraint on the sign of the determinant, or projecting approximately to the more stringent constraint, for example by using linear combinations of the initial position and the projection to the hyperplane. The optimal choice might depend on how overdetermined the problem at hand is.

Fixed distribution/histogram

The distribution or histogram constraint requires values in an array of length n to follow, at least approximately, a given distribution. This distribution can be specified by an exact list of values $\{\mathbf{z}_i\}_{i=1,\ldots,n}$, in which case the constraint set is discrete and formed by all the permutations of the $\{z_i\}$. It can also be specified by a number of disjoint bins, with specifications on the number of elements that have to lie in each bin.

Fixed distribution or histogram constraints can apply for example to the noise in a measurement when the statistics of the noise are known. In such a case, the distribution of noise values will typically be one dimensional. The expected distribution can be approximated by an expected histogram. The implementation of the projection simply requires sorting the input, assigning the lowest element to the lowest bin of the expected histogram, and keep filling up the lowest unfilled bins with the lowest remaining elements. The procedure scales as $O(n \log n)$.

As with the discrete case, the projection is equivalent to an assignment problem. In dimension $m > 1$ this assignment problem can be solved in $O(n^3)$ using the Hungarian algorithm. This requires finding the distance between each point and each bin, a process that scales in the worst case as $O(n^2 m)$.

Fourier amplitude

The Fourier amplitude constraint was used in many diffraction imaging applications, and was therefore discussed in [66, 75]. Given a complex-valued array of length n, this constraint specifies the magnitudes of the *Fourier transform* of the array. The two key observations leading to efficient projection to this constraint are that distances are preserved by the Fourier transform, and that the fast Fourier transform (FFT) provides efficient

translation between real and Fourier space. A projection can therefore be implemented by going to Fourier space, projecting to the amplitude constraint, and transforming back to real space, the whole procedure scaling as $O(n \log n)$.

This procedure can be applied to any constraint formulated in Fourier space, or in any space related to real space by a metric-preserving transformation that is both easy to compute and to invert. Examples of such transformations include the discrete wavelet transform [69] and, more generally, all orthonormal linear transforms.

Support

The support constraint is used on the pixel values of images in diffraction imaging. It requires that the pixel values be zero outside a finite, known region called *support*. The constraint set is a hyperplane obtained by setting variables outside the support to zero. The implementation is straightforward:

$$y_i = \begin{cases} x_i, & \text{if } i \text{ is inside the support} \\ 0, & \text{otherwise.} \end{cases} \tag{4.9}$$

This scales as $O(n)$ or, if it is performed in place, as $O(n - m)$, where m is the dimension of the support.

Fixed area

This constraint on arrays of length n is a weaker version of the support constraint and is also used in diffraction imaging. It requires that only a finite number of variables have nonzero values, without specifying which variables. The implementation involves sorting the pixel value amplitudes, and setting the output values corresponding to the

smallest amplitudes to zero. It is very similar to that of the discrete select-d constraint, above. It also scales as $O(n)$ in the typical case.

Energy in a step potential

The problem of finding low energy states of a set of n particles interacting through pairwise potentials can be broken down into two constraints, one of which is equivalent to a constraint specifying the maximal the total energy E_T of $n(n-1)/2$ independent pairs of particles. This constraint is easy to satisfy (provided the potentials are simple), but the projection to this constraint requires figuring out how to minimally displace the particles to reach the target energy. Because of this, it is not possible to treat each pair separately, as with constraints in the usual divide and concur approach. However, for some choices of pair potentials this energy projection is nevertheless rather straightforward. The step potential is one of them:

$$V_{ij}(r) = \begin{cases} \infty, & \text{if } r < r_0 \\ -|a|, & \text{if } r_0 < r < r_1 \\ 0, & \text{otherwise.} \end{cases} \tag{4.10}$$

Because this potential can take only two finite values, the energy constraint amounts to specifying that at least $|E_T/a|$ pairs have distance between r_0 and r_1. The projection to this constraint is very similar to that of the area constraint or the select-d constraint: it suffices to sort the pair distances and make sure that the $|E_T/a|$ closest pairs are at a distance r between r_0 and r_1. Calculating all the pair distances scales as $O(mn)$.

4.5 Symmetries and projections

A feature common to most of the constraints in the previous section is the existence of simple symmetries of the constraint set. The most commonly featured are translational symmetry (in all linear constraints) and rotation symmetry (as in the amplitude constraint or the pairwise exclusion constraint). Discrete constraints such as reflection and discrete rotation symmetry are present in most discrete constraints discussed. More subtle symmetries occur for example in the Fourier amplitude constraint: in this case, the symmetry is a local gauge invariance in Fourier space. This situation is reminiscent of integrable Hamiltonian systems, where the presence of symmetries is related to the existence of conserved quantities (through Noether's theorem), and to analytical solvability of the Hamiltonian. It would be interesting to find whether there is an equivalent, for projection operators, to the relation between symmetry and solvability in Hamiltonians. If this is the case, one might consider a systematic classification of constraint sets in terms of their symmetry groups (or conserved quantities), similarly to what is done for integrable Hamiltonians. There is considerable literature on the classification of integrable Hamiltonians. Some of our work on this topic was presented in References [76, 77]. For a recent review see, e.g., Reference [78].

4.6 Constraints without simple projection operators

Many of the constraints listed in the previous section are ideal for use with iterated projection schemes: exact projection operators for these constraints are efficient and easy to implement. However, many constraints do not have such simple projection operators. In this section we discuss how many such constraints can nevertheless be accommodated in the iterated projection framework.

In many cases the optimal approach might simply be to use approximate projection operators. As long as the approximate operator returns a value within the constraint set, many of the desirable properties of the difference map are maintained. In particular, fixed points are still guaranteed to lead to solutions.

It is useful at this point to emphasize the fact that the intuition behind the difference map is based on the behavior of the search process near the solution. An operator that approximates the projection operator when applied to a point near the constraint will not only guarantee that fixed points may be retrieved from solutions, but also that most of the convergence properties of the difference map will be maintained.

Many approximate projection operators can therefore be defined simply by following a greedy approach such as steepest descent or conjugate gradients towards the individual constraint set. The landscape in which the steepest descent is performed can be any cost function that is zero at the boundary of the constraint set. An example application of this method would be to variables obeying an arbitrary relation of the form $\mathbf{y} = f(\mathbf{x})$. If f is linear the conjugate gradient method can solve this problem exactly. If f is nonlinear the conjugate gradient can still be used to greedily minimize the squared distance

$$d^2 = (\mathbf{x} - \tilde{\mathbf{x}})^2 + (\mathbf{y} - \tilde{\mathbf{y}})^2 = (\mathbf{x} - \tilde{\mathbf{x}})^2 + (\mathbf{y} - f(\tilde{\mathbf{x}}))^2 \,. \tag{4.11}$$

Greedy minimization (i.e., minimization by steepest descent) of the squared distance does not guarantee that a global minimum of the distance has been found. However, the local minima guarantees that the vector joining the starting point $(\mathbf{x}, \mathbf{y})$ to the local minima $(\mathbf{x}_m, \mathbf{y}_m)$ is orthogonal to the surface of the constraint set.

Another example of an efficient greedy approach is Lloyd's heuristic, also known as the *k*-means method, discussed earlier in the context of the concurrence by cluster

constraint. This clustering problem has received considerable attention, and is now rarely solved using complete methods; the greedy approach is probably sufficient in most $D - C$ applications.

4.7 Characterizing the search process

We have so far explained the motivation and the definition of the difference map. In the next chapter we will provide examples of the applications of the difference map and divide and concur to constraint satisfaction problems, and show how they can lead to very efficient heuristics for solving some difficult constraint satisfaction problems. Readers interested mostly in applications are encouraged to proceed immediately to the next chapter. In this and the next sections we will be interested mostly in understanding the dynamics of the search process itself.

We are currently lacking a thorough understanding of the global search properties of iterated projection-based solvers. Most of our understanding of the search process is based on the 'local' properties of the maps or, more precisely, on properties of the maps for linearized constraints. In this section we delve deeper into the search procedure itself and describe different intuitive approaches to understanding the search process.

Empirical evidence has shown that attractive limit cycles do exist for the difference map in problems of low dimensionality, particularly in 2 dimensions (see Figure 4.8). When they occur, limit cycles can cause the search to fail. However, such limit cycles were rarely identified in higher dimensional problems. The dimension of the search space therefore appears to be an important factor in the success of iterated projection-based search strategies.

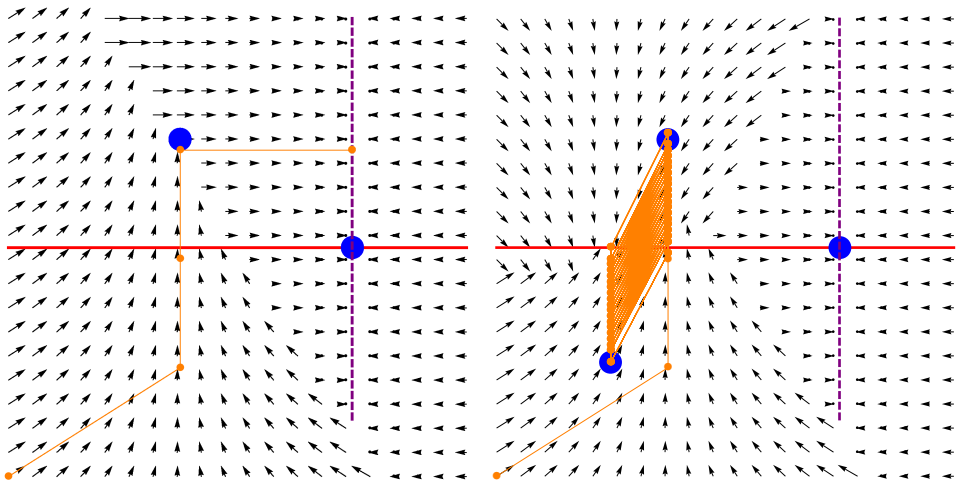

Figure 4.8: An example of a success and a failure of the difference map in few dimensions. One constraint space is the points on the red line, the second constraint set is the union of the blue points. The fixed points corresponding to the solution are indicated by a purple dashed line. The orange points correspond to the trajectory of the difference map with $\beta = 1$, and the arrows indicate the magnitude and direction of the displacement vector.

There are many ways in which a high dimensional search space can affect the search behavior. The existence of 'accidental' stable limit cycles in high dimension is made less likely simply by the high number of directions in which the iterate can 'escape' the limit cycle. The extra dimensions also provide more opportunities for independent constraints to be orthogonal; the likelihood that two randomly selected vectors are approximately orthogonal increases rapidly with the dimension of the vectors. Since the parameters of the difference map were optimized for orthogonal constraints, this choice of parameters might become a better choice, on average, in high dimensions.

A larger dimensional space can also be challenging since more dimensions also mean more space to explore. If the dimension of the target (i.e., the space of fixed points) grows with the dimension of the search space, this might not be a major problem. An indication of the difficulty of the problem can be obtained by comparing the dimension of the fixed point variety to that of the space effectively explored by the search

process.

4.7.1 Dimension of the space of fixed points

When two constraints C_1 and C_2 are orthogonal linear spaces, the dimension of their space of fixed points is simply the codimension of the space spanned by the constraints. In discrete (or Boolean) problems, where the primary constraint sets consists of the union of isolated points, the divide constraint has dimension 0, while the concur constraint has dimension N_v (the number of variables). The set of fixed points thus has dimension $N - N_v$, where N is the dimension of the search space. In this case, the increase in the number of dimensions resulting from the use of divide and concur (from N_v to N) is compensated exactly by the increase in the number of dimensions of the set of fixed points!

4.7.2 Dimension of the effective search space

A typical successful $D - C$ search process, when monitored by the size of the difference vector (or *error*), can be broadly separated in three parts (see Figure 4.9). The first part resembles a transient relaxation mode, where the error decreases steadily from a relatively high starting point. The second part resembles a search mode, where the error fluctuates (sometimes very strongly), but without any apparent systematic progress in the error. The last stage is usually the abrupt decrease of the error as the search converges to a fixed point. In easier problems the second stage is usually not observed as the progress towards a fixed point is more or less systematic. However, in difficult problems, most of the time is spent 'searching.'

It would therefore be useful to characterize the dimension of the space explored in the search mode. This 'explored space' cannot be strictly speaking described as an attractor, as the iteration will in many cases eventually reach a fixed point. However, it is often possible to modify only slightly the problem in order to make it unsatisfiable (or *unsat*). The search process in the unsat problem is quite similar to that of the initial problem - a statistically different search behavior would imply that the search process somehow already knows that a solution exists while in the searching mode. The unsat problem, on the other hand, does have a well-defined, nontrivial attractor, and the dimension of this attractor corresponds to the dimension of the effective search space explored in the original problem. Numerical simulations can provide an idea of the size of the search space (see, e.g., reference [68]). However, we can use a contracting property of the difference map to obtain a simple upper bound to the effective dimension of the search space.

4.7.3 Continuity and contraction

The difference map is not a continuous map as its building blocks, the projection operators, are themselves not continuous. However all projection operators considered in this work are at least continuous *by parts*. As a results, the difference map itself is continuous by parts, and one can think of the vector field defined by the difference map as a collection of domains within which the map is continuous, separated by discontinuity walls. For simplicity we will consider problems where one constraint is linear with dimension N_v (the concurrence constraint), and one constraint is discrete (the divide constraint). An example of such a problem, discussed in Section 5.1, is Boolean satisfaction, or kSAT. For such problems the behavior in individual cells is quite simple: all points in a cell are mapped to the same hyperplane of codimension N_v. That is to say that the effective

search space in this case is contained within a large but finite collection of hyperplanes of dimension $N_{lits} - N_V$. The dimension of the search space is therefore smaller than or equal to the dimension of the space of fixed points. Furthermore, the displacement parallel to the hyperplane is the same for all points in the cell. After the one iteration that brings all points to the same hyperplane, all points within a cell will have the same displacement vector as long as they are still within the cell. A small neighborhood V of a point in the search space will therefore be projected onto successive planes, a non-expanding operation, as long as the image of V does not straddle the boundary of two cells. If the image does encounter a boundary, the next iteration will split the image into two disconnected regions that might then be taken to different parts of the search space. By successive divisions the image of V can therefore become well distributed in space even though its d-volume can only remain constant or decrease. The d-volume is simply the volume in d dimensions: a flat surface in three dimension has 3-volume 0, but a 2-volume equal to its area. Any object with nonzero 3-volume has infinite 2-volume.

4.7.4 Low dimensional runtime monitors of the search process

The problems considered in this work are all high dimensional in nature, and this complicates data visualization. In some cases, particularly in imaging applications, this visualization can be achieved rather naturally by plotting the current value of the iterate. Such plotting can be computationally demanding, especially if done regularly, and is not always very informative. In this section we describe low-dimensional *monitors*, that is, quantities that can calculated easily and provide information on the state of the search.

The most important such monitor is the magnitude of the vector $P_1(f_2) - P_2(f_1)$. Since this vector can be represented as a line joining the two constraints, its magnitude

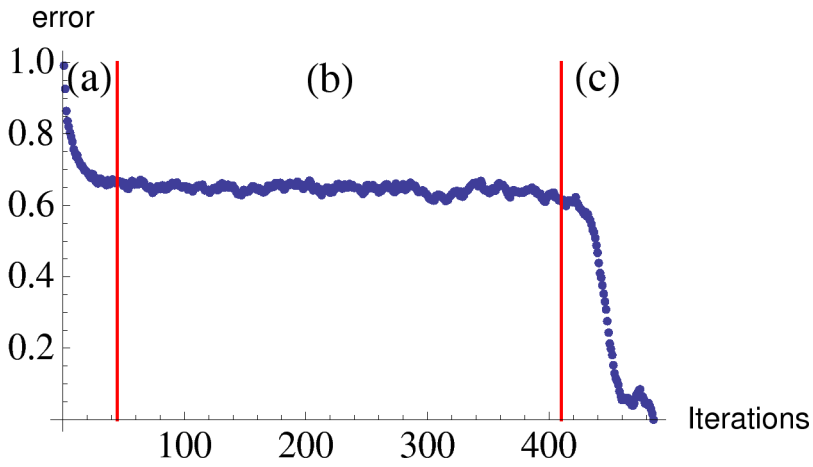

Figure 4.9: A typical error plot for a successful run of the difference map showing (a) a steady decrease in the magnitude of the error, (b) a 'search mode' where the error fluctuates but does not decrease systematically, and (c) an abrupt decrease in the error magnitude when converging to the solution. This error plot was generated while decoding error-correcting codes (see section 5.4, below).

(which we refer to as the *error*) provides an indication of how far the two constraints are from each other in the explored area. A typical progression of the error as a function of the iteration count is shown on Figure 4.9.

An error of zero, or less than a suitably small threshold, indicates that a fixed point and a solution have been found. This is often used as a termination criteria for the search process. The error also provides a computationally inexpensive monitoring of the search process. For example the average size of the error during the search provides an indication of how close the constraints are in the explored area of search space. A persistent change in the average value of the error during the search might indicate a significant change in the space explored by the iterate. Comparing the size of the variations of the error to the mean error can be used as a thoroughly unrigorous indication of the ergodicity of the search.

In divide and concur, one can also keep track of the *number* of unsatisfied con-

straints, rather than the error related to the Euclidean distance between the constraints. This can be achieved by keeping track of the number of constraints that required variable change during the divide projection. It can also be achieved by testing the result of the concurrence projection for satisfaction of the individual constraints. This way, one obtains an 'unsatisfaction count' for an actual configuration of the system, with only one value per variable. In cases where the variables are discrete (or Boolean), one can round the result of the concurrence projection before evaluating the constraints. If the number of violated constraints reaches zero, the problem has been solved and the search can be interrupted. Note that this can happen even if the error is nonzero, but always happens if the error reaches 0. If the cost of evaluating the number of unsatisfied constrained is low enough, this can be used as an efficient termination criteria. We used this criterion, for example, in the random Boolean satisfaction problem discussed in section 5.1, since the search sometimes reached limit cycles (with nonzero error) even though no unsat constraints remained.

A third monitor is the number of variables that get updated at each iteration. The same error can result from multiple variables changing a little or from a few variables changing a lot. In the random 3SAT problem, it was observed that only a small fraction of the variables got updated at each iteration, and that this fraction grew smaller as problem size increased. Such information might be useful not only in understanding the search process, but also in implementing efficient code taking advantage of the sparse update structure.

4.8 Optimizing iterated searches

4.8.1 The difference map, and parameter optimization

When the difference map is applied to two constraints C_1 and C_2, the displacement vector is proportional to the difference between a point on C_1 and a point on C_2. This important difference feature, which gives its name to the difference map, ensures that to each fixed point of the map corresponds to an intersection of the constraints. In this section we explain how requiring this feature leads to the difference map, equation 4.1. Given the projections P_1 and P_2 associated with C_1 and C_2, we define the map

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \beta \left(P_1 \left[f_2(\mathbf{x}_n) \right] - P_2 \left[f_1(\mathbf{x}_n) \right] \right). \tag{4.12}$$

where β is an arbitrary real or complex parameter. This map has the difference feature for arbitrary functions f_1 and f_2. We first take these functions to be linear combinations of $\mathbf{x}$, $P_1(\mathbf{x})$ and $P_2(\mathbf{x})$:

$$\begin{aligned} f_1(\mathbf{x}) &= a_{11} P_1(\mathbf{x}) + a_{12} P_2(\mathbf{x}) + a_{13}\mathbf{x}, \\ f_2(\mathbf{x}) &= a_{21} P_1(\mathbf{x}) + a_{22} P_2(\mathbf{x}) + a_{23}\mathbf{x}. \end{aligned} \tag{4.13}$$

Because projections depend only on the distance between the starting point and the constraint set, coordinate changes M that leave distances invariant (such as rotations and translations) also leave the projections invariant, in the sense that $\tilde{P}_i(M\mathbf{x}) = MP_1(\mathbf{x})$, where $\tilde{P}_i$ is the projection operator in the transformed coordinate system. This ensures that the f_i are invariant under rotations R: $\tilde{f}_i(R\mathbf{x}) = R f_i(\mathbf{x})$, where $\tilde{f}_i$ is f evaluated in the rotated frame. However, equations 4.13 do not guarantee that the f_i are also invariant under translations T of the coordinate system. Indeed $\tilde{f}_i(T\mathbf{x}) = (\sum_j a_{ij})T\, f_i(\mathbf{x})$.

To ensure that translations of the coordinate system do not affect the search process,

we therefore require

$$\sum_j a_{ij} = 1, \qquad i = 1, 2. \tag{4.14}$$

Moreover, we require that the search be efficient in the simple case of orthogonal linear constraints. In this case the search space K can be separated in orthogonal spaces $K = K_1 + K_2 + K_\perp$. Points in K can similarly be decomposed: $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_\perp$. We choose coordinates so that the constraints are expressed as

$$\begin{aligned} C_1 &= \{\mathbf{Y_1} + 0 + \mathbf{b} \mid \mathbf{Y_1} \in K_1\}, \\ C_2 &= \{0 + \mathbf{Y_2} + 0 \mid \mathbf{Y_2} \in K_2\}, \end{aligned} \tag{4.15}$$

where $\mathbf{b}$ is a constant element of $K_\perp$. The constraints intersect at the origin if and only if $\mathbf{b} = 0$. After one iteration, $\mathbf{x}$ is mapped onto

$$\mathbf{x}_1 \left[1 + \beta\left(a_{21} + a_{23}\right)\right] + \mathbf{x}_2 \left[1 - \beta\left(a_{12} + a_{13}\right)\right] + \mathbf{x}_\perp + \mathbf{b}. \tag{4.16}$$

The perpendicular component is incremented, after each iteration, by $\mathbf{b}$. If $\mathbf{b} \neq 0$, this eventually leads the iterate away from the local minima in the distance between the constraints. If $\mathbf{b} = 0$, on the other hand, the perpendicular component is fixed, so that all points in $K_\perp$ are fixed points. Actually, the whole of K can be fixed if one adjusts the parameters such that $a_{12} = -a_{13}$ and $a_{22} = -a_{23}$. However, just as having an arbitrary small nonzero $\mathbf{b}$ is enough to turn the fixed point variety $K_\perp$ into an unstable manifold, small departures from linearity or orthogonality might destroy most of these fixed points and leave no guarantee that the remaining fixed points are locally attractive, even when the constraints are still intersecting.

On the other hand, the choice $a_{21} + a_{23} = 1 - a_{22} = -1/\beta$ and $a_{12} + a_{13} = 1 - a_{11} = 1/\beta$ ensures that in one iteration all points are mapped to fixed points in $K_\perp$. Small deviations from orthogonality in this case will leave $K_\perp$ fixed and attractive. The optimal choice of

the relative size of a_{21} vs a_{23} and a_{12} vs a_{13} cannot be deduced from the optimization of linear constraints. For simplicity, the difference map uses $a_{12} = a_{21} = 0$. It is possible that different choices result in better search behavior in many applications.

If smooth constraints can always be approximated by a linear space near the solution, not all constraints are approximately orthogonal near the solution. In most cases where the constraints are independent and of low dimension relative to the search space (i.e. $\dim C_1 + \dim C_2 < \dim K$), the constraints will be orthogonal to a good approximation, and the parameter choice described above is justified.

Information on the ensemble from which the constraints are drawn can be used to refine the choice of parameters, and this has been done in [75]. However, this involves significant problem-dependant development, and in all applications discussed below the choice of parameters described above was made, resulting in the difference map as presented above:

$$\begin{aligned} \mathbf{x}_{n+1} = f(\mathbf{x}_n) &= \mathbf{x}_n + \beta \left(P_1 \left(f_2(\mathbf{x}_n) \right) - P_2 \left(f_1(\mathbf{x}_n) \right) \right), \\ f_i(\mathbf{y}_n) &= (1 + \gamma_i) P_i(\mathbf{x}_n) - \gamma_i \mathbf{x}_n, \qquad i = 1, 2, \end{aligned}$$

with $\gamma_1 = -1/\beta$ and $\gamma_2 = 1/\beta$.

4.8.2 Optimizing β

The optimization procedure described in the previous section does not specify the magnitude nor the sign for β. However, in many applications, the choice of β strongly affects the search performance. For most applications optimal values of β have magnitude between .5 and 1.4. In some cases, optimal performance can only be achieved by carefully adjusting β (see, e.g., section 5.1). This is quite similar to the behavior of stochastic local searches, where performance depends sensitively on the balance between greediness

and randomness [79].

The determination of the optimal choice of β, in the applications discussed in chapter 5, was achieved through experimentation. One strategy is to optimize β for small, tractable problems and assume that the optimal value of β depends weakly on problem size. Another approach is to try to estimate, from the early behavior of runtime monitors (see section 4.7.4), which β produces the most efficient search. Both these approaches require rather strong assumptions, and the experimentation with different values of β was one of the most time consuming parts of the problem-specific development for these applications. Obtaining a priori information on the optimal value of β, or a criterion to optimize β dynamically during the search process would be quite useful. Adaptive procedures for the noise parameter have been proposed for stochastic local searches such as WALKSAT (see, e.g., reference [79]) and might be a source of inspiration for a similar procedure for the difference map.

One *a priori* criterion for the optimization of β can be obtained from an extension of the procedure described in section 4.8.1. In section 4.8.1 we have found values of the γ_i that ensure that the difference map converges to a fixed point in a single iteration for linear orthogonal constraints. Similarly, we can find a value for β which ensures that errors due to slight non-orthogonality of the constraints is minimized.

To describe a pair of nearly orthogonal linear spaces L_1 and L_2, we first define the space $L_\perp$ orthogonal to both L_1 and L_2. We then define the space $\tilde{L}_2$, orthogonal to both L_1 and $L_\perp$. $\tilde{L}_2$ is therefore nearly parallel to L_2. A basis of for L_2 can then be written as $\{\mathbf{y}_i\}_{i=1,\ldots,d_2} = \{\tilde{\mathbf{y}}_i + \epsilon_i \mathbf{v}_i\}_{i=1,\ldots,d_2}$, where d_2 is the dimension of L_2, the ϵ_i are small parameters, and $\tilde{y}_i$ and v_i are unit vectors in $\tilde{L}_2$ and L_1, respectively. We can now decompose arbitrary vectors in orthogonal components: $\mathbf{x} = \mathbf{x}_1 + \tilde{\mathbf{x}}_2 + \mathbf{x}_\perp$.

Restricting ourselves to maps which are of second order in the projection operators, we use the fact that the constraints are linear and the projections idempotent to write the most general map based on projections P_1 and P_2 as

$$\mathbf{x}_{n+1} = a_{00}\mathbf{x} + a_{01}P_1(\mathbf{x}) + a_{02}P_2(\mathbf{x}) + a_{12}P_1(P_2(\mathbf{x})) + a_{21}P_2(P_1(\mathbf{x})). \tag{4.17}$$

To linear order in the ϵ_i, we find

$$\begin{aligned}
\mathbf{x} &= \mathbf{x}_1 + \tilde{\mathbf{x}}_2 + \mathbf{x}_\perp && (4.18)\\
P_1(\mathbf{x}) &= \mathbf{x}_1 && (4.19)\\
P_2(\mathbf{x}) &= \tilde{\mathbf{x}}_2 + \sum_i \epsilon_i \left(\mathbf{x}\cdot\mathbf{v}_i\right)\mathbf{y}_i + \sum_i \epsilon_i \left(\mathbf{x}\cdot\mathbf{y}_i\right)\mathbf{v}_i && (4.20)\\
P_1(P_2(\mathbf{x})) &= \sum_i \epsilon_i \left(\mathbf{x}\cdot\mathbf{y}_i\right)\mathbf{v}_i && (4.21)\\
P_2(P_1(\mathbf{x})) &= \sum_i \epsilon_i \left(\mathbf{x}\cdot\mathbf{v}_i\right)\mathbf{y}_i && (4.22)
\end{aligned}$$

Requiring that this map reaches fixed point $\mathbf{x}_\perp$ after one iteration for orthogonal constraints ($\epsilon_i = 0$) imposes $a_{00} = -a_{01} = -a_{02} = 1$. Requiring that all terms linear in ϵ_i vanish imposes $a_{12} = a_{21} = -a_{02}$. This leaves us with the map

$$\mathbf{x}_{n+1} = \mathbf{x} - P_1(\mathbf{x}) - P_2(\mathbf{x}) + P_1(P_2(\mathbf{x})) + P_2(P_1(\mathbf{x}). \tag{4.23}$$

Note that this map, which minimizes the error due to slight non-orthogonality of the constraints, corresponds to a path similar to the Z-box, the integration path chosen to minimize error in the effective pair potential calculations (see Figure 4.10)

We have not yet imposed that this map has the difference property. Thanks to the linearity of the constraints, this can be achieved simply, for example by writing

$$\mathbf{x}_{n+1} = \mathbf{x} + P_1\left(P_2(\mathbf{x}) - \mathbf{x}\right) - P_2\left(\mathbf{x} - P_1(\mathbf{x})\right). \tag{4.24}$$

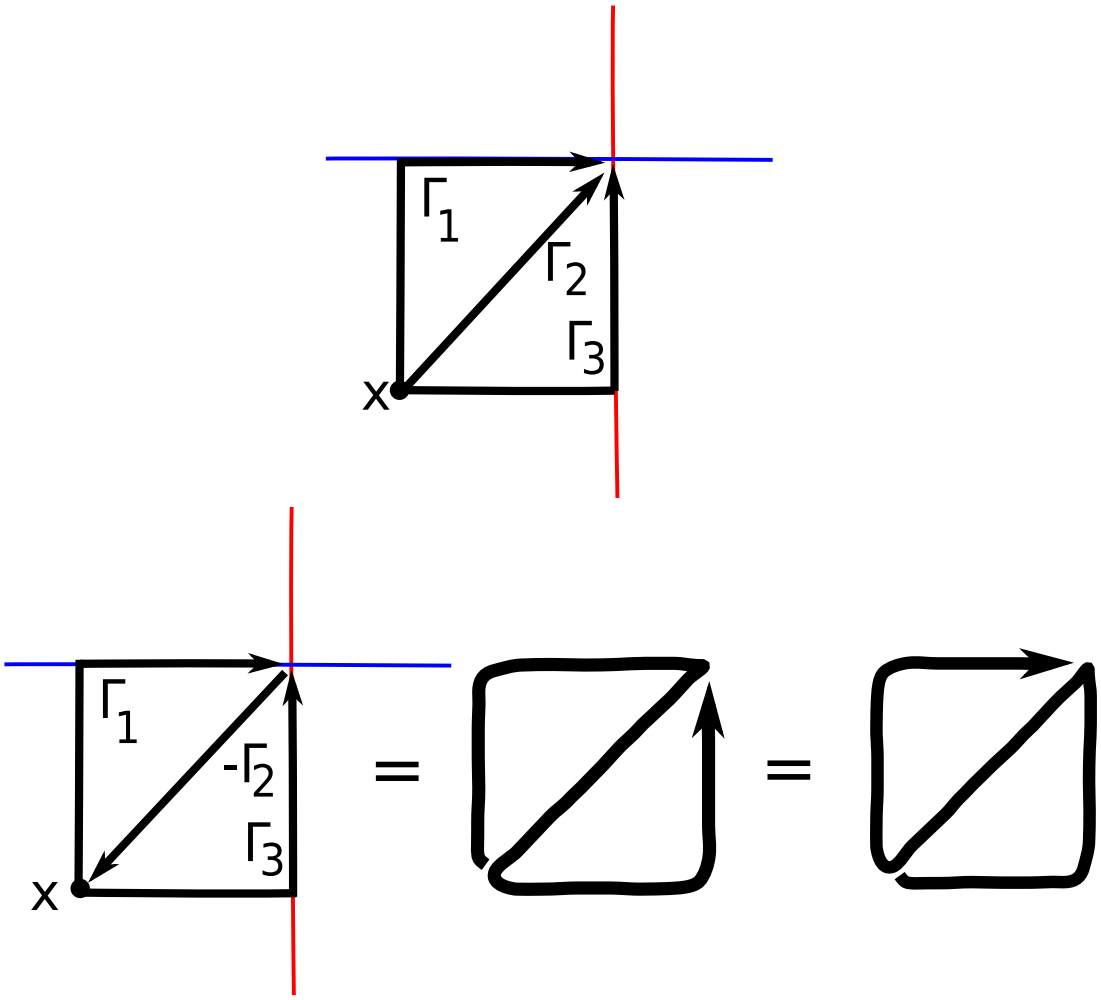

Figure 4.10: (top) Three estimates of the position of the intersection of constraints 1 and 2, illustrated by a blue and red line, respectively. Starting from point **x**, we have $\Gamma_1 : P_2(P_1(\mathbf{x}))$, $\Gamma_2 : P_1(\mathbf{x}) + P_2(\mathbf{x}) - \mathbf{x}$, $\Gamma_3 : P_1(P_2(\mathbf{x}))$. (bottom) Linear combination of the three estimates that ensures optimal cancellation of errors when the constraints are nearly orthogonal. The corresponding paths are identical to those resulting from parameter path integral formalism in Section 3 (see, in particular, Figure 2.10).

This map has the difference property and converges rapidly to a fixed point for constraints which are linear spaces and are approximately orthogonal. It corresponds to the limiting case of the difference map with $\beta \to 0$. If it is applied to constraints which are not linear spaces, it has the serious drawback that it then depends on the choice of the origin. The difference map with very small β does not depend on the choice of the origin, and according to this analysis would be the best choice to ensure rapid conver-

gence for nearly orthogonal constraints. The fact that empirically observed optimal β are usually closer to 1 than to 0 indicates that other factors play an important role in determining the efficiency of the difference map search.

4.8.3 Optimization of divide and concur

The optimization procedures mentioned above were performed with generic applications of the difference map in mind. Optimal parameter choices for divide and concur might be different from those determined for such generic difference map applications. In particular, the divide and concur constraints are not random linear constraints, and the orthogonal assumption fails for a small number of constraints. For a set of N_C intersecting orthogonal linear primary constraints, the minimal angle θ_{DC} between the divide and concur constraints is given by $\cos\theta = 1/\sqrt{N}$ unless all primary constraints are discrete, in which case the divide and concur constraints are indeed orthogonal. This can be demonstrated by considering the dot product between arbitrary vectors satisfying the divide constraint, $X_D = \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \cdots \oplus \mathbf{x}_N$ and the concur constraint $X_C = \bar{\mathbf{x}} \oplus \bar{\mathbf{x}} \oplus \cdots \oplus \bar{\mathbf{x}}$. Using the orthogonality of the primary constraints, one can decompose $\bar{\mathbf{x}}$ in components parallel to the different constraints, or orthogonal to all constraints: $\bar{\mathbf{x}} = \bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2 + \cdots + \bar{\mathbf{x}}_N + \bar{\mathbf{x}}_\perp$. We also need to define $\mathbf{x} = \sum \mathbf{x}_i$ and the angle $\theta_{\mathbf{x}\bar{\mathbf{x}}}$ between $\mathbf{x}$ and $\bar{\mathbf{x}}$.

The angle θ_{DC} between X_D and X_C is given by

$$\begin{aligned} \cos\theta_{DC} &= \frac{X_D \cdot X_C}{|X_D|\,|X_C|} \\ &= \frac{\sum_i \mathbf{x}_i \cdot \bar{\mathbf{x}}_i}{\sqrt{N}\,|\bar{\mathbf{x}}|\,|\mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_N|} \\ &= \frac{1}{\sqrt{N}} \frac{\mathbf{x} \cdot \bar{\mathbf{x}}}{|\mathbf{x}|\,|\bar{\mathbf{x}}|} = \frac{1}{\sqrt{N}} \cos\theta_{\mathbf{x}\bar{\mathbf{x}}}. \end{aligned} \tag{4.25}$$

Since $\mathbf{x}$ and $\bar{\mathbf{x}}$ are arbitrary, they can be chosen to be parallel, yielding a maximum value of 1 for $\cos\theta_{\mathbf{x}\bar{\mathbf{x}}}$, and of $1/\sqrt{N}$ for $\cos\theta_{DC}$. The only exception occurs if the divide constraint has dimension 0, implying $|X_D| = 0$. In this case, the divide constraint is a single point, the divide and concur constraints are by definition orthogonal.

The orthogonal approximation is therefore valid for continuous constraints only in the limit where the number of constraints is large.

Another feature of the divide and concur approach not shared by generic difference map applications is that the concurrence constraint is always linear, unless additional constraints are added onto the concurrence constraint. For this reason one can set for example the parameter a_{12} to zero without loss of generality.

Finally, and perhaps most importantly, the divide and concur scheme allows one to adjust easily the relative importance of the different constraints, by introducing weights in the concurrence projection.

Optimizing the weights in divide and concur

As mentioned in sections 4.3, the concurrence projection in divide and concur comes with its own lot of adjustable parameters, as one is free to adjust the relative weights λ_i of each constraint when taking the average over replicas. Interestingly, changing the weights in this context is equivalent to changing the metric of the search space. Giving more weight to a constraint is equivalent to increasing the metric for the variables involved in it, which makes the volume of the constraint set for this constraint larger.

There is little hope here to obtain a general strategy to assign optimal weights *a priori*; this optimal choice may depend strongly on the particular position of the iterate

in the search space. An example of this is the disk packing problem discussed in section 5.2.1 below. In this problem, all constraints are formally equivalent, yet during the search some constraints become effectively irrelevant to the optimal next move.

However we will show that dynamical updating schemes for the weights are quite useful in finding dense disk packings. Our efforts therefore focused on finding adaptive methods and heuristics for determining the optimal weights. The idea here is to adjust the metric to the current searching conditions, but to adjust it slowly enough so that the desirable convergence properties of the difference map are not affected.

Adjusting the weights can achieve two goals. First, irrelevant constraints can be effectively discarded and their contribution neglected, while still keeping track of them in case they become important again as the search continues. Second, constraints that are consistently violated during the search process can be given extra weight, making their immediate satisfaction more important and asking more change from constraints that might have more 'slack' available.

A simple way to achieve this is to increment the weight of a constraint by a constant value each time a constraint is violated (i.e., each time the divide projection has a nontrivial effect on the variables involved). Another strategy is to make the weight of a constraint proportional to the norm of the total displacement vector of the variables involved in the constraint. In sphere packing problems, we found it more efficient to set the weight in proportion to the exponential of the amount by which a constraint is violated (see section 5.2.1).

The choice of the weight updating scheme has significant impact on performance for disk packing problems. However, none of the weight update schemes we applied to the random Boolean satisfaction (kSAT) problem resulted in performance improve-

ments. Obtaining a general criterion, or even a heuristic, for determining the optimal reweighting strategy would be quite useful.

4.8.4 Preconditioning

For all applications of divide and concur discussed in this dissertation, the search was initiated at a random position in the search space. In this section we discuss possible simple ways of choosing the initial conditions that might improve search performance.

The simplest approach would be to initiate the search from a random position, but proceed with a number of greedy steps to get the iterate near the constraints before using divide and concur or the difference map. The idea in this case being that the machinery of the difference map, useful in avoiding local minima, might not be the most efficient in the approach phase of the search process.

A more intriguing possibility would be to initiate the search using statistical information about the problem. Message passing algorithms such as belief propagation (BP) and survey propagation (SP) are designed to obtain statistical estimates of the expectation values of different variables in the ensemble of solutions to the problem. Taking these expectation values as a starting point to divide and concur is an appealing possibility. The current approach for using information from BP and SP to solve constraint problems is decimation: variables with very polarized expectation values are set to either true or false, and the procedure is repeated until the problem is small enough to be passed on to a local search algorithm such as WALKSAT. Divide and concur has an advantage over WALKSAT that it can actually use the information provided by the continuous expectation values of each variables. Discrete local algorithms, on the other hand, require a rounding of all the polarizations before proceeding.

However, we have tried such an approach in a decoding problem of error-correcting codes (see section 5.5), without convincing evidence that the ‘preconditioning’ with BP improved the convergence rate of $D - C$ in problems where BP failed to find a solution. However in this application the solution is unique; failure of BP to find a solution might be an indication that the marginals are only weakly correlated to the solution. It would be interesting to try this approach in a problem such as random Boolean satisfaction (kSAT), where BP typically converges to a point in the search space where the density of solutions is expected to be higher than at a random point.

CHAPTER 5

SELECTED APPLICATIONS OF DIVIDE AND CONCUR

The application of divide and concur to Boolean satisfaction and some sphere packing problems are described in reference [80]. We include in this section the relevant results. We also include preliminary results for additional applications that are related to Boolean satisfaction (hypervertex covers, decoding of low-density parity check codes) and discuss the possible application to energy minimization for particles interacting via pair potentials.

5.1 Boolean satisfaction: the kSAT problem

The kSAT problem (or k-satisfiability) is one of the best studied constraint satisfaction problems, and was the first problem that was shown to be NP-complete. The challenge is to find an assignment for N_v Boolean variables that satisfies a list of N_c Boolean constraints, or *clauses*. Each clause is an OR statement involving *k literals*. For 3SAT (kSAT with k=3), a clause reads $\ell_1 \vee \ell_2 \vee \ell_3$, where each literal ℓ_i represents either the truth value of one of the N_v Boolean variables or its negation.

In this problem each of the constraints is a logical OR, for which we have defined projection operators above. A $D - C$ formulation of 3SAT is therefore obtained by associating a real-valued search variable to each 3SAT literal, where the values $\{1, -1\}$ are taken to mean $\{\text{True}, \text{False}\}$. The constraint D requires that each clause is satisfied; that is, each literal must have value ± 1, with at least one literal per clause having value 1. In other words, to each clause corresponds a variable triplet, which is projected by P_D to the nearest of the seven satisfying assignments for this clause. Geometrically, these correspond to seven vertices of a cube, as illustrated in Figure 4.6. In this application

P_C ensures that all of the literals associated with the same Boolean variable concur with due regard to negations . Since each constraint (clause) involves only three variables, the search space has dimension $3N_c$. For simplicity, we give equal weight to each constraint ($\lambda_i = 1$).

We compared the performance of the $D - C$ algorithm with WALKSAT [58] on a collection of 3SAT problem instances ranging from $N_v = 50$ to $N_v = 25600$, with fixed ratio $\alpha \equiv N_c/N_v = 4.2$, a value for which randomly generated instances are expected to be difficult [81]. Random instances were generated using the program MAKEWFF (distributed with WALKSAT), and instances that were not solved by either WALKSAT or the $D-C$ algorithm were discarded. Each algorithm was applied 10 times to each instance, starting from different random initial conditions. The median number of variable updates required to find the solution is plotted in Figure 5.1. The number of variable updates in WALKSAT equals the total number of flips of the Boolean variables. In the $D-C$ algorithm it is the total number of nonzero updates of any of the real-valued search variables, that is, literals.

In this application we verify every few iterations whether $P_A(\mathbf{x})$, rounded to the nearest integer, is a solution to the initial problem. If it is, the search is interrupted even when no fixed point has yet been found. Because the verification is computationally simple it does not add significantly to the computational burden. It does allow to resolve an issue that was observed particularly in small problems, namely, that limit cycles exist near the solution and occasionally prevent the search from reaching a fixed point. Since solutions could be retrieved from the limit cycles by the verification procedure just described, the appearance of limit cycles did not lead to a failure of $D - C$ in this application.

Figure 5.1 shows that WALKSAT (with the 'noise' parameter fixed at the value $p = 0.57$, the choice resulting in the best observed scaling behavior) and the $D-C$ algorithm

(with $\beta = 0.9$) have similar performance behavior. Not only do they both find the same problems easy and the same problems hard (which is not unexpected), but the scaling of the number of variable updates needed to reach the solution, as a function of problem size, is also similar. Such a similarity is surprising, considering the difference in search strategies. WALKSAT uses pseudorandom processes (or 'noise') to update the variables asynchronously. In $D - C$, on the other hand, the update rule is completely deterministic and is applied synchronously to many variables. Figure 1 also shows that choosing suboptimal parameters for either algorithm results in rapid performance degradation for large problem sizes near the critical region. In the case of WALKSAT, this had been observed previously (see, e.g., reference [79]). Even though the scaling of the variable updates are similar for WALKSAT and $D - C$, our implementation of $D - C$ required significantly more CPU time (between 4 and 200 times, depending on the instance) than WALKSAT.

5.2 Sphere packing

5.2.1 Disks in a square

Another constraint problem that has been extensively studied is the packing of n spheres in a finite D-dimensional volume (see, e.g., references [82–86] and references therein). The constraint formulation of this problem is more directly geometrical than Boolean satisfiability. Since each sphere must avoid $n - 1$ other spheres and lie within a certain volume, there are altogether n constraints per sphere. The search space requires one D-dimensional variable replica for every sphere participating in a constraint, for a net search space dimensionality of Dn^2.

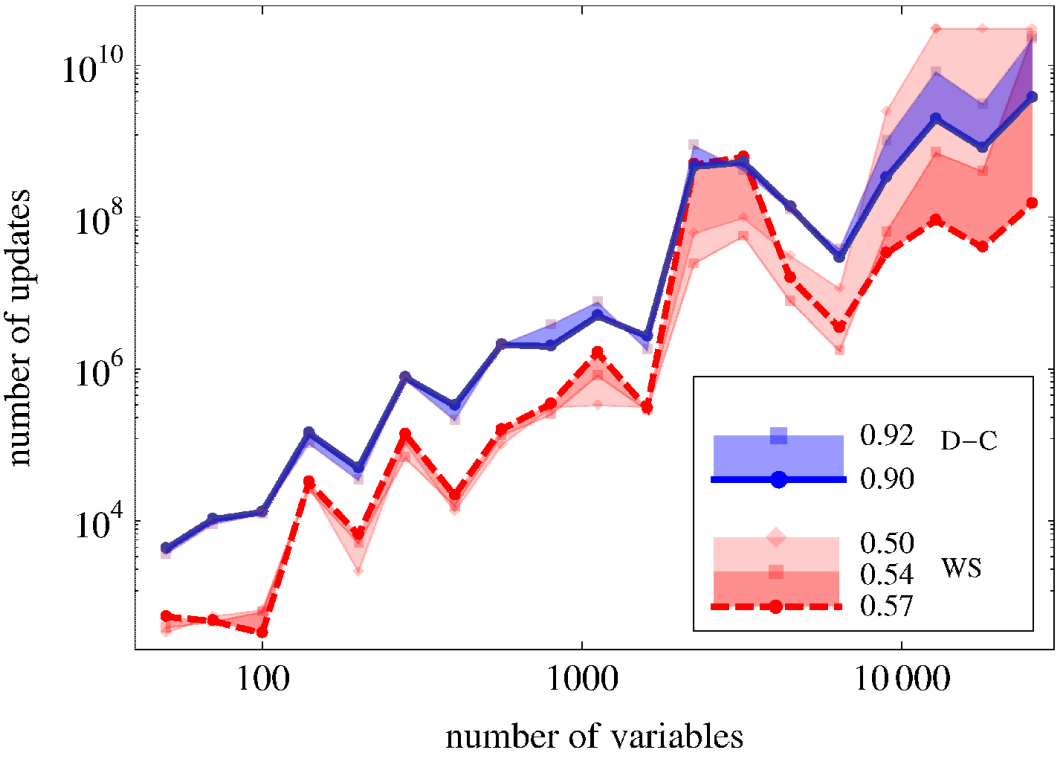

Figure 5.1: Median number of variable updates needed to find a solution for WALKSAT (WS) and divide and concur ($D-C$) on the *same* set of random 3SAT instances with $\alpha = 4.2$. Each median was calculated by solving the same instance 10 times starting from different random initial guesses, for parameter values $\beta = 0.9$ ($D-C$) and $p = 0.57$ (WALKSAT). Variations resulting from changing β and p are indicated by the shaded areas; both methods exhibit parameter sensitivity for problems with more than 10^4 variables. A point at the top edge indicates that the median exceeded the cutoff on the number of updates, 3×10^{10}

Just as every Boolean variable is constrained by each of the clauses where it occurs, every sphere in a packing has a volume exclusion relationship with each of the other spheres in the packing: $\|\mathbf{x}_a - \mathbf{x}_b\| > m_{ab}$. By simply replacing the Boolean OR projections by volume exclusion projections in the $D-C$ scheme, one goes from solving kSAT to solving sphere packing problems. This similarity and the success of $D-C$ with 3SAT is strong motivation to apply $D-C$ to the sphere packing problem.

Near the solution of any n-sphere packing problem, the number of relevant exclusion

constraints (contacting pairs) grows only as n (for fixed D) while the total number of constraints is $O(n^2)$. In the $D-C$ approach it is possible to increase the weight of these relevant pairs by dynamically adjusting the corresponding metric weight λ_{ab}. This results in considerable performance improvement. At the end of each DM step we used $\lambda_{ab} \to \sigma\lambda_{ab} + (1-\sigma)\exp(-\alpha\, d_{ab})$, where d_{ab} is the current distance between the pair[1] We used the value $\sigma = 0.99$ to ensure that the metric update is quasi-adiabatic (i.e., slow on the time scale of variable updates), and $\alpha \simeq 30$.

We first consider the problem of finding the densest packing of n equal disks of diameter m in a unit square. This problem is quite challenging, due to the coexistence of many different arrangements with similar density. We tested the $D-C$ algorithm for each value of n in the range 3-199. For each n, we generated up to 400 random initial guesses. For each initial guess, a small value of the diameter m was chosen, and a packing was sought using $\beta = -1$. When a solution was found, m was increased, and the process was repeated until the algorithm failed to find a packing, or until current the best known packing diameter $m^\star$ (from reference [83]) was reached. In the latter case the target was increased beyond $m^\star$ with the hope of finding a denser packing. No information about the best known packings was used, apart from their densities.

For 143 of the 197 values of n a packing with diameter close to the optimal packing ($m > m^\star - 10^{-9}$) was found. More surprisingly, improved packings were found in 38 cases. The smallest n for which an improved packing was found is 91. The largest improvement was for $n = 182$, for which a packing was found with $m = m^\star + 4.6\times10^{-5}$. For 28 values of n a packing was found with $m > m^\star + 1\times10^{-6}$. An example of such an improved packing is shown in Figure 5.2. Other improved packings can be found in appendix A.

[1] To get a unique pair distance one uses coordinates given by the concurrence term of the difference map, $P_C \circ f_D$.

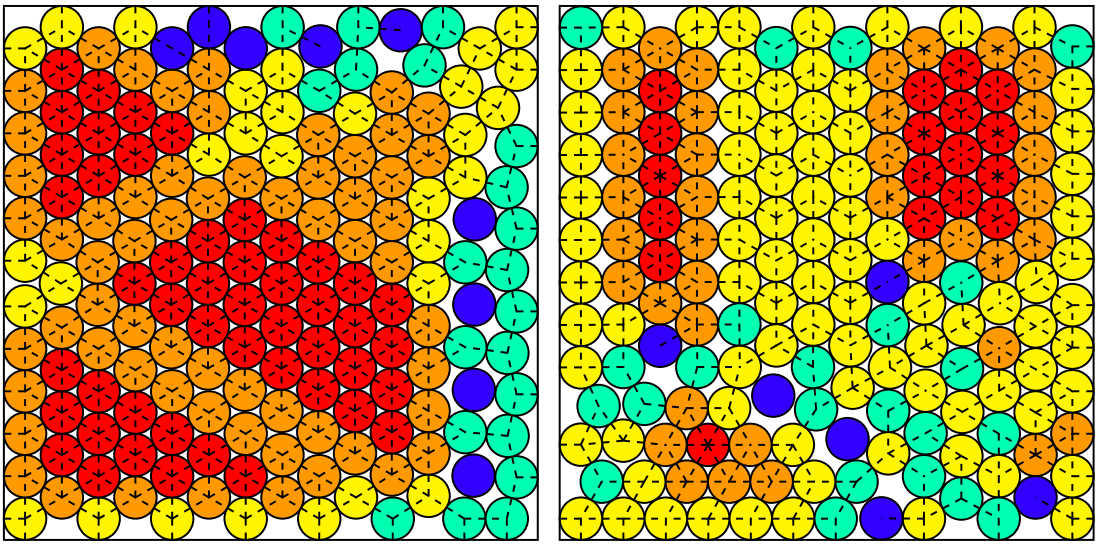

Figure 5.2: An example of an improved packing for 169 disks in a square found by the $D-C$ algorithm. The figure on the left shows the previously best known packing [83], with density 0.8393. The density of the improved packing shown on the right is 0.8399. Contacts are shown with dotted lines; colors indicate the number of contacts.

Note even though the algorithm assigned negligible weights to irrelevant pairs, it nevertheless checked at every iteration for possible overlaps between every single pair in the problem. Efficient handling of irrelevant pairs might therefore result in additional performance improvements.

5.2.2 Kissing numbers

When packing many disks the optimization challenge is easy to identify as a contest between close-packing in the bulk and an efficient match to the boundary. In higher dimensions the structure of the solution is not so easily characterized, and we can look to the $D-C$ method as an unbiased tool for exploration. A classic problem in geometry is to determine *kissing numbers* τ_D: the maximum number of unit spheres that can be packed in D-dimensions, so that each contacts a given unit sphere. Early investigations

of this problem were stimulated by a debate between Newton and Gregory, who disputed the value of τ_3. A thorough review of this problem can be found in Reference [82]. The only known kissing numbers to this day are $\tau_1 = 2$, $\tau_2 = 6$, $\tau_3 = 12$, $\tau_4 = 24$, $\tau_8 = 240$, and $\tau_{24} = 196560$. In dimension 1-8, and also 16-24, the best known lower bounds on τ_D are given by the number of minimal vectors in the unique laminated lattice of the same dimension [82]. For dimension 9-15 the best bounds are obtained from constructions based on error-correcting codes [82]. Discoveries of novel packings in higher dimensions has for the most part been achieved through mathematical inspiration. Unbiased searches, defined only by the basic constraints, have to our knowledge not been attempted beyond dimension 5 [87]. This raises the possibility that interesting packings in high dimensions may have escaped detection only for lack of imagination.

With minimal adjustment to the above procedure for finding disk packings, we were able to find kissing arrangements as good as the best known in dimension 2-4, 6, and 8. After introducing just the assumption of inversion symmetry, optimal packings were obtained in all dimensions up to 8. Our searches in higher dimensions have so far revealed an interesting new packing in dimension 10. It is easy to understand why this packing was missed. Constructions based on integral lattices and error-correcting codes all have the property that the cosine of the angle subtended by any two spheres is rational. The packing of 378 spheres discovered by the $D - C$ algorithm has all cosines in a set that includes irrational numbers: $\{\pm 1, \pm 1/2, (\pm 3 \pm \sqrt{3})/12, 0\}$. An analysis of the coordinates obtained by the algorithm has revealed that these 378 sphere positions are expressible as unique integer multiples of a basis of 12 vectors. The construction has a strong relationship to quasicrystals, where the excess dimension of the basis accounts for irrational relationships in the geometry. This relationship is further explored in reference [88]. The algorithm, of course, had no knowledge of quasicrystal geometry.

This ‘irrational’ structure emerged as soon as the number of spheres was increased above 372, the largest known kissing number for 10-dimensional lattices [82]. The same irrational arrangement was also found for up to 384 spheres; the 6 additional spheres were accommodated in holes of the structure (and have continuously variable cosines). Finally, the algorithm has so far been unsuccessful in discovering the best known kissing arrangement in 10 dimensions, with kissing number 500.

5.3 Hypervertex cover and constrained kSAT problems

Given a graph $G = (V, E)$, where V is a set of N_V vertices and E is a set of N_E unordered pairs of vertices (or *edges*), the N-vertex cover problem is to find a subset $S \subset V$ with cardinality N such that each edge contains at least one vertex that belongs to S. The minimal vertex cover problem is simply the problem of finding the smallest N for which this is possible. This definition generalizes directly to hypergraphs, where the edges are unordered sets of arbitrary number of vertices.

A natural search space for this problem has one variable x_i per node i, with $x_i = 1$ if $i \in S$ and $x_i = 0$ otherwise. The constraint that an edge (x_i, x_j) should be covered is equivalent to a logical OR constraint on the variables x_i and x_j. Finding a cover of a graph (with no restriction on the size of the cover) is therefore a particularly simple example of a kSAT problem, where no variable appears negated. This problem always admits the trivial solution $S = V$. The N-vertex cover problem is therefore a particular modification of the kSAT problem, where an additional constraint imposes that a maximum of N variables are allowed to be True in the solution.

The minimal vertex cover problem is a well-studied NP-complete problem. A review of methods to find minimal vertex covers can be found in [89]. Message-passing

approaches such as belief propagation and survey propagation are particularly efficient at finding small covers for graphs generated from a random ensemble. The work presented in this section was prompted by an article by Mézard and Tarzia [90] describing the application of survey propagation to an ensemble of random regular hypergraphs, where each variable is connected to exactly L edges, and each edge connects exactly K variables. For this ensemble, survey propagation could find smaller covers than belief propagation and a greedy approach. In this section we explain how $D - C$ can be used to find small covers, and find it to be competitive with survey propagation.

There are a few ways to approach the N-vertex cover problem using $D - C$. One can treat the constraint on the number of True variables as a primary contraint. Since this extra constraint involves all N_V variables, this results in the addition of N_V dimensions to the search space. This constraint is equivalent to the 'select-at-least-d' constraint discussed above and can be implemented efficiently.

Another possibility is to modify the concurrence constraint to a 'constrained concurrence' ensuring that at most N variables are True in the solution. This can be done for example by forcing $N - N_V$ variables to be False as in the 'select-at-least-d' constraint. Since the divide constraint already imposes that the variables are discrete, another option for the constrained concurrence can simply require that the sum of the variable values is smaller than N. The projection to the 'constrained concurrence' constraint is straightforward: one first projects to the concurrence constraint, ensuring that each variable has a definite value. The additional constraint on the sum of these values is equivalent to a 'dot product or weighted sum' constraint, described in section 4.4. Here the sum has uniform weight only if all variables occur the same number of times in the problem. This implementation is very similar to the kSAT solver: the only additional code is a few lines in the concurrence constraint operator to enforce the weighted sum constraint.

In order to search for minimal covers (rather than N-covers), a few extra lines can be added to update the target cover size once the previous target has been found.

We have implemented two algorithms, one treating the constraint on the number of True variables as a primary contraint, and the other one using a 'constrained concurrence' imposing a maximum sum to the concurred values. The algorithms are provided with an initial target cover size N_0. When a cover of the appropriate size is found, the target cover size is reduced by one and the search continues from the current iterate value. Giving too high an initial target N_0 sometimes results in the algorithm stalling. We usually started the search with a target cover size N just 3% above the predicted critical value below which covers are no longer expected to exist for this problem [90].

Even though both solvers are able to find covers as small as the ones reported in [90], the solver based on the constrained concurrence is significantly faster and found slightly smaller covers than the one based on additional primary constraint. The performance difference between the two approaches might be partly due to the fact that the constrained concurrence algorithm depends on only one adjustable parameter (β) whereas the one based on additional primary constraint depends on two (β, and the relative weight between the edge covering constraints and the minimality constraint w.) The process of optimizing the parameters was thus simpler in the constrained concurrence approach.

Another possible heuristic explanation for the performance difference is related to the way information propagates on the factor graph representation of these implementation. Adding an extra primary constraint creates longer loops in the problem, slowing down the propagation of information between all constraints.

In the remainder of this section we present the result of the constrained concurrence

implementation of $D-C$, and compare the results to available results from Mézard and Tarzia [90]. As mentioned above, it was found in [90] that belief propagation can find smaller covers compared to a greedy approach, and that survey propagation could find even smaller ones. Detailed results were provided only for an instance with $k = 6$, $L = 4$ and $N_V = 12288$, where the density $\rho = N/N_V$ of the smallest cover found was 0.212 for the greedy approach, 0.186 for belief propagation, and 0.182 for survey propagation. The density threshold at which no more covers are expected to be found is estimated in reference [90] at 0.178.

To generate random hypergraphs with an approximately uniform distribution, we used a method described in [91], and also used in [90]. A hypergraph can be seen as a regular bipartite graph, where nodes and hyperedges are represented as the two types of nodes in the bipartite graph. Each node of the bipartite graph has a number of connections available (k for the 'edge' nodes, and L for the 'node' nodes). Edges are added sequentially to the bipartite graph, and the probability of adding an edge between nodes i and j is proportional to the product $\hat{d}_i\hat{d}_j$ of the remaining free connections $\hat{d}_{i,j}$ of nodes i and j, respectively.

The application of both implementations of $D-C$ to multiple different random instances drawn from the same ensemble allowed to find covers with density 0.182 for problem size $N_V = 12288$, on par with the reported survey propagation result. The $D-C$ algorithm did not use information about the ensemble from which the random graphs were generated. A detailed comparison on identical instances would be in order to draw firm conclusions on the respective merits of each approach. However, the implementation (and in particular the optimization) of SP is not straightforward [92], and we have therefore not yet carried such a comparison. We provide on Figure 5.3 the minimal hypervertex covers found for $k = 6$ and $L = 4$ for various problem sizes.

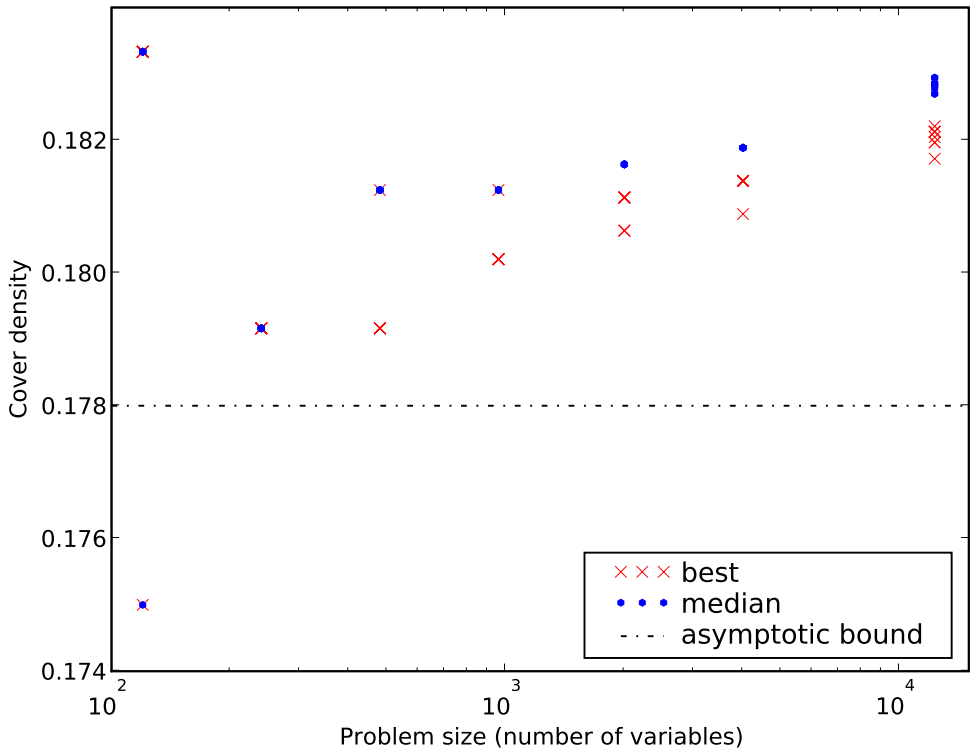

Figure 5.3: Performance of $D - C$ on the hypervertex cover problem for $k = 6$ and $L = 4$ for various problem sizes. For each size 10 instances were generated, and for each instance minimal covers were sought from 20 different random initial conditions. Most points lie on top of each other on this figure, indicating that the search consistently finds the same cover sizes. The maximum number of iterations allowed was 10^6. The asymptotic bound is the density below which problems are expected to have no covers, in the limit of large N_V, according to [90].

Ten different problem instances were generated for each problem size, and each one was solved twenty time with $\beta = -0.85$ and a maximum of 10^6 iterations. The search was also interrupted if the error failed to vary by more than 6% over 2500 consecutive iterations, which in this application happens frequently.

In Figure 5.4, we show the minimal cover size found as function of iteration numbers for twenty restarts for a single instance with $N_V = 12288$. The initial target was $N_0 = 2245$.

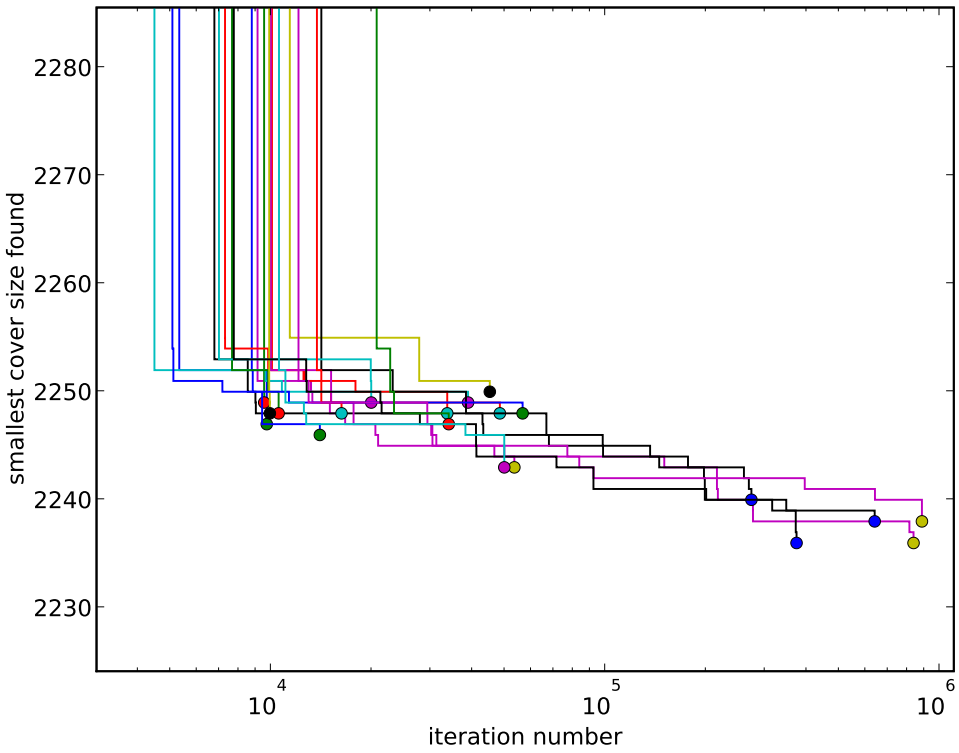

Figure 5.4: Minimum vertex cover found by $D-C$ as a function of iteration number for various initial conditions, for a problem with $k = 6$, $L = 4$, and $N_v = 12288$. The initial target was set to $N_0 = 2245$. The search was interrupted if the error failed to vary by more than 6% over 2500 consecutive iterations, or after 10^6 iterations. The dots indicate the last improvement made for each run. The best reported cover found using SP [90] for an instance of the same size had density 0.182, corresponding to $N = 2236 \pm 6$.

5.4 Low density parity-check codes

Another problem that can be addressed by a slight modification of a $D-C$ kSAT solver is the decoding of low density parity check codes [93]. Low density parity check codes are error-correcting codes, that is, sets of binary sequences (or *code words* that are distributed in such a way that the distance between any two code words is large. If the noise is small enough, a code word transmitted through a noisy channel can be reconstructed exactly from the noisy word received by finding the code word closest to the received

word. This task is referred to as *decoding*, and the low-density error-correcting codes owe part of their success to the fact that such decoding can be achieved efficiently using a belief propagation algorithm.

A low density parity check (LDPC) code is specified by a sparse binary matrix M. Code words are the vectors $\mathbf{x}_i$ that satisfy $M \cdot \mathbf{x}_i = 0 \mod 2$, which can be formulated as a list of binary XOR constraints. The decoding challenge is therefore to retrieve a LDPC word $\mathbf{x}$ of length N that has been transmitted through a noisy channel. It is assumed that the noise in the channel follows a known error distribution. In the simplest model, the error probability for each bit transmitted is p. The task is to retrieve the most likely word $\mathbf{r}$, given received message $\mathbf{y}$ and error probability p. Simple Bayesian analysis shows that the likelihood of word $\mathbf{r}$, given a signal y, is proportional to $p^E(1-p)^{(N-E)}$, where E is the number of differences between $\mathbf{r}$ and $\mathbf{y}$.

We express this problem in terms of a reconstructed error vector $\epsilon = \mathbf{r} - \mathbf{y} \mod 2$. Since $\mathbf{r}$ is a word and therefore satisfies $M \cdot \mathbf{r} = 0 \mod 2$, the problem can be simply formulated as that of finding a binary vector ϵ satisfying $M \cdot \epsilon = M \cdot \mathbf{y} \mod 2$ with the least possible amount of nonzero elements. Since $M \cdot \mathbf{y}$ is known, each row of M imposes a constraint on ϵ. Because we are working in a $\mod 2$ base and M is sparse, each constraint has the form of a short *XOR* statement. The formulation of this problem is therefore similar to the hitting set problem discussed above, with the OR constraints replaced by XOR constraints. However, the solution landscape is expected to be different in both problems: whereas the solutions in random kSAT problems are believed to form clusters, the XOR constraint makes the existence of such clusters of solutions very unlikely. This is precisely why XOR constraints are used to define efficient error-correcting codes.

In this problem we expect a unique solution to exist for each signal transmitted with

a low enough noise. In Figure 5.5 we compare results from belief propagation (BP) , as implemented in the package PYCODES [94] to $D-C$ for 150 bits words, with 150 checks involving 3 variables each, and $p = 0.27$. In the $D-C$ implementation the maximum number of allowed errors was set to be uniformly 48, independent of the actual number of errors, which is assumed unknown by the decoder. In this problem, as in the vertex cover problem, two implementations are possible, depending on whether the constraint on the number of errors is imposed as a primary constraint or as a modification of the concur constraint. Since the latter approach was the most successful in the hypervertex cover problem, it would be interesting to try it also on LDPC codes. However, we have not yet carried such calculations, and we present below results for the former approach.

We used $\beta = .8$ and set the weight of the 'number of errors' constraint to be 0.3 times that of the XOR constraints. The search was interrupted when a valid codeword was found regardless of whether it satisfied the 'number of errors' constraint. Imposing a smaller maximum number of errors reduces iteration count significantly for lower error numbers.

The two approaches exhibit different behavior as the number of variables is varied. The BP approach is efficient for decoding words with little error, while its performance degrades abruptly when the number of errors passes a threshold. Beyond this threshold only the $D-C$ succeeds at decoding words reliably. This remains true if the expected error probability is varied.

BP converges efficiently towards the solution in easier problems, and the $D-C$ approach succeeds at finding the solution in more difficult problems. This suggests the use of a hybrid approach, starting the search with *BP* iterations and following suit with $D-C$. Results with this hybrid approach, which can be seen as a preconditioning of the $D-C$ search with BP iterations, are also shown on Figure 5.5. The method matches,

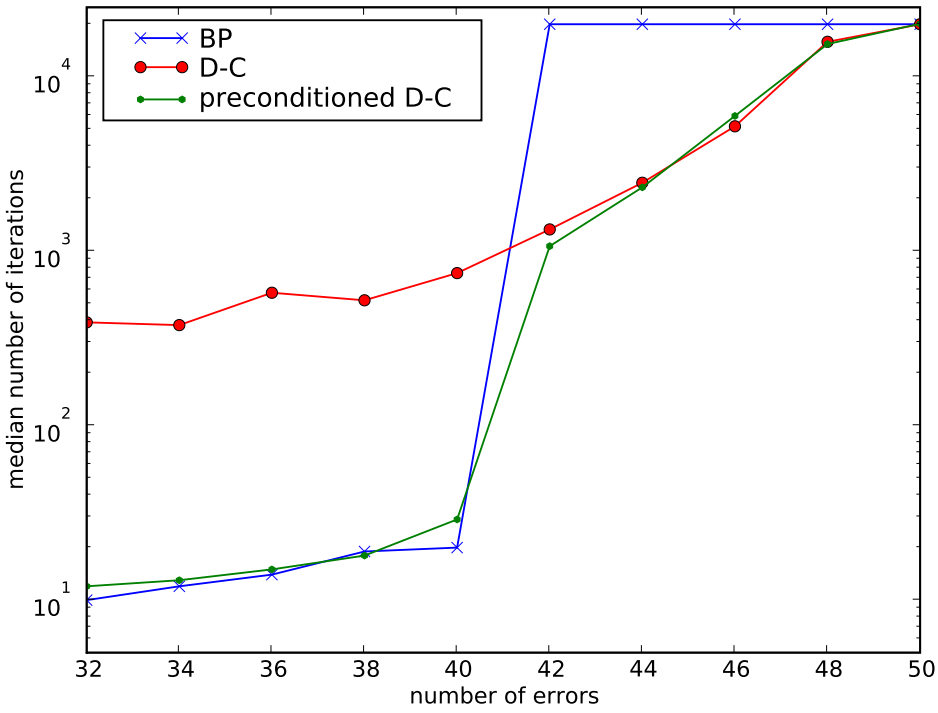

Figure 5.5: Median number of iterations for belief propagation (blue) and divide and concur (red) on a set of decoding problems using Gallager low-density parity check codes. Each data points is the median iteration number for ten decoding attempts, each for different codes at fixed error numbers. Codes were generated for 150 variables and 150 parity checks, each check involving 3 variables. The error probability was taken to be $p = 0.27$, corresponding to an average of 40.5 errors. The green curve represents the total iterations of a mixed solver, preconditioning the $D - C$ search with at most 120 BP iterations. Points at the upper right have reached the cutoff of 20000 iterations without finding a solution.

but does not improve significantly upon, the best of the two approaches. In this case, preconditioning the $D - C$ search did not result in faster convergence of $D - C$.

5.5 Ground state of particles interacting via a pairwise potential

In this section we provide the outline of an application we mentioned in sections 4.3.2 and 4.4.2, without results, to provide an example of how an energy minimization problem can be treated through a variation of divide and concur.

The application is finding low energy states of a set of n particles interacting through pairwise potentials. As usual the optimization problem is formulated as a constraint problem by using an energy target E_T. The total energy is divided into $n(n-1)/2$ contributions from each pair. Each pair is assigned its own variables, so that each particle has $n - 1$ replicas, and

$$E = \sum_{i<j} \phi_p(\mathbf{x}_{ij} - \mathbf{x}_{ji}).$$

As usual, the concurrence constraint enforces that all replicas of a given variable are in agreement: $\mathbf{x}_{i1} = \mathbf{x}_{i2} = \cdots = \mathbf{x}_{in}$. The divide constraint enforces the energy target condition in this extended variable space. Because of the introduction of the replicas, the energy of each pair can be adjusted independently: the energy constraint is now a constraint over the total energy of $n(n+1)/2$ pairs living in different variable spaces. However, because we are looking for a projection to a configuration where the *total* energy is below E_T, we still need to determine how much of the energy reduction $E - E_T$ should be assigned to each pair in order to minimize the distance to the constraint set.

For some choices of pair potentials this is rather straightforward. Consider for ex-

ample n identical particles interacting via a step potential:

$$V_{ij}(r) = \begin{cases} \infty, & \text{if } r < r_0 \\ -|a|, & \text{if } r_0 < r < r_1 \\ 0, & \text{otherwise.} \end{cases} \tag{5.1}$$

The constraint on the target energy is equivalent to a constraint on the number of particles at distance smaller than r_1. The projection to this constraint is easy to implement and is discussed in section 4.4.2.

This provides an simple example of a restraint problem which can be reformulated in terms of simple constraints, and be amenable to a projection-based solving strategy.

CHAPTER 6

CONCLUSION

We emphasized in the text a similarity between patterns occurring in the two projects discussed in this dissertation. These patterns are illustrated on Figures 2.10 and 4.10. The former describes the optimal way to 'turn on' a pair of perturbations to an initially well-understood Hamiltonian in order to retrieve the ground state energy, whereas the latter (reproduced here on Figure 6.1) describes the optimal way to retrieve, from an arbitrary starting point, the intersection of two linear spaces by using projection operators. In both cases, different estimates to the quantity of interest are obtained by following the three different 'trajectories' Γ_1, Γ_2, and Γ_3, each trajectory corresponding to a specific order in which the different building blocks of the problem are taken into account. Different approximations can be used along each trajectory, taking advantage of the relative simplicity of the individual building blocks of the problem.

When looking for a linear combination of the three resulting estimates that ensures optimal error cancellation, a single combination stands out, and corresponds to the path $Z = \Gamma_1 - \Gamma_2 + \Gamma_3$. The fact that the same trajectory is optimal in the two different projects described in this dissertation hints at broader applicability. The most likely candidates for the use of trajectory Z are difficult problems built from symmetric building blocks or, more precisely, problems where considering any building block independently results in a simple problem, but where the simultaneous treatment of all building blocks is challenging.

This interesting similarity aside, the two projects discussed in this thesis have little in common. We now move on to conclusions and open questions concerning the individual projects.

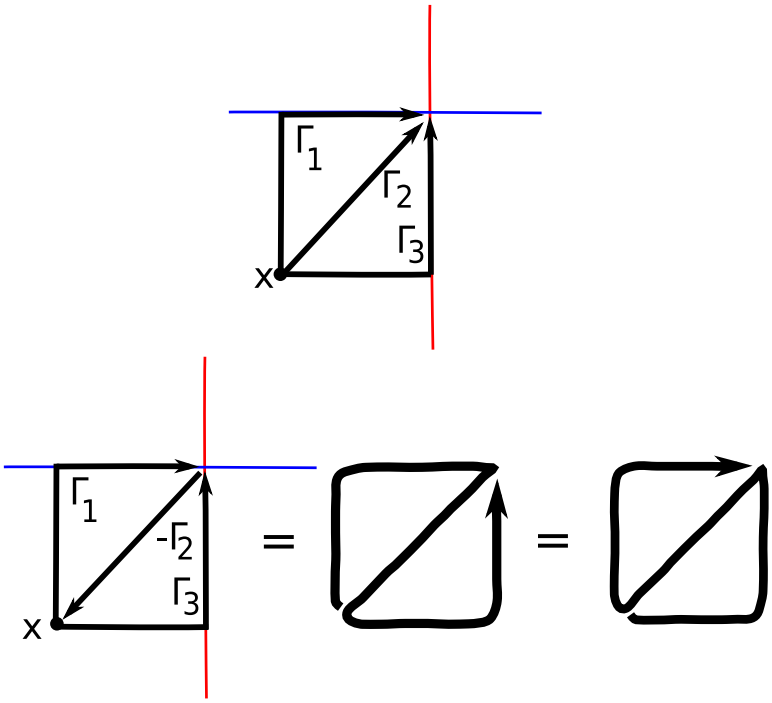

Figure 6.1: The Z-box pattern, recurring in this dissertation. (top) Three different simple trajectories to reach the intersection of the blue and red lines, starting from point x. (bottom) Order in which these paths should be travelled to ensure maximal error cancellation in the two projects discussed in this dissertation.

6.1 Symmetrized successive perturbation method

In Chapter 2, we derived and discussed the symmetrized successive perturbation (SSP), an interesting identity relating the variations in electronic density induced by the presence of individual ions in an electron gas to the effective, electron-mediated interaction between the same ions. We showed that this identity could be used to obtain more precise effective pair potentials than those derived from quadratic response theory alone. Discrepancies were observed between results from SSP and ab-initio methods for the effective pair potentials between hydrogen atoms in metallic environments, emphasizing the difficulty of treating protons as weak perturbations of the electron gas.

Perturbation theory is nowadays mostly used to understand qualitative features of

metallic systems. The fact that SSP relates relates such physically meaningful and intuitive quantities as induced electronic density and effective interactions is therefore interesting in its own right. The parallel between the perturbative approach and the Heitler-London model of the hydrogen molecule we outlined in section 2.3.3, for example, would not have been so easily noticeable in a straightforward perturbation approach, nor the fact that a cutoff of perturbation theory at arbitrary order systematically underrepresents electron-electron interactions.

The main advantage of SSP is that it allows to incorporate symmetric quantities (the induced densities of individual atoms) into the calculation of quantities with fewer symmetries (the pair potential). Induced densities from Kohn-Sham DFT, or even from experiments, can be used in conjunction with SSP to obtain simple expressions for the effective ionic interactions. This is, in our opinion, the most promising possible application of the results from Chapter 2.

In this dissertation we have only presented results for SSP at linear order in perturbation, and for pair potentials. The new interpretation of SSP in terms of path integrals presented in Section 2.6.1 simplifies considerably the task of carrying SSP to higher order in perturbation and for many-center potentials, both for hydrogen and for other systems. This might allow to resolve the discrepancy between SSP and ab-initio results for hydrogen-hydrogen pair potentials.

The path integral interpretation of the SSP result did not explain the simplicity of the electron-electron interaction term (equation (2.12)). The proof that this contribution took a simple form required significant diagrammatic bookkeeping. Obtaining a simple argument for the simplicity of this term would be a useful step forward, and might result in further generalizations of the SSP results.

6.2 Divide and concur

We presented in Chapter 4 the divide and concur ($D-C$) approach to the solution of constraint satisfaction problems. We discussed some discrete and continuous applications in Chapter 5 and showed that for some important problems, in particular Boolean satisfaction and disk packing problems, $D-C$ competes with or improves upon the state-of-the-art methods. Many open questions remain about the performance and behavior of $D-C$. As a conclusion to this work, we present a few research directions which could help answer some of these questions.

We have mentioned in the text a number of applications where $D-C$ is applicable, without providing much numerical data. There is therefore a natural continuation of this work, which simply involves applying $D-C$ to more standard constraint problems and compare its performance with existing algorithms.

Additional work is also needed in order to accurately assess the performance of $D-C$ on standard benchmark problems. For example, even though the $D-C$ algorithm exhibits similar scaling behavior to WALKSAT in the solution of random $3SAT$ problems, the actual solution time of our implementation of $D-C$ is significantly worse than that of WALKSAT. This could be due either to an intrinsic superiority of the WALKSAT strategy for this particular problem, or to a better optimization of WALKSAT code. A careful optimization of divide and concur is in order to obtain a more detailed performance comparison, particularly for very large problems where optimal memory use is crucial.

The process of dynamically updating constraint weights, described in section 4.8.3, still needs to be better understood. Such a dynamic weight updating resulted in considerable performance improvement in the disk packing problem, but not in discrete problems such as kSAT. Whether efficient weight update schemes can be implemented

for kSAT remains an open problem. More generally, obtaining an efficient, generally applicable update mechanism for the constraint weights would be useful.

The efficiency of $D - C$ also depends strongly on the availability of efficient projection operators to individual constraint sets. In Section 4.4 we provided a list of constraints for which simple projection operators could be defined, and observed that a common feature of these constraints was that they exhibited simple symmetries. It would be interesting to determine whether a simple relation between symmetry and solvability exists, as is the case for exactly solvable Hamiltonians. Then one could envision a systematic classification of constraint spaces in terms of their symmetries, similarly to the program aimed at classifying integrable Hamiltonians.

Theoretical questions aside, the success of a constraint satisfaction approach should not be measured only by its performance on standard benchmark problems, but also by how successful it is in solving practical constraint satisfaction problems occurring in research or in industry. The fact that the difference map was originally designed to tackle such a practical research problem, namely the phase retrieval problem in diffraction imaging, indicates that projection-based algorithms can be applied successfully to real-life constraint satisfaction problems. Other experimental setups requiring numerical data assembly are therefore likely candidates for practical applications of divide and concur. Examples discussed in the text are protein structure determination through NMR spectroscopy and the determination of the structure of protein complexes by combining a variety of measurements [95]. The configuration space to be explored in these problems is large. Furthermore, the cost of acquiring additional data imposes limits on how overconstrained the reconstruction problem can become. The combination of large search space and slightly overconstrained problems makes for very challenging reconstruction and requires very efficient search heuristics. $D - C$, which can handle large

problems near the overconstrained to underconstrained transition region, is therefore a promising approach to such problems.

APPENDIX A

DISK PACKINGS

In this chapter we provide a list of the improved packings of disks in a square mentioned in section 5.2.1. In the diagrams below N is the number of disks, and m is the diameter of the circles, assuming the *center* of the disks are constrainted to a unit box. The outside bounding box therefore has sides $1 + m$. Dashed lines join disks contacting, to within working precision. The color coding emphasizes the number of contacts per disk: red for 6 neighbors, orange for 5, yellow for 4, and aqua for 3. Two or less contacts are similarly colored blue, since a sphere with two or less contacts can always be moved slightly and be made contactless. Some configurations (such as that for N=91) are minor modifications of packings previously found. The majority are significantly different from previously found packings. Decimals of m whose value is different in the improved packing from what it was in the previously best known packing are indicated in red. Last digits are rounded down.

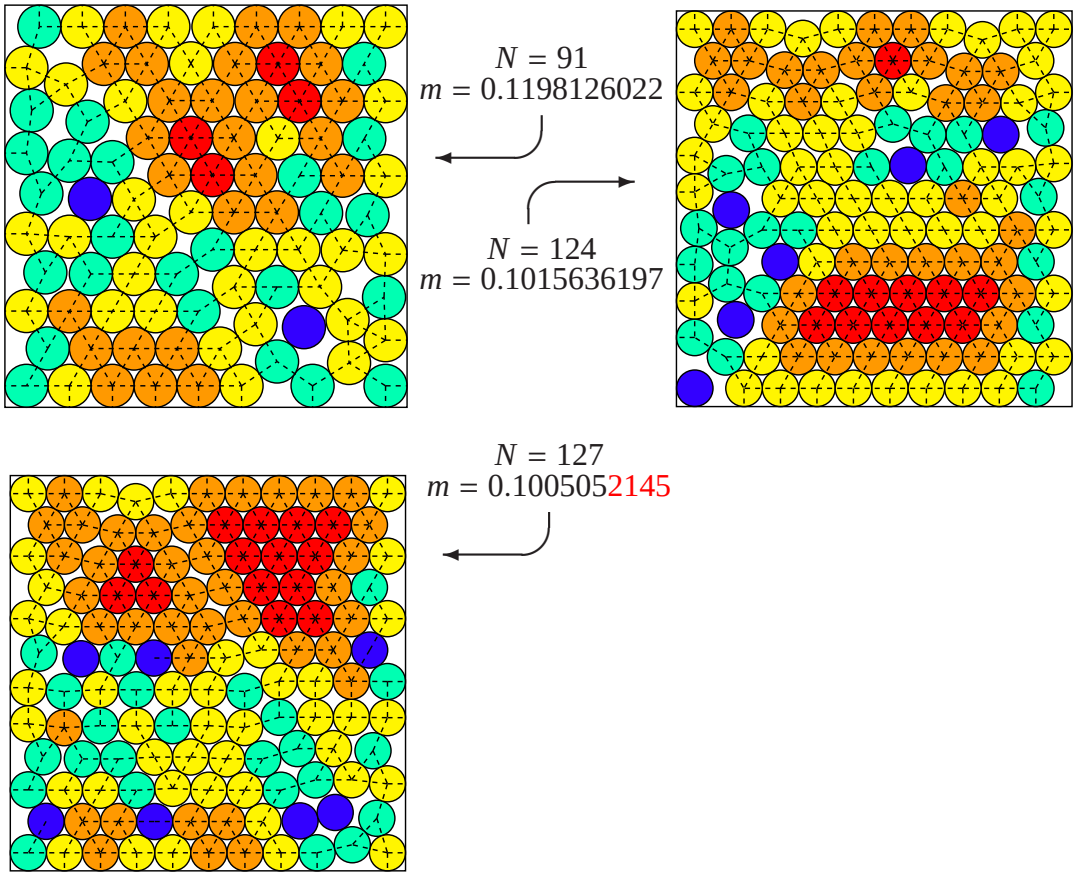

Figure A.1: Improved packings of 91 to 127 disks in a square.

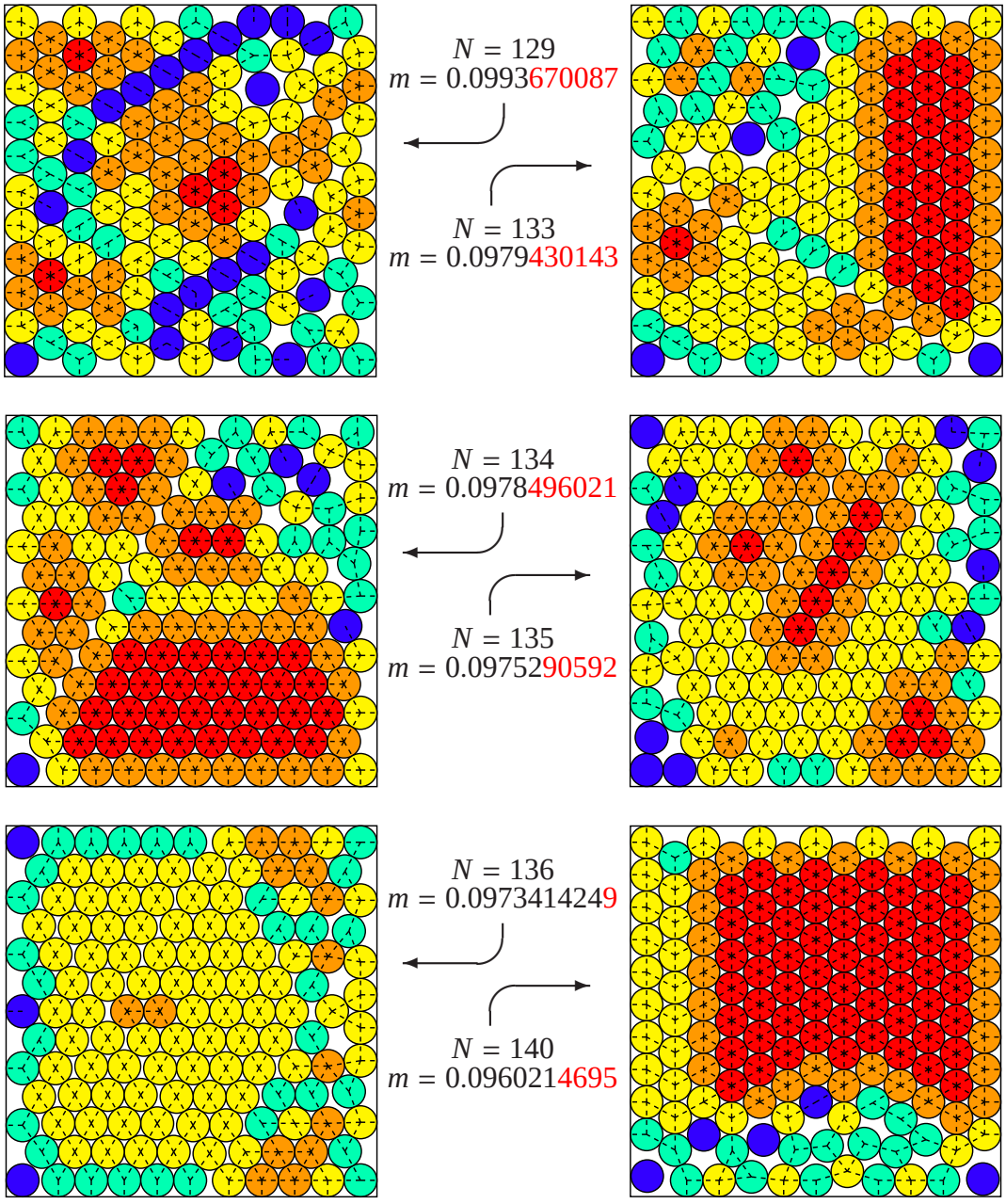

Figure A.2: Improved packings of 129 to 140 disks in a square.

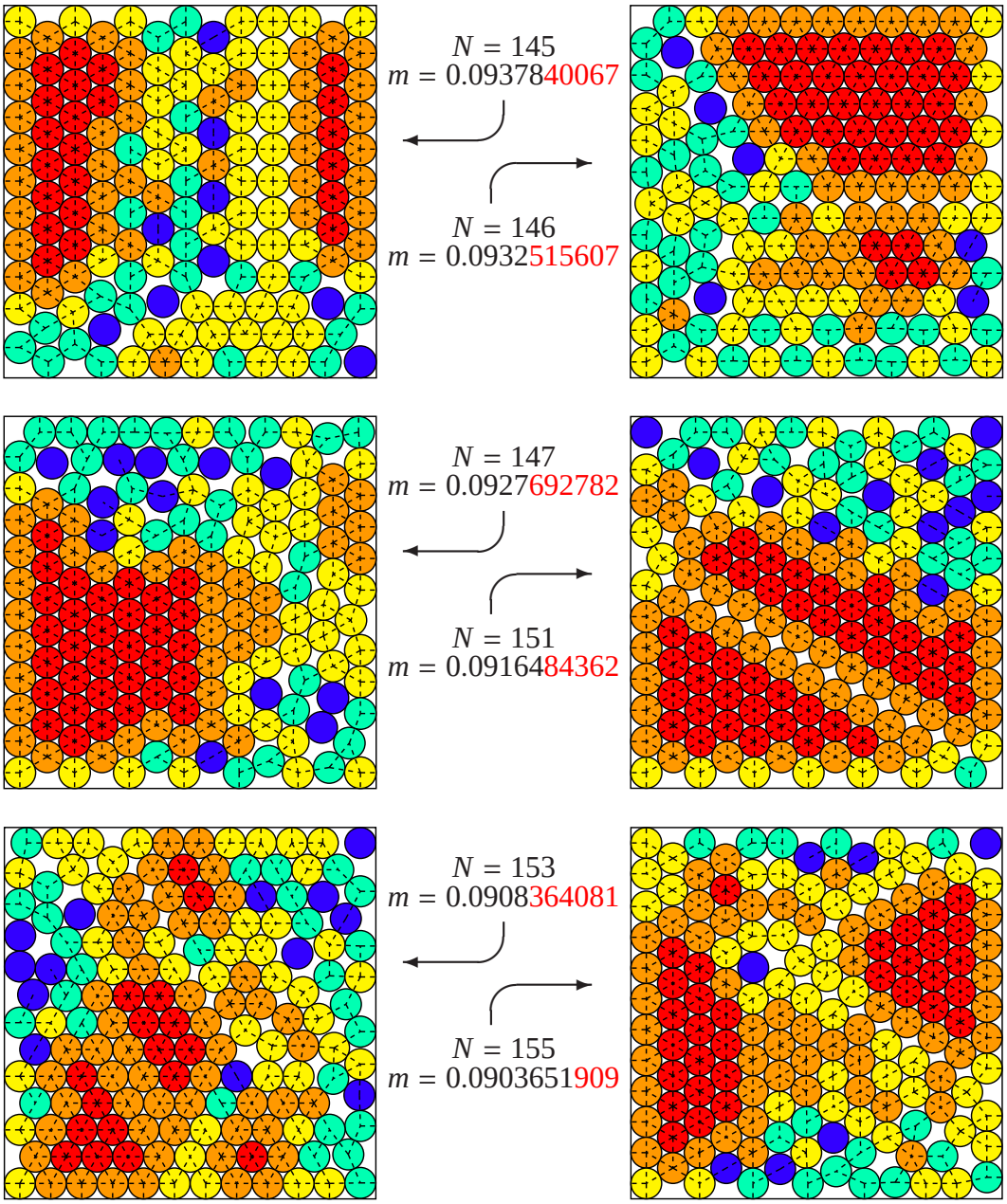

Figure A.3: Improved packings of 145 to 155 disks in a square.

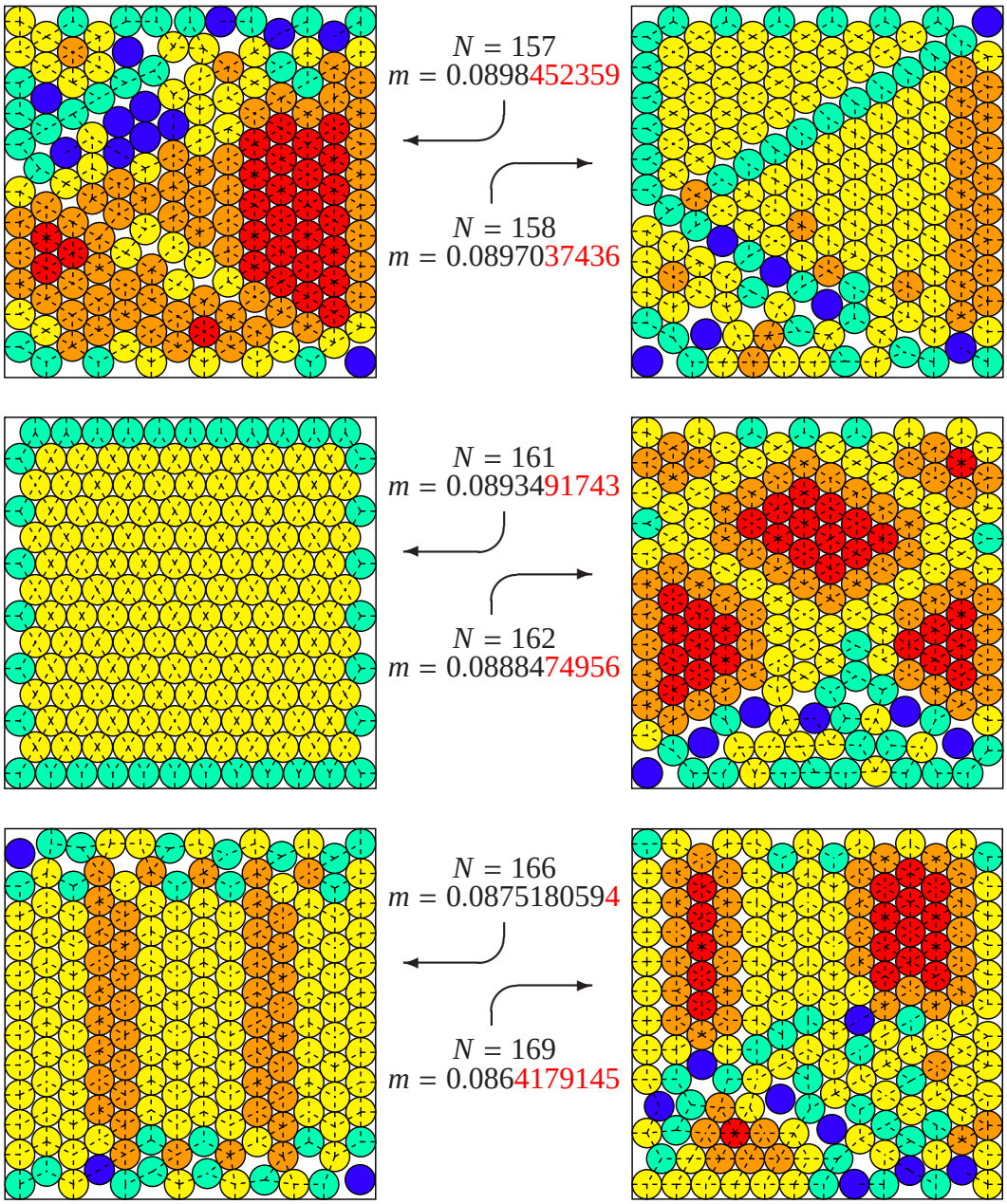

Figure A.4: Improved packings of 157 to 169 disks in a square.

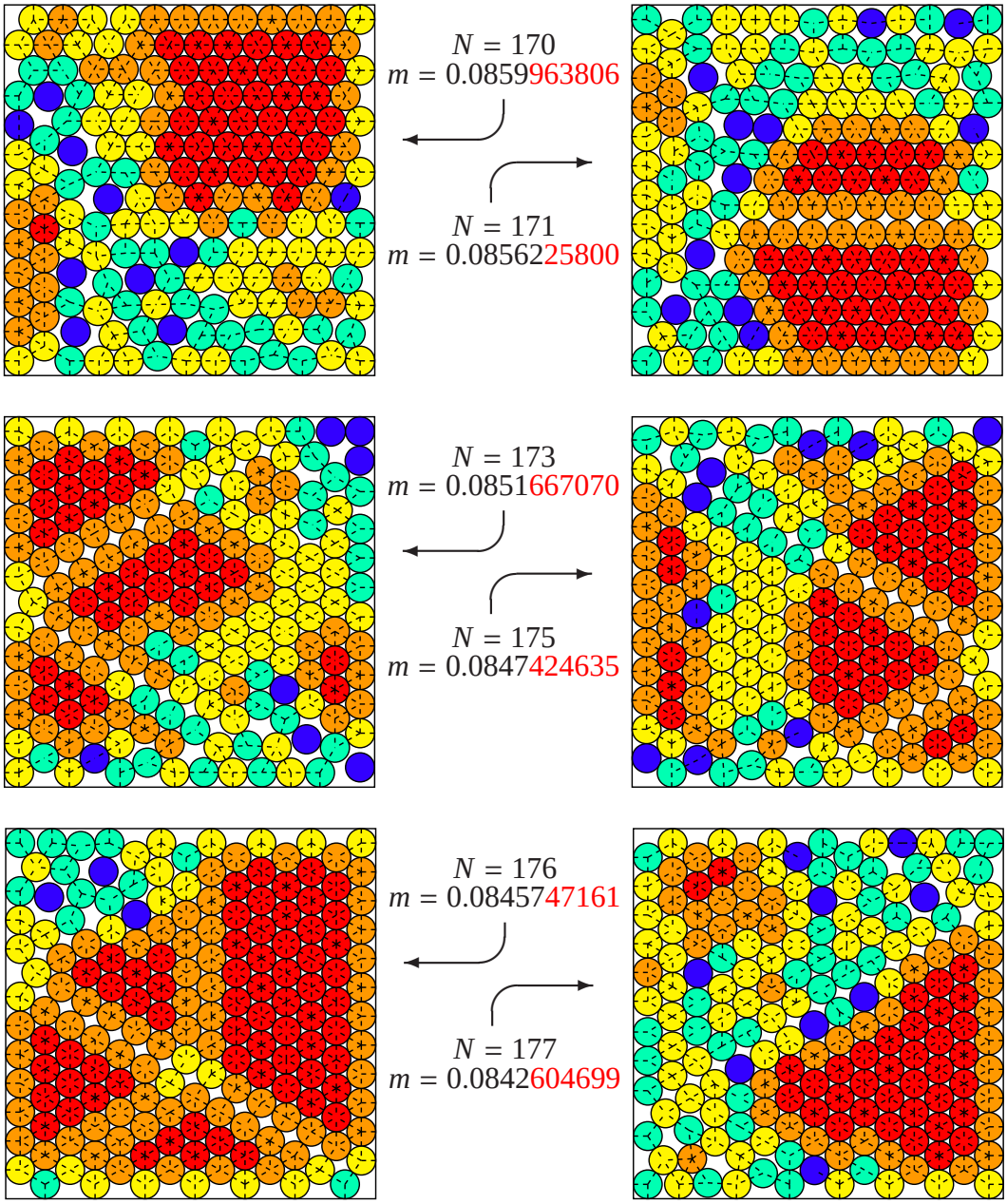

Figure A.5: Improved packings of 170 to 177 disks in a square.

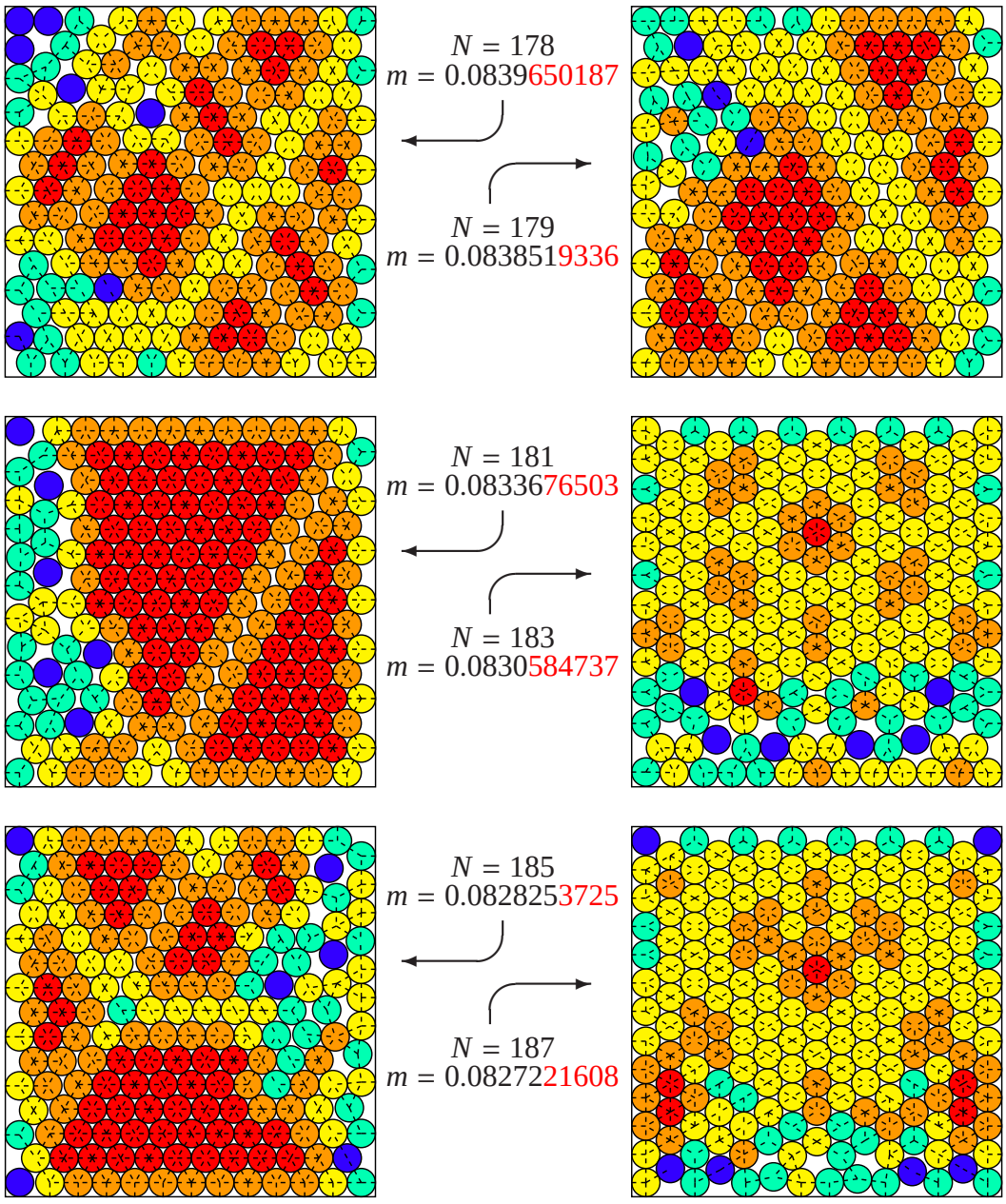

Figure A.6: Improved packings of 178 to 187 disks in a square.

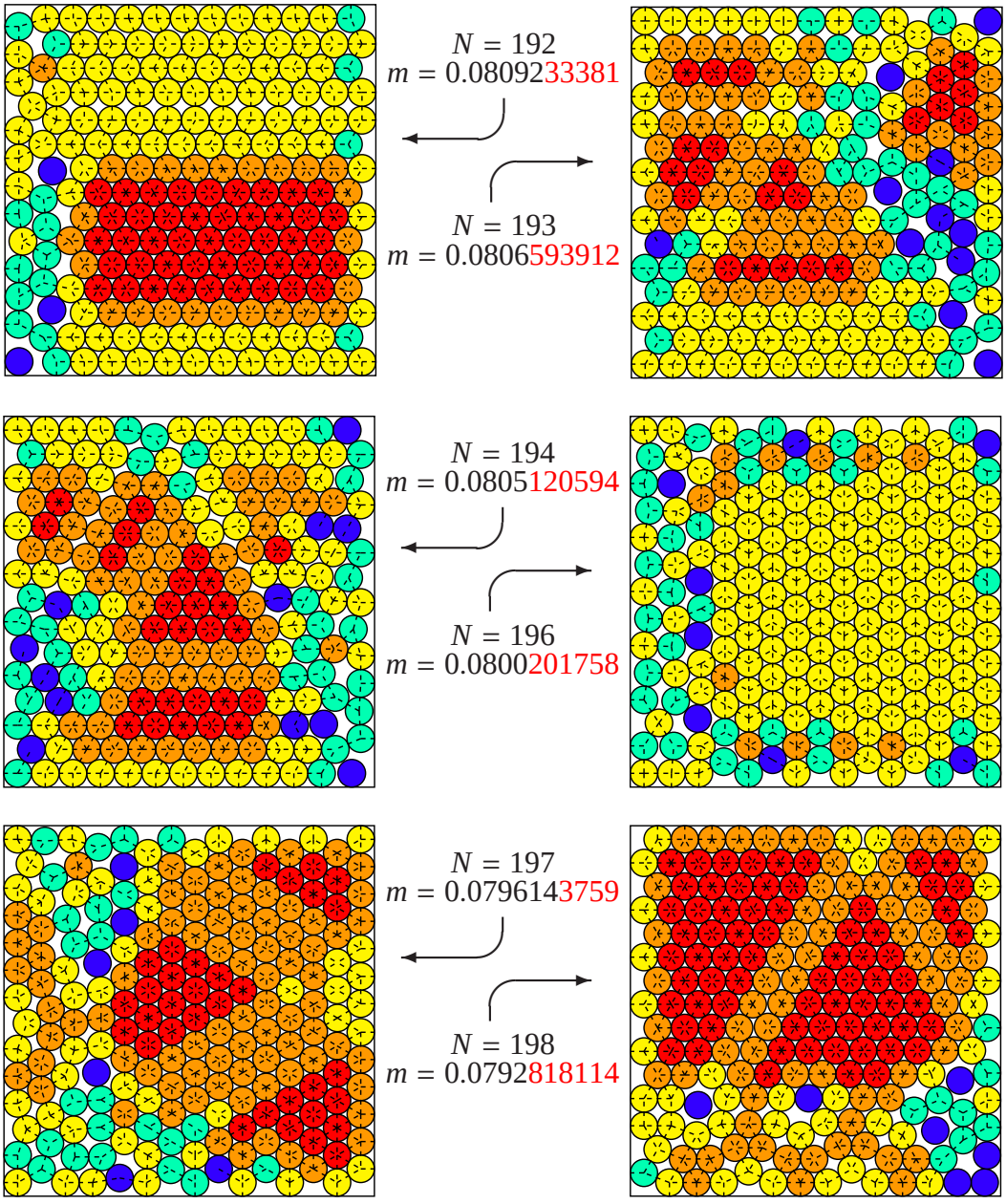

Figure A.7: Improved packings of 192 to 198 disks in a square.

BIBLIOGRAPHY

[1] E. G. Brovman and Yu. M. Kagan. Phonons in nontransition metals. *Soviet Physics Uspekhi*, 17:125, 1974. (Usp. Fiz. Nauk., 112: 369, 1974).

[2] J. Miao, P. Charalambous, J. Kirz, and D. Sayre. Extending the methodology of X-ray crystallography to allow imaging of micrometre-sized non-crystalline specimens. *Nature*, 400(6742):342, 1999.

[3] D. Shapiro, P. Thibault, T. Beetz, V. Elser, M. Howells, C. Jacobsen, J. Kirz, E. Lima, H. Miao, A. M. Neiman, and D. Sayre. Biological imaging by soft x-ray diffraction microscopy. *PNAS*, 102(43):15343, 2005.

[4] J. Cavanagh. *Protein NMR Spectroscopy: Principles and Practice*. Academic Press, 1996.

[5] S. Gravel and N. W. Ashcroft. Nonlinear response theories and effective pair potentials. *Phys. Rev. B*, 76(14):144103, 2007.

[6] Marvin Cohen, Volker Heine, and D. Weaire. The fitting of pseudopotentials to experimental data and their subsequent application. *Solid State Physics*, 24, 1970.

[7] Jürgen Hafner. *From Hamiltonians to Phase Diagrams*. Springer-Verlag, 1987.

[8] Charles Kittel. *Solid State Physics*. Academic, New York, 1968. p. 1 see also erratum in Appendix A (p.295).

[9] B. V. Derjaguin and L. Landau. Theory of the stability of strongly charged lyophobic sols and of the adhesion of strongly charged particles in solutions of electrolytes. *Acta Physicochim. URSS*, 14:633, 1941.

[10] E. J. W. Verwey and J. T. G. Overbeek. *Theory of the Stability of Lyophobic Colloids*. Dover Publications, 2000.

[11] S. Asakura and F. Oosawa. On interaction between two bodies immersed in a solution of macromolecules. *J. Chem. Phys*, 22(7):1255, 1954.

[12] S. Asakura and F. Oosawa. Interaction between particles suspended in solutions of macromolecules. *J. Polym. Sci.*, 33(126):183, 1958.

[13] W. Kohn and LJ Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev*, 140(4A):A1133–8, 1965.

[14] G. G. N. Angilella, F. Siringo, and R. Pucci. Symmetry breaking and restoring under high pressure: the amazing behaviour of the ”simple” alkali metals. *Eur. Phys. J.*, 32:323, 2003.

[15] J. M. Pitarke, R. H. Ritchie, and P. M. Echenique. Quadratic response theory of the energy loss of charged particles in an electron gas. *Phys. Rev. B*, 52(19):13883, 1995.

[16] A. Lodder. Calculations of the screening of the charge of a proton migrating in a metal. *Phys. Rev. B*, 74(4):045111, 2006.

[17] S. A. Kauffman. *The origins of order: Self-Organization and Selection in Evolution*. Oxford University Press New York, 1993.

[18] E. G. Maksimov, M. V. Magnitskaya, and V. E. Fortov. Non-simple behavior of simple metals at high pressure. *Physics-Uspekhi*, 48(8):761, 2005.

[19] Bruno Rousseau and N. W. Ashcroft. Interstitial electronic localization. *Phys. Rev. Lett.*, 101(4):046407, 2008.

[20] S. T. Weir, A. C. Mitchell, and W. J. Nellis. Metallization of Fluid Molecular Hydrogen at 140 GPa (1.4 Mbar). *Phys. Rev. Lett.*, 76(11):1860, 1996.

[21] Paul Loubeyre, Florent Occelli, and René LeToullec. Optical studies of solid hydrogen to 320 GPa and evidence for black hydrogen. *Nature*, 416(6881):613, 2002.

[22] E. Babaev, A. Sudboe, and N. W. Ashcroft. A superconductor to superfluid phase transition in liquid metallic hydrogen. *Nature*, 431(7009):666, 2004.

[23] N. W. Ashcroft. Hydrogen dominant metallic alloys: High temperature superconductors? *Phys. Rev. Lett.*, 92(18):187002, 2004.

[24] A. A. Louis and N. W. Ashcroft. Extending linear response: Inferences from electron-ion structure factors. *Phys. Rev. Lett.*, 81(20):4456–4459, 1998.

[25] Kazutaka Nagao, S. A. Bonev, A. Bergara, and N. W. Ashcroft. Enhanced Friedel structure and proton pairing in dense solid Hydrogen. *Phys. Rev. Lett.*, 90(3):035501, 2003.

[26] S. A. Bonev and N. W. Ashcroft. Hydrogen in jellium: First-principles pair interactions. *Phys. Rev. B*, 64(22):224112, 2001.

[27] K. W. Jacobsen, J. K. Nørskov, and M. J. Puska. Interatomic interactions in the effective-medium theory. *Phys. Rev. B*, 35(14):7423, 1987.

[28] J. M. Ziman. *Electrons and phonons*. Oxford University Press New York, 2001.

[29] David Pines and Philippe Nozières. *The theory of quantum liquids I*. Benjamin, New York, 1966. p. 297.

[30] R. Elber. *Recent Developments in Theoretical Studies of Proteins*. World Scientific, 1996. p.43.

[31] Z. D. Popovic, M. J. Stott, J. P. Carbotte, and G. R. Piercy. Theory of the heat of solution of hydrogen in Al and Mg using nonlinear screening. *Phys. Rev. B*, 13(2):590, 1976.

[32] Y. Yafet. Ruderman-Kittel-Kasuya-Yosida range function of a one-dimensional free-electron gas. *Phys. Rev. B*, 36(7):3948, 1987.

[33] Gabriele F. Giuliani, Giovanni Vignale, and Trinanjan Datta. RKKY range function of a one-dimensional noninteracting electron gas. *Phys. Rev. B*, 72(3):033411, 2005.

[34] M. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Table*. Courier Dover Publications, 1965.

[35] Adriaan Anthony Louis. *Quantum dissipation from phonons; Metallic hydrogen; Electron-ion mixtures*. PhD thesis, Cornell University, 1998.

[36] A. Galindo, I. Nagy, R. Díez Muiño, and PM Echenique. Curvature of the total electron density at critical coupling. *Phys. Rev. B*, 72(12):125113, 2005.

[37] J. K. Nørskov. Electron structure of single and interacting hydrogen impurities in free-electron-like metals. *Phys. Rev. B*, 20(2):446, 1979.

[38] R. Díez Muiño and A. Salin. Self-consistent screening of diatomic molecules in an electron gas. *Phys. Rev. B*, 60(3):2074, 1999.

[39] O. B. Christensen, P. D. Ditlevsen, K. W. Jacobsen, P. Stoltze, O. H. Nielsen, and J. K. Nørskov. H-H interactions in Pd. *Phys. Rev. B*, 40(3):1993, 1989.

[40] G. Kresse and J. Furthmüller. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. Rev. B*, 54(16):11169, 1996.

[41] G. Kresse and J. Furthmüller. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Computational Materials Science*, 6(1):15, 1996.

[42] G. Kresse and D. Joubert. From ultrasoft pseudopotentials to the projector augmented-wave method. *Phys. Rev. B*, 59(3):1758–1775, 1999.

[43] P. E. Blochl. Projector augmented-wave method. *Phys. Rev. B*, 50(24):17953–17979, 1994.

[44] JP Perdew. Electronic Structure of Solids 91. *Akademie-Verlag, Berlin*, page 11, 1991.

[45] M. Methfessel and A. T. Paxton. High-precision sampling for Brillouin-zone integration in metals. *Phys. Rev. B*, 40(6):3616, 1989.

[46] A. Milchev and R. Pickenhain. The quadratic response of a Fermi gas. *Phys. Status Solidi*, 79:549, 1977.

[47] P. Vashishta and K. S. Singwi. Electron Correlations at Metallic Densities. V. *Phys. Rev. B*, 6(3):875, 1972.

[48] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev*, 136(864):B864, 1964.

[49] N. D. Mermin. Thermal properties of the inhomogeneous electron gas. *Phys. Rev*, 137:A1441, 1965.

[50] S. Lundqvist and N.H. March. *Theory of the inhomogeneous electron gas*. Plenum Press, 1983. p.124.

[51] A.C. Hewson. *The Kondo Problem to Heavy Fermions*. Cambridge University Press, 1997.

[52] James Porter. *Pair and triplet interactions in wide band systems*. PhD thesis, Cornell University, 2001.

[53] David W. Sims, Emily J. Southall, Nicolas E. Humphries, Graeme C. Hays, Corey J. A. Bradshaw, Jonathan W. Pitchford, Alex James, Mohammed Z. Ahmed, Andrew S. Brierley, Mark A. Hindell, David Morritt, Michael K. Musyl, David Righton, Emily L. C. Shepard, Victoria J. Wearmouth, Rory P. Wilson, Matthew J. Witt, and Julian D. Metcalfe. Scaling laws of marine predator search behaviour. *Nature*, 451(7182):1098, 2008.

[54] S. Goss, S. Aron, J. L. Deneubourg, and J. M. Pasteels. Self-organized shortcuts in the Argentine ant. *Naturwissenschaften*, 76(12):579, 1989.

[55] O. G. Berg, R. B. Winter, and P. H. Von Hippel. Diffusion-driven mechanisms of protein translocation on nucleic acids. 1. Models and theory. *Biochemistry*, 20(24):6929, 1981.

[56] M. Davis and H. Putnam. A Computing Procedure for Quantification Theory. *Journal of the ACM (JACM)*, 7(3):201, 1960.

[57] M. Davis, G. Logemann, and D. Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5(7):394, 1962.

[58] B. Selman, H. Kautz, and B. Cohen. Local search strategies for satisfiability testing. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 26:521, 1996.

[59] S. Seitz, M. Alava, and P. Orponen. Threshold Behaviour of WalkSAT and Focused Metropolis Search on Random 3-Satisfiability. *Proc. 8th Intl. Conf. on Theory and Applications of Satisfiability Testing*, 2005.

[60] E. H. L. Aarts and J. K. Lenstra. *Local Search in Combinatorial Optimization*. Princeton University Press, 2003.

[61] M. Alava, J. Ardelius, E. Aurell, P. Kaski, S. Krishnamurthy, P. Orponen, and S. Seitz. Circumspect descent prevails in solving random constraint satisfaction problems. *arXiv:0711.4902*, 2007.

[62] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. *Exploring Artificial Intelligence in the New Millennium*, page 239, 2003.

[63] M. Mézard, G. Parisi, and R. Zecchina. Analytic and Algorithmic Solution of Random Satisfiability Problems. *Science*, 297(5582):812, 2002.

[64] A. Braunstein, M. Mezard, and R. Zecchina. Survey propagation: an algorithm for satisfiability. *Random Structures and Algorithms*, 27(2):201, 2005.

[65] Heinz H. Bauschke and Jonathan M. Borwein. On projection algorithms for solving convex feasibility problems. *SIAM Review*, 38(3):367, 1996.

[66] V. Elser. Phase retrieval by iterated projections. *J. Opt. Soc. Am.*, 20(1):40, 2003.

[67] V. Elser and I. Rankenburg. Deconstructing the energy landscape: Constraint-based algorithms for folding heteropolymers. *Phys. Rev. E*, 73(2):26702, 2006.

[68] V. Elser, I. Rankenburg, and P. Thibault. Searching with iterated maps. *PNAS*, 104(2):418, 2007.

[69] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C: the art of scientific computing*. Cambridge University Press, 2002.

[70] M. Deleglise and J. Rivat. Computing π (x): the Meissel, Lehmer, Lagarias, Miller, Odlyzko method. *Math. Comp*, 65(213):235, 1996.

[71] C. M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, and B. Reed. Probabilistic Algorithms for Testing Primality. *J. Number Theory*, 12(10):21, 1980.

[72] H. W. Kühn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83, 1955.

[73] J. Munkres. Algorithms for the assignment and transportation problems. *J. Soc. Ind. Appl. Math.*, 5(1):32, 1957.

[74] J. A. Hartigan. *Clustering Algorithms*. John Wiley & Sons, 1975.

[75] V. Elser. Random projections and the optimization of an algorithm for phase retrieval. *J. Phys. A*, 36(12):2995, 2003.

[76] S. Gravel and P. Winternitz. Superintegrability with third-order integrals in quantum and classical mechanics. *Journal of Mathematical Physics*, 43:5902, 2002.

[77] S. Gravel. Hamiltonians separable in Cartesian coordinates and third-order integrals of motion. *Journal of Mathematical Physics*, 45:1003, 2004.

[78] P. Tempesta, P. Winternitz, J. Harnad, W. Miller Jr, G. Pogosyan, and M. A. Rodrígez, editors. *Superintegrability in Classical and Quantum Systems*. CRM proceedings and lecture notes, AMS, 2004.

[79] H. H. Hoos. An adaptive noise mechanism for WalkSAT. *Proc. of the 18th Nat. Conf. in Artificial Intelligence (AAAI-02)*, page 655, 2002.

[80] S. Gravel and V. Elser. Divide and concur: A general approach to constraint satisfaction. *arXiv:0801.0222*, 2007. (to appear in Phys. Rev. E).

[81] P. Cheeseman, B. Kanefsky, and W. M. Taylor. Where the really hard problems are. *Proceedings of the 12th IJCAI*, page 331, 1991.

[82] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups*. Springer, 1999.

[83] P. G. Szabó, M. C. Markót, T. Csendes, E. Specht, L. G. Casado, and I. García. *New Approaches to Circle Packing in a Square*. Springer-Verlag, New York, 2007.

[84] D.W. Boll, J. Donovan, R.L. Graham, and B.D. Lubachevsky. Improving Dense Packings of Equal Disks in a Square. *Electron. J. Combin*, 46:2, 2000.

[85] R.L. Graham and B.D. Lubachevsky. Repeated patterns of dense packings of equal disks in a square. *Electron. J. Combin*, 3:1–16, 1996.

[86] I. Castillo, F.J. Kampas, and J.D. Pintér. Solving circle packing problems by global optimization: Numerical results and industrial applications. *Eur. J. Oper. Res.*, 191(3):786–802, 2008.

[87] K. J. Nurmela. *Constructing Spherical Codes by Global Optimization Methods*. Helsinki University of Technology, 1995.

[88] V. Elser and S. Gravel. Laminating lattices with symmetrical glue. *arXiv:0802.0730*, 2008. To be published in Phys. Rev. E.

[89] A. K. Hartmann and M. Weigt. *Phase Transitions in Combinatorial Optimization Problems*. Wiley-VCH, 2006.

[90] M. Mézard and M. Tarzia. Statistical mechanics of the hitting set problem. *Phys. Rev. E*, 76(4):41124, 2007.

[91] M. Bayati, J.H. Kim, and A. Saberi. A Sequential Algorithm for Generating Random Graphs. *Lecture notes in computer science*, 4627:326, 2007.

[92] M. Tarzia. private communication.

[93] R. Gallager. Low-density parity-check codes. *IEEE Trans. Inf. Theory*, 8(1):21, 1962.

[94] Documentation and source code for the PYCODES package are available online at http://www.mit.edu/~emin/source_code/pycodes/.

[95] Frank Alber, Svetlana Dokudovskaya, Liesbeth M. Veenhoff, Wenzhu Zhang, Julia Kipper, Damien Devos, Adisetyantari Suprapto, Orit Karni-Schmidt, Rosemary Williams, Brian T. Chait, Andrej Sali, and Michael P. Rout. The molecular architecture of the nuclear pore complex. *Nature*, 450(7170):695–701, 2007.

INDEX