Diss. ETH No. 13096

The Structure of Dense Polymer Systems: Geometry, Algorithms, Software

Dissertation

submitted to the
Swiss Federal Institute of Technology
Zurich

for the degree of
Doctor of Technical Sciences

presented by

Matthias Müller

Dipl. Informatik-Ing. ETH
born on October 30th, 1969
citizen of Küsnacht ZH

Accepted on the recommendation of
Prof. Dr. Jürg Nievergelt, examiner
Prof. Dr. Ulrich Suter, coexaminer

1999

Contents

List of Figures

Abstract

We present new algorithms for generating polymer structures for starting Molecular Dynamics simulations in dense phase. High quality polymer structures are characterized by low potential energy as well as consistency with experimental statistical data.

In contrast to common methods that consider energy potentials, our approach to finding such structures consists of the transformation of the chemical problem into a geometric optimization problem. We solve this problem by means of heuristic search algorithms. The efficiency of the search procedures is based mainly on the new parallel-rotation (ParRot) technique as well as on the concept of a horizon which allows splitting the global packing problem into a sequence of problems with increasing difficulty.

Our packing algorithms, which are available as a software package, are applicable for general polymer systems. This permits us to tackle the problem of packing long chains into large boxes (up to $50\mathring{A}$) in a few hours on current workstations. We have succeeded in packing systems consisting of up to $100,000$ particles five times faster than the best known methods. Furthermore, we succeeded, for the first time, in generating dense conformations of polycarbonate and polystyrene chains that correspond to experimental statistical data and which maintain these properties during subsequent Molecular Dynamics simulations.

Kurzfassung

Wir präsentieren neue Algorithmen zur Berechnung von dichten Polymerstrukturen, welche in Moleküldynamik-Simulationen benötigt werden. Strukturen von hoher Qualität zeichnen sich aus durch tiefe potentielle Energie sowie durch ihre Übereinstimmung mit experimentell ermittelten statistischen Daten.

Im Gegensatz zu den üblicherweise verwendeten Verfahren, welche mit dem Energiepotential arbeiten, beruht unsere Methode, solche Stukturen zu erzeugen, auf der Transformation des chemischen Problems in ein geometrisches Optimierungsproblem. Dieses lösen wir mit Hilfe von heuristischen Suchalgorithmen. Effizienz erreichen wir hauptsächlich durch zwei neue Konzepte, die parallele Rotation (ParRot) und die Idee eines Horizont-Parameters, welcher es erlaubt, das globale Packungsproblem in eine Sequenz von Problemen mit aufsteigendem Schwierigkeitsgrad aufzuteilen.

Unsere Algorithmen, welche in Form eines Software-Pakets erhältlich sind, können Polymer-Systeme allgemeiner Form verarbeiten. Sie erlauben es, in kurzer Zeit (einigen Stunden) lange Ketten in grosse periodische Zellen (bis $50\mathring{A}$) auf üblichen Workstations zu packen. Wir konnten beispielsweise ein System von Polymerketten mit insgesamt $100'000$ Partikeln fünf mal schneller generieren, als dies mit der besten bisher bekannten Methode möglich war. Es gelang uns auch, zum ersten Mal dichte Konformationen von Polykarbonat- und Polystyrol-Ketten zu erzeugen, die den experimentell gemessenen statistischen Daten entsprechen und diese Eigenschaft auch während anschliessenden Moleküldynamik-Simulationen beibehalten.

Acknowledgments

The outstanding collaboration with the members of Professor Ulrich Suters group from the Materials Science Department made this thesis possible. I wish to sincerely thank all the people who assisted me during my work.

Jürg Nievergelt, my advisor, initiated this interdisciplinary project. He established the important and valuable contacts to chemists and materials scientists. In my research, he left me complete freedom and provided me with enough computing power for my experiments and calculations.

Ulrich Suter, my coexaminer supported our project with his expert advice and his positive feedback. His enthusiasm for our new geometric approaches was a great encouragement.

Serge Santos recognized the benefit of our geometric algorithms to the polymer packing problem. He introduced me to polymer science and made our methods known to many scientists of his department. As a result of our intensive and fruitful collaboration, we succeeded in upgrading the packing software to a valuable tool in polymer science.

Pierre Robyr, the expert for polystyrene and polycarbonate statistics competently post processed and analyzed our results. With his support, we managed to generate innovational polymer conformations.

Martin Kröger raised my interest in simplified polymer models. His new efficient algorithm `GenPol` to generate dense amorphous systems was a great challenge for our own program `PolyGrow`.

Peter Schorn was my predecessor in the molecular modeling project. Some important techniques, implemented in the final software, are based on the embedding algorithm he suggested.

I have always enjoyed working in the pleasant atmosphere of our research group. *Silvania Avelar*, my office mate, introduced me to the Brazilian way of living. *Thomas Lincke* taught me patience at lunch time and at other occasions. *Ambros Marzetta*, the Unix guru helped me laying out my thesis in LaTeX. *Fabian Mäser*, one of the five best XBlast players in Switzerland, was always my most serious competitor. *Nora Sleumer* encouraged me to stay fit and well dressed. *Christoph Wirth* positively influenced my programming

style during the development of our commercial product Stöck Wyys Stich.

Finally, I thank my parents for making my education possible.

Chapter 1

Introduction

1.1 The Polymer Packing Problem

In the field of Materials Science, computer simulation of atomistic systems has become a very useful and important alternative to real experiments. Molecular Dynamics and Monte Carlo simulations are common methods to reveal and explain properties of materials. A large number of substances such as plastics, rubbers, synthetic fibers and materials from all living organisms are composed of polymers. These macromolecules, which consist of tens of thousands of atoms, have the shape of long chains. Polymeric materials in the amorphous state are composed of densely packed, randomly entangled chains. The chemical (covalent) structure of these chains is completely determined by the type of material used. It does not change during the simulation. In contrast, the conformation of the system, that is, the positions of the atoms in space is altered by the simulation process.

Since it is difficult to completely change the initial three dimensional shape during the simulation of dense atomistically detailed polymer systems, one should start with a reasonably "good" conformation. The problem consists of creating a packing of chains that do not intersect and whose three-dimensional shape corresponds to statistical data derived from real experiments. The high density together with the connectivity of polymer chains make this problem a difficult one.

1.2 Drawbacks of Today's Methods

Today's methods used to generate starting structures of dense polymer systems usually start with coarse initial guesses which are condensed and relaxed by molecular dynamics simulation and potential energy minimization. The

problem mainly lies in the generation of the initial-guess structure. The initial guess does not match experimental statistical data, it loses its statistical properties during condensation and relaxation. Thus, as an example, no satisfactory conformation of the widely used polycarbonate has been available.

1.3 A New Approach

In this thesis, we describe alternative approaches for generating dense polymer systems that avoid overlap, and guarantee the chains to obey the proper chain statistics.

Our approach is based on the idea of transforming the chemical packing problem into a geometrical optimization problem which is solved by heuristic search algorithms. The algorithms work at the target density from the start. They reduce intersections and consider statistical properties simultaneously. We split the delicate packing problem into a sequence of problems with increasing difficulty. A horizon parameter defines the range beyond which the atoms do not perceive each other. It is increased to eventually attain its full range where each atom can "see" all others. Besides this incremental modification of the non-bonded interactions between the atoms, the performance of the search algorithm relies heavily on the efficiency of the moves used during the exploration in the search space of conformations. We devised a new move, the parallel rotation (ParRot) for efficiently relaxing the torsion angles that are deep inside long chains in dense systems.

1.4 Results

Two considerable advantages of our packing software are its efficiency and the quality of the structures it generates. We have succeeded in packing systems of up to $100,000$ particles (periodic box size up to $50\mathring{A}$) within hours on a workstation. For large systems, our method solves the packing problem up to five times faster than the best of todays methods. Our software is able to generate conformations of polymer systems that obey statistical data derived from real experiments. We show, for the first time, that dense conformations of polycarbonate and polystyrene chains that correspond to experimental statistical data and that preserve these properties during subsequent Molecular Dynamics simulation can be generated.

1.5 Structure of This Thesis

In chapter 2 we give the necessary background on polymers and computer simulations of polymer materials. We introduce two simulation methods, the Molecular Dynamics method (MD) and the Monte Carlo method (MC). They both ask for 'reasonable' starting conformations of dense polymer systems. We give a survey of current approaches to the problem of generating such structures.

Chapter 3 is devoted to our new approach to the polymer packing problem. First, we introduce a geometric model of polymers. We show how force field parameters are transformed to geometric and statistical constraints. We explain our algorithms `PolyGrow` and `PolyPack` for generating simplified and atomistically detailed polymer systems respectively. Both algorithms perform heuristic searches in conformational space. We describe the internal structure and the usage of the polymer packing software in Chapter 4. Then we show its efficiency as well as the quality of the generated conformations by means of diverse case studies (Chapter 5).

Chapter 2

The Context

2.1 Chemical Background

The work presented in this thesis is the result of a fruitful interdisciplinary collaboration between computer scientists and chemists. For those readers not belonging to the latter group, I will first give a brief introduction to the world of polymers. For a more detailed survey see [10].

2.1.1 Polymers Here, There and Everywhere

Molecules range in size from a few atoms, such as H_2O, to tens of thousands of atoms. Giant molecules, called polymers, play an important role in the biological and artificial world and in many fields of research.

A variety of materials are composed of polymers, such as:

- Plastic materials (including bullet proof vests)
- Rubber and all its products
- Synthetic fibers (optic fibers, clothes)
- Cellulose (wood and paper)
- DNA, proteins (all living organisms)

Polymer physics and the mathematics of macromolecules were developed mainly by chemists in the 1940s and 1950s. The most notable among them was P. Flory (1908-1982) [7]. For his pioneering work in polymer physics, he received a Nobel prize in 1974. Polymer physics has eventually grown into an independent field of research. All of its concepts and models have been used successfully both in physical chemistry and in molecular biology.

2.1.2 What Does a Polymer Molecule Look Like?

Anyone who wants to understand the properties of polymer materials should first have an image of their molecules. Polymers are long molecular chains. A helpful image is a long, entangled, three-dimensional rope or wire. These chains are composed of monomer units. A monomer unit is a small group of just a few atoms connected to other monomers by covalent bonds. A large number of these monomer units are connected to form a chain. Monomer units are also called repeat units. Figure 2.1 shows schematically the structure of the simplest polymer, a polyethylene chain. To be considered a polymer, a molecule must consist of a large number R of repeat units. Artificially synthesized molecules contain from hundreds to tens of thousands of units, while natural polymers like the DNA molecule are much larger. They are composed of up to ten billion monomer units.

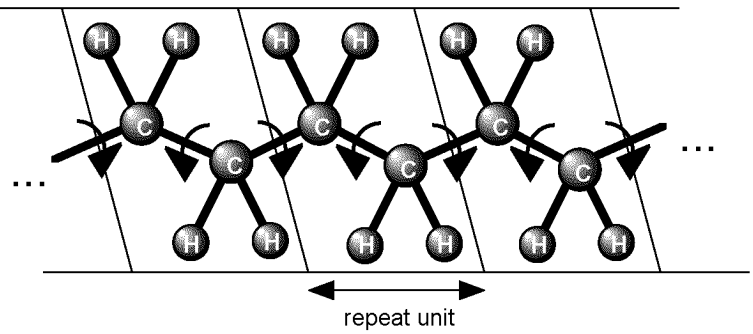

Figure 2.1: The covalent structure of polyethylene.

So far, we only described the way in which atoms are connected to form a polymer. This is called the *primary* or *covalent structure* of a molecule. The covalent structure of polymers used in experiments or computer simulations is always known in advance.

However, the crucial information about polymers in simulations is their three-dimensional shape, called conformation or *tertiary structure*. The problem of computing the conformation of polymers in a dense environment is the subject of this thesis.

The term *secondary structure* is used mainly for proteins to describe their three-dimensional structure as a sequence of regular, periodic, spatial shapes such as alpha helices and beta sheets.

At first glance it seems reasonable to assume that a polymer chain looks like a straight line (Fig. 2.1). However, under most conditions, this is not

true. A polymer diluted in a solvent, like water, gets tangled up into a random, loose, three-dimensional coil. This is a result of the chain's flexibility. Although the atoms in the molecule are joined together by strong covalent bonds, their position in space with respect to each other need not be fixed. The majority of the most commonly used synthetic polymers, as well as all protein molecules, have single C-C chemical bonds along their backbone. Such molecules appear flexible because parts of a molecule may rotate around the single bonds. For example, a monomer unit of a polyethylene chain contains two C-C bonds which leads to a total of $2R$ (freely) rotatable torsion angles per chain. This large number of degrees of freedom make the chain look like a random walk in vacuum or diluted in a solvent.

2.1.3 Possible States of Polymer Substances

The three simplest states of ordinary matter are: solid (crystal), liquid, and gaseous. The gas and the crystal state are not typical for polymers. However, polymers do not only exist in a liquid state. There is much more diversity in polymer substances: There are plastics and rubber, fibers, timber and paper, polymer films and all the various polymers found in nature. This variety is due to the fact, that polymer substances are composed of very long, strongly entangled molecular chains. Depending on the kind and strength of interactions between the monomers, polymers can exist in four different states:

- *Viscous state, polymer melt:* The polymer is a liquid of macromolecules. In thermal motion, the chains easily move with respect to each other.

- *Elastic state:* The chains are joined together with covalent chemical bonds or other forces to form a polymer network. Thus, they can not flow freely. But at small scale, the mobility of the chains is not constrained by the cross-links.

- *Semi-crystalline state:* If the temperature of a polymer melt is decreased slowly, the thermal motion is enough to enable the polymer chains to form crystalline regions.

- *Polymer glass:* Polymers that are unable to crystallize at low temperatures, tend to become a glass. Any thermal motion at any scale larger than the size of a monomer does not exist. In contrast to the semi-crystalline state, the chains are entangled randomly showing no order or large scale structure.

In all the states, the material consists of densely packed, multiple entangled polymer chains. We will focus mainly on polymer melts and glasses.

2.2 Computer Simulation of Polymers

To explain how polymer materials behave, theoreticians have tried to construct simple mathematical models of polymers and polymer systems. The development of powerful computers made it possible to simulate these models and analyze them numerically. The first simulation of a liquid was carried out in 1953 by Metropolis, Rosenbluth, Teller and Teller at the Los Alamos National Laboratories in the United States [25]. The rapid development of computer hardware made computer simulation possible on most of todays workstations. A survey of the history of polymer simulation and todays methods is given in [1].

Computer simulation provides a direct route from the microscopic details of a system (the masses of the atoms, the interactions between them, molecular geometry etc.) to macroscopic properties of experimental interest (the equation of state, transport coefficients, structural order parameters, and so on.). This type of information is technologically useful, because:

- It may be difficult or impossible to carry out experiments under extremes of temperature and pressure, while a computer simulation would be perfectly feasible.
- The high speed of molecular events is often an experimental difficulty but it presents no hindrance to the simulator.
- Subtle catalysis or enzyme action are difficult to probe experimentally, but can be extracted readily from a computer simulation.

When its results are compared with those of real experiments, computer simulation is also a test of theories and the underlying model.

2.2.1 Mathematical and Computational Models

Basically, an atomistic model of a material consists of the following two entities:

- A set of N particles. A particle may be an atom in a detailed model, or a whole monomer in simpler models. The microscopic state of the system is specified in terms of the positions and momenta of these particles. $\mathbf{r} = (r_1, \ldots, r_N)$, $\mathbf{p} = (p_1, \ldots, p_N)$.

- A potential energy function $V(\mathbf{r})$ which depends on the coordinates of the particles. The forces acting on each particle can be calculated from the potential energy function. It thus governs the entire time-evolution of the system and all its mechanical properties.

The approach used almost universally in computer simulation is to break up the potential energy into terms depending on the coordinates of individual particles, pairs, triplets etc.:

$$V(\mathbf{r}) = \sum_i v_1(\mathbf{r_i}) + \sum_i \sum_{j>i} v_2(\mathbf{r_i}, \mathbf{r_j}) + \sum_i \sum_{j>i} \sum_{k>j>i} v_3(\mathbf{r_i}, \mathbf{r_j}, \mathbf{r_k}) + \dots \quad (2.1)$$

The first term $v_1(\mathbf{r_i})$ in eqn (2.1) represents the effect of an external field on the system (including, for example, the container walls). The remaining terms represent particle interactions. Three-body and higher terms are only rarely included in computer simulations. Firstly because summation over triplets of atoms is time-consuming. Secondly because the average three-body effects can be partially included by defining an 'effective' pair potential. Four-body and higher terms are expected to be small in comparison with v_2 and v_3. We may thus rewrite eqn (2.1) in the form

$$V(\mathbf{r}) \approx \sum_i v_1(\mathbf{r_i}) + \sum_i \sum_{j>i} v_2^{eff}(r_{ij}). \quad (2.2)$$

where $r_{ij} = |\mathbf{r_i} - \mathbf{r_j}|$. For the interaction of two atoms we distinguish between non-bonded (intermolecular) and bonded (intramolecular) atom pairs.

A typical potential function of two non-bonded atoms is shown in figure 2.2. There is an attractive tail at large scale separations, due to correlation between the electron clouds surrounding the atoms ('van der Waals' dispersion) and a steeply rising repulsive wall at short distances coming from the non-bonded overlap between the electron clouds. The idealized Lennard-Jones potential, commonly used in computer simulations, approximates this behavior:

$$v^{LJ}(r) = 4\epsilon((\sigma/r)^{12} - (\sigma/r)^6) \quad (2.3)$$

where r is the distance between the atoms and ϵ and σ are parameters that are depend on the type of the atoms involved. Chemical bonds are, in principle, also inter-atomic potential energy terms. Since bond vibrations are of very high frequency but of low amplitude, the potential energy surface is often replaced by a rigid distance constraint of fixed length. The same applies to atoms of adjacent bonds. Bond angles are often kept fixed during

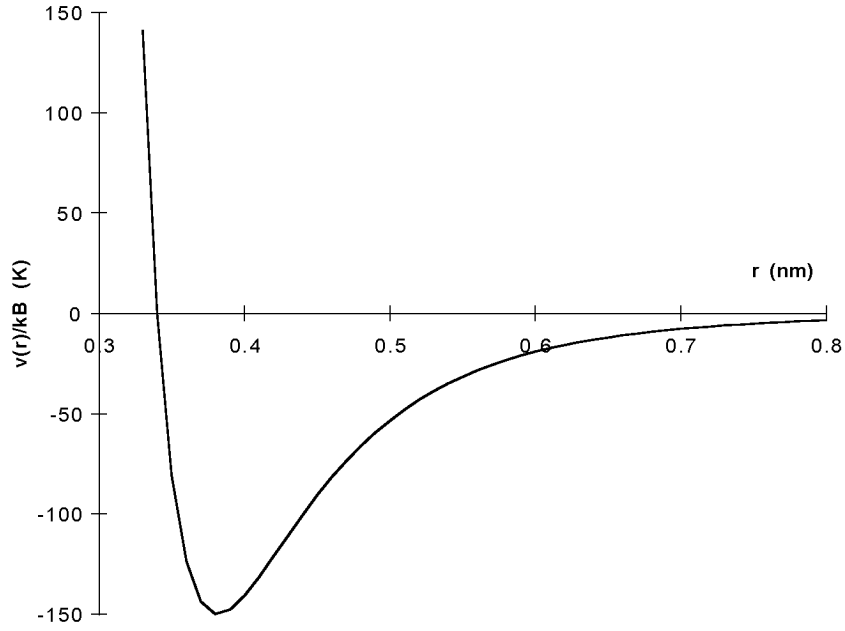

Figure 2.2: The pair potential for argon (Maitland and Smith [21]).

the simulation. However, torsional motion about bonds cannot in general be neglected since these motions involve energy changes comparable with normal thermal energies. Moreover, torsional motions are the main degrees of freedom allowing a single molecule to change its 3-dimensional shape.

2.2.2 Periodic Boundary Conditions

In comparison to macroscopic systems ($N \approx 10^{23}$), computer simulations are usually performed on a small number of particles ($N \approx 10^{6}$). In such small systems, atoms tend to drift apart. A potential representing a container may hold the system together, but the large fraction of molecules on the surface will experience quite different forces from molecules in the interior.

The problem of surface effects can be overcome by implementing periodic boundary conditions. A cubic box is replicated throughout space to form an infinite lattice. When a particle moves in the original box, all its periodic images move exactly the same way. Thus, as a molecule leaves the central box on one side, one of its images will enter through the opposite face. It is sufficient to store the coordinates of the particles in the central box. If the cubic box is large enough, the particles are not able to 'sense' the symmetry of the periodic lattice. The common experience in simulation work is that periodic boundary conditions have little effect on the equilibrium thermody-

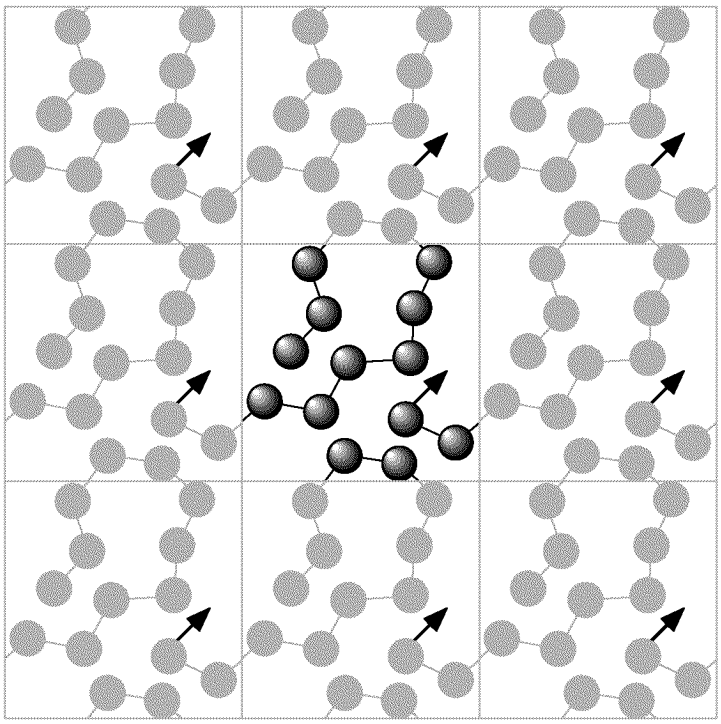

Figure 2.3: Periodic boundary conditions: A cubic box is replicated throughout space.

namic properties and structures of polymer systems. Thus, cubic periodic boundary conditions are used almost exclusively in simulations of polymer melts and glasses.

2.2.3 Conformational Space

The positions $\mathbf{r}$ of the particles of a system, can be thought of as coordinates of a single point in a multidimensional space termed *conformational space*. For a system of N particles, this space has $3N$ dimensions. The potential energy function $V(\mathbf{r})$ defines a complex hyper-surface on conformational space. It usually contains a huge number of local minima and energy barriers of various sizes. Let us use the term $\mathbf{r}(t)$ for the point which describes the state of the system at time t. As the system evolves in time, $\mathbf{r}(t)$ describes a trajectory on the hyper-surface of the potential energy function.

2.2.4 The Molecular Dynamics Method (MD)

The most obvious way to simulate a many-particle system is to compute its real world time evolution by solving the equations of motion (Newton's equations) derived from the potential energy function $V(\mathbf{r})$. In most cases, this cannot be done analytically. To solve the equations of motion for a set of Lennard-Jones particles, an approximate, step-by-step procedure is needed, since the forces change continuously as the particles move. Molecular dynamics (MD) is the term used for the technique to compute the time-evolution and the dynamic properties of many-particle systems.

Computer simulation generates detailed information at the microscopic level. Methods of statistical mechanics are used to convert this information into macroscopic terms as pressure and internal energy. Suppose that we can write the instantaneous value of some property $F(\mathbf{r})$ (for example the end-to-end distances of the chains) as a function of the actual conformation $\mathbf{r}$. The system evolves in time, so that $\mathbf{r}$, and hence $F(\mathbf{r})$ will change. It is reasonable to assume that the experimentally observable 'macroscopic' property F_{obs} is really the *time average* of $F(\mathbf{r})$ taken over a long time interval t_{obs}. Since the equations of motion are solved on a discrete step-by-step basis, the time average may be written in the form

$$F_{obs} \approx < F >_{time} = \frac{1}{t_{obs}} \sum_{t=1}^{t_{obs}} F(\mathbf{r}(t)). \tag{2.4}$$

The practical question regarding the method is whether or not a sufficient region of conformational space is explored by the system trajectory to yield satisfactory time averages within a feasible amount of computer time.

Since the system is not likely to cross high energy barriers, the Molecular Dynamics method will only explore a rather small region of conformational space. In other words, a system of entangled, densely packed polymer chains is not likely to change its 3-dimensional shape significantly. Consequently $< F >_{time}$ will converge very slowly towards the observable value F_{obs}.

2.2.5 The Monte Carlo Method (MC)

A much more efficient method is to replace the time average by the ensemble average. This is done in Monte Carlo simulations. Instead of following the path $\mathbf{r}(t)$ governed by the equations of motion, we choose a sequence of randomly chosen conformations $\mathbf{r}_1 \ldots \mathbf{r}_N$ to compute the observable P_{obs}. But now we have to be careful because the conformations do not appear with equal probability along the time trajectory $\mathbf{r}(t)$ in eqn (2.4). The probability for a conformation $\mathbf{r}$ to appear at a temperature T is

$$p(\mathbf{r}) \sim e^{-V(\mathbf{r})/T} \tag{2.5}$$

Thus, P_{obs} must be computed as the expected value

$$F_{obs} \approx < F >_{ensemble} = \frac{\sum_{k=1}^{N} F(\mathbf{r}_k) p(\mathbf{r}_k)}{\sum_{k=1}^{N} p(\mathbf{r}_k)}. \tag{2.6}$$

Unfortunately, a randomly chosen conformation is most likely to have a very high energy and thus, very low probability, because there will be pairs of atoms that are very close to each other, resulting in high Lennard-Jones energy terms (eqn (2.3)).

In eqn (2.6) we would thus sum mainly properties of non-relevant conformations and $< F >_{ensemble}$ would converge very slowly towards F_{obs}. The technique that solves this problem is called *importance sampling*. The sequence of conformations $\mathbf{r}_1 \ldots \mathbf{r}_N$ is chosen randomly, but according to the probability distribution p. In this case we may rewrite eqn (2.6) as a simple average

$$F_{obs} \approx \frac{1}{N} \sum_{k=1}^{N} F(\mathbf{r}_k). \tag{2.7}$$

The most common method to generate a sequence of conformations according to a probability distribution p is the Metropolis acceptance-rejection algorithm [25]:

procedure MC;
begin
  Choose a starting conformation $\mathbf{r}$;
  loop
    $\mathbf{r}' \leftarrow$ *locally changed* $\mathbf{r}$;
    probability $\leftarrow \min(1, p(\mathbf{r}')/p(\mathbf{r}))$;
    if random() < probability **then**
      $\mathbf{r} \leftarrow \mathbf{r}'$;
    endif
  endloop
end

where `random()` returns a random number between zero and one. The current conformation $\mathbf{r}$ is changed locally by a move. Examples for Monte Carlo moves are the displacement of a single atom or the modification of a torsion angle inside a molecule. The new conformation $\mathbf{r}'$ is accepted with

probability $\min(1, p(\mathbf{r}')/p(\mathbf{r}))$ or rejected otherwise. A Monte Carlo move should not completely change the current conformation because such a move would have a very low acceptance probability. Monte Carlo moves therefore perform local changes keeping most of the particles frozen.

With this restriction, the problem of locality arises as it did within the molecular dynamics method. The sequence of conformations is not likely to cross high energy barriers and the starting conformation cannot be changed completely.

2.2.6 The Need of Starting Structures

In principle, all simulations should be ergodic (produce all conformations according to their probability) and it should not matter with which structure one begins. In pratice, however, since it is difficult to completely change the starting structure during the simulation of dense polymer systems one should start with a reasonably "good" structure of low potential energy. The problem consists of creating a packing where the chains do not intersect themselves and also do not intersect their periodic images. Additionally, statistical properties like torsion angle distributions and end-to-end distances must agree with those experimentally evaluated.

The task of generating dense polymer systems is formidable due to the high density and the connectivity of the chains. Generating starting structures poses an interesting geometric and combinatoric problem which is the main subject of this thesis.

2.3 Survey of Current Approaches and Results

Much effort has been put into constructing "reasonable" amorphous packings starting from atomistic models of single polymer chains. Common approaches can be roughly divided into four groups:

- *Chain growing:* Methods that grow chains into dense phases by adding new segments according to a choice criterion for achieving the target chain properties [24, 34].

- *Lattice methods:* Coarse "initial guesses" of the polymer chains are created on a lattice and afterwards relaxed by potential energy minimization. [2].

- *Condensation methods:* This method departs from structures with very low densities that are then condensed step by step during NpT-MD simulation to experimental densities of polymer systems [11, 23, 3].

- *Polymerization:* This native technique, described recently, starts by preparing the monomer liquid in the periodic box and then polymerizing the monomers to a chain [15].

All these methods start with an "initial guess" structure that is subsequently relaxed by potential energy minimization or by simulated annealing where the temperature is gradually reduced to that of interest.

The main problem lies still in the ad hoc nature of the initial-guess. The structures generated simply do not provide an ensemble in the sense of statistical-mechanics. The quality of these structures is very much a function of the constructor and the care of testing. The initial guess does not correspond to experimental statistical data, or it loses its imposed statistical properties during "condensation" and relaxation.

2.4 A New Approach to the Packing Problem

This thesis presents alternative approaches for generating polymer systems that avoid severe overlaps, *and* guarantee the chains to obey the proper chain statistics. The main ideas implemented in the packing algorithms which differ from common approaches are:

- *Geometric constraints:* The energy potential function to be minimized is approximated by geometric constraints such as distance constraints, torsion angle states and torsion angle distributions. Generalized coordinates, the hard sphere model as well as the rotational isomeric states model (RIS), are used to derive a set of constraints from an energy potential function.

- *Discrete combinatorial search:* The packing algorithm starts with a random configuration which is improved iteratively by means of a heuristic search algorithm which ultimately furnishes the target configuration. In contrast to energy minimization and simulated annealing techniques, the search does not necessarily follow a path driven by physical forces. It rather tries to solve the combinatorial problem regardless of the physical problem it is derived from.

- *Combination of techniques:* The packing algorithm `PolyGrow` combines a chain growing process with an iterative technique to remove local overlaps of atoms. This hybrid algorithm succeeds in generating large systems of simplified polymer chains.

- *Principle of horizon:* Since it is difficult to manipulate an atomistically detailed, highly connected system, an incremental modification of the non-bonded interaction is implemented in `PolyPack`. A horizon value defines the scope beyond which the atoms are "ghosts" to each others. It is increased to eventually attain its full range where each atom can "see" all others.

- *The parallel rotation move:* The performance of the search algorithm `PolyPack` relies heavily on the efficiency of the moves used during the exploration of conformational space. We devised a new efficient method, the ParRot move [33], for efficiently relaxing torsion angles that are deeply inside long chains in dense systems. We show that this move contributes to a large extent to the success of our algorithm.

These techniques are embodied in our packing algorithms `Embed`, `PolyPack` and `PolyGrow` which are described in detail in the next chapter.

Chapter 3

Packing Algorithms

This chapter is devoted to the presentation and analysis of algorithms to generate dense polymer systems that avoid overlap and agree with given chain statistics. In contrast to most of the commonly used packing techniques that focus on energy, our algorithms are designed to solve a geometric, combinatorial problem. Therefore, the energy potential function, which actually is to be minimized, is replaced by a set of geometric and statistical constraints. In this new, simplified model of a polymer system, the problem of finding a low energy starting structure becomes a combinatorial problem of finding an arrangement of points which satisfies a set of geometric constraints. After defining possible geometric models, packing techniques for atomistically detailed and more simplified models are presented.

3.1 Geometric Models of Polymer Systems

As described in section 2.2.1, a model for a polymer system consists of a set of N particles, given by their coordinates $\mathbf{r} = (r_1, \ldots, r_N)$ and a potential energy function $V(\mathbf{r})$ that governs the time-evolution of the system.

In contrast, a geometric model of a polymer system is composed of a set of points and a set of constraints, concerning these points. The concept of periodic boundary conditions to simulate a dense environment (see section 2.2.2), is adopted one-to-one. However, the potential energy function $V(\mathbf{r})$ is replaced completely by a set of geometric and statistical constraints.

3.1.1 Generalized Coordinates

Instead of defining the conformation of a system by the Cartesian coordinates of the particles $\mathbf{r}$, one can describe the same system by "internal" coordinates

(i.e. bond lengths, bond angles, torsion angles and a few position and orientation coordinates). The use of these coordinates, popularized by Flory [7] and others, drastically reduces the number of parameters.

In the case of polymer chains it has been shown that bond lengths and bond angles (the angle between two adjacent bonds) are usually confined to a narrow range about a mean value, and they can therefore often be considered to be constant for most analyses of flexible chains. Fixed bond lengths and bond angles give each molecule a locally rigid structure where torsional rotations about bonds are the remaining degrees of freedom. We restrict ourself by considering the torsion angles to be the only degrees of freedom of the system.

Torsion angles cannot always be varied independently. When they belong to a cyclic substructure, such as a ring of five or six atoms, they are subject to constraints. Therefore, we consider cyclic substructures to be rigid.

A molecular structure can always be split into a set of rigid groups connected by torsion angles. Rigid groups are sets of atoms whose relative positions cannot be modified by changing the torsion angles. These sets are overlapping because two adjacent groups share the two atoms of the common free bond. The bond (with variable torsion angle) connecting two rigid groups is termed a free bond. Rigid groups and free bonds build a tree structure where rigid groups are the vertices and free bonds the edges.

A rigid group $g^{(i)} := \{r_1^{(i)}, \dots r_{N_i}^{(i)}\}$ in the molecule i is defined as the set of relative coordinates $r_j^{(i)}$ for the N_i atoms comprised in the rigid group. Because the relative coordinates of the atoms are not changed by isometries in $\mathbf{R}^3$, $g^{(i)}$ rather represents the class of atom coordinates related to each other by translation and rotation operations. To construct the configuration of a molecule, the algorithm traverses the tree and attaches every group encountered according to the current torsion angle. The tree structure is composed of T free bonds and in the absence of cyclic structures $T + 1$ rigid groups (see figure 3.1). A vector of T values for the torsion angles are sufficient to describe the internal conformation of the molecule, whereas a position vector and three Euler angles describe the molecules location and orientation, respectively.

We can entirely describe the molecule i, composed of a set of rigid groups $g_k^{(i)}$, by an origin vector $r_0^{(i)}$, three Euler angles $\alpha^{(i)}, \beta^{(i)}$ and $\gamma^{(i)}$ and a vector of N values for the torsion angles:

$$D^{(i)} := \{r_0^{(i)}, \alpha^{(i)}, \beta^{(i)}, \gamma^{(i)}, \Phi_1^{(i)}, \dots, \Phi_N^{(i)}\} \tag{3.1}$$

Then, the conformation of a system of M molecules is specified by $D := \{D^{(1)}, \dots, D^{(M)}\}$.

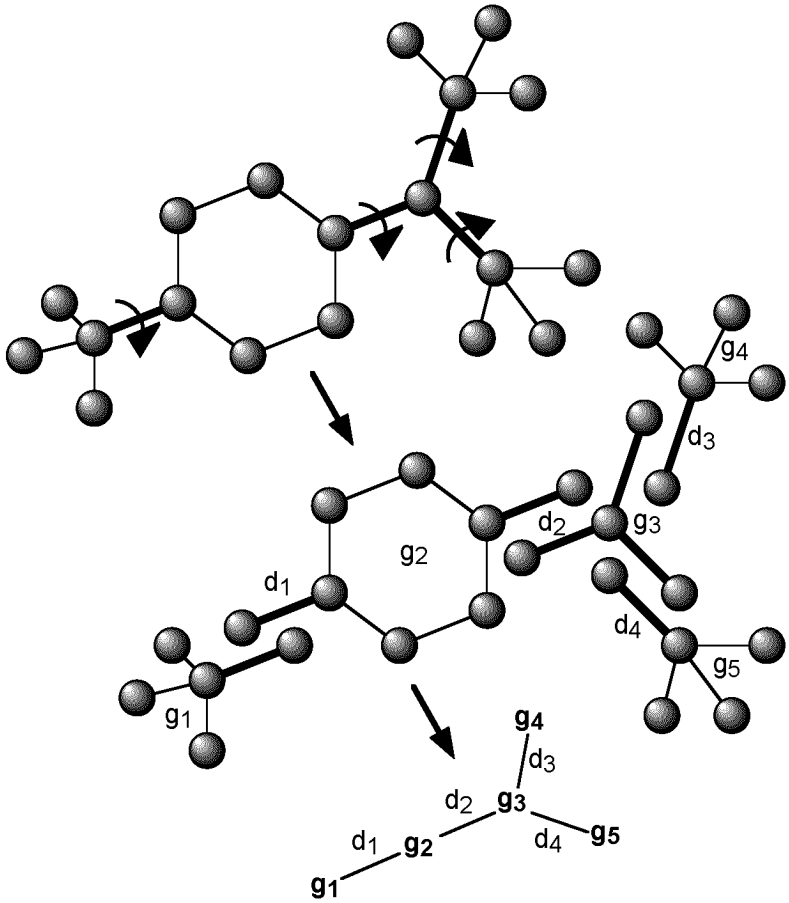

Figure 3.1: Tree view of a molecule: Nodes represent rigid groups of atoms. They are connected by torsion angles that can be freely rotated.

3.1.2 Rotational Isomeric State Model (RIS)

A geometric model of a polymer system should account for every kind of particle interactions contained in the potential energy function $V(\mathbf{r})$. The forces between atoms that belong to the same or adjacent bonds are replaced by the constraints of fixed bond lengths and bond angles. These constraints leave free choices for values of the remaining degrees of freedom, the torsion angles. However, forces, acting between pairs of atoms far apart in the same chain influence the distribution of the torsion angle values. Certain torsion angles as well as certain combinations of adjacent torsion angles are more likely than others.

To account for these effects, the so called Rotational Isomeric States (RIS) model [7, 22, 30] is used. In the RIS model, the possible values for torsion

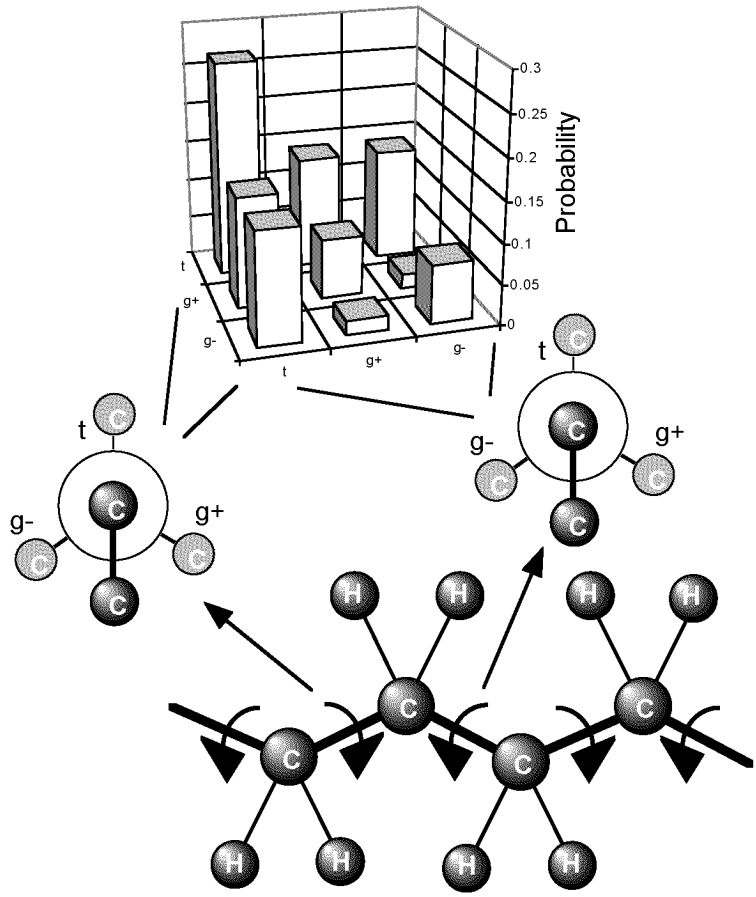

Figure 3.2: Rotational Isomeric States model for polyethylene.

angles are restricted to discrete values. These RIS states together with values for the correlations of (usually) two adjacent torsion angles along the polymer backbone allow to properly account for the chemical details of different polymer structures. We somewhat relaxed the RIS model insofar as the torsion angles are not required to exactly assume a given RIS state, but are allowed a certain tolerance around that state (say, $\pm 20^o$). The model makes possible the description of the conformational behavior of macromolecules with proper attention to the details of the chemical structure of the chain. Figure 3.2 shows a RIS model for polyethylene. Torsion angles are restricted to the three discrete states *trans* ($t = 0^o$), *gauche plus* ($g^+ = 120^o$) and *gauche minus* ($g^- = -120^o$), often encountered in many kinds of polymers. Besides the restriction to these values, a 3×3-matrix describes the distribution of all possible pairs of states of two adjacent torsion angles.

3.1.3 Excluded Volumes Interaction

The distance between particles that do not belong to the same rigid group, can be modified by changing the torsion angles. The energy terms in $V(\mathbf{r})$ of such non-bonded pairs of particles is given by the van der Waals potential function (eqn (2.3)). We replace this function by a lower distance constraint for each non-bonded pair of atoms. A radius is assigned to each type of atom, such that the sum of the radii of two atoms equals the distance where the van der Waals potential has its minimum. Such an approximation is also called an excluded volumes interaction or a hard-sphere potential. Atoms are replaced by hard-spheres which must not overlap in a dense packing. The hard-sphere potential is a local type of interaction, and, thus, requires relatively little computation.

3.1.4 Atomistically Detailed Polymers and Simplified Chains

So far, we have always equated particles and atoms. Indeed, an atomistically detailed model replaces each atom by an individual particle. However, coarser models [16, 40] that represent polymers by sequences of spheres, make possible the simulation of very large systems on most of todays workstations. Recent results have shown that simulations can be used to predict the behavior of real polymers even if the simulation model does not contain specific chemical details [18, 8, 12]. In contrast to the atomistically detailed model, entire monomers, usually composed of 5 to 30 atoms, are replaced by one single spherical particle. A polymer is, thus, modeled by a sequence of spheres which are connected by bonds of fixed length, like a pearl necklet.

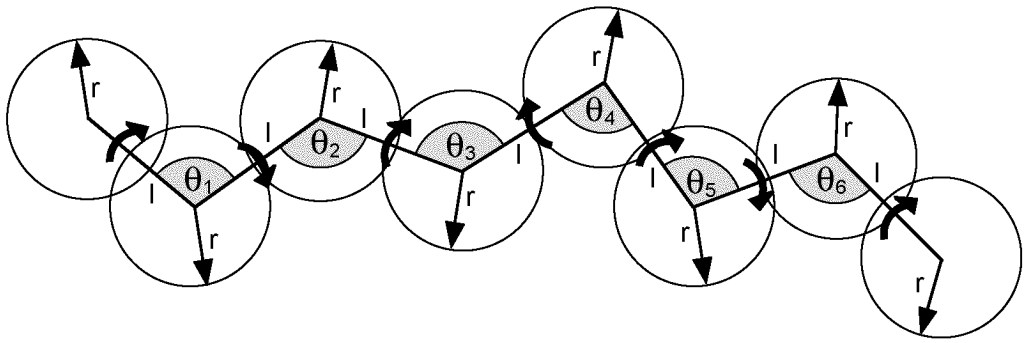

Figure 3.3: Simplified model of a polymer chain.

We investigate both simplified and atomistically detailed models. Figure 3.3 shows the details of the model for simplified polymer chains we used. Spheres with uniform radius r are connected by bonds of fixed length l. The bond angles θ_i may also be uniform or varying according to a distribution function which defines the stiffness of the chain (see section 4.1). The torsion angles are unconstrained and constitute the main degrees of freedom.

3.2 The Polymer Packing Problem (PP)

After having introduced several concepts to derive a set of constraints from a potential energy function, we are now ready to formulate the geometric Polymer Packing problem (PP):

A polymer system is a set of polymer chains. A chain is composed of hard-spheres joined with each other in rigid groups. Each sphere represents an atom or a monomer. The only degrees of freedom are the torsion angles, the values of which, the RIS states, are restrained to a discrete and finite set of possible angles. These discrete RIS states are, to second-order, pairwise correlated and obey given distributions. Given a target density, a number of polymer chains, and the corresponding RIS states with their distribution; find values for all torsion angles for which the following conditions hold:

1. The density of the periodic system equals a given value.

2. Hard-spheres do not overlap.

3. The pairwise distribution for the torsion angles is respected.

In some instances, sphere overlap bounded by a constant is allowed. This can also be modeled by reduced radii. The difficulty of finding dense polymer conformations which obey all three conditions is caused by a conflict between conditions 2 and 3. The only way to remove a collision between two atoms of the same chain is to manipulate the torsion angles and thus, their distribution. On the other hand, choosing the torsion angles according to a given distribution may cause new overlaps of the atoms.

This is one reason why dense structures generated with common techniques (i.e. energy minimization or simulated annealing) often do not respect postulated, experimentally measured torsion angle distributions.

3.3 The Combinatorial Chain Packing Problem (CP)

In order to analyze the Polymer Packing Problem theoretically, we introduce the Combinatorial Chain Packing Problem (CP). This combinatorial optimization problem models a simplified subproblem of the Polymer Packing Problem. It enables us to show theoretically the difficulties of packing large, highly connected systems.

A system consists of a set of chains. In the simplified case, a chain is a sequence of hard-spheres with fixed bond lengths and bond angles. The degrees of freedom, the torsion angles, are restrained to a discrete and finite set of possible angles. Given the description of such a system, find values for all torsion angles such that the hard-spheres do not overlap.

3.3.1 CP is NP-complete

Because CP is a discrete, combinatorial optimization problem, we are able to analyze its theoretical time complexity. Complexity theory distinguishes two main classes of problems: Those which can be solved by polynomial time algorithms and those for which a polynomial time algorithm does not exist, to the best of current knowledge. The latter set of problems constitutes the class of NP-hard problems [9, 41]. The combinatorial chain packing problem is NP-complete. In practice, this means that the only way to find an optimal solution in general is to try all possible combinations of torsion angle values. The time complexity of this exhaustive, brute force algorithm is exponential in the number of torsion angles.

CP belongs to the class of non-deterministic polynomial problems (NP). A non-deterministic machine would guess a set of torsion angle values and verify the resulting conformation, both in polynomial time.

Reduction

To prove that CP is NP-complete, we must show that CP can be reduced in polynomial time to some known NP-complete problem. We choose the Partition problem: Given a list of positive integers $z_1, \ldots, z_n$, does there exist a partition of $\{1, \ldots, n\}$ into disjoint subsets I and J such that the sums over these subsets are equal: $\sum_{i \in I} z_i = \sum_{j \in J} z_j$? This is one of the earliest known NP-complete problems [14].

We need to show that there is a transformation that maps every instance of the Partition problem into some instance of the Chain Packing problem,

with two properties. First, the transformation must be computable in polynomial time. Secondly, a partition must exist for the instance of the Partition problem if and only if a conformation without overlap exists for the system generated by the transformation of this instance. Having such a transformation, we can formulate the following algorithm:

function Partition($z_1, \ldots, z_n$):Boolean;
begin
  system ← TransformPartitionIntoSystem($z_1, \ldots, z_n$);
  conformation ← Pack(system);
  return *conformation is solution*;
end

If we can find an appropriate transformation, in case we have a polynomial-time algorithm for the Chain Packing problem, we have as well a polynomial-time algorithm for Partition, namely the one above. Given a vector of positive integers $z_1, \ldots, z_n$ we construct a molecule as figure 3.4 shows:

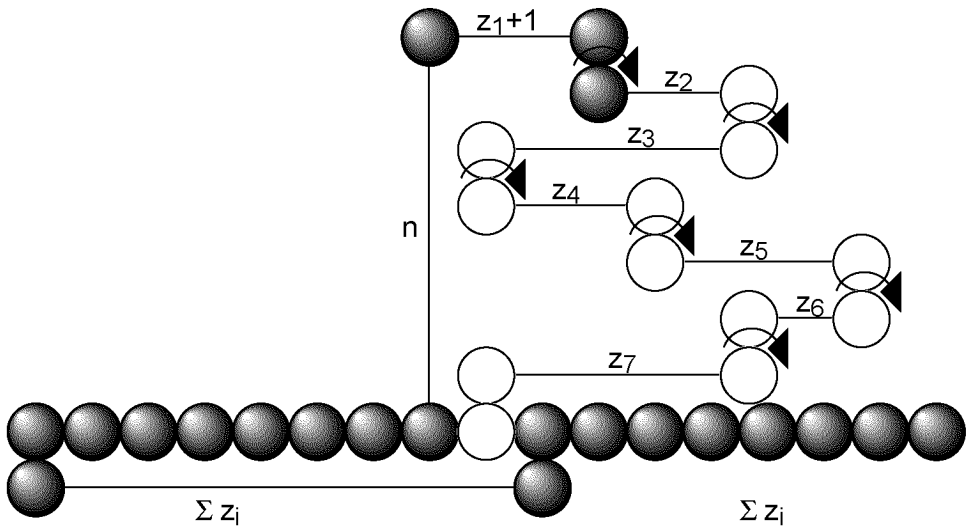

Figure 3.4: Transformation of the Partition problem into the Chain Packing problem.

All atoms have a diameter of one unit. The stationary part of the molecule is a straight chain of twice as many atoms as the sum of $z_1, \ldots, z_n$, with a gap one atom wide in the middle. A bond of length n units leaves the atom left of the gap at a right angle. The atom at the other end of this bond is the first atom of a mobile chain. The bonds connecting this chain have

lengths $z_1+1, 1, z_2, 1, z_3, 1, ...z_{n-1}, 1, z_n, 1$. All torsion angles are static (have only one admissible value) except the bonds of length 1 that are part of the mobile chain. They are allowed to be in *cis* (0^o) or *trans* (180^o) state. The molecule's structure forces the last atom of the mobile chain to move along the stationary part, producing collisions with the stationary atoms. There is only one way to avoid such a collision, namely if the last atom fits exactly into the gap. If and only if such a packing is possible, a partition for the numbers $z_1, \ldots, z_n$ exists. This partition can be read off the mobile chain. If the bond of length z_i points to the left, the value z_i becomes a member of the first sum, if it points to the right, it is included in the second sum. Bonds that point to the right increase the distance to the gap, the others reduce this distance. If the two sums are equal, the last atom fits perfectly into the gap. This transformation has the postulated properties. CP is thus NP-complete.

Implications

The proof of the NP-completeness of CP shows where the difficulty of packing has its origin. At first glance, the molecule generated by the transformation from the Partition problem does not seem to have a realistic covalent structure as a whole. Nevertheless it can be interpreted as part of a dense system. The moving part of the molecule represents a new chain to place, while the stationary part can be interpreted as surrounding, densely packed molecules, leaving only small gaps open. This corresponds exactly to the situation we encounter when trying to fit a molecule into the interstice left over by surrounding molecules.

One procedure used to squeeze a set of molecules into a tiny box is the so called Chain Growing algorithm (see 3.6). It starts with a small part of a chain and attaches further parts one by one. At early stages of this process, no collisions occur, since the box is empty, allowing the chain to grow in any direction. However, at later stages, the remaining parts need to be fitted exactly into gaps left over by early stages. At this point, the algorithm usually gets in trouble. We have now found the theoretical reason for this. There is, in general, no better strategy than just trying all possible combinations of torsion angles to avoid later collisions. Our theoretically derived molecule causes the same problems. No matter how the mobile torsion angles are chosen, no collisions occur in upper parts of the mobile chain. However the last atom must fit exactly into the only gap at the origin. This consideration leads to the conclusion that there is no smart way to foresee in early stages how to let a chain grow in order to solve later packing problems.

Given the fact that CP is NP-complete, what conclusions should be

drawn? The NP-completeness of CP does not imply that *all* instances of CP are hard to solve. It just means that some of them are difficult, i.e. the ones generated by the transformation from instances of the partition problem. We now know that there is no smart algorithm to solve the problem universally, for arbitrary instances. However, there might be a fast algorithm which finds solutions for most of the instances and fails only for a few of them. NP-completeness, thus, justifies the use of *heuristic* search techniques. Heuristic procedures usually offer parameters to tune the algorithm for specific problem instances.

3.4 The Complexity of Packing Algorithms

We introduce three packing algorithms: `Embed`, `PolyGrow` and `PolyPack`. They all belong to the class of heuristic search methods. The efficiency of these algorithms can be analyzed in two ways. In this chapter, we discuss their worst case time complexity. The time needed to pack specific instances of the Polymer Packing Problem (PP) is investigated in chapter 5.

The theoretical time complexity of an algorithm is defined by the computing time required as a function of the size of the problem to be solved (see [4]). The size of a polymer system is given by the total number of atoms contained in the system. There are usually many problem instances of the same size. The computing time for a certain problem size is thus determined by the time needed to solve the worst case instance. Worst case analysis leads to the fact that the chain packing problem is NP-complete (see section 3.3.1).

However, it may be the case that the problem instances which cause the original problem to be NP-complete are not of interest. This is true for the Polymer Packing Problem. In chapter 5 we show that `Embed`, `PolyGrow` and `PolyPack` are able to pack realistic problem instances in polynomial time.

Our packing algorithms all have the structure of an iterative search. To analyze their time complexity independently of specific problem instances, we split the computing time into two separate factors: The time to compute one iteration step and the number of iterations needed. The time complexity of one iteration step can be derived from the structure of the algorithm while the number of iterations is difficult to predict in our cases.

3.5 Embed: A Simple Embedding Algorithm

As a first embedding algorithm, we present Embed, which is similar to a steepest descent energy minimization algorithm, but driven by geometric constraints. The algorithm is simple and general in the sense that it can handle problems of the following form: Given a set of N points $\mathbf{p}_1 \ldots \mathbf{p}_N$ and a set of G geometric constraints $c_1 \ldots c_G$, find coordinates for $\mathbf{p}_1 \ldots \mathbf{p}_N$ such that all constraints are satisfied. Examples of geometric constraints are:

- An exact distance constraint: $|\mathbf{p}_i - \mathbf{p}_j| = d$
- An lower bound on angles: $\measuredangle(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k) > a$
- A chirality constraint: $Vol(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k, \mathbf{p}_l) < 0$

Embed starts with an arbitrary conformation. The conformation is improved iteratively until all geometric constraints are satisfied, or until we run our of patience:

```
procedure Embed;
begin
  Initialize p_1 ... p_N; iters ← 0;
  repeat
    solutionFound ← true;
    forall p ∈ p_1 ... p_N do
      △p ← 0;
      k ← 0;
      forall c ∈ c_1 ... c_G concerning p do
        if p does not satisfy c then
          p' ← location nearest to p which satisfies c;
          △p ← △p + (p' − p);
          k ← k + 1;
          solutionFound ← false;
        endif;
      endfor;
      if k > 0 then
        p ← p + △p / k;
      endif
    endfor
    iters ← iters + 1;
  until solutionFound or iters > maxIters;
end
```

Unfortunately, the process can get stuck. It may be the case that the actual embedding cannot be improved although there are still unsatisfied constraints. In this case, the search needs to be stopped and restarted from a new starting conformation.

The outermost loop of `Embed` implements the iterative search process. It is repeated until a solution is found. Inside the iteration loop, the actual conformation is improved by two nested loops. The first loop visits all points $\mathbf{p}_1 \ldots \mathbf{p}_N$. The innermost loop runs over all the constraints concerning the current point $\mathbf{p}$. If $\mathbf{p}$ violates constraint c, the new position $\mathbf{p}'$ for $\mathbf{p}$ is evaluated, as the point nearest to $\mathbf{p}$ that satisfies c.

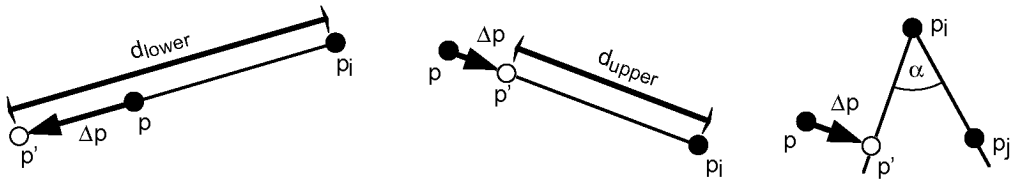

Figure 3.5: Computation of $\Delta\mathbf{p}$ for distance and angle constraints.

Figure 3.5 shows how $\mathbf{p}'$ is determined for distance and angle constraints. Each constraint c concerning $\mathbf{p}$ leads to a different displacement vector $\triangle\mathbf{p}$ for $\mathbf{p}$. At the end of the innermost loop, $\mathbf{p}$ is moved by the arithmetic mean of all these displacement vectors. Instead of computing a simple mean value, it would be feasible to evaluate a weighted average of the displacement vectors if certain constraints are more important than others. As a further extension, overrelaxation could be implemented by stretching the final displacement vector by a factor greater than one. The advantages of this algorithms are:

- It is efficient for systems up to about 100 points (see section 5.2.1 for detailed analysis).
- It is easy to understand and to implement.
- No numerical problems arise. Apart from the evaluation of the new position $\mathbf{p}'$, which is dependent on the specific geometric constraints, `Embed` makes use only of simple vector additions and substractions.
- The method can be adopted to many different kinds of geometric problems, because it can handle many types of geometric constraints.

3.5.1 Analysis of Embed

It is difficult to theoretically analyze Embed in general because its behavior depends on the structure of the specific embedding problem. In general, an iterative search results in one of three patterns of behaviour: Convergence, divergence and oscillation. Divergence is impossible in this case because repulsive forces (e.g. lower bounds on distances) only act up to a certain range.

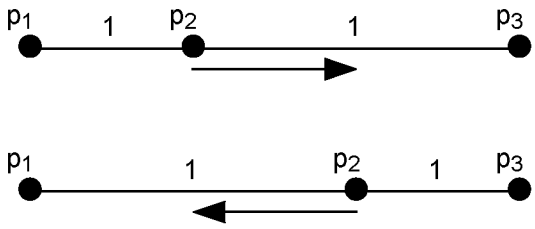

Figure 3.6: Oscillating states of Embed.

An example of two oscillating states is shown in figure 3.6. Given three points $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ in the plane and two distance constraints $|\mathbf{p}_1 - \mathbf{p}_2| = 1$ and $|\mathbf{p}_2 - \mathbf{p}_3| = 1$. If the positions of the points are updated in the order $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_2, \mathbf{p}_1 \ldots$ and $|\mathbf{p}_1 - \mathbf{p}_2| = 1$ and $|\mathbf{p}_2 - \mathbf{p}_3| = 2$ at the beginning, Embed does not find a solution but oscillates between the two shown states. However, if the points are updated in the natural order $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3 \ldots$ Embed finds the exact solution in only two steps. As our experiments show, oscillation occures very seldom with realistic embedding problems.

Embed converges almost always but not necessarily to a solution of the embedding problem. Figure 3.7 shows an embedding problem with two different stable fixpoints.

Given four points $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4$ in the plane and six distance constraints $|\mathbf{p}_1 - \mathbf{p}_2| = 1, |\mathbf{p}_2 - \mathbf{p}_3| = 1, |\mathbf{p}_3 - \mathbf{p}_4| = 1, |\mathbf{p}_4 - \mathbf{p}_1| = 1, |\mathbf{p}_1 - \mathbf{p}_3| = \sqrt{2}, |\mathbf{p}_2 - \mathbf{p}_4| = \sqrt{2}$, the unit square is the unique solution of the embedding problem. It is only one of two fixpoints of Embed. The second is shown in figure 3.7. Since the displacement vectors of all points annihilate each other, Embed gets stuck in this state. When started with points randomly selected from the unit square Embed finds the correct solution about 55% of the time.

In three dimensions, Embed overcomes local minima more easily as its application to chain packing problems shows. Embed succeeds more than 98% of the time in packing a simplified polymer chain of 100 spheres of diameter 0.9 at density 0.9 with fixed bond angle of 60^o and bond length 0.9. Density 1.0 means one sphere per cube unit. (See section 5.2.1).

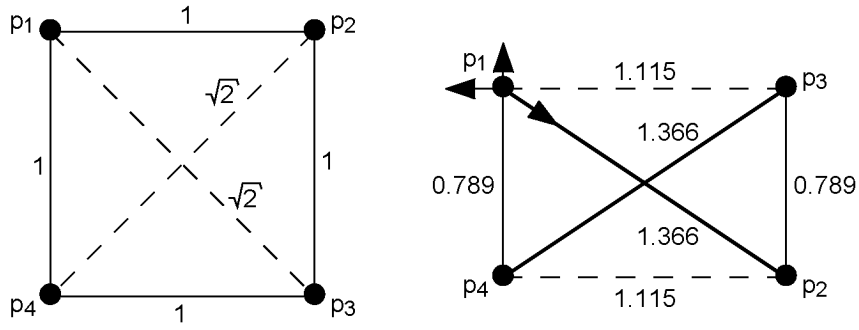

Figure 3.7: Two fixpoints of `Embed`: The solution and a local minimum.

As a further example we computed the coordinates of C_{60}, a Fullerene that has the shape of a soccer ball:

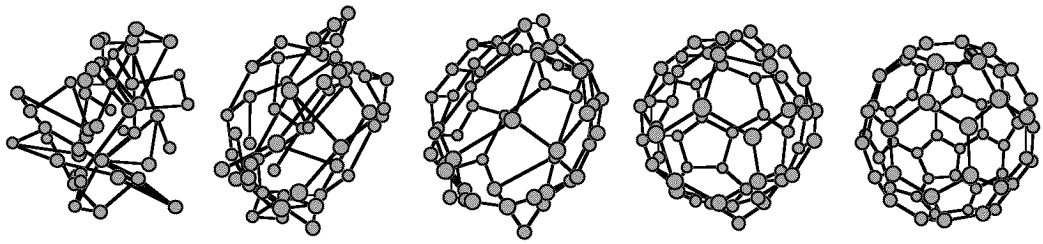

Figure 3.8: `Embed` computes the coordinates of the soccer ball iteratively.

The soccer ball is composed of 60 points and all 90 exact distance constraints due to bond lengths. In order to get a ball, 30 additional lower distance bounds on the diagonals are needed. 20 runs from different randomly chosen starting conformations led to the following result: Within 100 iteration steps, `Embed` always finds a conformation whose internal distances do not differ by more than 5% from the imposed ones. To compute 100 iterations takes about 5 seconds on a SGI R5000 workstation.

3.5.2 The Complexity of `Embed`

As mentioned in section 3.4, in analyzing the complexity of `Embed`, it is best to distinguish two distinct issues: The time to compute one iteration step and the number of iterations needed. We evaluate the complexity of one iteration step by analyzing the structure of `Embed`. The number of iterations depends on the structure of the embedding problem. We measured it for specific packing problems in section 5.2.1.

Let us assume that the number of points involved in a single geometric constraint is limited by a constant k. For distance and angle constraints: $k = 3$. Not more than $k \cdot G$ displacement vectors need to be computed in each iteration step. To average them and to move the points takes not more then another $k \cdot G$ time steps. The time complexity of an iteration step is therefore of the order $O(G)$.

In the case of polymer chains, there are N^2 lower bounds on distances representing the excluded volume condition of the atoms. The time complexity of an iteration step is therefore of the order $O(N^2)$. However, if a grid structure is used, the complexity can be reduced to $O(N)$ because not all constraints need to be considered (see section 4.2.3).

3.6 PolyGrow: Packing Simplified Polymers

In section 3.1.4 a simple model for large systems of polymers was introduced. A polymer chain is represented by spherical particles connected by bonds of fixed length. `PolyGrow` is an algorithm to generate dense systems of simplified polymers. Our hybrid algorithm combines the iterative algorithm `Embed`, described in the previous section, with the chain growing technique.

3.6.1 Chain Growing

`PolyGrow` starts with an empty box. The system is generated by growing chains atom by atom. During the chain growing process, `PolyGrow` uses backtracking [13] to look ahead a fixed number of steps. This way, the growing chain does not turn into dead end regions too easily. If there is no way to continue the current chain, `PolyGrow` tries to repair the collision generated by the new atom by applying `Embed` locally in the region of the collision. In other words, earlier placed chains are pushed away to make space for the new chain. The main procedure of `PolyGrow` looks as follows:

```
procedure PolyGrow:Boolean;
begin
  create empty box;
  for chain ← 1 to nrOfChains do
    trials ← 0;
    while trials < chainTrials do
      if AddChain() then break;
      RemoveChain();
      trials ← trials +1;
    endwhile;
    if trials = chainTrials then return false;
  endfor;
  return true;
end;
```

The main loop of `PolyGrow` tries to add a given number of chains to the empty box. The procedure `AddChain`, which generates a new chain, may get stuck when it tries to place a new chain in a disadvantageous, densely packed region. Since both the starting location as well as the chains conformation are randomly chosen, it is reasonable to restart `AddChain` a fixed number of times. Let us look at the procedure to add a new chain in detail:

```
procedure AddChain:Boolean;
begin
  len ← 0;
  while len < chainLen do
    if AddAtom() then
      len ← len +1;
    else
      if len < cutLen then return false;
      trials ← trials +1;
      if trials > cutTrials return false;
      remove cutLen atoms; len ← len − cutLen;
    endif;
  endwhile;
  return true;
end;
```

`AddChain` places a chain by adding atoms one by one. The procedure `AddAtom` may fail when there is no way to elongate the actual chain. In this case, `AddChain` removes a fixed number of atoms at the chain's end hoping to find a better way through the dense environment the next time. In contrast to an exhaustive backtracking search, `AddChain` cuts the chain not more than a constant number of times. The main ideas of `PolyGrow` are implemented in the procedure `AddAtom`:

```
procedure AddAtom:Boolean;
begin
  forall p ∈ possible positions do
    if LookAhead(p, searchDepth) then
      attach new atom at position p;
      return true;
    endif;
  endfor;
  forall p ∈ possible positions do
    attach new atom at position p;
    if RelaxRegion(p) then return true;
    remove new atom;
  endfor;
  return false;
end;
```

First, a discrete set of possible positions for the new atom is evaluated.

This step depends on the particular model of the polymer system. The positions are tested by the procedure `LookAhead` in random order. `LookAhead` performs a backtrack search of constant depth. It checks whether the chain can be elongated via the new position by a certain number of atoms without overlaps. This is important because the chains are not completely flexible. Their stiffness is determined by the the choice of the bond angle distribution (see section 4.1).

```
procedure LookAhead(p, depth):Boolean;
begin
  if depth ≤ 0 then return true;
  forall p_succ ∈ possible successors of p do
    if p_succ does not overlap then
      if LookAhead(p_succ, depth-1) then return true;
    endif;
  endfor;
  return false;
end;
```

If a promising position **p** is found inside the first loop of `AddAtom`, a new atom is attached at this location. Otherwise another strategy is applied: The second loop tests all possible positions in the order of their expected quality. Such an order can be generated by using the knowledge gained during the first loop. The new atom is attached at position **p** although it may generate overlaps with existing atoms. If the new atom does not overlap with any other atoms, `AddAtom` returns successfully. Otherwise, the newly produced overlaps need to be removed. This is not a trivial task, since the colliding atoms belong to earlier generated chains. If such atoms are moved, adjacent bond lengths as well as bond angles will change. However, neither bond lengths, nor bond angle statistics should be disturbed during the collision correction. The only way to modify the positions of atoms inside a chain while keeping bond lengths and bond angles fixed, is a simultaneous rotation of at least seven torsion angles. This complex operation is discussed in section 3.7.6 in more detail. It needs to be performed simultaneously on all the chains, contributing to the collisions of the newly positioned atom.

3.6.2 Local Relaxation

Fortunately there is a way to avoid such complex calculations. Since the atoms which need to be moved form a rather small and local system of particles, `Embed` is perfectly suitable to solve the local packing problem. First,

all atoms within a certain distance to the new atom are marked to be moved by `Embed`. Additionally, if an atom is marked, two atoms before and after it on the same chain are marked also. This step guarantees that there is enough flexibility to change the positions of atoms inside the chain. The geometric constraints consist of exact distance constraints derived from bond lengths and bond angles, and of lower distance bounds from the hard-spheres condition. Thus, the `RelaxRegion` call simply applies `Embed` on the set of marked atoms.

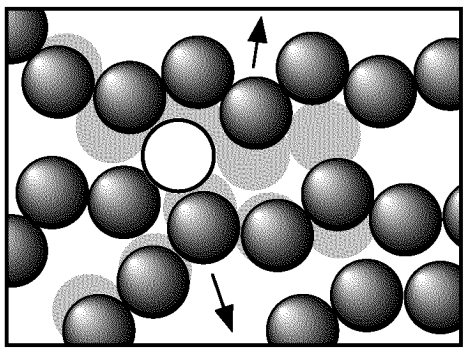

Figure 3.9: `Embed` is used to make space for a new atom, keeping all bond angles and bond lengths fixed. The gray circles show the positions of the atoms before the relaxation.

3.6.3 The Complexity of `PolyGrow`

To analyze the time complexity of `PolyGrow`, we split the computing time into the time used by `AddAtom` to add a new atom to the system and the number of times `AddAtom` is called.

The time used by `AddAtom` is independent of the problem size. The number of possible positions for the new atom is constant. For each position, a search of constant depth is started. If there is no space for the new atom, `RelaxRegion` is called at most once for each new position. The number of particles `RelaxRegion` moves is independent of the problem size. Therefore, the time used to add one atom is independent of the total number of particles in the system.

Since `AddAtom` can fail, it is difficult to predict how many times it is called during the packing process. If N is the number of particles, `AddAtom` is called exactly N times in case it never fails. This leads to linear time complexity $O(N)$ in the best case.

As our experiments show, `PolyGrow` reaches linear time complexity for realistic systems of simplified polymer chains (see section 5.2.2).

3.7 PolyPack: Packing Detailed Polymers

PolyGrow is efficient in packing simplified polymer systems. However, if atomistically detailed conformations of dense polymer systems are of interest, PolyGrow is not a suitable algorithm. The success of PolyGrow is based on the efficiency of the local relaxation technique. In case of simplified chains, local relaxation is efficient because an entire monomer is represented by a single sphere. The smallest number of torsion angles that need to be modified simultaneously along a chain, in order to keep bond angles and bond lengths fixed is seven (see section 3.7.6). A simultaneous rotation of seven subsequent torsion angles relocates only four spheres of a simplified chain, but six rigid groups if single atoms are considered. The first and the last group are not translated, but rotated about their own axes. The rigid groups of polycarbonate, for example, consist of three to twelve atoms. Thus, at least 40 atoms of every chain involved in a collision, together with a large number of additional constraints to keep the groups rigid, form a system too large to be relaxed by Embed. To generate dense, atomistically detailed polymer systems, we present an alternative algorithm, called PolyPack. In contrast to the previously discussed methods, PolyPack uses generalized coordinates (see section 3.1.1). This way, the number of parameters is reduced drastically. A monomer unit of polycarbonate, for example, consists of 33 atoms (99 degrees of freedom when Cartesian coordinates are used), but only 8 torsion angles. Consequently, PolyPack performs an iterative, heuristic search in torsion angle space.

3.7.1 The Optimization Target

PolyPack tries to reach two competing goals, the agreement of the torsion angle values with the proposed distribution and the elimination of atom overlaps in a dense, periodic environment. To assess the quality of a system conformation, we introduce a penalty or cost function inspired by the hard-sphere type of interaction. The collision function coll between two atoms k and l is measured as the fractional overlap of the two hard-spheres relative to the sum of the two hard-sphere radii:

$$\mathrm{coll}(k,l) := \max\left\{0, 1 - \frac{\parallel \mathbf{r}_k - \mathbf{r}_l \parallel_L}{r_k^0 + r_l^0}\right\} \tag{3.2}$$

where r_k^0 and r_l^0 are the hard-sphere radii for the atoms k and l respectively, and $\mathbf{r}_k$ and $\mathbf{r}_l$ their location. The distance $\parallel \mathbf{r}_k - \mathbf{r}_l \parallel_L$ is measured according to the minimal image convention for a system in a periodic box of side length

L. An overall-cost function may now be defined as the maximum over all atom collisions

$$\text{cost}^{\infty}(D) := \max_{(k,l)\text{: atom pair}} \text{coll}(k,l) \tag{3.3}$$

where D is the conformation, defined in torsion angle space (see 3.1.1). However, it is hardly possible in practice to find a "reasonable" conformation by starting a search with a random conformation and taking into account all collisions from the beginning. To overcome this problem, we reduce at first the number of collisions considered and gradually increase that number to eventually include the full range of interactions among the atoms.

The Horizon Parameter

This can be achieved by limiting the "horizon" h beyond which pairs of atoms are "ghosts" to each others. If two atoms of the same molecule are considered, they are linked by a series of free bonds. If the number of free bonds along the shortest path joining these two atoms is larger than the horizon value h, the collision contribution of this atom pair is ignored.

In the case of a pair of atoms belonging to different molecules, the intermolecular collision is considered only if both atoms are separated by less than $h/2$ free bonds from their molecule's central free bond. That way collisions between atoms near the center of a molecule can be eliminated in early stages of the search. Figure 3.10 shows all atom pairs whose overlaps are considered at horizon 2.

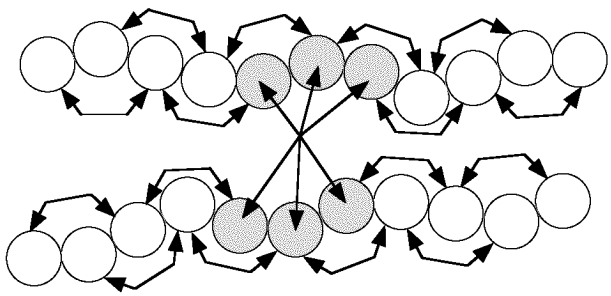

Figure 3.10: The considered collisions at horizon 2.

The overall cost function $\text{cost}(D,h)$ of a conformation D is defined as the maximum value of the collision function over all possible atom pairs within a given horizon h. The cost function reads

$$\text{cost}(D,h) := \max_{(k,l)\text{: atom pair within } h} \text{coll}(k,l). \tag{3.4}$$

3.7.2 The Optimization Strategy

PolyPack addresses the problem of minimizing the cost function $\text{cost}(D, h)$ under the constraint of torsion angle distributions, given by the RIS model. The search begins with an arbitrary conformation. An outer loop initializes the horizon value with zero and increases it step by step until it reaches its maximum value, where all possible collisions are considered. At each stage, the inner loop of PolyPack tries to minimize the cost function for the fixed horizon h by performing moves on the actual conformation D. Besides the simple move of changing a single torsion angle, we implemented a variety of other moves to manipulate D in torsion angle space. The choice of acceptance or rejection of these moves is driven by the cost function $\text{cost}(D, h)$ and the probability of the resulting conformation $\text{prob}(D)$ given by the RIS model.

3.7.3 The Basic Optimization Step

The basic optimization step consists of the optimization of all torsion angles in random order. When a torsion angle ϕ_i is chosen, the algorithm systematically scans the set of its RIS states, and attempts to eliminate all the values of ϕ_i that cause collisions that exceed a set threshold c_{goal}. In a second step, the algorithm picks one value for ϕ_i of the remaining values according to its RIS probabilities $\text{prob}(D, \phi_i)$. If none of the RIS states decreases the collision below the threshold c_{goal}, then the one angle giving the lowest collision value is chosen (see figure 3.11). The parameter c_{goal} is analogous to a temperature parameter in a physical system. It permits to overcome cost-function barriers.

3.7.4 The Search Loops

The search starts with an arbitrary conformation. In the innermost loop, the molecular structure is iteratively improved by performing optimization steps based on different types of moves. The iteration continues until the collision function $\text{coll}(D, h)$ attains a local minimum of reaches a lower limit.

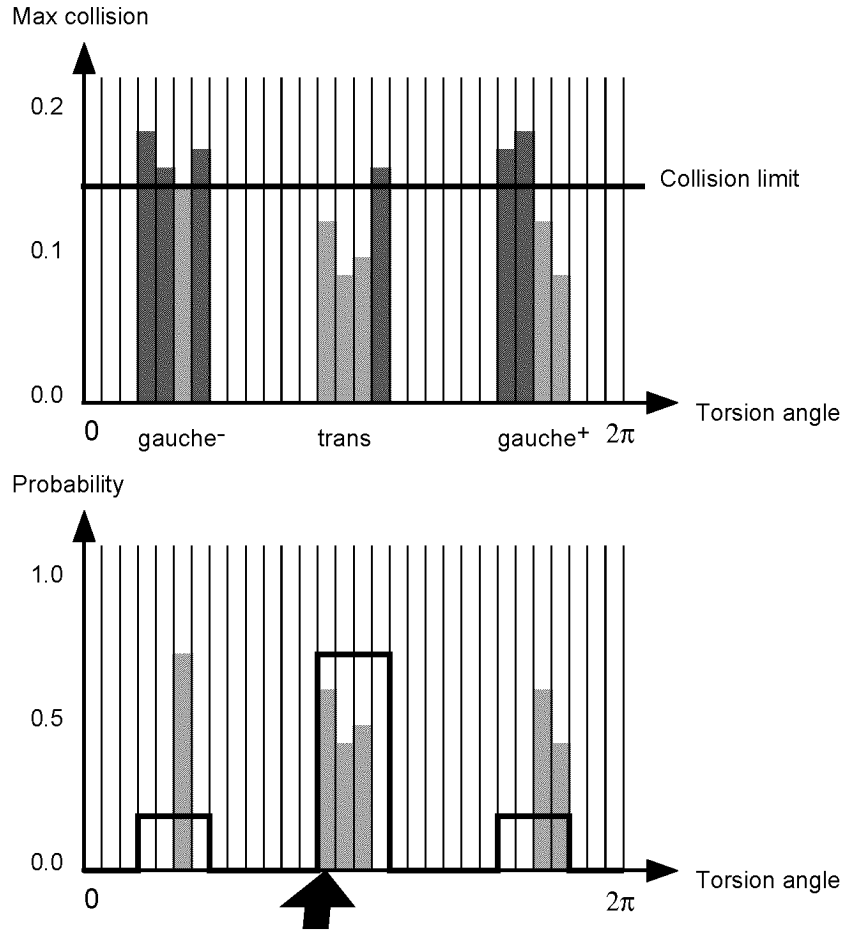

Figure 3.11: Selection of a new torsion angle value considering atom collisions and torsion angle probabilities.

procedure Optimize(D, c_{goal}, h);
begin
  repeat
    forall move $\in$ set of moves **do**
      OptimizationStep(D, move, c_{goal},h);
    endfor
  until (cost(D, h) $\leq c_{goal}$) **or** configuration D not changed;
end.

If none of the attempted optimization steps decreases cost(D, h) below c_{goal}, the search path is trapped in a minimum (almost certainly local). To overcome the barriers of the cost function in the high-dimensional conforma-

tional space, we opted for a local shaking to randomize the system, similar to a sudden increase in temperature in the simulated annealing technique. The "shake" procedure consists of an instantaneous increase of the collision limit c_{goal} followed by a step-by-step reduction of c_{goal} until its original value is reached again. It turns out to be a robust method to escape local minima:

procedure Shake(D, c_{goal}, h);
begin
  $D_{save} \leftarrow D$;
  $c_{shake} \leftarrow (\text{cost}(D, h) + c_{offset})$;
  repeat
    Optimize(D, c_{shake}, h);
    if cost(D, h) $\leq c_{shake}$ **then** $c_{shake} \leftarrow (c_{shake} - c_{delta})$;
  until $c_{shake} \leq c_{goal}$ **or** configuration D not changed;
  if cost(D_{save}, h) $<$ cost(D, h) **then** $D \leftarrow D_{save}$;
end.

In the case where the "shake" algorithm fails to improve the structure, the procedure automatically returns the original structure.

We now have all ingredients to formulate the complete embedding algorithm `PolyPack`. At the beginning, the horizon h is set to zero. In other words, no collisions are taken into account. The starting configuration is constructed by choosing torsion angle values according to the RIS probabilities. Then the centers of gravity of the chains are spread evenly throughout the box.

The outer loop consists of increasing the horizon h by an amount h_{step} until h reaches its maximum, where all possible collision pairs are considered. For each value of h, an optimization step is performed. If one of those optimization steps does not succeed, the "shake" procedure is used. To summarize the entire algorithm:

procedure HorizonLoop(D, c_{goal});
begin
  Construct an appropriate starting structure D;
  for $h \leftarrow 0$ **to** h_{max} **by step** h_{step} **do**
    $trial \leftarrow 1$;
    repeat
      Optimize(D, c_{goal}, h);
      if cost(D, h) $> c_{goal}$ **then**
        Shake(D, c_{goal}, h);
      endif
      $trial \leftarrow (trial + 1)$;
    until $trial > trial_{max}$ **or** cost(D, h) $< c_{goal}$;
  endfor
end.

Figure 3.12 illustrates the principle of increasing the horizon value. At early stages, the packing problem is easy to solve because only few atom collisions are considered. With increasing horizon, the problem becomes more and more difficult. At each stage, the search procedure uses the result from the previous stage as starting conformation.

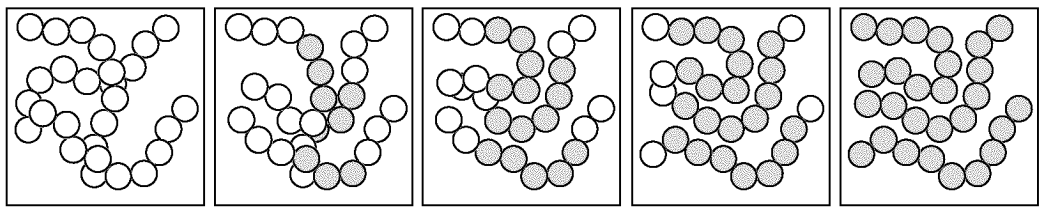

Figure 3.12: Gradually increasing problem difficulty by using horizon parameter.

The efficiency of the algorithm strongly relies on the capabilities of the moves performed at each optimization stage to relax the torsion angles.

3.7.5 The Moves

Our algorithm uses five moves. The simplest one in the framework of generalized coordinates is to modify one torsion angle at a time; it is called a single rotation (see figure 3.13).

To overcome problems caused by the manipulation of single torsion angles inside long chains, we devised an alternative move, the Parallel Rotation move (section 3.7.6).

The two next moves treat a single rigid object. The entire molecule can be translated by a small displacement vector, or rotated around its center of

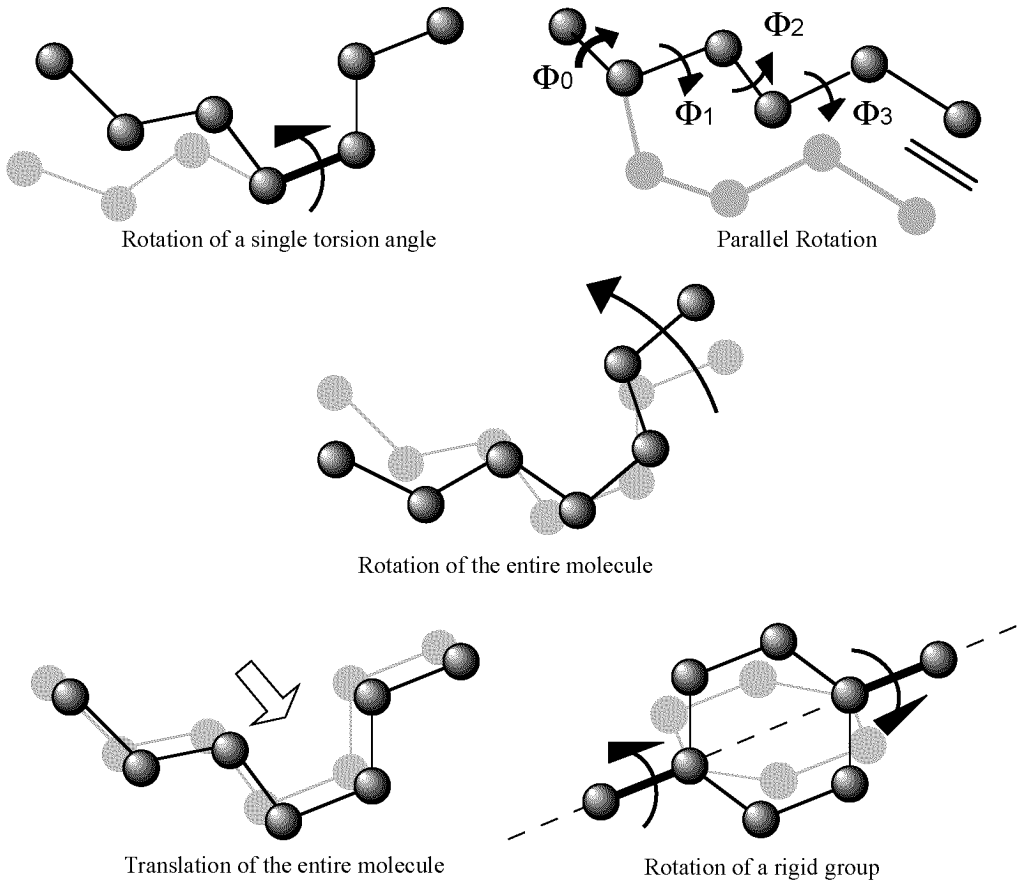

Figure 3.13: The five moves used in `PolyPack`.

mass. The first move modifies the global position of the chain, the second one its orientation. No torsion angles are changed.

The last move has been introduced specifically for those polymer chains that consist of rings along their backbone. The rigid group comprising the ring is connected to its neighboring rigid groups by two collinear bonds. When the two torsion angles around the ring are varied by the same amount but in opposite directions, the ring rotates while the rest of the molecule remains in place. This group rotation move is of high efficacy for particular molecular structures.

3.7.6 Parallel Rotation (ParRot)

Since `PolyPack` employs the model of generalized coodinates, the internal 3-dimensional structure of a molecule can only be changed by modifying its

torsion angles. Unfortunately, the simplest move, the modification of a single torsion angle, suffers major drawbacks. First, the displacement of atoms far from the rotating angle is likely to be large. This leverage effect makes impossible to correct overlaps by a single rotation without evoking a large number of new collisions. Second, the rotating part of the molecule changes its orientation with regard to the periodic box. This change of orientation on its own can cause severe inter and intra-molecular atom overlaps due to the periodicity of the system.

The concerted rotation move offers a solution to these problems [6]. Instead of changing a single torsion angle, a series of seven torsion angles is changed simultaneously. One angle may be modified freely while six angles complensate its rotation as figure 3.14 shows.

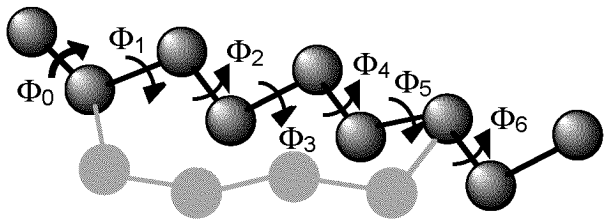

Figure 3.14: The Conrot move: $\phi_1, \ldots, \phi_6$ compensate the rotation of the driver angle ϕ_0.

The concerted rotation `ConRot` affects only the positions of the atoms between ϕ_0 and ϕ_6. Six torsion angles are needed to compensate the rotation of ϕ_0 because the location (3 degrees of freedom) as well as the orientation (another 3 degrees of freedom) of the chain following ϕ_6 must remain fixed. Given the seven original values $\phi_0, \ldots, \phi_6$ and a new value ϕ'_0 for the "driver" angle, the new values $\phi'_1, \ldots, \phi'_6$ are computed such that atoms subsequent to ϕ_6 do not move while all bond lengths and bond angles are kept fixed. This delicate problem needs to be solved numerically since it has no analytical solution [6]. A lot of programming and computing effort is needed to perform this complex move.

In view of these difficulties, we devised the much simpler parallel rotation move `ParRot`. Instead of correcting both, the location *and* orientation of the subsequent chain, only its orientation is kept fixed. Consequently, only three compensating torsion angles ϕ_1, ϕ_2 and ϕ_3 are involved in a parallel rotation move. The chain beyond ϕ_3 moves parallel to its original position and all its atoms are translated by the same vector, independently of their distance to to rotation center.

The mayor advantages of `ParRot` are:

1. The geometric problem has an analytical solution that can be proven to

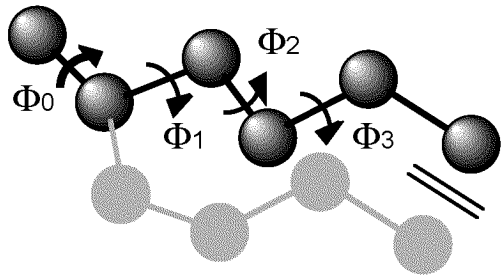

Figure 3.15: The ParRot move: ϕ_1, ϕ_2, ϕ_3 compensate the rotation of the driver angle ϕ_0.

have at most two solutions. This leads to its high success rate and efficiency as part of `PolyPack`. Furthermore `ParRot` has particularly high acceptance rates when it is used as move in Monte Carlo algorithms [33].

2. `ParRot` does not suffer any leverage effect, because of the conservation of the orientation of the moving chain segment.

3. In contrast to the rotation of a single torsion angle, `ParRot` causes no additional intra molecular collisions, because the moving chain segment remains parallel to its original orientation.

4. Unlike `ConRot`, `ParRot` is a global move which is able to change the entire conformation of a molecule, even in a dense environment.

A typical ParRot move consists of the following steps: (i) a bond with torsion angle of initial value ϕ_0 is selected within the chain; (ii) one of the two chain parts starting from that bond is selected and the direction vectors $\mathbf{u}$ and $\mathbf{u}_\perp$ defining its orientation are calculated; (iii) a new value ϕ_0^{new} is assigned to this torsion angle; (iv) sets of new values $\phi_1^{\text{new}}, \phi_2^{\text{new}}$, and ϕ_3^{new} for the three torsion angles ϕ_1, ϕ_2, ϕ_3 consecutive to ϕ_0 on the side of the moving chain part are calculated so that $\mathbf{u}$ and $\mathbf{u}_\perp$ remain unchanged. Only the four values $\{\phi_0, \phi_1, \phi_2, \phi_3\}$ are changed during the move. In this formulation, we will use the torsion angle ϕ_0 as the "driver angle" for the ParRot move.

The three compensating torsion angles $\{\phi_1^{\text{new}}, \phi_2^{\text{new}}, \phi_3^{\text{new}}\}$ satisfy a system of two equations. They account for the constraints of both the vectors $\mathbf{u}$ and $\mathbf{u}_\perp$ being constant for the ParRot move. We have

$$\mathbf{u} = \mathbf{T}(\phi_0^{\text{new}})\mathbf{T}(\phi_1^{\text{new}})\mathbf{T}(\phi_2^{\text{new}})\mathbf{e}_x = \mathbf{T}(\phi_0)\mathbf{T}(\phi_1)\mathbf{T}(\phi_2)\mathbf{e}_x \tag{3.5}$$

$$\mathbf{u}_\perp = \mathbf{T}(\phi_0^{\text{new}})\mathbf{T}(\phi_1^{\text{new}})\mathbf{T}(\phi_2^{\text{new}})\mathbf{T}(\phi_3^{\text{new}})\mathbf{e}_y = \mathbf{T}(\phi_0)\mathbf{T}(\phi_1)\mathbf{T}(\phi_2)\mathbf{T}(\phi_3)\mathbf{e}_y \tag{3.6}$$

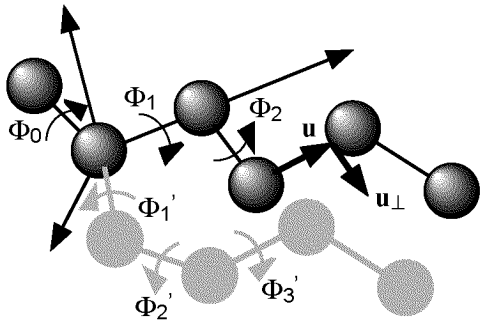

Figure 3.16: The ParRot move leaves $\mathbf{u}$ and $\mathbf{u}_\perp$ unchanged

where the rotation $\mathbf{T}(\phi) = \mathbf{R_x}(\phi)\mathbf{R_z}(\pi + \theta)$ - as defined by Mattice and Suter [22] - is composed of two rotations specifying the transformation of a vector in the frame of reference of bond $i + 1$ into the frame of bond i. The angle θ denotes the bond angle between the bond i and $i + 1$ that must not necessarily be the same for all the bond junctions.

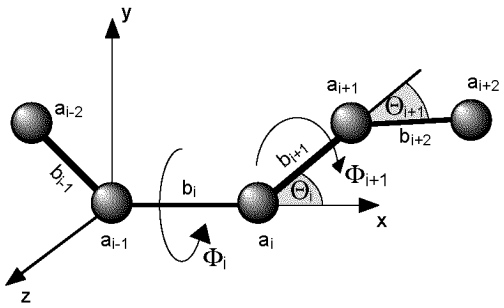

Figure 3.17: Denotations of a bond's local frame of reference

Solving eqn (3.5) provides us with a solution for ϕ_2 which reads

$$\cos\phi_2^{\text{new}} = \frac{\cos\theta_1\cos\theta_2 - \mathbf{v}_x(\phi_0^{\text{new}}, \phi_0, \phi_1, \phi_2)}{\sin\theta_2\sin\theta_1} \tag{3.7}$$

where $\mathbf{v}_x$ is the x-component of the vector

$$\mathbf{v}(\phi_0^{\text{new}}, \phi_0, \phi_1, \phi_2) = \mathbf{T}(\phi_0^{\text{new}})^{-1}\mathbf{u}(\phi_0, \phi_1, \phi_2). \tag{3.8}$$

Eqn (3.7) has either zero, one, or two solutions depending on the absolute value of its right-hand side being larger than, equal to, or smaller than 1.

Upon substituting the values of ϕ_2^{new} into eqn (3.6), one obtains a single equation giving for each value of ϕ_2^{new} a corresponding value for ϕ_1^{new} being

the unique solution of the linear equation

$$\begin{bmatrix} \cos\phi_1^{\text{new}} \\ \sin\phi_1^{\text{new}} \end{bmatrix} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \frac{1}{1-\mathbf{v}_x^2} \begin{bmatrix} \mathbf{v}_y \\ \mathbf{v}_z \end{bmatrix} \tag{3.9}$$

where the parameters $a := \sin\theta_1\cos\theta_2 + \cos\phi_2^{\text{new}} + \cos\theta_1\sin\theta_2$ and $b := \sin\theta_2\sin\phi_2^{\text{new}}$ depend on the new value for the second torsion angle ϕ_2^{new}.

For each existing solution of eqn (3.7), eqn (3.9) has exactly one solution. Consequently the number of possible sets of torsion angles fulfilling the condition in eqn (3.6) can be analytically determined to be either zero, one, or two, by solving eqn (3.7). A typical set of trajectories of the moving torsion angles for a driving angle varying from 0 to 2π and the relative distance to the original position of the moving chain segment are shown in figure 3.19. Note that this displacement is the same for all moved atoms beyond the third flexible bond and that its amplitude for a varying driving torsion angle almost always remains below the typical van der Waals radius of the atoms involved.

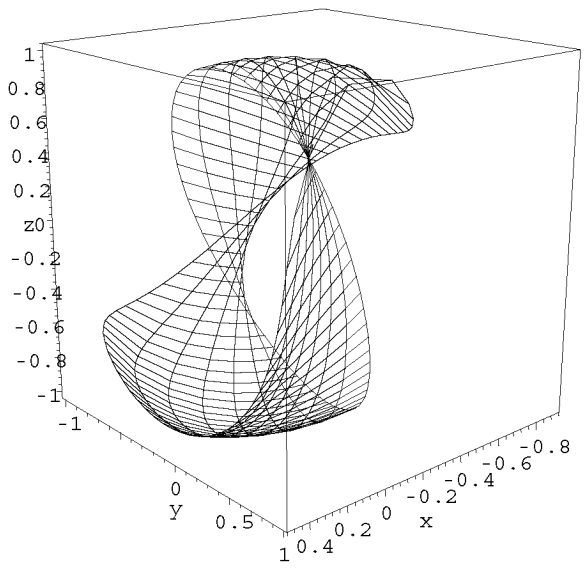

Figure 3.18: 3d-trajectories generated by the moving chain segment of `ParRot` ($b_i = 1.0,\ \theta_i = 120^o,\ \phi_1 = -50^o, \phi_2 = 10^o \ldots 80^o, \phi_3 = 30^o, \phi_0' = 0 \ldots 360^o$)

The proposed packing algorithm has as target the distribution of torsion angles. To obtain the distribution desired, one has to correct for the bias in the choice of the torsion angles introduced by the ParRot move. This bias results from the simultaneous rotation of the four involved torsion angles that are dependent of each other via eqn (3.5) and eqn (3.6). A uniform

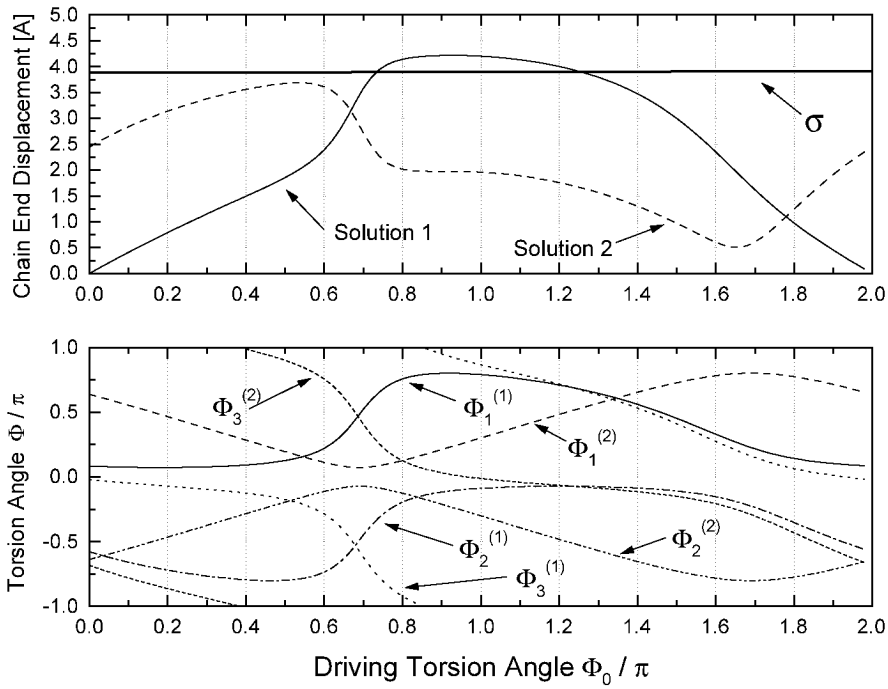

Figure 3.19: Displacement distance of moved chain segment in ParRot

distribution for the driving angle does not lead to a uniform distribution for the three driven torsion angles. A relative weight, the Jacobian weight [6], for the set of torsion angles $\phi_0, \phi_1, \phi_2, \phi_3$ that is a solution of eqn (3.7) and eqn (3.9), has to be considered in the choice of a configuration. This Jacobian weight is given by [33]

$$J(\phi_2) = \mid \mathbf{u} \cdot (\mathbf{u}_1 \times \mathbf{u}_2) \mid \tag{3.10}$$

where $\mathbf{u}_1$ and $\mathbf{u}_2$ are the unit bond vectors at the torsion angles ϕ_1 and ϕ_2, respectively.

To conclude, we have devised a new simple and robust move that has been shown [33] to be very efficient in relaxing all the torsion angles even deeply inside a long chain in a dense environment.

3.7.7 General Parallel Rotation

So far, we described the parallel rotation move in connection with a simple chain of free bonds connecting single atoms. In this model, two adjacent free bonds meet at the center of the common atom, i.e. at a single point. However, the tree model (figure 3.1) used in `PolyPack`, which decomposes

a molecule into free bonds and rigid groups asks for a more general ParRot move. As shown in figure 3.20, adjacent free bonds do not need to meet at a single point.

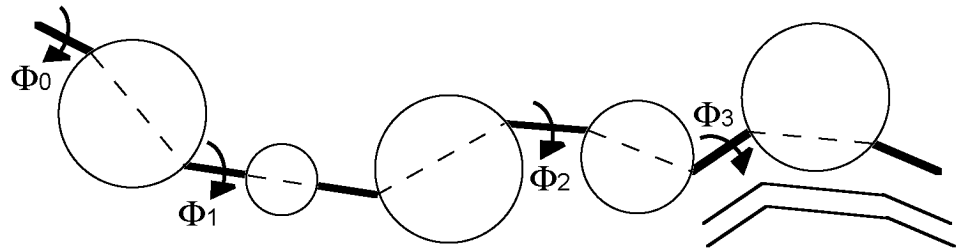

Figure 3.20: Generalized parallel rotation: Virtual bonds with fixed torsion angles need to be added while collinear bonds are excluded. Rigid groups are drawn as white spheres.

A rigid group, thus, cannot be replaced by a singe atom at the intersection point. It must be replaced by a virtual bond (the dashed lines) connecting the free bonds. Between the four free bonds with variable torsion angles $\phi_0 \ldots \phi_3$, three virtual bonds need to be inserted as terms in equations 3.5 and 3.6. Fortunately, their torsion angles are constant. Thus, the resulting equations can be solved the same way as the original ones. A second issue which needs to be taken into account when dealing with general molecules, is shown in figure 3.20. The free bond following ϕ_1 is collinear to ϕ_1. Collinear bonds appear in polycarbonate for example (see section 5.3.6). The collinear torsion angle does not constitute a separate degree of freedom and thus, cannot contribute to the compensation of the rotation of driver angle ϕ_0. Therefore, it is combined with ϕ_1 for the computation of the ParRot move. The introduction of virtual bonds together with the elimination of collinear bonds make possible the application of the ParRot move to arbitrary molecular structures.

3.7.8 The Complexity of `PolyPack`

`PolyPack` uses generalized coordinates. It packs systems of M identical molecules with T torsion angles each. We define the problem size as the total number of degrees of freedom, i.e. the number of torsion angles $M \cdot T$. The total number of atoms is proportional to the number of degrees of freedom for realistic molecules, since the size of rigid groups is independent of the problem size and varies from two to a maximum of about 30 atoms.

A single move consists of the rotation of a set of atoms and the computation of the new maximum collision after the move. The maximum of rotating

atoms is proportional to the size T of the molecules. The number of pairs of atoms which need to be checked for new collisions is proportional to $M \cdot T^2$ since all rotating atoms have to be checked against all atoms in the system. The introduction of a grid (see section 4.2.3) reduces the cost of the collision check procedure to the order T. The complexity of the execution of a single move is thus proportional to T.

A single torsion angle is optimized by performing moves for a constant number of angles given by the RIS model. Therefore, its complexity remains proportional to T. The time needed to optimize all $M \cdot T$ torsion angles is thus of the order $M \cdot T^2$.

The number of optimization steps performed at a fixed horizon value cannot be derived directly from the structure of the algorithm. It is dependent on the characteristics of the problem i.e. the density of the system and the covalent structure, the number (M) and the size (T) of the molecules. We isolate this non predictable factor i. The horizon loop is executed h_{max} times. Since h_{max} is proportional to T, we end up with a time complexity of the order $O(i \cdot M \cdot T^3)$ for `PolyPack`.

Chapter 4

Polymer Packing Software

The packing algorithms described in the previous chapter have been implemented as comprehensive and user-friendly tools for generating starting conformations for any molecular structures. The software is written in ANSI C. This language guarantees high performance and portability because optimizing C-compilers are available on almost every type of machine and operating system.

`PolyGrow` and `PolyPack` are the two main applications for computing systems of simplified chains and packings of atomistically detailed polymer systems, respectively. Both programs feature interactive visualization based on the X-Window system. They are also available in a command line version convenient for batch jobs and systems without X-Window support.

This chapter describes the usage of the applications from a users point of view as well as part of the program's internal structure. A user does not need to understand or modify the applications source code. General packing problems as defined in section 3.2 are specified via text files or command line parameters.

Since a few parameters are sufficient to define the properties of simplified polymer chains, `PolyGrow` is easy to use. The packing process is started with just a set of command line parameters. However, in atomistic models, the covalent structure as well as detailed geometric information of the molecules comprised in the system of interest need to be specified. Consequently `PolyPack` interacts with existing biochemical software and exchanges information via files of standardized formats.

4.1 PolyGrow

The application `PolyGrow` is invoked by the following command:

```
polygrow N chainLen
```

The user provides two mandatory and various optional command line parameters. The first parameter N determines how many particles need to be squeezed into a cubic periodic box. The side length L of the periodic box depends on the number of particles N as well as on the target density ϱ of the system:

$$L = \sqrt[3]{\frac{N}{\varrho}} \tag{4.1}$$

The density is measured in particles per cubic unit and is set to 1.0 by default, unless another value is specified via the optional parameter `-d`.

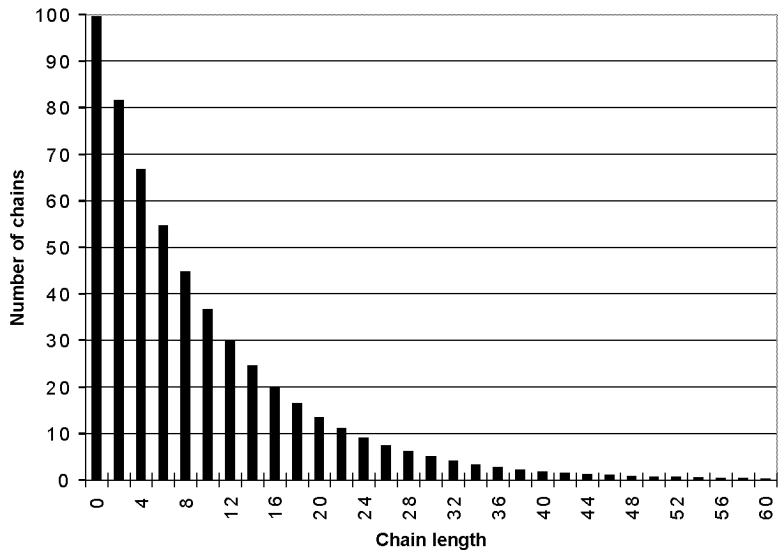

Figure 4.1: Distribution of chain lengths of a system of 10000 particles.

The second mandatory parameter fixes the number of particles per chain. If this value is set to zero, the chain lengths are distributed exponentially, where the number of chains of length l is given by

$$n_l = e^{\frac{-l}{k}} k(1+k) e^{\frac{-1}{k}} \tag{4.2}$$

The constant k can be evaluated from the system size, namely $k = \sqrt[4]{N}$, such that the number of particles enclosed in all chains equals the total number of particles given by the parameter N.

The optional command line parameters of `PolyGrow` are summarized in table 4.1. Since the particles of a simplified chain usually do not represent

single atoms but entire monomer units, the angles between subsequent bonds need not be fixed at a typical bond angle value. Some applications require to generate chains whose bond angles vary according to a certain distribution function. This way, chains of diverse stiffnesses can be simulated by tuning the distributions standard deviation. The two optional parameters `-a` and `-k` of `PolyGrow` may be used to choose among those two different models. If a bond angle value is specified via the parameter `-a`, chains of fixed bond angles are generated. In contrast, if a value `kappa` is stated by the parameter `-k`, the bond angles will be distributed according to

$$\text{prob}(\theta) \propto e^{\texttt{kappa}(1-\cos(\theta))} \tag{4.3}$$

Figure 4.2 shows the distribution of bond angles for different values of `kappa`. A stretched bond angle is defined to have zero degrees. Thus, the larger `kappa` becomes, the more angles will be nearly stretched and the less flexible the chains will be. On the other hand, small values for `kappa` allow the chains to bend almost freely because larger angles become more probable.

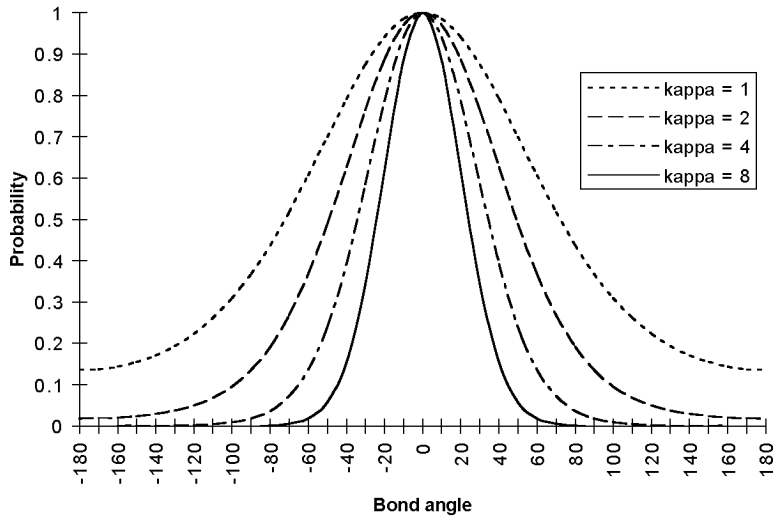

Figure 4.2: Distributed bond angles. The parameter `kappa` determines the stiffness of the chains.

After the start of `PolyGrow` with the appropriate parameters, the coordinates of the specified system are generated by using the chain growing algorithm described in section 3.6. This process is visualized if the X interface is activated by the `-x` parameter. Various buttons allow the user to trace the algorithm step by step or in a fast forward mode. The chains are displayed as wireframes to visualize their structure or as sequences of spheres

Parameter	Meaning	Default
`-a bondAngle`	Use fixed bond angles. The size must be specified in degrees ($0 \ldots 360$).	60^o
`-b`	Generate polymer brush. All chains grow from $z = 0$. The box is periodic only in x and y dimensions. Chains, thus, cannot cross the $z = 0$ plane.	no
`-d density`	The density must be specified in particles per cubic unit. It determines the size of the cubic box.	1.00
`-k kappa`	Use distributed bond angles. The parameter `kappa` describes the stiffness of the chains (see figure 4.2).	fixed angles
`-l bondLen`	The distance between consecutive particles.	0.9
`-n nrAngles`	Number of possible positions generated when a new particle is attached to a chain.	20
`-o maxOverlap`	The maximum admitted overlap of two particles in the final conformation ($0.0 \ldots 1.0$)	0.10
`-r atomRadius`	Radius of the particles.	0.50
`-s searchDepth`	Search depth of the procedure `LookAhead`, described in section 3.6.	10
`-t chainTrials`	Defines, how many times a chain is removed and regrown before `PolyGrow` stops (see section 3.6). 0 means infinity.	0
`-x`	Display interactive X-Window interface.	no

Table 4.1: Command line parameters of `PolyGrow`.

to give an impression of the density of the system. When the algorithm succeeds in packing all particles, their coordinates are written to a text file of the following format:

```
Grid parameters:
  box_len     = 10.772174
  atom_radius = 0.500000
  bond_len    = 0.900000
  brush       = no
  angle_fixed = yes
  bond_angle  = 1.047198
  kappa       = 4.000000

Grow parameters:
  nr_particles    = 1000
  chain_len       = 100
  max_overlap     = 0.200000
  nr_angles       = 20
  search_depth    = 10
  nr_chain_trials = 0
100
     8.0792    0.5384    4.8462
     8.3654    0.1757    4.0738
     8.0073    0.4415    3.2920
[...]
100
     4.8475    7.0019    7.0019
     5.1723    6.1666    7.0844
     6.0176    6.0382    7.3655
[...]
```

The header section contains all parameters needed to describe the properties of the system. The rest of the file is composed of blocks for every chain. Each block begins with an integer value, specifying the length of the chain followed by the coordinates of the individual particles.

4.2 PolyPack and PolyCmd

The packing algorithm `PolyPack` is available as a software package which contains four tools. To generate conformations of atomistically detailed polymer systems, we provide a visual, interactive application called `PolyPack` and its

command line version `PolyCmd`, suitable for batch job mode. Additionally we offer a file conversion tool `mdf2pp` and `PolyStat` for analyzing the generated conformations. Two separate sections are devoted to the two latter tools.

The description of the packing problem of an atomistically detailed polymer system is too extensive to be specified by a few command line parameters. `PolyPack` and `PolyCmd` read the specification of the problem from a description file with the extension `".PP"`. The format of this file is explained in detail in appendix B. It contains the geometry of the molecules involved, as well as the parameters of the RIS model. `PolyPack` is able to pack multiple molecules with identical covalent structures. The problem description file therefore contains the specification of just one molecule. Such a description is composed of a list of atoms, their radii, the covalent structure (the molecular graph defined by the covalent bonds), the decomposition of this graph into rigid groups of atoms and rotatable torsion angles, and the 3-dimensional shape of every rigid group given by a set of coordinates. In addition, the list of torsion angles contains the corresponding RIS-states and a pointer into the list of RIS matrices, also contained in the problem description file. Given an appropriate `.PP`-file, `PolyPack` computes coordinates for all the atoms, comprised in the system, and writes them to an output file.

4.2.1 `PolyPack` and Existing Biochemical Software

The problem description file of `PolyPack` is unsuitable to be created or edited by hand. Since there are user friendly tools to construct molecules on the screen, it is much more convenient to use such a tool and convert its files to the `.PP`-format. For this purpose, we developed the conversion tool `mdf2pp`. One of the most commonly used software used in polymer science is MSI/Biosym's `Insight`. This application offers commands to construct various polymers. The monomer units can be selected from a large library. A molecule is stored in two files. Molecule description files with the extension `.MDF` basically contain a list of atoms with their specific attributes and a list of bonds, defining the covalent structure of the molecule. The coordinates of the atoms, declared in the molecule description file, are stored in a separate file with the extension `.CAR`. Our conversion tool reads the molecule description file and a corresponding coordinates file. The appropriate RIS model can be chosen by a command line parameter from a list of available models which are coded in `mdf2pp`. These three components suffice to generate a complete problem description file.

Figure 4.3 illustrates how `PolyPack` interacts with Biosyms biochemical software. The automatically generated problem description file is read by `PolyPack` or `PolyCmd`. These packing programs store their results as coordi-

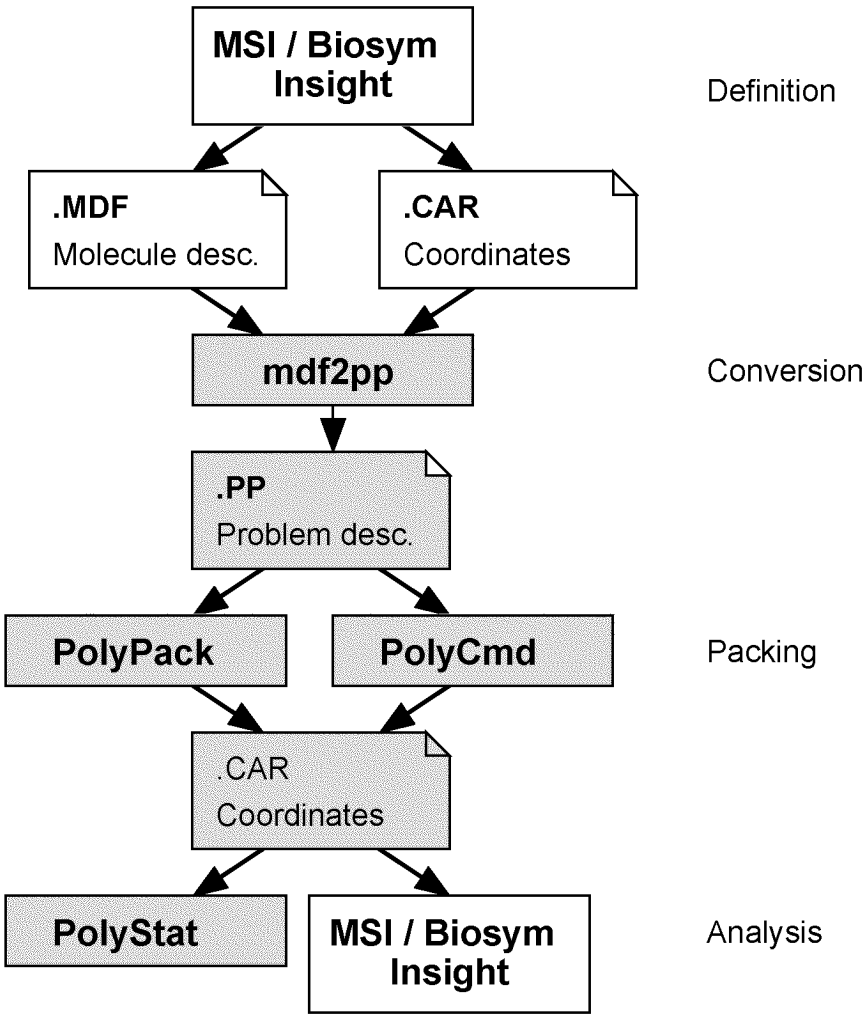

Figure 4.3: `PolyPack` and existing biochemical software

nate files in `.CAR`-format, which then can be either analyzed by `PolyStat` or further processed by `Insight`.

4.2.2 Usage of PolyPack and PolyCmd

The interactive program `PolyPack` is started without any command line parameters. Problem description files and coordinate files are loaded and saved via menu commands. All additional parameters about the system and packing process can be tuned interactively as well. In contrast, `PolyCmd` reads the name of the problem description file and other parameters from the command line. It is invoked by the following command:

```
polycmd <problem>.pp
```

A variety of additional parameters, explained in table 4.2, must be specified in order to completely define the packing problem.

Parameter	Meaning	Default
`-a nrAngles`	Number of discrete positions, a torsion angle is divided into	20
`-c horizon`	A checkpoint file is read. The packing process continues with the corresponding horizon value.	none
`-d density`	The density must be specified in g/cm^3. It determines the size of the periodic box using the total weight of all atoms.	1.00
`-g goal`	The desired maximal overlap (0.0 ... 1.0).	0.10
`-i increment`	The increment by which the horizon is increased	1
`-m minRange`	The minimal number of torsion angles that must lie between two atoms of the same molecule such that their overlap is considered.	1
`-p period`	The size of the periodic box in Angstroms.	10
`-s nrStructures`	The number of molecules contained in the system.	1
`-t nrShakes`	The number of times `Shake` is called before the horizon is increased.	2

Table 4.2: Command line parameters of `PolyCmd`.

Most of the parameters of `PolyCmd` are explained in section 3.7, where the algorithm `PolyPack` is illustrated. However, there is a mechanism concerning its implementation on real machines. Because the packing process can take hours or even days, it may happen that the program needs to be stopped during computation. Therefore, a possibility of checkpointing is of great importance. Checkpointing means saving the state of a computation in external memory so that the computation can be interrupted and resumed. Every time, the packing at a certain horizon is completed, `PolyCmd` writes its actual coordinates to a file called `checkpoint.x.car` where `x` represents the actual horizon. This way, the computation can be restarted at the saved stage. If `PolyCmd` is executed with the parameter `-c horizon`, it reads the file `checkpoint.<horizon>.car` and continues where it was stopped before.

4.2.3 Implementation Details

The architecture of both `PolyPack` and `PolyCmd` is shown in figure 4.4. The two applications have all non-visual software layers in common. These layers and thus, `PolyCmd` as a whole, can be compiled on a large number of Unix platforms because no additional packages other than standard C-libraries are needed. However, to compile `PolyPack`, X-Window and `Motif`-libraries must be available.

Data structures

The information needed to describe the molecular system can be split into a static and a dynamic part. On the one hand, the description of the problem, containing the covalent structure of the molecules as well as the information about rigid groups and rotatable torsion angles, remains unchanged during the embedding process. This static data needs to be stored only once as unique template, since we are dealing with a set of identical molecules.

On the other hand, the dynamic information is specific to the conformation of the molecules, and, thus, must be stored for each molecule individually even for identical molecules. These data are composed of the values of the torsion angles supplemented with the location and orientation of the molecule as a whole. The Cartesian coordinates of the atoms can be gained from both the static and the dynamic data.

Detecting Atom Collisions

Having to consider atom pair interactions usually leads to calculations where most of the computational effort is spent in detecting atom collisions. It is,

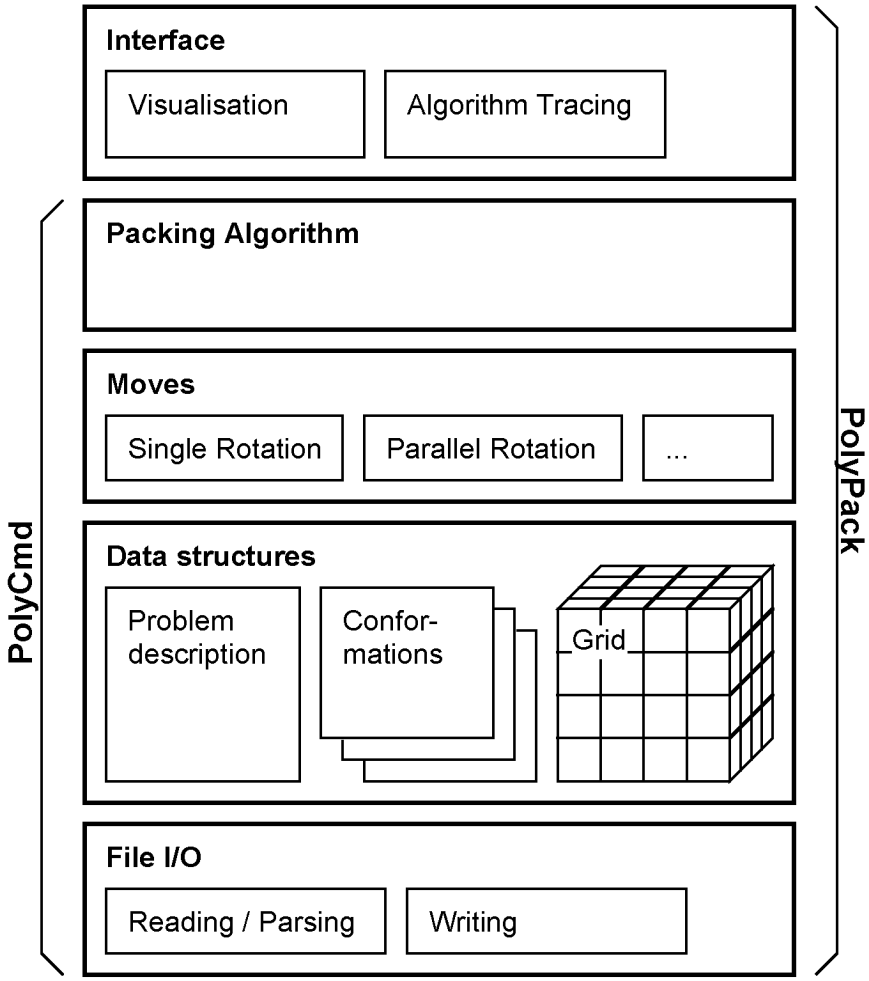

Figure 4.4: `PolyPack` architecture.

thus, important to optimize this procedure. The effort of collision detection can be drastically reduced by introducing a grid which divides the simulation box into cubic cells of edge length w. The grid spacing is chosen in such a way that potentially colliding atom pairs can only be found in either the same or two neighbor cells. Therefore w must be greater or equal to twice the maximal hard sphere radius r_{max}. This choice ensures that collisions among atoms that are more that one cell apart are impossible.

Since w is independent of the problem size, the volume of these cells and the average number of atoms in one cell are constant. To find atoms colliding with a specified one, 27 cells need to be scrutinized. Thus, the time to determine the overall maximal collision after a move scales linearly with

the number of moved atoms.

A further improvement of the grid algorithm can be exploited. As described in section 3.7.2, at each stage of the search, a new embedding is chosen accordingly to its associated RIS probability among a set of embeddings in which the maximum collision value is below the current collision limit c_{goal}. The collision of two hard spheres of radii r_1 and r_2, respectively, is defined as $(r_1 + r_2 - d)/(r_1 + r_2)$ where d is the distance between the two atoms. Therefore the upper limit d_{max} of d of two atoms generating a collision value larger than c_{goal} is given by

$$d_{max} = 2(r_1 + r_2)(1 - c_{goal}) \tag{4.4}$$

As collisions below c_{goal} need not be detected, a grid of cells of width $w = d_{max}$ suffices for a given c_{goal}. In our algorithm, the grid is dynamically updated to adapt to the current value c_{goal}. The number of atoms in one cell decreases as $1/c_{goal}^3$, which accordingly speeds up the collision detection. Such an adaptable grid greatly improves the efficiency of the algorithm.

Software Layers

As we have seen so far, the data structure layer of `PolyPack` and `PolyCmd` consists of three main entities: A static problem description object, a set of dynamically changing conformation objects, specifying the 3-dimensional shape of every individual molecule, and a cubic grid, used to detect atom collisions efficiently. The grid object is used only temporarily during the packing process. In contrast, there must be a way to save and reload the problem description as well as the set of conformations. The input-output layer therefore contains procedures to read and write files in `.PP` and `.CAR` format (see section 4.2.1).

The basic operations to manipulate the system, stored in the data structure layer, are five moves described in section 3.7.5. Namely, there is the rotation about a single bond (SingleRot), the concerted parallel rotation (ParRot), the rigid group rotation and two moves to change the molecules position and orientation. The packing algorithm modifies the molecular system only via these five moves.

A special interface layer, enclosed only in `PolyPack`, enables visualization and step by step tracing of the packing algorithm. Interactivity and tracing is helpful to analyze the effects of parameters on the packing process.

4.3 The Conversion Tool MDF2PP

The first stage of any molecular simulation consists in creating a template for the molecular structure to be simulated. In our case, as only systems of homogeneous material are considered, a unique template for one of the molecules suffices.

We assume this structure template to comply to the MSI/Biosym file formats. A detailed description of the file formats is given in appendix B. Our conversion tool `mdf2pp` parses a `.MDF`- and a corresponding `.CAR`-file and generates the tree representation `.PP` needed by the `PolyPack` software bundle. It first reads the adjacency list contained in the `.MDF`-file and constructs a weighted graph of atoms and bonds. The weights, also contained in the molecule description file, determine, whether a bond is single (no attribute) double (2.0) etc, or part of a ring structure (1.5).

In a first step, a graph traversal procedure detects all cycles and marks all bonds, belonging to one or more cyclic subgraphs. Such bonds are not freely rotatable and, thus, considered to be rigid in our model. Additionally, bonds with weights, other than 1.0 as well as bonds which connect only a single atom to the molecule, are excluded from the list of free bonds.

The obtained set of free bonds defines a unique decomposition of the molecule into rigid groups. In a second step, the corresponding `.CAR`-file is used to define the internal, three dimensional shape of each identified rigid group. It contains coordinates for every atom, declared in the `.MDF`-file. The coordinates file may represent any folding of the molecule, as long as its internal geometry, that is, bond lengths, bond angles and chiralities correspond to the model of interest.

Two important components can be extracted neither from the molecule description file nor from the coordinates file, namely the hard sphere radii and the RIS states for each torsion angle together with their probabilities and pairwise correlations matrices. Hard sphere radii of atoms, appearing in polymers are hardwired in `mdf2pp` as constants. The conversion tool also contains a library of RIS models which are selected by a command line parameter:

```
mdf2pp <filename> (-b|-e|-p|-s1|-s2|-c|-m|-n|-f <model>.RIS)
```

The first parameter specifies the filename of the template, which must exist as the two files `<filename>.MDF` and `<filename>.CAR`. Seven RIS models are included, namely those for poly**b**ead, poly**e**thylene, poly**p**ropylene, poly**s**tyrene in two versions, poly**c**arbonate, polyM**m**a or **n**one. If the user

wants to add a new type of polymer he chooses the `-f` option. The corresponding RIS-model must be specified in a description file called `<model>.RIS`. The syntax and semantics of this file are described in appendix B.

The generated `.PP`-file is flexible enough to specify a general geometric packing problem as defined in section 3.2.

4.4 Generating Statistics with PolyStat

To analyze coordinate files generated by `PolyPack`, a tool called `PolyStat` is provided. It computes overlap statistics and torsion angle distributions which can be compared to the target data. `PolyStat` is invoked by the following command:

```
polystat <filename.pp> <filename.car> [-p period] [-m minRange]
```

Besides the specification of the `.PP`-file and its corresponding coordinates file, two optional parameters may be added. The first parameter is used to override the box size defined in the coordinates file. The second parameter defines the minimal number of torsion angles that must lie between two atoms of the same molecule such that their overlap is considered. `PolyStat` writes the following entities to standard output:

- Parameters of the system such as the size of the periodic box, the number of rigid groups and torsion angles, the maximum horizon value etc.
- The Euler representation of the chains conformations: For each chain, this representation consists of three coordinates specifying the location of the first atom, three angles defining the orientation of the entire chain and a set of torsion angles (see section 3.1.1).
- Average distances: For all integers d less than the number of atoms per chain, the average distance between atoms $a_i^{(j)}$ and $a_{i+d}^{(j)}$ for all chains j is plotted.
- Distance distribution: For all real $r = i\Delta r$, the number n of atom pairs who's distance is within $i\Delta r \ldots (i+1)\Delta r$ is plotted. The value n is scaled by the factor $v/(\pi \Delta r N(N-1)r^2)$, where v is the volume of the periodic box and N the total number of particles of the system.
- The average square radius, which is the average over the squares of all distances between atom pairs, belonging to the same chain.

- Atom collisions, namely the maximum collision (defined in section 3.7.1) as well as a histogram of all collisions in the system.
- Angle distribution: A history of all occurring torsion angle values.
- Correlations: The actual and the postulated pairwise torsion angle distribution matrices.

Chapter 5

Case Studies

5.1 Selection of Examples

The first part of this chapter is devoted to the packing of various simplified polymer systems. Simplified polymer chains have a plain and fixed covalent structure. However, there are diverse attributes of the system itself which are variable, such as the system size, the density, the lengths of the individual chains and their stiffness. All those parameters have notable impacts on the difficulty of the corresponding geometric packing problem. The behavior and time complexity of `PolyGrow` are, thus, discussed with respect to these system attributes.

The subject of the second part of this chapter are systems of atomistically detailed polymers. We discuss three polymer structures among a vast number of existing materials: *Polyethylene*, *polycarbonate* and *polystyrene*. Polyethylene represents simple structures with no side chains. Polycarbonate is a very famous structure for which a lot of theoretical and experimental data is available. Finally polystyrene has a rather complex covalent structure composed of flexible side chains who make the question of tacticity and chirality an important issue. For the latter two polymers, all known 3-dimensional conformations deviate from the experimentally measured torsion angle distributions. Therefore, they constitute a great challenge for out packing algorithm `PolyPack`. The characteristics of the three polymer structures we chose cover a wide spectrum of materials which are investigated today.

5.2 Simplified Polymers

The characteristics of a system of simplified polymer chains are determined by a set of parameters: The total number of particles (atoms), the lengths of the chains, the density which determines the size of the periodic box, the diameter of the particles, the bond length and the size of the bond angles or a bond angle distribution function (see section 3.1.4). We investigated the behavior and the time used by `Embed` and `PolyGrow` in function of these parameters.

5.2.1 Analysis of `Embed`

A system of M simplified polymer chains with fixed bond angles, containing a total of N particles leads to a packing problem with N points, N^2 lower bounds on distances and $2N - 3M$ exact distance constraints. The lower bounds ensure, that the points do not get closer than twice the atoms radius. The $N - M$ bonds and the $N - 2M$ bond angles are kept fixed by as many exact distance constraints.

As starting conformations (the input of `Embed`), we generated random chains with the proper bond lengths and bond angles but without considering atom overlaps.

Figures 5.1 and 5.2 show the performance of `Embed` versus the size and the density of the systems respectively. For our test runs, we chose the following parameters: Atom diameter 0.9, bond length 0.9, bond angle 60^o and number of chains 10. All spatial dimensions are specified without unit. They are interpreted relatively to the system of density 1.0 with one sphere of diameter 1.0 per cube unit. The results shown in the diagrams are average values of 20 runs at each case.

The left diagram in figure 5.1 shows the number of iterations and the time needed to compute one iteration step in function of the number of particles in the system. The density is kept constant at 0.9 particles per cube unit. The time spent in each iteration step grows linearly because the time needed to rearrange a singe particle is independent of the problem size. The introduction of a grid (see section 4.2.3) reduces the number of lower distance bounds which have to be considered for a singe particle from N to a constant independent of N. The fact that the number of iterations needed to solve the embedding problem grows linearly with the system size can not be explained so easily. It is caused by the characteristics of the particular class of packing problems we investigated, namely systems of simplified polymer systems. The multiplication of the two linear curves leads to a quadratic time complexity of `Embed`, shown in the diagram on the right.

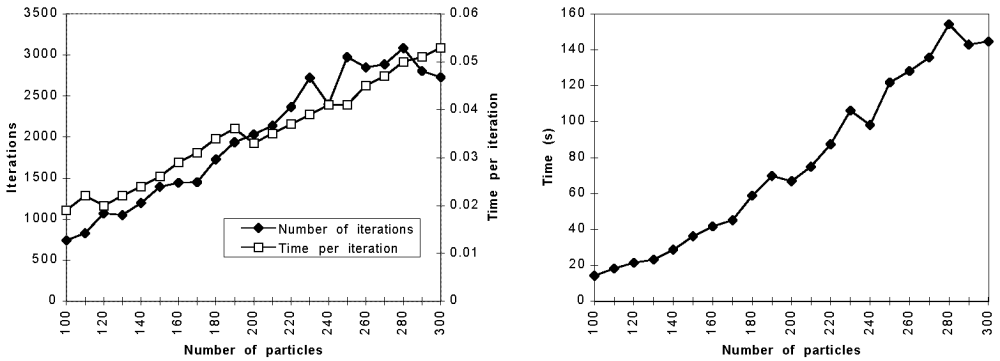

Figure 5.1: Computing time of `Embed` versus the size of the problem (MIPS R5000 at 180 MHz).

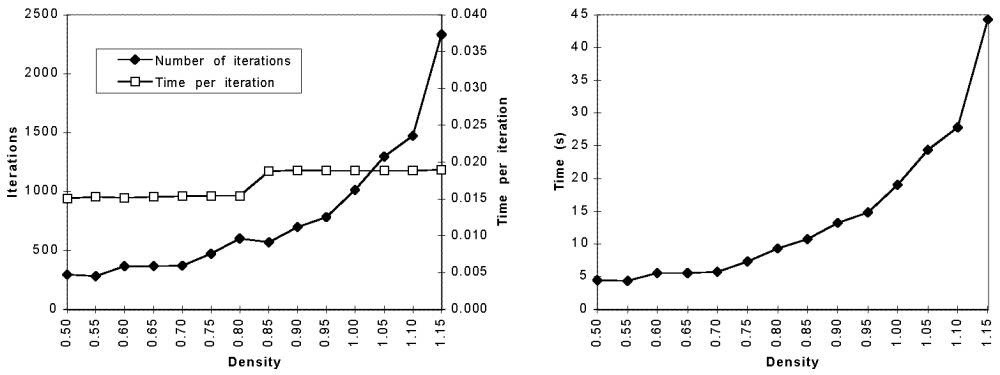

Figure 5.2: Computing time of `Embed` versus the system density. (MIPS R5000 at 180 MHz).

Figure 5.2 shows the same properties as figure 5.1 but in function of the density of the system while the number of particles is constant (100) at each case. The time-per-iteration-curve jumps at density 0.80 because the size of the grid cells has to be adjusted at this point. The total computing time shows a non linear behavior. The packing problem is easy to solve up to a density of about 0.95. Beyond this point, the difficulty of the problem increases rapidly. Above 1.30, `Embed` is unable to solve the packing problem.

If we compare these results (figure 5.1) to the ones obtained with `PolyGrow` (figure 5.3) we see that `Embed` is efficient only for small systems containing up to about 100 particles. However, as part of `PolyGrow`, `Embed` is the perfect alternative to the complex computation of a set of concerted rotations. There, only local problems need to be solved where most of the geometric constraints are already satisfied.

5.2.2 Analysis of `PolyGrow`

To analyze the time complexity of `PolyGrow` we choose a base system composed of 1000 particles of diameter 0.90, chains of length 100, system density 0.90 and a bond angle distribution parameter κ of 4.0. These parameters are adopted from [17]. By varying a single parameter, the dependence of the execution time in function of this particular parameter can be evaluated.

Linear dependence on system size

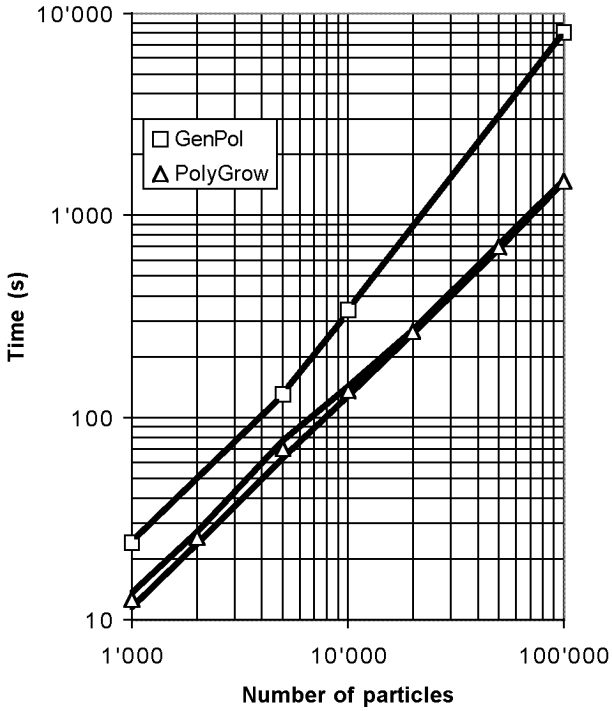

Figure 5.3: Computing time of `PolyGrow` (MIPS R5000 at 180 MHz) compared to results obtained with the program `GenPol` (MIPS R10000 at 195 MHz).

Figure 5.3 shows the time used by `PolyGrow` to pack system of various sizes. The time increases linearly with the system size. We compare our results with the performance of `GenPol` [17], the fastest method to generate simplified amorphous polymer systems at present. On the bi-logarithmic scale polynomials become straight lines. The gradient corresponds to the degree of the polynomial. `PolyGrow` shows a linear behavior while the time used by `GenPol` increases slightly steeper.

The white squares show the time used by `GenPol`, the black triangles represent average values over 5 runs of `PolyGrow`. The two lines above and

below the triangles show the standard deviation. `PolyGrow` is between two and five times faster than `GenPol` depending on the system size.

Non linear dependence on the systems density

The dependence of the execution time on the systems density is not linear. Above a certain value the problem even becomes unsolvable because there is no way to pack the chains without overlapping spheres. Up to the conventional density of 0.9 particles per cube unit, the execution time does not increase dramatically as figure 5.4 shows.

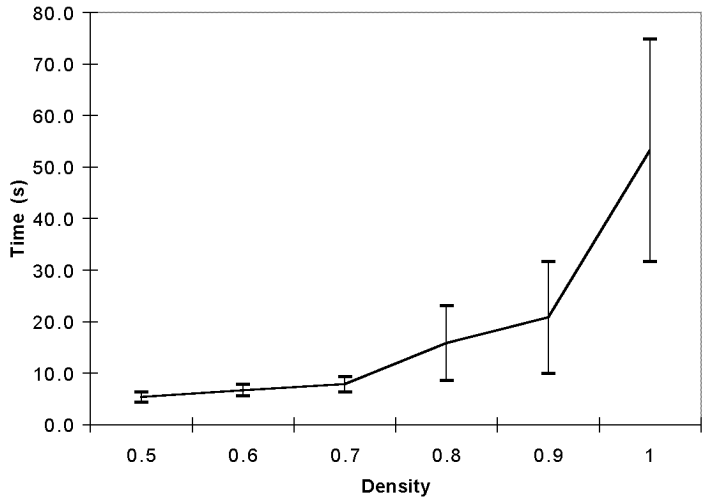

Figure 5.4: Computing time of `PolyGrow` versus the system density of the system to be packed (Ultra-Sparc-II at 248 MHz).

However, `PolyGrow` is unsuitable for densities above 1.0. For very high densities, `GenPol` is more suitable. It is able to pack systems up to 1.5 particles per cube unit.

Influence of the stiffness of the chains

The stiffness of the simplified polymer chains is determined by the parameter κ which defines the bond angle distribution (see section 4.1).

The time needed to pack systems of different bond angle distributions is shown in figure 5.5. The graph shows the average time as well as the standard deviation over 20 runs. The problem becomes more difficult with increasing κ. `PolyGrow` is also able to pack polymer chains with fixed angles. The last bar in figure 5.5 shows the time used to pack a system with a typical bond angle of 60^o. The only degrees of freedom of such a system are the torsion angles. This makes the packing problem more difficult for `PolyGrow`.

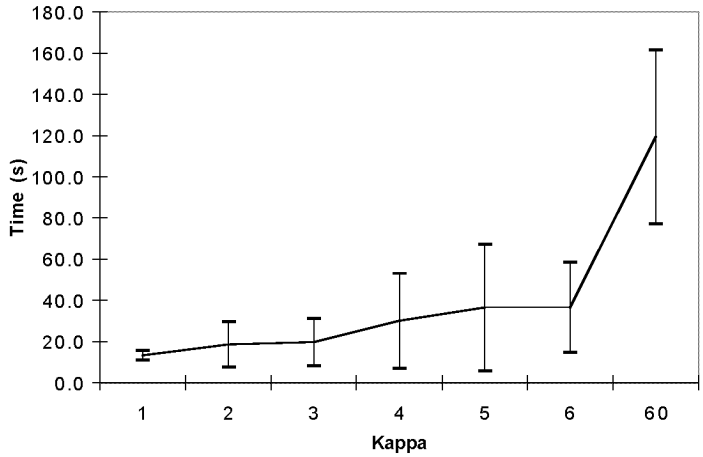

Figure 5.5: Computing time of `PolyGrow` versus the parameter κ of the bond angle distribution (Ultra-Sparc-II at 248 MHz).

Independence of the lengths of the chains

The final parameter to be analyzed is the chain length. As figure 5.6 shows, the execution time of `PolyGrow` is independent of the lengths of the chains.

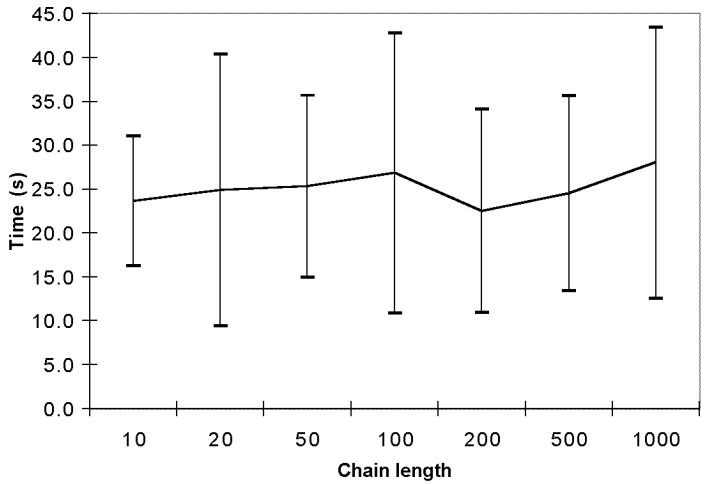

Figure 5.6: Computing time of `PolyGrow` versus the length of the chains. (Ultra-Sparc-II at 248 MHz)

The standard deviations over 20 runs is much higher than the differences in the average values. The task to finish a chain and to begin an new one during the chain growing process does neither take too much time, nor does this additional degree of freedom simplify the problem significantly.

5.3 Atomistically Detailed Systems

5.3.1 Polyethylene

In order to test the `PolyPack`-algorithm, we applied it to three widely used polymers structures. The simplest structure we investigated is polyethylene. Its covalent structure is shown in figure 5.7. Polyethylene is composed of a simple chain of carbon atoms completed by the necessary hydrogen atoms. We used this molecule to analyze various properties of `PolyPack`. As a typical density of a polyethylene glass we chose $0.892 g/cm^3$ for all test runs.

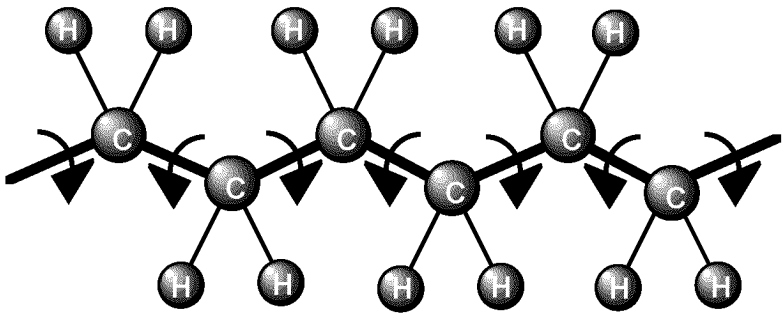

Figure 5.7: Covalent structure of polyethylene.

First, we measured the scaling behavior of the computing time with respect to the number of chains and the length of these chains. Secondly, we tested the effects of the horizon technique, the parallel rotation and the shake technique by packing the same system first with and then without the specific method. Thirdly, we ascertained the effectiveness of different moves at different stages of the embedding process. The more complex polymers polycarbonate, and polystyrene are discussed separately at the end of this section.

5.3.2 Measured Time Complexity

We applied `PolyPack` to systems of various size in order to study its scaling behavior with respect to the number of chains and the length of these chains.

On a bi-logarithmic scale, a polynomial time complexity is represented by a straight line. Its gradient equals the order of the asymptotic growth. Figure 5.8 shows the growth of computing time in function of the number of chains. The computational effort versus the length of the chains is shown in figure 5.9. The time is measured in seconds on an Ultra-Sparc-II at 248 MHz. For large systems, the points lie almost perfectly on a straight line.

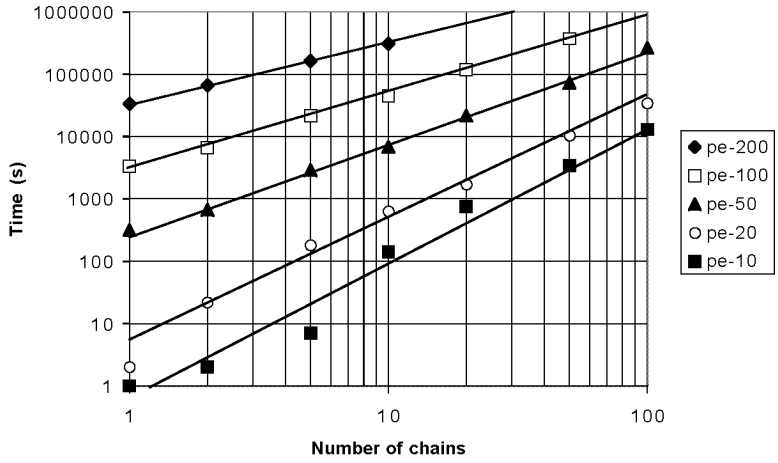

Figure 5.8: Computing time of `PolyPack` versus number of chains.

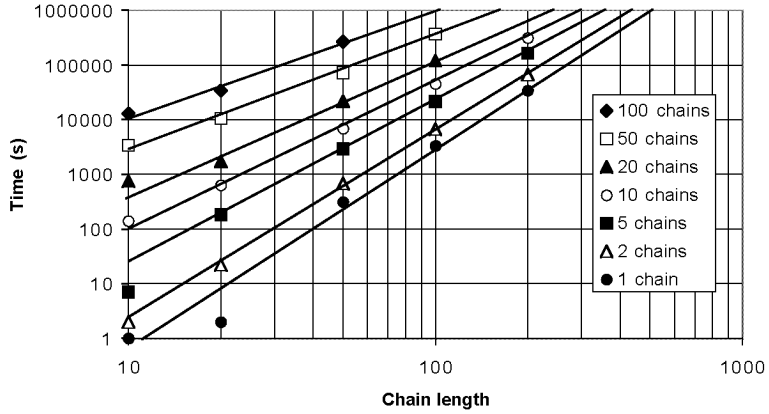

Figure 5.9: Computing time of `PolyPack` versus chain length.

The relative computing time t is thus polynomial both in the number of molecules M and the length of the molecules T:

$$t \, \alpha \, M^{\mu} \cdot T^{\nu} \tag{5.1}$$

with $\mu = 1.5 \pm 0.2$ and $\nu = 2.8 \pm 0.2$. In section 3.7.8 we derived a time complexity of the order $O(i \cdot M \cdot T^3)$. The influence of the unknown value i for these specific problem instances can now be estimated by comparing the two results.

The efficiency of `PolyPack` seems to be low when compared to the one of `PolyGrow` shown in figure 5.3. However, the chain length of polyethylene is measured in monomers instead of particles (atoms). A polyethylene-100-chain is composed of 302 atoms and 100 torsion angles. `PolyPack` is designed to handle complex atomistically detailed molecules. Since it operates in

torsion angle space, the computing time depends mainly on the number of torsion angles rather than on the number of particles. Therefore, `PolyPack` is more efficient for complex structures with many atoms per degree of freedom. Additionally torsion angles on side chains, not belonging to the backbone, are modified very efficiently by `PolyPack`. Thus, polycarbonate and polystyrene are better examples than simple polyethylene to demonstrate the efficiency of `PolyPack`.

5.3.3 Effects of Packing Techniques

In section 3.7, we described a variety of methods to improve the packing of macromolecules. Since most of these methods have a heuristic character, it is difficult to predict their effect on the convergence of the algorithm theoretically. We measured the effects of the most important techniques (ParRot, Shake and the introduction of a horizon parameter) and compared their performances on a given system of 10 polyethylene chains of 50 repeat units each. The system contains 1520 atoms and $M \cdot T = 500$ torsion angles.

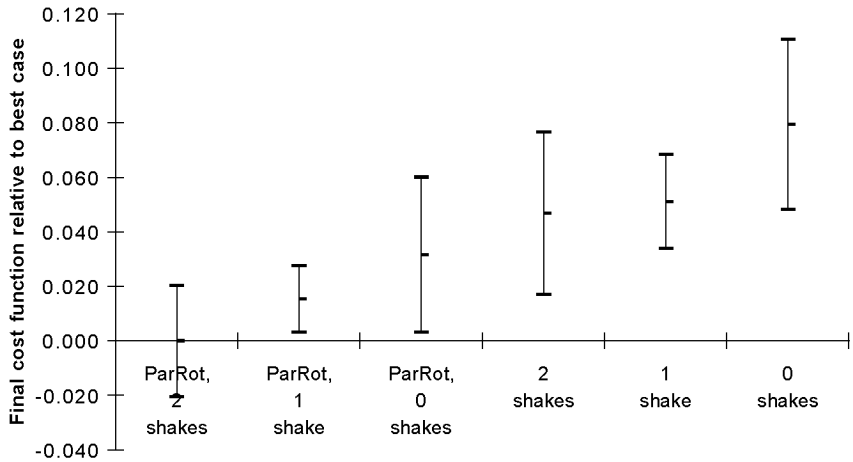

Figure 5.10: Effects of ParRot and Shake.

To measure the effectiveness of ParRot and Shake, we started `PolyPack` with and without using the ParRot move and with zero, one, or two shakes at each horizon value. For all six variants, we packed the system of polyethylene chains 20 times, each time with a different starting conformation. Each run produced a packing with a final maximum overlap value. The average overlaps as well as the standard deviations over the 20 runs are shown in figure 5.10. The omission of the ParRot move as well as the shake method each increases the final overlap by about one standard deviation, indicating that both techniques improve the performance of `PolyPack` significantly.

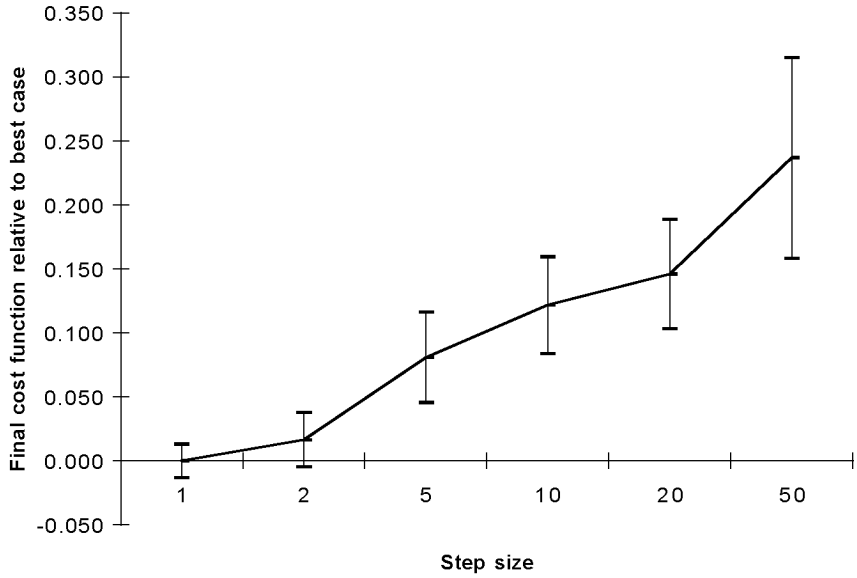

Figure 5.11: Influence of increment step size of the horizon parameter.

In section 3.7.4 the parameter h_{step} was introduced. It defines the amount by which the horizon parameter h is increased during the outermost loop of the packing algorithm. Since the chains of our test system consist of 50 repeat units and 50 torsion angles on the backbone, the maximum value for the horizon parameter h_{max} is 50. If h_{step} is set to 1, the system is optimized for all horizon values from 0 up to 50. This strategy is very time consuming, but it produces the most accurate results. On the other hand, if h_{step} is set to 50, the concept of a horizon parameter is eliminated since the system is optimized only once, considering all atom collisions from the beginning. h_{step} is used to control the trade off between the computing time and the quality of the generated structure. Figure 5.11 presents the average over collision values of 10 runs, as well as their standard deviation for several step sizes ranging from 1 to 50.

The benefit of the horizon method is shown even more impressively in figure 5.12. We packed a system of 10 polyethylene chains of 20 monomers without increasing horizon and with a horizon step size of 1. In the first case, when all collisions are taken into account from the start, the packing process gets stuck at more than 50% maximum overlap. In the latter case, a conformation with only 12% maximum overlap ist found. The peaks on the collision curve are caused each time the horizon is increased and new collisions are considered.

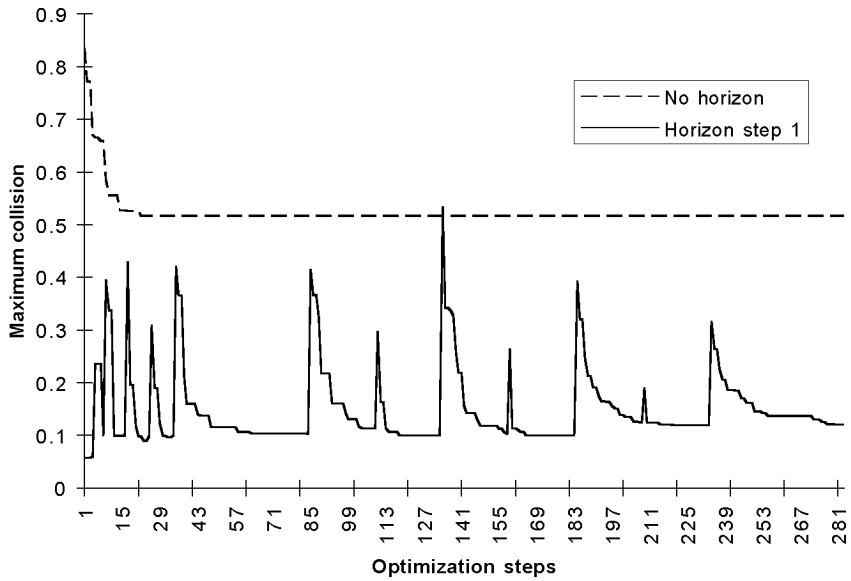

Figure 5.12: Devolution of the maximum collision with and without the horizon method.

5.3.4 Effects of Moves

Our algorithm makes use of five different moves (section 3.7.5). We measured the effectiveness of a particular move as the ratio of the number of times it succeeded in improving the system (reducing the maximum collision), relative to the total number of times it was employed during the packing process. Figure 5.13 shows the success ratio for all moves described in function of the value of the horizon parameter. As expected [33], the ParRot move outperforms the other moves, especially at the late stages of the packing process when most of the collisions are considered.

5.3.5 Accuracy of Results

The accuracy of a conformation is measured by two criteria, the maximum overlap of spheres and the agreement of the distribution of the torsion angles with the distributions given by the RIS model.

According to the standard RIS model of polyethylene [22], torsion angles on the backbone are limited to the three RIS states *trans*, $gauche^+$ and $gauche^-$. Thus, nine states, a pair of two adjacent angles can occupy. The RIS a priori probabilities for these states to occur are shown in figure 5.14 as white bars.

We generated 20 different conformations of a system of 10 polyethylene chains composed of 50 building blocks each at $0.892 g/cm^3$. The mean value

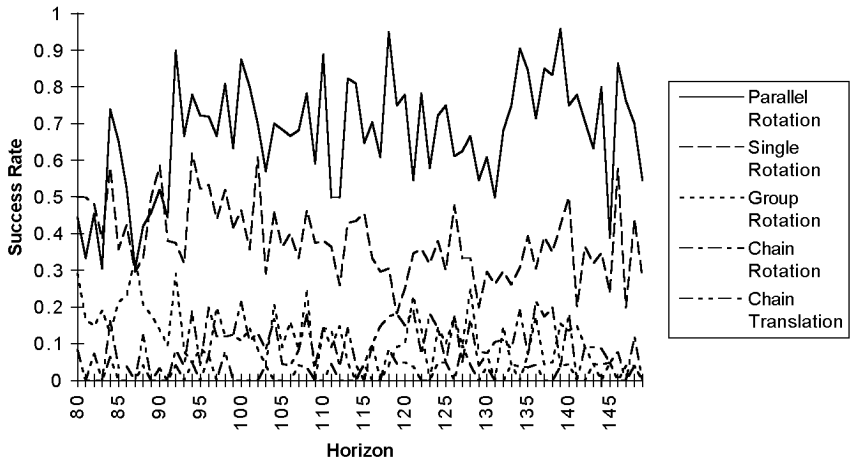

Figure 5.13: Success ratio of all five moves used in `PolyPack`.

over the maximum overlaps of these 20 structures was 0.216, which means a maximal encountered atom overlap of 21.6% of their hard core radii on average. The average distribution over all pairs of torsion angles of all computed structures is shown in figure 5.14 as black bars. The small horizontal bar indicates the mean value while the upper and lower bars represent the standard deviation. The figure shows that `PolyPack` is able to reproduce a given distribution accurately even at hight densities.

Another way to assess the end structure after packing is to minimize the total system energy (with flexible bond lengths, bond angles, and continuous torsion angles) using an appropriate force field, and then to compare the three-dimensional shape before and after the minimization. A small difference between the two structures shows that the geometric constraints imposed in the packing algorithm form an appropriate model for the bonded and non-bonded forces. The conformation of a system of five polyethylene chains of 200 repeat units before and after energy minimization is shown in figure 5.15 (parent chains only). The local remaining overlaps between atom pairs are removed during the minimization procedure without significantly modifying the chain configuration. This is to say that, for instance, the end-to-end distance or gyration radius, which are strongly dependent on the RIS model, are free of the perturbations suffered in the usual packing approaches. The new algorithm can faithfully reproduce a desired dihedral angle distribution, provided by RIS models or possibly experimental results.

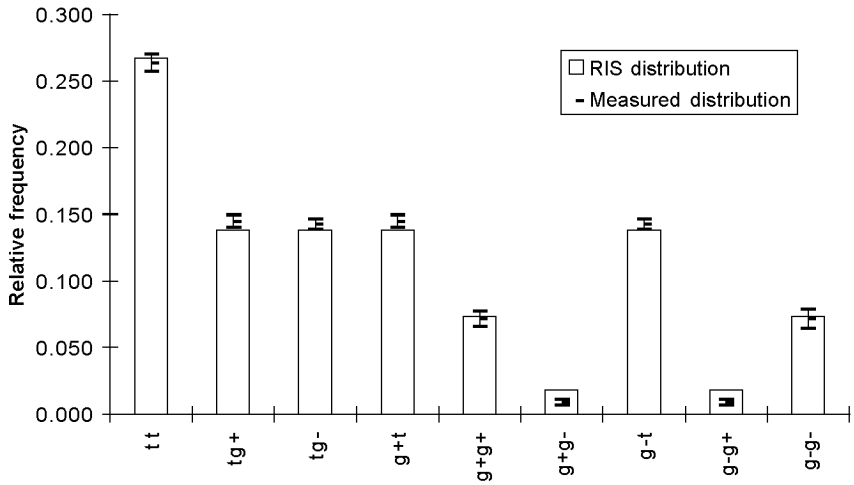

Figure 5.14: RIS distribution and measured distribution of polyethylene.

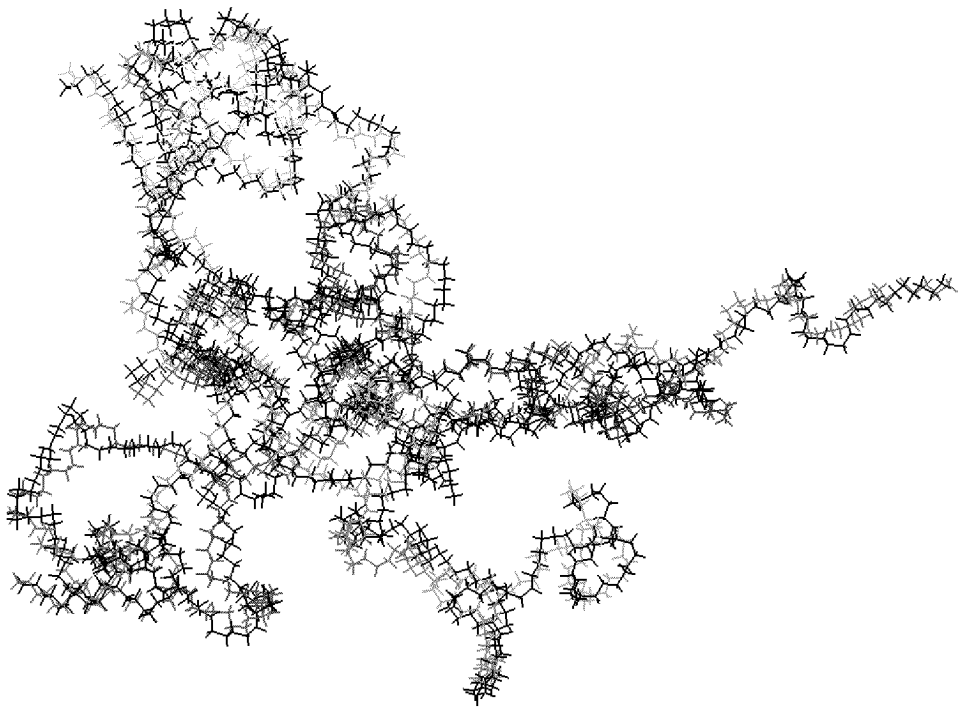

Figure 5.15: Five polyethylene chains before (black) and after (gray) energy minimization.

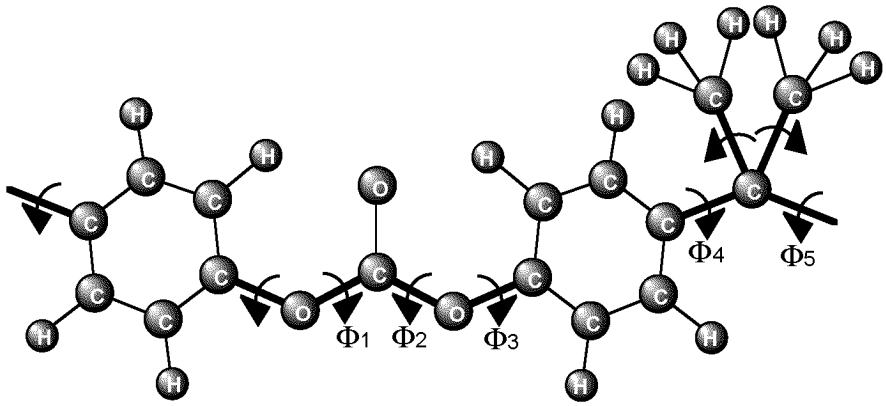

Figure 5.16: Covalent structure of polycarbonate.

5.3.6 Polycarbonate

The covalent structure of polycarbonate is shown in figure 5.16. Nuclear Magnetic Resonance (NMR) experiments show that at most 15% of the torsion angles belonging to the carboxyl group (ϕ_1 and ϕ_2) are in the *cis*-state ($-60^o \leq \phi \leq 60^o$) [39]. Other references [36] state that the quota of angles in *cis* state is even near 10%. In all present atomistic conformations, used in computer simulations, the percentage of torsion angles in *cis* state is above 25% [42]. One of the best results has been achieved with the so called "amorphous cell" method [35]. In [37], this method was applied to a system of polycarbonate chains. 26% of the carboxyl angles contained in the resulting conformation are in *cis* state. The detailed distribution of $\phi_1(\phi_2)$ is shown in diagram (4) of figure 5.17.

We used `PolyPack` to generate a system of seven polycarbonate chains containing 25 monomer units each. The system contains $5,789$ atoms and $1,400$ torsion angles (degrees of freedom). We packed it into a periodic box of size $39.5\mathring{A}$ corresponding to a density of $1.20g/cm^3$. The target torsion angle distributions we used are derived from the standard RIS model of polycarbonate [22, 30]

$$P_{12} = \begin{bmatrix} 0.910 & 0.045 \\ 0.045 & 0.000 \end{bmatrix} P_{23} = \begin{bmatrix} 0.2387 & 0.0113 \\ 0.2387 & 0.0113 \\ 0.2387 & 0.0113 \\ 0.2387 & 0.0113 \end{bmatrix} \tag{5.2}$$

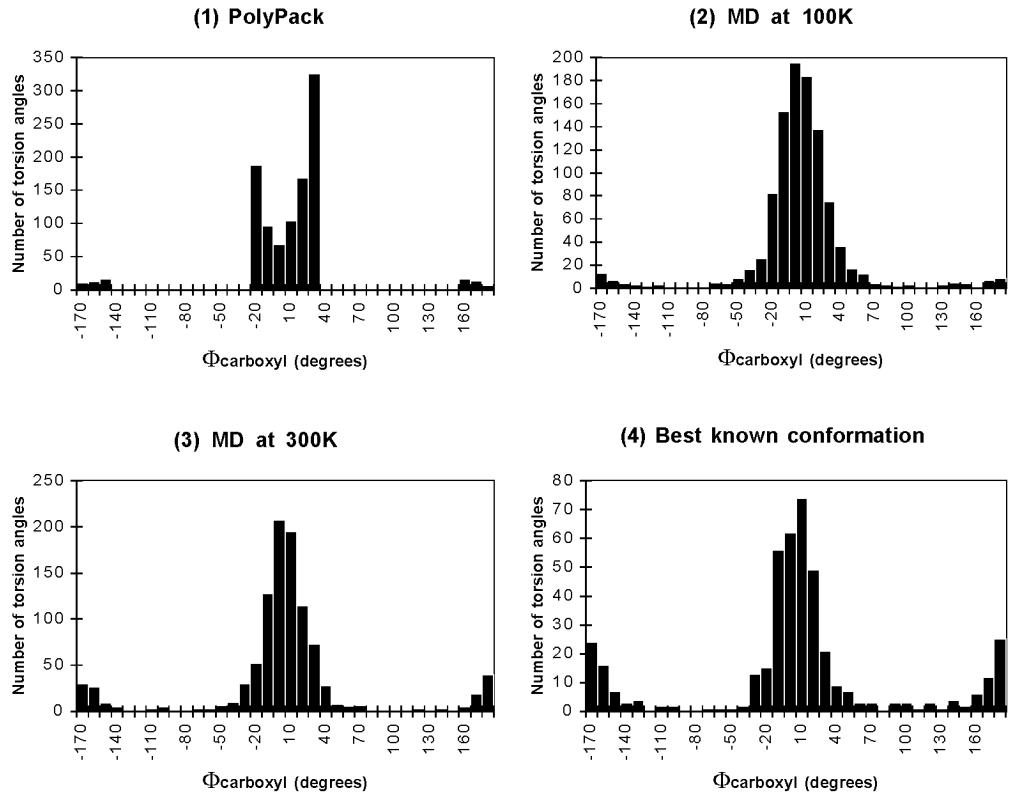

Figure 5.17: Distribution of $\phi_{carboxyl}$ of conformation generated with `PolyPack` compared to best known conformation.

$$P_{34} = \frac{1}{8}\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad P_{45} = \frac{1}{16}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{5.3}$$

where P_{ij} describes the distribution of torsion angle states of ϕ_i and ϕ_j. The packing process took 139 hours on an Ultra-Sparc-II at 248 MHz. `PolyPack` succeeded in generating a conformation with 9% of the carboxyl angles (ϕ_1 and ϕ_2) being in *cis* state. Diagram (1) in figure 5.17 shows the distribution of carboxyl angles of the system generated by `PolyPack`. The angles appear strictly separated in clearly defined state intervals. `PolyPack` does not care about the distribution within these intervals but it cares for the percentage of angles inside each interval. The geometric model, used by `PolyPack` is very strict in two ways. Firstly, it only allows torsion angles to be inside certain intervals and secondly, it keeps all bond lengths and bond angles to be fixed. Such a strict conformation is transformed from the

geometric world back to the chemical world by energy minimization and a molecular dynamics run.

We first applied the pcff91-forcefield (included in Biosym Discover) and minimized the energy using the steepest descent method. We selected a maximum atom displacement of $0.2\mathring{A}$ per step and iterated $10,000$ steps. Continuing, we started a Molecular Dynamics run of 100 pico seconds at a temperature of 300K and 100K. Finally we minimized the energy a second time. The distribution of carboxyl torsion angles after these three steps is show in Diagram (2) and (3) of figure 5.17 respectively.

Surprisingly the percentage of *cis* conformations increased but only to 16% which is within an acceptable range. Our resulting structure is now used in various simulations as the conformation which best corresponds to experimental data.

5.3.7 Polystyrene

As a third example of an atomistically detailed polymer we investigate polystyrene, a molecule with rotatable side groups attached to the main chain. Figure 5.18 shows its covalent structure as well as all rotatable torsion angles.

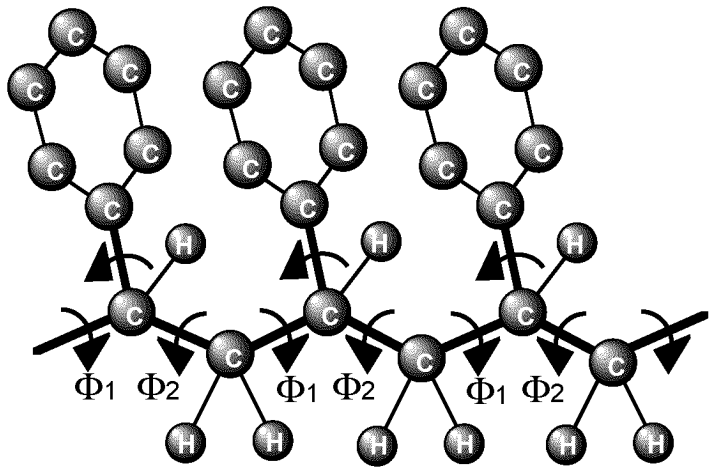

Figure 5.18: Covalent structure of polystyrene.

In contrast to polycarbonate and polyethylene, a monomer unit of polystyrene can assume two distinguishable conformations of different chirality because the side chain can be on the left or on the right side of the adjacent hydrogen atom. Figure 5.19 shows the view along the backbone of the two conformations together with the two states *trans* and *gauche*$^+$. The direction in which the torsion angle is measured depends on the position of the side

chain. A polymer is called *isotactic* if all monomers have the same orientation and *atactic* otherwise. The results we present in this section are obtained with isotactic polystyrene.

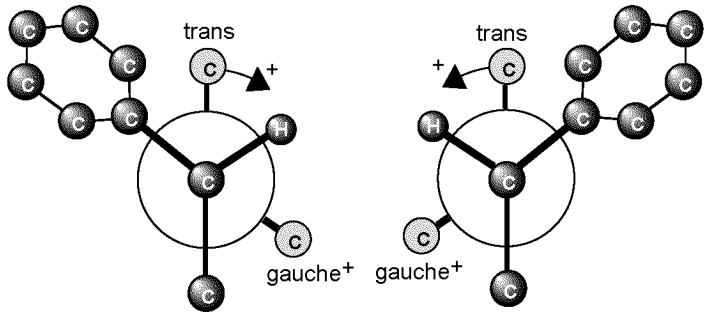

Figure 5.19: Left- and right handed conformations of a monomer of polystyrene.

The relative orientations of the rings in polystyrene can be measured experimentally by replacing single carbon atoms of the rings by carbon-13 atoms. The spins of these labeled atoms are stimulated and the polaristation-transfer between the nuclei is evaluated by Nuclear Magnetic Resonance spectroscopy (NMR) (see [31]). A two dimensional NMR spectrum of amorphous polystyrene is shown in diagram (a) of figure 5.20. The height of a point at (ω_A, ω_B) is given by

$$f(\omega_A, \omega_B) = \frac{N}{n_A n_B} \sum_{i \in A} \sum_{j \in B} \frac{1}{r_{ij}^6} \tag{5.4}$$

where N is the total number of spins, n_A and n_B are the number of spins that belong to group A and group B, and r_{ij} is the distance between spin i and spin j. A spin belongs to group A or group B if its resonance frequency is within $\omega_A \ldots \omega_A + d\omega_A$ or $\omega_B \ldots \omega_B + d\omega_B$ respectively. High values near the diagonal indicate, that proximate rings tend to be parallel to each other.

The NMR spectrum of atomistically detailed models of polystyrene can be computed using formula 5.4. The spectrum calculated from an average over 24 atomistic simulations of polystyrene is shown in Diagram (b) in figure 5.20. The spectra obtained from experimental data and from atomistic simulations differ significantly, because the underlying conformations used in simulations do not correspond the angle distributions given by the RIS model of polystyrene. For example, the values in the upper right corner of diagram (b) are much too high due to an overpopulation of *trans-trans* states along the backbone.

We used `PolyPack` to generate conformations of isotactic polystyrene that correspond to given torsion angle distributions. We used a system composed

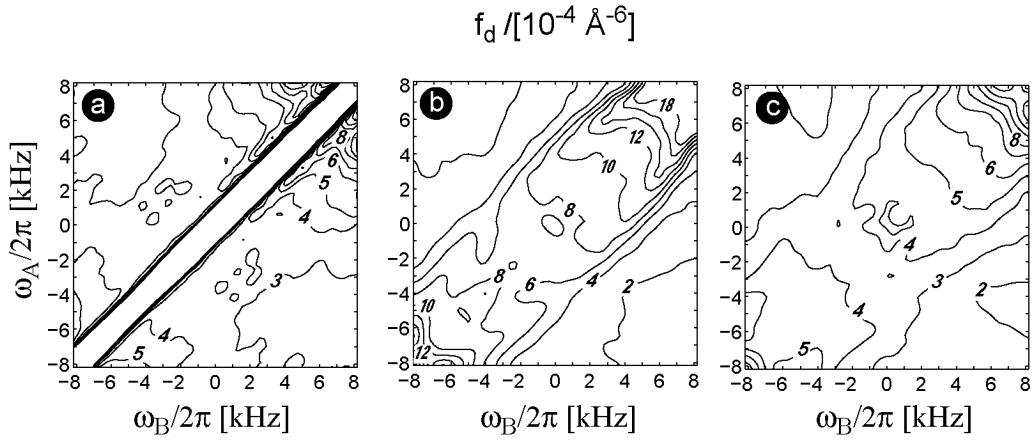

Figure 5.20: NMR measurements of polarization-transfer between carbon-13 nuclei in specifically labeled amorphous polystyrene. Conformations generated with `PolyPack` (c) show higher correspondence to experimental data (a) than those generated with conventional methods (b).

of 9 chains of 40 monomers each at a density of $1.05 g/cm^3$ which corresponds to a box size of 39Å. There are 1080 torsion angles and 5778 atoms comprised in the system. The packing process took 18 hours on an Ultra-Sparc-II at 248 MHz.

$$P_{12} = \begin{bmatrix} 0.000 & 0.435 \\ 0.435 & 0.130 \end{bmatrix} \tag{5.5}$$

$$P_{21} = \begin{bmatrix} 0.050 & 0.410 \\ 0.410 & 0.130 \end{bmatrix} \tag{5.6}$$

P_{12} and P_{21} specify the pairwise distribution of the *trans* (t) and *gauche* (g^+) states of torsion angle pairs attached to the same and to neighboring side chains respectively. The matrices are derived from Rapold and Suter's 2-state RIS model of polystyrene at 300K [22, 30]. We set the probability for *trans-trans* states in P_{12} to zero to make sure that they stay rare even after energy minimization.

As mentioned in the previous section, the moves used by `PolyPack` are more restricted than those in typical force fields. Bond lengths and bond angles are kept fixed and torsion angles are restrained to certain intervals. Thus, `PolyPack` is not able to remove all non-bonded overlaps completely. In subsequent energy minimization and molecular dynamics runs, these overlaps are balanced at the expense of local bond length, bond angle and torsion angle motions.

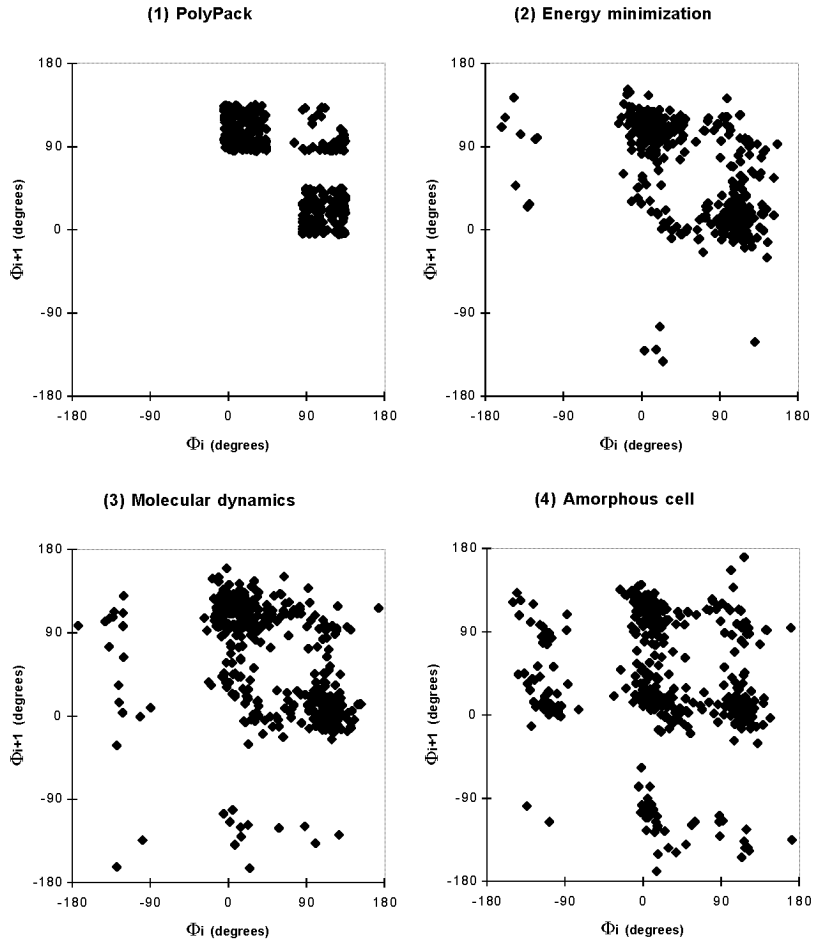

Figure 5.21: Correlation matrices of adjacent torsion angles in glassy amorphous polystyrene.

Diagram (1) of figure 5.21 gives the distribution of torsion angles of the meso dyads generated by `PolyPack` with $\triangle\phi = \pm 25^o$. It corresponds perfectly to the matrix P_{12}. The relaxation of the strict conformation is illustrated in diagram (2) and (3). They show the distribution of torsion angles after energy minimization and after $20ps$ of molecular dynamics simulation at 300K. Diagram (4) exhibits the distributions of a conformation generated with the commonly used amorphous cell method [35]. In contrast to the distribution in diagram (3), there are significantly more points near the center which corresponds to the *trans-trans* state, typical for conformations computed with conventional packing methods.

We succeeded in generating conformations of isotactic polystyrene that

correspond well to the RIS model (less than 10% *trans-trans* states). They maintain this property in molecular dynamics runs. Moreover diagram (c) in figure 5.20 shows, that the NMR spectra derived from our conformations show significantly better agreement with the experimental data than those, generated by conventional methods.

Chapter 6

Conclusions

6.1 Interdisciplinary Collaboration

The work and results we presented in this thesis have their source in a fruitful interdisciplinary collaboration between chemists and computer scientists. Interdisciplinary cooperation causes some difficulties not necessarily present in the teamwork of scientists from the same field. People from different disciplines have different backgrounds and knowledge and they speak different languages. For a chemist, for example, the direction in which a torsion angle is measured inside a polymer with side chains, is not worth mentioning. A computer scientist on the other hand, takes for granted, that all torsion angles are measured the same way, either clockwise or anti-clockwise. Consequently, both parties need to learn to communicate problems precisely without unspoken assumptions.

However, the advantages of interdisciplinary work surmount these drawbacks easily. Computer scientists and chemists for example, regard a problem like polymer packing differently, influenced by their different backgrounds. The pooling of the knowledge opens many new ways of solving such a problem. A computer scientist does not care whether intermediate states generated by the search algorithm are physically possible states or whether the forces that change the conformation during the packing process have a physical explanation. All he cares about is that the resulting conformation meets the requirements stated by the chemist.

6.2 Geometry is an Effective Filter

Apart from implementation details, there is basically only one way to simulate on an atomistic level the behavior of materials over time. The actual

forces and velocities of the particles lead to their new coordinates in the next step of the simulation. In contrast, any way to generate the initial conformation is reasonable, as long as the resulting conformation meets the stated requirements. Thus, polymer packing is an open problem and there is no unique way how to solve it. We approach it as a geometric optimization problem. No significant information gets lost when the chemical world is transformed into a set of geometric and statistical constraints. This transformation is even an effective filter that simplifies the problem by selecting only relevant properties of the initial system.

6.3 Parrot - A Universal Move

The study of the polymer packing problem from a geometric point of view asked for a more efficient move than simple torsional rotation. The parallel rotation technique allows relaxation deep inside long chains in a dense environment. The performance of the search algorithm relies heavily on the efficiency of the ParRot move. Although ParRot was devised especially for the polymer packing algorithm, we found that it has many more applications. All algorithms that need to change the conformation of dense, highly connected systems profit from this kind of move. We showed in [33] that the integration of parallel rotation into existing Monte Carlo algorithms increases their efficiency significantly. It may be the case that even in nature, densely packed polymers move somehow similar to parallel rotation.

6.4 Effective Heuristics

We showed that the combinatorial chain packing problem is NP-complete. Many discrete combinatorial problems turn out to be NP-complete. The NP-completeness property results from worst case analysis. It justifies the use of heuristic techniques which may be effective for a certain class of instances which are of interest. However, for arbitrary instances, no theoretical guarantee for efficiency can be given.

Among the heuristic techniques we integrated into our packing algorithms, two turned out be especially successful. We implemented the concept of a horizon which allows the splitting of the global packing problem into a sequence of problems with increasing difficulty. The idea of solving simpler problems first and using their outputs as input for more difficult ones is applicable in various other optimization problems.

For packing simplified polymer systems, we combined a constructive chain

growing process with an iterative technique to remove local overlaps of atoms. This hybrid algorithm succeeds in generating very large polymer systems. Chain growing is very efficient but it may run into troubles during the packing process. On the other hand, relaxation is much more expensive but it is needed only locally in the environment of the newly placed atom. The combination of constructive and iterative processes turns out to be a suitable way of constructing large, simplified polymer systems.

6.5 Software for Polymer Packing

The goal of this project was not only to devise new methods for packing polymers theoretically, but to offer a user-friendly software package that is compatible with existing biochemical software, that can be used effectively and intensively by material scientists. The real case studies in the previous section show, that our software finally reached this goal. The programs `PolyPack` and `PolyGrow` became important tools used in the process of generating realistic starting structures because without realistic starting structures, there is no realistic simulation!

Bibliography

[1] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids.* Clarendon Press, Oxford, 1986.

[2] W. Brostow and J. Kubat. Molecular-dynamics simulation of stress relaxation on a triangular lattice. *Phys. Rev. B*, 47(13):7659, 1993.

[3] D. Brown, J. H. R. Clarke, M. Okuda, and T. Yamazaki. The preparation of polymer melt samples for computer simulation studies. *J. Chem. Phys.*, 100(9):6011, 1994.

[4] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* The MIT Press, 1990.

[5] G. M. Crippen and T. F. Havel. *Distance Geometry and Molecular Conformation.* Research Studies Press LTD, Taunton, England, 1988.

[6] L. R. Dodd, T. D. Boone, and D. N. Theodorou. A concerted rotation algorithm for atomistic monte carlo simulation of polymer melts and glasses. *Mol. Phys.*, 78(4):961, 1993.

[7] P. J. Flory. *Statistical Mechanics of Chain Molecules.* Hanser Publishers, Munich, 1989.

[8] J. Gao and J. H. Weiner. Stress relaxation in a polymer melt of freely-rotating chains. *J. Chem. Phys.*, 97:8698, 1992.

[9] M. R. Garey and D. S. Johnson. *A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, 1979.

[10] A. Y. Grosberg and A. R. Khokhlov. *Giant Molecules Here, There, and Everywhere...* Academic Press, New York, 1997.

[11] A. A. Gusev, M. M. Zehnder, and U. W. Suter. Elasticity of solid polymers as a result of thermal motions. *Macromolecules*, 27:615, 1994.

[12] S. J. Hess. Rheological properties via nonequilibrium molecular dynamics. from simple towards polymeric liquids. *Non-Newt. Fluid Mech.*, 23:305, 1987.

[13] T. C. Hu. *Combinatorial Algorithms.* Addison-Wesley, 1982.

[14] R. M. Karp. On the complexity of computational problems. *Networks*, 5:45–68, 1975.

[15] M. Kotelyanskii, N. J. Wagner, and M. E. Paulatis. Building large amorphous polymer structures: Atomistic simulation of glassy polystyrene. *Macromolecules*, 29:8497, 1996.

[16] O. Kratky and G. Porod. Röntgenuntersuchung gelöster fadenmoleküle. *Rec. Trav. Chim.*, 68:1105, 1949.

[17] M. Kröger. Efficient hybrid algorithm for the dynamic creation of womlike chains in solutions, brushes, melts and glasses. *submitted to Elsevier Science*, 1998.

[18] M. Kröger, W. Loose, and S. Hess. Structure and rheology of polymer melts via nonequilibrium molecular dynamics. *J. Rheol.*, 37:1057, 1993.

[19] E. Leontidis, J. J. de Pablo, M. Laso, and U. W. Suter. A critical evaluation of novel algorithms for the off-lattice monte carlo simulation of condensed polymer phases. *Adv. Pol. Sci.*, 116:285–300, 1994. MMC, CCB, CONROT, NVT/NPT.

[20] G. C. Maitland, M. Rigby, E. B. Smith, and W. A. Wakeham. *Intermolecular Forces - Their Origin and Determination.* Clarendon Press, Oxford, 1981. Ueli's library.

[21] G. C. Maitland and E. B. Smith. The intermolecular pair potential of argon. *Mol. Phys.*, 22:861–868, 1971.

[22] W. L. Mattice and U. W. Suter. *Conformational Theory of Large Molecules The Rotational Isomeric State Model in Macromolecular Systems.* ISBN 0-471-84338-5. Wiley, New York, 1994.

[23] J. I. McKechnie, D. Brown, and J. H. R. Clarcke. Methods of generating dense relaxed amorphous pollymer samples for use in dynamics simulations. *Macromolecules*, 25:1562, 1992.

[24] H. Meirovitch. Computer simulation of self-avoiding walks: Testing the scanning method. *J. Chem. Phys.*, 79(1):502, 1983.

[25] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087, 1953.

[26] J. C. Meza and M. L. Martinez. Direct search methods for the molecular conformation problem. *Journal of Computational Chemistry*, 15:627–632, 1994.

[27] M. Müller, J. Nievergelt, S. Santos, and U. Suter. A novel geometric embedding algorithm for efficiently generating dense polymer structures. *submitted to J. Chem. Phys.*, 1998.

[28] J. Nievergelt and K. Hinrichs. *Algorithms & Datastructures.* Prentice Hall, 1993.

[29] V. J. Rayward-Smith, I. H. Osman, C. R. Reeves, and G. D. Smith, editors. *Modern Heuristic Search Methods.* John Wiley and Sons, 1996.

[30] M. Rehahn, W. L. Mattice, and U. W. Suter. *Rotational Isomeric State Models in Macromolecular Systems.* Springer, Berlin, 1997.

[31] P. Robyr, Z. Gan, and W. Suter. Conformation of racemo and meso dyads in glassy polystyrene from 13c polarization-transfer nmr. *Macromolecules*, 1998.

[32] G. C. Rutledge and U. W. Suter. Detailed atomistic simulation of oriented pseudocrystalline polymers and application to a stiff-chain aramid. *Macromolecules*, 24:1921, 1991.

[33] S. Santos, U. Suter, M. Müller, and J. Nievergelt. A novel parallel-rotation algorithm for atomistic monte-carlo simulation of polymer melts and glasses. *submitted to J. Chem. Phys.*, 1998.

[34] D. N. Theodorou and U. W. Suter. Atomistic modeling of mechanical properties of polymeric glasses. *Macromolecules*, 19:139, 1985.

[35] D. N. Theodorou and U. W. Suter. Detailed molecular structure of a vinyl polymer glass. *Macromolecules*, 18:1467, 1985.

[36] M. Thomaselli, M. M. Zehnder, P. Robyr, C. Grob-Pisano, R. R. Ernst, and U. W. Suter. Local conformations of glassy polycarbonate. *Macromolecules*, 30:3579, 1997.

[37] A. Tiller, A. A. Gusev, and U. W. Suter. in preparation. *J. Chem. Phys.*, 1999.

[38] R. Unger and J. Moult. Finding the lowest free energy conformation of a protein is an np-hard problem. *Bulletin of Mathematical Biology*, 55(6):1883–1198, 1993.

[39] M. Utz. Measurement of structural distribution functions in disordered systems. *J. Chem. Phys*, 109:6110, 1998.

[40] H. R. Warner. Kinetic theory and rheology of dilute suspensions of finitely extendable dumbbells. *Ind. Eng. Chem. Fund.*, 11:379, 1972.

[41] I. Wegener. *Theoretische Informatik*. B. G. Teubner, Stuttgart, 1993.

[42] M. M. Zehnder. *Atomistic Simulation of the Elasticity of Polymers*. PhD thesis, ETH Zurich, No 12497, 1997.

Appendix A

Glossary

Chemistry

Amorphous system: A system which is not in an ordered crystalline state but consists of randomly entangled chains (section 2.1.3).

Cis state: A state of a torsion angle where both adjacent bonds point to the same side (section 3.1.2).

Concerted rotation: *ConRot*: The simultaneous rotation of seven adjacent torsion angles which changes a polymer chain locally, while keeping all bond angles and bond lengths fixed (section 3.7.6).

Conformation: The 3-dimensional shape of a molecule (section 2.1.2).

Covalent structure: The way the atoms of a molecule are joined together by bonds (section 2.1.2).

Euler angles: Three angles which specify the orientation and rotation of a body in 3-dimensional space (section 3.1.1).

Excluded volumes: The non-bonded interaction of atoms is modeled by hard spheres which are not allowed to overlap. (section 3.1.3).

Gauche state: A state of a torsion angle between *cis* and *trans* (section 3.1.2).

Generalized coordinates: The specification of the 3-dimensional shape of a molecule by its torsion angles. (section 3.1.1).

Glass: A non-crystallized amorphous polymer system. (section 2.1.3).

Molecular Dynamics: Simulation of the time evolution of molecules by solving the equations of motion (section 2.2.4).

Monte Carlo: Generation of a sequence of conformations according to their probability to appear (section 2.2.5).

Parallel rotation: *ParRot*: The simultaneous rotation of four torsion angles which conserves the orientation of the moving part of the chain (section 3.7.6).

Parent chains: The chains contained in the original box of a periodically replicated grid (section 2.2.2).

Periodic boundary conditions: A cubic box is replicated throughout space. The periodic images move like the particles in the original box. This way a dense environment is simulated. (section 2.2.2).

Polymer: A chain-like molecule composed of a sequence of uniform repeat units (section 2.1.2).

Primary structure: The way the atoms of a molecule are joined together by bonds (section 2.1.2).

RIS model: Rotational Isomeric States model. Defines a set of states for all torsion angles and their pairwise distribution (section 3.1.2).

Relaxation: The minimization of the energy of a molecular system (section 3.6.2).

Repeat unit: One unit of a polymer which is replicated to form a chain. (section 2.1.2).

Tertiary structure: The 3-dimensional shape of a molecule (section 2.1.2).

Trans state: A state of a torsion angle where adjacent bonds point to opposite sides. (section 3.1.2).

Van der Waals interaction: The force which acts between two non-bonded atoms. (section 2.2.1).

Computer Science

Backtracking:	A method for traversing a search tree. (section 3.6).
Complexity:	The complexity of an algorithm is measured by the number of time (or space) units it uses with respect to the size of the problem to be solved. (section 3.4).
Heuristic search:	A search method which uses insight and knowledge about a class of problem instances. (section 3.3.1).
Iterative search:	A search method which starts with an initial guess and improves it repeatedly until it finds the solution or a local optimum. (section 3.5).
NP-complete:	A problem is NP-complete if it belongs to the class of Non-deterministic Polynomial problems and if it is at least as complex as all problems in NP. For an NP-complete problem, no polynomial time algorithm exists to the best of todays knowledge. (section 3.3.1).
O-Notation:	A function $f(n) : \mathbb{N} \rightarrow \mathbb{N}$ is of the order $O(g(n))$ if $\exists c > 0, n_0 : \forall n > n_0 : f(n) \leq c \cdot g(n)$.
Polynomial time:	An algorithm is polynomial if its complexity is of the order $O(n^k)$ where n is the size of the input and k an arbitrary constant.
Worst case analysis:	Only those problem instances that cause the maximum effort are considered. (section 3.4).

Notation

G: Number of geometric constraints.

h: Horizon: Number of free bonds separating two atoms. (section 3.7.1).

L: Side length of the cubic periodic box.

M: Number of molecules (polymer chains).

N: Total number of particles or atoms.

R: Number of repeat units per polymer chain.

T: Number of rotatable torsion angles (free bonds) per polymer chain.

θ: Bond angle.

ϕ: Torsion angle.

Appendix B

File Formats

MSI/Biosym's Molecule Description File (.MDF)

A `.MDF`-file mainly contains the covalent structure of the molecule together with additional atom and bond attributes as this clipping shows:

```
!BIOSYM molecular_data 4
[..]
@molecule POLYCARB_B0
CARB_1:C            C  cp   [..]  C1/1.5 C5/1.5 HC
CARB_1:HC           H  hc   [..]  C
CARB_1:C1           C  cp   [..]  C/1.5 C2/1.5 H1
CARB_1:H1           H  hc   [..]  C1
CARB_1:C2           C  cp   [..]  C1/1.5 C3/1.5 C7
CARB_1:C3           C  cp   [..]  C2/1.5 C4/1.5 H3
CARB_1:H3           H  hc   [..]  C3
CARB_1:C4           C  cp   [..]  C3/1.5 C5/1.5 H4
CARB_1:H4           H  hc   [..]  C4
[..]
```

The `.MDF`-file is used by the conversion tool `mdf2pp` to generate the decomposition of the molecule into rotatable torsion angles and rigid groups.

MSI/Biosym's Coordinates File (.CAR)

A `.CAR`-file contains coordinates for every atom, declared in the corresponding `.MDF`-file:

```
!BIOSYM archive 3
PBC=OFF
C    3.313660622   -2.504962206  -11.698267937 CARB 1  [..]
HC   2.429594755   -2.005253792  -12.065854073 CARB 1  [..]
C1   4.420852184   -1.754677892  -11.284504890 CARB 1  [..]
H1   4.390906811   -0.676178515  -11.332902908 CARB 1  [..]
C2   5.566863537   -2.402448654  -10.808005333 CARB 1  [..]
C3   5.605681896   -3.800502777  -10.745265961 CARB 1  [..]
H3   6.489747524   -4.300211430  -10.377680779 CARB 1  [..]
C4   4.498489857   -4.550786972  -11.159029007 CARB 1  [..]
H4   4.528435707   -5.629286766  -11.110631943 CARB 1  [..]
[..]
```

RIS-Model Description File (.RIS)

The `.RIS`-file specifies the states and correlations for a specific polymer. The following example shows a RIS model for a-tactic polystyrene:

```
RIS 1  {A-tactic polystyrene}

  BACKBONELIST 2
    CA C1

  DIHEDRALLIST 1
    DIHEDRAL CA C1
      SIDECHAIN C2
      PREDCORR
        MM 2 rowindex
        MR 3 rowindex
        RM 4 rowindex
        RR 5 rowindex
      SUCCCORR
        M 0 rowindex
        R 1 rowindex
      ILIST 2
        INTER  -5   45  { 20  trans }
        INTER  85  135  { 110 gauche+ }

  CORRELATIONLIST 2
```

```
CORRELATION 2 2
   0.0000  0.4350
   0.4350  0.1300

CORRELATION 2 2
   0.0500  0.4100
   0.4100  0.1300
```

First, all atoms belonging to the backbone are listened. For each type of dihedral (free torsion angle), there is an entry in the list of dihedrals. The angle is identified by the two atoms of the free bond. Their order defines the direction, the predecessor and the successor of the bond. If there is a side chain attached to one of the two atoms, this atom must be declared first. The first atom of the side chain adjacent to the bond must also be specified. The correlation with the predecessor and the successor angle are specified by the number of the correlation matrix and one of the two identifiers `rowindex` or `columnindex` which define whether the state of the angle corresponds to the row or column index of the correlation matrix. For a-tactic polymers, four correlations with the predecessor and two correlations with the successor may be specified. The letters `M` and `R` identify meso and racemo dyads respectively. The RIS states are defined by an angle interval. The number of RIS states must correspond to the dimension of the selected correlation matrices. The list of RIS correlation matrices, which are referenced by the free bonds, completes the `.RIS`-file.

`PolyPack` Problem Description File (.PP)

The `.PP`-file, generated by `mdf2pp` contains the complete description of a polymer packing problem. It starts with a list of atoms:

```
POINTLIST 332
  POINT     0 'C'  0 'CARB_1' 1  1.5000
  POINT     1 'HC' 0 'CARB_1' 0  1.0700
  POINT     2 'C1' 0 'CARB_1' 1  1.5000
  POINT     3 'H1' 0 'CARB_1' 0  1.0700
  POINT     4 'C2' 0 'CARB_1' 1  1.5000
[..]
```

A list of distances specifying adjacent atoms, type, weight and length follows:

```
DISTANCELIST 351
  DISTANCE     0     2 BOND  1.50  ILIST  1
    INTER  1.4000  1.4000
  DISTANCE     0     9 BOND  1.50  ILIST  1
    INTER  1.4000  1.4000
  DISTANCE     0     1 BOND  1.00  ILIST  1
    INTER  1.0800  1.0800
  DISTANCE     2     4 BOND  1.50  ILIST  1
    INTER  1.4000  1.4000
  DISTANCE     2     3 BOND  1.00  ILIST  1
    INTER  1.0800  1.0800
[..]
```

Free bonds are identified by four atoms which define their torsion angle value. A type identifier is followed by two integers which reference the two rigid groups, connected by the bond. After the weight, four integers define the correlations, the free bond is involved in. The first and third number identify the correlation matrices valid for the current bond in connection with its predecessor and its successor along the backbone. The second and fourth number are interpreted as boolean values and specify, whether the current bond defines the row or column index in the corresponding matrix. This information is important in case of non-symmetric matrices. A list of intervals on the torsion angle values define the RIS states. The dimension of the referenced correlation matrices must correspond to the number of declared RIS states.

```
DIHEDRALLIST 79
  DIHEDRAL 2 4 15 29 OMEGA 0 1   0.00 1 1  0 1  ILIST 4
    INTER -2.6180 -2.0944
    INTER -1.0472 -0.5236
    INTER  0.5236  1.0472
    INTER  2.0944  2.6180
  DIHEDRAL 4 15 11 12 OMEGA 1 2   0.00 -1 1  -1 1  ILIST 0
  DIHEDRAL 4 15 16 17 OMEGA 1 3   0.00 -1 1  -1 1  ILIST 0
  DIHEDRAL 4 15 29 20 OMEGA 1 4   0.00 0 1  1 1  ILIST 4
    INTER -2.6180 -2.0944
    INTER -1.0472 -0.5236
    INTER  0.5236  1.0472
    INTER  2.0944  2.6180
[..]
```

The list of torsion angles is followed by the list of rigid groups. Every group is composed of a list of atoms and their coordinates which describe its internal 3-dimensional structure.

```
RIGIDGROUPLIST 80
  RIGIDGROUP 8
        4  5.5669 -2.4024 -10.8080
        2  4.4209 -1.7547 -11.2845
        0  3.3137 -2.5050 -11.6983
        9  3.3525 -3.9030 -11.6355
        7  4.4985 -4.5508 -11.1590
        5  5.6057 -3.8005 -10.7453
        6  6.4897 -4.3002 -10.3777
        8  4.5284 -5.6293 -11.1106
  RIGIDGROUP 5
       15  6.7839 -1.5777 -10.3532
        4  5.5669 -2.4024 -10.8080
       11  6.3099 -0.8918 -9.0584
       16  7.2905 -0.7282 -11.5336
       29  7.9162 -2.5079 -9.8830
[..]
```

The list of RIS correlation matrices, which are referenced by the free bonds, completes the `.PP`-file.

```
CORRELATIONLIST 4

  CORRELATION 4 4
     0.1250  0.0000  0.1250  0.0000
     0.0000  0.1250  0.0000  0.1250
     0.1250  0.0000  0.1250  0.0000
     0.0000  0.1250  0.0000  0.1250
[..]
  CORRELATION 4 2
     0.2387  0.0113
     0.2387  0.0113
     0.2387  0.0113
     0.2387  0.0113
[..]
```

Appendix C

Curriculum Vitae

1983–1988	Mathematisch-Naturwissenschaftliches Gymnasium Zürich.
1988–1993	Major in computer science and minor in control systems ETH Zürich, resulting in the degree of Dipl. Informatik-Ing. ETH.
1995–1998	Post graduate study of teacher education in computer science, Höheres Lehramt, ETH Zürich.
1993–1999	Assistant and Ph.D. student in the research group of Prof. Nievergelt, Institute of Theoretical Computer Science, ETH Zürich.