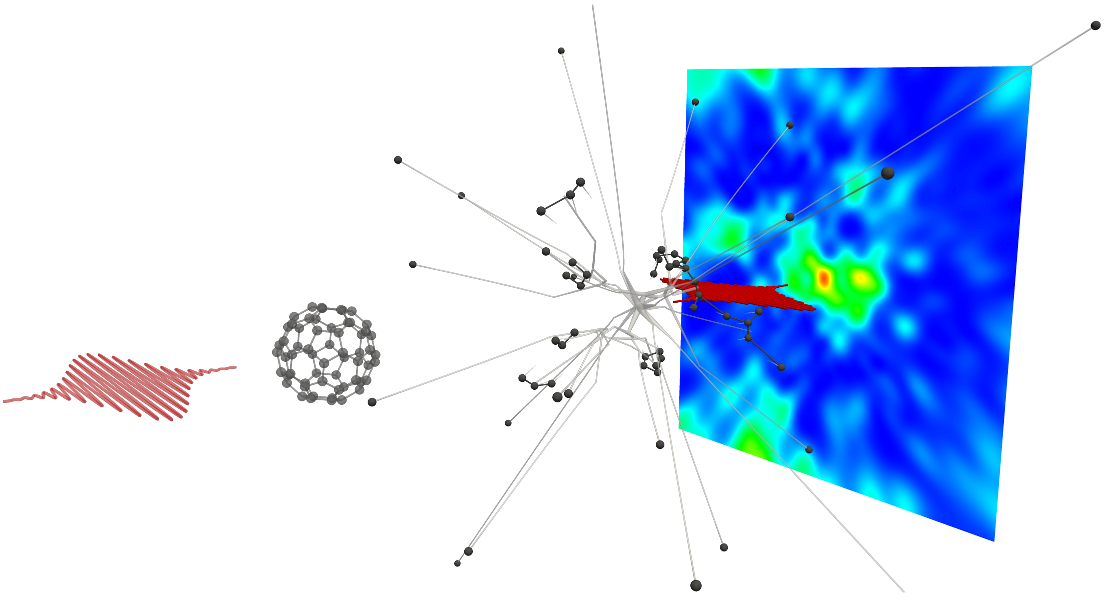
Visualization and Simulation of Laser-Induced Fullerene Fragmentation

Master's Thesis by
Kai Sdeo



Matriculation Number:	3317794
Submission:	May 03, 2017
Supervisor:	Prof. Dr. Filip Sadlo[1]
Co-Supervisor:	Prof. Dr. Thomas Pfeifer[2]

[1] Ruprecht-Karls-University Heidelberg
Visual Computing Group
[2] Max Planck Institute for Nuclear Physics Heidelberg
Quantum Dynamics and Control

Erklärung

Ich versichere, dass ich diese Master-Arbeit selbstständig unter Anleitung verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Abgabedatum: 03. Mai 2017

Acknowledgements

First of all, I would like to thank my supervisors Prof. Dr. Thomas Pfeifer and Dr. Christian Ott from the Max Planck Institute for Nuclear Physics for offering me working on this interesting project and for helping me judging the plausibility of the results of the simulation. I also thank my supervisor Prof. Dr. Filip Sadlo who apart from the physics enriched the topic by his ideas of visualization approaches. Combining precise simulations with a multi-faceted visual analysis really helped to create an interesting and worthwhile project. For me it has been the first time working on a project without clear guideline and without knowing the outcome of the work. I must confess that I have been a little overwhelmed when finding ideas for suitable visualization techniques. Filip guided me through the process of finding and realizing the techniques, presented in this work. The development of the realized techniques of this work has been a preparation for my future PhD position. Further persons I consulted in questions of the experiment are Dr. Sven Augustin, Georg Schmid, and Dr. Kirsten Schnorr, which also provided images for the physical background. From the visualization section Dr. Bastian Rieck also helped me, especially in questions regarding writing this thesis. Thanks to all of you, the cooperation has been a great experience.

My former studying colleagues Adrian Geißler advised me in the choice of the truncated normal distribution I used to model the intensity of the laser pulse. Thanks and good luck with your PhD position. I also want to thank my mother. She has always been a very worrying and caring person, supporting me with all she got. Last but not least I want to mention everyone that contributed to the spelling correctness by proofreading my thesis: Dennis Aumiller, Jan Falb, Adrian Geißler, Dr. Christian Ott, Dr. Bastian Rieck, Prof. Dr. Filip Sadlo, Sophie von Schmettow, Georg Schmid, Marcel Schneider, and Vanessa Sdeo.

Zusammenfassung

Die Erforschung eines der variationsreichsten Bindungselementen Kohlenstoff, das essenziell für das Leben ist, schafft nützliches Wissen in vielen Bereichen der Forschung, wie auch in der Industrie. Ein Molekül, das sich besonders durch seine Kugelhüllenform darin eignet, auch komplexere Systeme von Kohlenstoffatomen zu untersuchen ist C_{60}, eines der bislang am weitest erforschten Fullerenen, dessen Dynamik jedoch längst noch nicht komplett verstanden wurde. Dies gilt insbesondere für dessen Wechselwirkung mit ultrakurzen und starken Laserpulsen. Simulationen können Einblicke in die Dynamik von Molekülen geben, die in Kombination mit Visualisierungen veranschaulicht, analysiert und nachvollzogen werden können. In Anlehnung an Laserexperimente mit Fulleren, die am LCLS durchgeführt wurden, um weitere Aufschlüsse zur Dynamik von Fulleren zu erlangen, werden im Rahmen dieser Arbeit einige deren Experimente in einer Moleküldynamiksimulation modelliert und deren Prozess durch spezifische Visualisierungsmethoden analysiert. Der Fokus hierbei liegt in der Veranschaulichung der Fragmentierungsdynamik, die durch die in den Experimenten verwendeten Laserpulsen ausgelöst wird. Die Ziele dieser Arbeit liegen in zwei Gebieten. Die Simulation soll das Experiment nachstellen. Dies beinhaltet die Modellierung von C_{60} selbst durch eine Wahl von Kraftfeldpotentialen und die Modellierung des Laserpulses mit dessen Intensität. Die Simulation wird hierbei auf die Ergebnisse der physikalischen Experimente abgestimmt. Die Ziele im Bereich der Visualisierung sind die Entwicklung neuer Analysetechniken für die Fragmentierungsprozesse von Fulleren, die Visualisierung der Fragmentdynamik durch Flexibilitätsmethoden, die Rekonstruktion von Beugungsbildern als weiteres Referenzmaterial für die experimentelle Auswertung, wie auch eine Analyse der in dieser Arbeit erzielten Ergebnissen.

Abstract

The benefit from the research of carbon, an element with one of the highest variety of binding possibilities that is essential for life, has a strong impact in many fields in science as well as in industry. A molecule that is suited to explore more complex systems of carbon atoms due to its highly symmetrical hollow sphere-like structure is C_{60}, one of the best known fullerenes. Still, its dynamics is far from being understood, especially its interaction with ultrashort and strong laser pulses. Simulations can help us to get insights into the dynamics of molecules. In combination with visualization, these dynamics can be analyzed and understood. Leaned to laser experiments with fullerene, performed at LCLS to get further insights into the dynamics of fullerene, this work examines some of their experiments by means of molecular dynamics simulations, which we analyze by our developed visualization techniques. The focus is on the fragmentation dynamics, induced by laser pulses that are used in the experiments. The contribution of this work can be summarized into simulation and visualization. Simulations are required to imitate the experiment, including the modeling of C_{60} by the choice of force field potentials, the modeling of laser pulses, and their intensities. The results of our simulations are adapted based on results from the experiments. Goals in the visualization are the development of novel analysis techniques. These techniques are for the fragmentation process of fullerene, the fragmentation dynamics by flexibility methods, the reconstruction of diffraction images, which can be used as additional medium for the physical analysis, as well as the analysis of the achieved results of this work.

Contents

1 Introduction

Carbon is an essential element in our world. Living organisms like humans, animals, and plants, as well as microorganisms like bacteria or viruses all consist of cells—and one of the most important components to build these complex structures is carbon and its rich variety of chemical bindings.

Fullerenes are a category of carbon-based molecules that have a hollow, sphere-like structure. They have semiconductive and superconductive properties and the probably best-known fullerene is the buckyball, or C_{60}, which consists of 60 carbon atoms. Its highly symmetrical structure makes this medium-sized molecule interesting for studying the properties of complex carbon bond systems.

Advancing the knowledge of the chemical binding properties of carbon enriches many fields. In many cases, one would like to be able to specifically break or create bonds in such molecules. Already today, we are able to influence bonds by lasers. In radiation therapy, X-rays can break covalent bonds in cancerous cells in order to destroy them. There are also applications in the field of materials science that include the discovery and design of new materials. Even fullerenes can be designed by radiating a surface of graphite with lasers.

Many applications involve laser-induced interaction with fullerenes, and the underlying processes can be studied by observing ensembles of fullerene molecules. In such setups, the laser excites the molecules, and depending on the input energy, molecules are ionized or even teared into molecule fragments. While the final states of the decay have already been studied extensively, the dynamics are still in research.

This works aims at a simulation-driven description and advanced visualization-aided analysis of laser-fullerene interaction with the focus on the fragmentation of fullerene. Three subtasks have been processed in this work. The starting point is the description of the process by a model. Molecular dynamics simulations are performed to obtain data of the processes. These data are compared in a second step with experimental data. In step three, novel visualization techniques are developed and used as the key tool for analyzing the dynamics of the fullerene.

The topic of this thesis is motivated by experiments done at the Linac Coherent Light Source (LCLS) at SLAC in Stanford in 2016. Optical lasers interacting with fullerene induce coherent electronic and vibrational excitations. For a certain level of excitation, the original structure of the molecule starts oscillating symmetrically or asymmetrically. The lifetime of these exited states have been determined only indirectly so far. The goal of the experiment was to measure these lifetimes directly by diffraction imaging. High laser intensities ionize the molecule until it decays into fragments. The charge and mass of the respective ions were determined by time-of-flight spectroscopy. They are essential for the validation of the quality of our simulation results. Details on fullerene and the experiments are given in Chapter 3.

Examining the laser-fullerene interaction based on simulations gives us further insights into the overall process. The simulation data extends our means to see and understand the motion of fullerenes and their fragmentation. Our simulations are performed using LAMMPS, a fast molecular dynamics framework. It enables us to model fullerene with a force field that is determined by potentials. The interaction with laser pulses is modeled by means of a probability distribution to ionize atoms in the molecule. The probability to ionize an atom is set proportional to the Gaussian-shaped laser pulse intensity. As a subsequent step, some adaption of the simulation results is necessary to improve the correlation to the results of the experiments. This has been performed by introducing simple bond-breaking approaches. Details about the simulation are given in Chapter 4.

Whereas direct depiction of the C_{60} molecule provides a basic view, visualization approaches rapidly reaches their limits. To understand the fragmentation of fullerenes advanced visualization techniques are necessary to understand the fragmentation of fullerenes. Besides standard atomistic and bond-centric visualizations, we resent a set of visualization tools to analyze the fragmentation dynamics. For decay classification, we introduce hole size analysis of the fullerene as an abstract measure. Analog to the experiments, virtual diffraction images are reproduced for further comparisons with experiments. All theses techniques are described in detail in Chapter 5.

All results belonging to simulation as well as visualization are given in Chapter 6. We show the behavior of the molecule for different force fields, show the influence of bond breaking approaches, presents the similarity between simulation and experimental results and present our results regarding the fragmentation dynamics. This work concludes with Chapter 7.

In this thesis, we specifically contribute:

- Simulation results of
  - ionized fullerenes with a molecule dynamic framework
  - diffraction images of fullerenes
- Visualization techniques for
  - Gaussian-shaped laser pulses
  - fragmentation dynamics of medium-sized molecules
  - tracking of fragments by tree-like trajectories
- Simulation quality analysis with experimental data by comparing
  - time-of-flight yields
  - simple bond-breaking approaches
- Fragmentation dynamics analysis of fullerene by
  - fragment property relations
  - holes of fullerene fragments as an abstract representation

2 Related Work

C_{60} has been the first fullerene discovered in 1985 [KHO$^+$85]. Since then it has been very popular for many researchers of a broad, interdisciplinary field. Many interesting notions and facts about C_{60} are found in the work of Hertel et al. [HLS05].

Many optical laser experiments with fullerene were performed, e.g., see [HLS05], [BCR03]. In their research, time-of-flight mass spectroscopy has also been used. One of the main result they provided was that the energy redistribution strongly depends on the laser wave length, pulse duration, and intensity. Diffraction imaging is a widely spread technique. Some theory is provided by Cao et al. [CW98]. Neutze et al. [NWvdS$^+$00] studied X-ray pulses and their damage on proteins. They made use of diffraction image simulations, but they did not analyze sequences of it.

In recent years, the use of simulations became popular in combination with experimental results to get better understanding of processes. There have also been works that simulated the interaction between laser and C_{60}. Yamazaki et al. [YNN$^+$14] for example explored highly charged fullerene explosions. A sophisticated simulation in combination with experimental results was performed by Murphy et al. [MOJ$^+$14]. Their work included observations of the dramatic influence of distinct pulse lengths. Both works analyzed the decay of C_{60}, but they restricted themselves to classical visualization techniques.

This work performs simulations, particularly molecule dynamic (MD) code to acquire molecule data. Elaborately information about MD simulations are given by Leimkuhler and Matthews [LM15]. They made suggestions for a couple of simulation frame works, including the chosen one in this work LAMMPS [Pli95]. A more detailed overview about molecule simulation framework is serves Vlachakis et al. [VBPK14].

Bond-breaking approaches are investigated for this specific work. For MD simulations, there might be more advanced approaches. Öhrn et al. [ÖOD96] proposed a bond-breaking approach to provide previews of chemical reaction processes since their exact computation usually are very lengthy.

The evaluations of the results are usually done by simple and intuitive means like scatter plots, histograms or bar charts. Commonly, also illustrations of molecular or atomic configurations are shown. Besides such simple approaches, more elaborate approaches are desired to be found during this work. The state of the art report for the visualization of biomolecular structures [KKF+16] served a broad overview. They introduced a taxonomy of visualization techniques that is briefly presented in Section 5.2. In the following, some advanced visualization techniques of this report are presented, grouped in categories.

The first category is flexibility. Uncertainty in the position can be shown as probability cloud [RJ99]. Schmidt-Ehrenberg et al. [SEH05] developed a technique to further depict atomic properties in such clouds. In context of thermal vibrations, a semi-transparent fuzzy layer around molecular surfaces was shown by Lee and Varshney [LV02]. Fioravante et al. [FSTR13] presented a technique that uses principal component analysis and covariance clustering to obtain motional correlations in proteins. Similarly, Ahlstrom et al. [ABE+13] used network visualization to show certain conformations during molecule dynamic simulations.

Aggregation techniques like the work of Falk et al. [FKRE10] visualized the signal density, obtained from the positions of signal proteins. Atoms that have been grouped according to certain properties were sampled into separate grids to show a spatial density in the work of Rozmanov et al. [RBT14]. Properties of molecules can be mapped onto spheres [KFS+17]. The texture containing the information can than be unfolded. Aggregated motion can also be shown as arrows or stream lines. This way, Chavent et al. [CRG+14] studied the diffusion of lipids and Ertl et al. [EKK+14] showed the motion of ions around DNAs in a nanopore.

Since the decay of fullerene can be seen as a link of reaction the visualization of reaction networks has also been listed here. Tools for modeling molecular reactions like *CellDesigner* [FMKT03] are able to visualize the quantitative change of reactants in time-intensity plots. Moreover, they also provide various approaches for network visualization. Wang et al. [WTM+14] shows the yield of reactants, which are ploted with respect to the size of the molecules and the reaction time. For some temporal regimes reactants were also directly depicted in 3D. In another figure the reactants were illustrated in a graph in which the nodes contained reactants as atom-stick model. By scaling the opacity, the most important reactants were emphasized. All nodes are connected by arrows that depict the main direction of the chemical reaction. The spatial depictions of chemical reactions is also possible by means of several tools. *ZigCell3D* [dHCKMK13], visualizing cellular models on the atom scale. A special feature of the tool is that ongoing reactions are highlighted. Since places of reactions are often crowded some events might be missed. To overcome this problem, Le Muzic et al. [MPSV14] invented a visualization procedure that observes the simulation process. It focuses on reaction spots once they are happening. This technique was further improved by slowing down the animation speed during active reactions [LMWPV15].

All these methods may be adaptable for this work, however have neither been designed to visualize molecule-laser interaction nor specifically presented the decay of fullerene for high laser intensities. Visualizations that include fullerenes have been performed for applications, like showing the properties of organic semiconductors [JBS+17]. The visualization framework, mainly used in this work are *ParaView* [AGL05, Aya15]. *PyMol* [Sch15] and *VMD* [HDS96] were also tried out.

3 Physical Background

The experiments performed at Linac Coherent Light Source (LCLS), which are the basis of this work, are described in this chapter. It starts with a short motivation of the experiments in Section 3.1, followed by the structure of the molecule of interest, C_{60}, in Section 3.2. Concluding, the experiments are described in detail in Section 3.3.

3.1 Motivation for the Experiment

The fullerene is a system that enables the exploration of the coupling between many atomic and electric degrees of freedom. Its symmetry also makes theoretical studies of such a complex structure possible. A high complexity also arises in the molecule-light interaction from excitation and energy-redistribution. In total, the fullerene has 174 vibrational modes. These and further details are provided by Hertel et al. [HLS05]. The coupling of electronic modes, vibrational modes, and coherent lifetimes during or after optical excitation is very complex and not understood. They have been experimentally observed in only very indirect ways by modulations of ion yields, laser pulse shapes, or by theoretical calculations. The experiments performed at the LCLS aims at determining the lifetimes of these vibrational modes by direct diffraction imaging. The minimum goal of the experiments is to determine which mode is preferentially excited by near infrared femtosecond laser pulses. More details on the vibrational modes are provided in Section 3.2.

The results of the experiments performed at the LCLS can lead to several other studies. Direct visualization of dynamical vibrational control by laser pulse sequences could be explored, as suggested by Naoyuki et al. [NKI^{+}12]. The experiments can be seen as proof-of-principle experiments for time-resolved imaging of laser-induced coherent molecular dynamics. Future experiments might be conducted in the same fashion with other medium-sized molecules interesting for other applications.

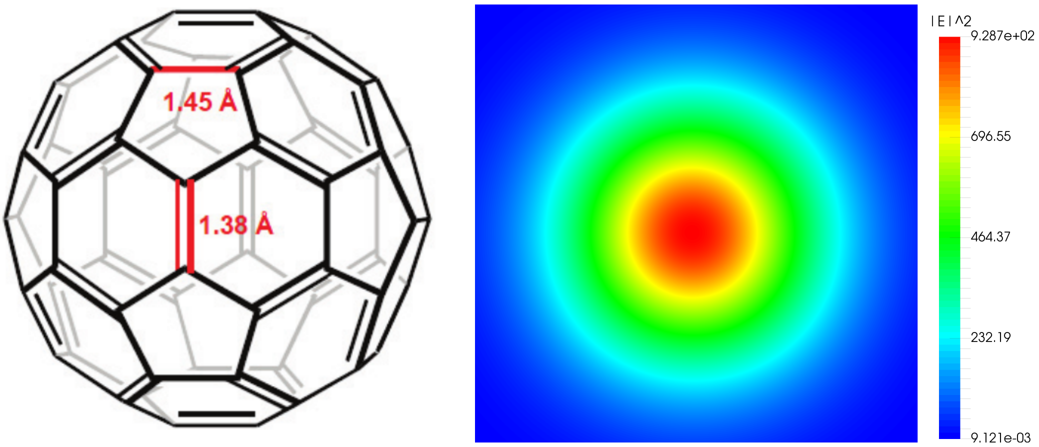


Figure 3.1: **Left:** Structure of C_{60} with 30 single bonds, spread on pentagonal rings, and 60 shorter double bonds. Image taken from [BdMB15]. **Right:** Computed diffraction image of C_{60} at rest.

3.2 The Fullerene C_{60}

The fullerene considered in this work is C_{60}. It consists of 60 carbon atoms and is depicted in Figure 3.1. It possesses 240 valence electrons, four for each carbon atom to form bonds. There are 30 single and 60 double covalent bonds. Each bond of the respective type is formed by two or four electrons. In total, the 240 valence electrons are spread on 90 bonds. The molecule builds a spherical surface, linked by 12 planar pentagons with uniform corner angles of 108°, and 20 planar hexagons with angles of 120°. The double bonds are located between adjacent hexagons. The single bonds form the pentagons, which are linked with the hexagons. The single bonds between two carbon atoms in C_{60} are longer than the double bonds. This has to be taken into account for the used binding potential (Section 4.3.1). There are contrary data about their exact length. Hedberg et al. [HHB+91] reported bond lengths of 1.458 Å and 1.401 Å. A more recent source [Car05] mentions lengths of 1.45 Å and 1.38 Å, which Berkai et al. [BdMB15] referred to who provided the parameters for the force field, used in the simulation. Monticelli [Mon12] the positions of the atoms and the bond connectivity of C_{60}. Since these data were used, we averaged the bond lengths by the distances of the provided positions to minimize the impact of oscillations in the molecule. Therefore, the bond lengths 1.48 Å and 1.38 Å have finally been used. The data also provided triplet lists for angular, and quartet lists for dihedral potentials, respectively (see Section 4.3.1).

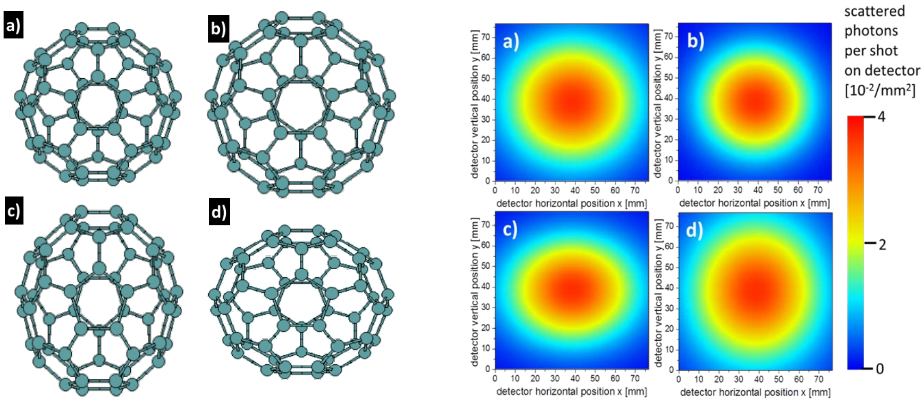


Figure 3.2: **Left:** Atom configurations of the vibration modes. On the top, the radially-symmetric breathing mode in starting state (a) and end state (b), and the prolate (c) and oblate mode (d). **Right:** Simulated diffraction images that have been computed by our project partners beforehand to obtain expected results as preparation for the experiments. Images (a)–(d) correspond to the respective atom configurations.

C_{60} molecules can be excited to vibrational modes. Already in the first femtosecond laser pulse experiments [DMS+93, HCF+94], the excitations of specific vibrational modes have been observed. These modes are the radially-symmetric breathing mode and the prolate/oblate mode. Figure 3.2 shows the vibrational modes of the breathing modes with their diffraction images (Section 3.3.2). Note that configurations with a high radius of the buckyball result in a smaller circular structure in the image.

Similarly, the stretching in vertical direction results in a stretching of the circle in horizontal direction. The images were simulated by the physicists with am assumed photon energy of 2 keV, a distance to the image plane of 69 mm, and detector size of 76.8 mm. They simulated these images using static geometry unlike this work, in which the molecule is simulated and the diffraction image of the resulting configuration is computed therefrom.

Theoretical calculations about the lifetimes of the vibrational modes were performed by Dresselhaus et al. [DDE96]. They report that the breathing mode has a oscillation period of 67 fs. Prolate and oblate modes posses a period of 122 fs.

3.3 Experiments

In the following, the experimental setup (Section 3.3.1) and the details of the two types of experiments (Sections 3.3.2 and 3.3.3) are described.

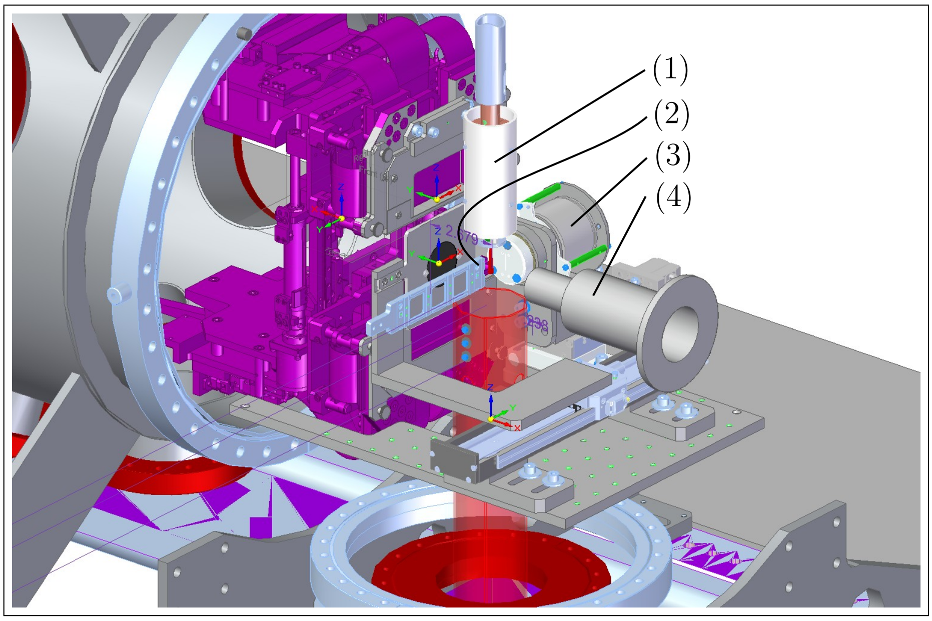


Figure 3.3: CAD image of the experimental setup. **(1)** Oven that sprays a jet of gaseous fullerene molecules. **(2)** Diagnostic paddle with repeller electrode. **(3)** Time-of-flight mass spectrometer. **(4)** Light baffle that is passed by both lasers, pointing from right to left. The image was provided by our project partner.

3.3.1 General Setup of the Experiment

The experiments have been performed at the LCLS. In the first type of experiment, diffraction images were recorded to have a spatial measure of the structure of C_{60} to study the behavior of the interaction with light. The second type of experiment uses a time-of-flight spectrometer (TOFMS) to measure the mass distribution of ionic reaction fragments.

Both experiments were done in the LAMP chamber. An insight view of the chamber is schematically illustrated in Figure 3.3. The fullerene molecules were filled into an oven, see Figure 3.3 (1). C_{60} becomes gaseous at heating temperatures of about 550° C. From the oven, an effusive jet was sprayed into the chamber. The jet provided densities of about 10^9 up to $10^{11}\,\mathrm{cm}^{-3}$ C_{60} molecules.

The experimental setup uses two different laser sources. Their beams are sent from right to left through a light baffle (Figure 3.3 (4)). A light baffle is a kind of pipe that reduces light scattering. The LCLS X-ray focusing mirrors were tuned so that the X-ray spot size in the interaction region is around $30\,\mu$m. The optical laser with a wavelength of 800 nm was focused collinearly to the X-ray beam to a

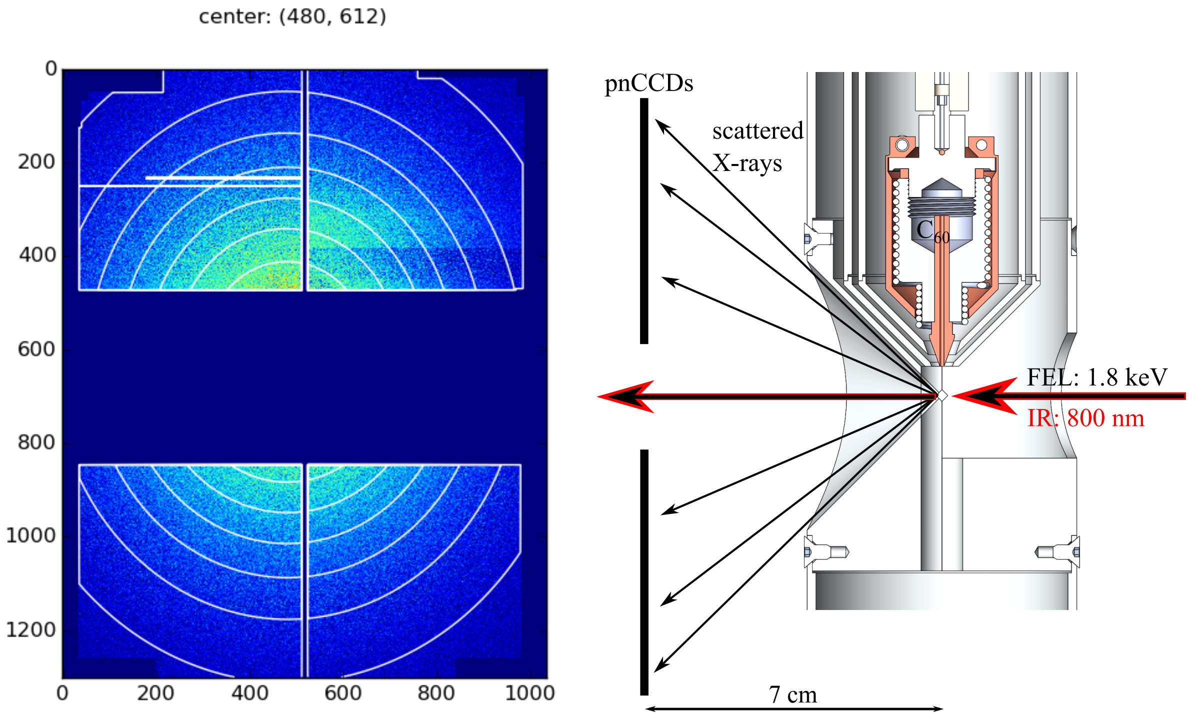


Figure 3.4: **Left:** Measured diffraction image, consisting of four areas of the pnCCD detector. Non-detector regions without data are colored dark blue. The circular white lines signal the contour lines of a Gaussian fit. **Right:** Illustration of the diffraction experiment on C_{60}. The laser pulses are shot coplanarly from right to left. The optical infrared (IR) laser is shown in red, the free-electron laser (FEL) and its inducing scattered X-rays in black. Both images were provided by our project partner.

focal spot size of 60 μm or more to assure that all sample molecules are completely hit. The optical laser intensity, needed for the vibrational excitation is between 10^{13} and $10^{14}\,\mathrm{W\,cm^{-2}}$ with pulse durations of 40 fs. For observing the transition to C_{60} breakup and fragmentation, intensities in the order of $10^{15}\,\mathrm{W\,cm^{-2}}$ are required.

The diagnostic paddle (Figure 3.3 (2)) and the spectrometer (Figure 3.3 (3)) are used for the time-of-flight spectroscopy and beam diagnostics, respectively.

3.3.2 Recording of Diffraction Images

Diffraction images can be recorded by pnCCD detectors that are sensitive to X-rays. The extends of the recorded diffraction images correspond to the sizes of 10.35 cm in horizontal and 8.34 cm in vertical direction. Note that the laser pulses must not hit the detectors directly, otherwise they will be damaged. This is the reason for the

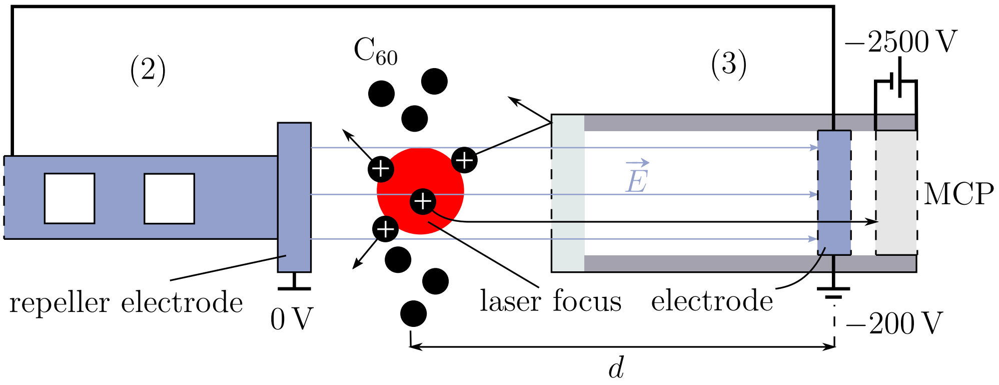


Figure 3.5: Schematic experimental setup for the time-of-flight spectroscopy with the diagnostic paddle (Figure 3.3 (2)) and time-of-flight mass spectrometer (Figure 3.3 (3)). C_{60} molecules are ionized by IR laser pulses at the focal spot. The electric field accelerates all ions toward the micro-channel plate (MCP), where their time-of-flight is measured with respect to the laser trigger.

missing data points in Figure 3.4.

Images are recorded during a time-resolved *pump-and-probe* experiment. Consider Figure 3.4 for a depiction of the experiment. First, an infrared laser pulse excites an ensemble of molecules (*pump*). Its pulse duration is 40 fs. After a defined time delay, the X-ray probe pulse scatters at the molecular ensemble and the corresponding diffraction images are recorded by the pnCCD detectors (*probe*). The pump-probe delay is scanned continuously.

3.3.3 Time-of-Flight Spectroscopy

Figure 3.5 shows the experimental setup. From the oven, fullerene molecules are evaporated into the vacuum chamber. For sufficiently high intensities, the infrared laser trigger ionizes fullerene molecules (marked as white plus) at the focal spot. A trigger that is synchronized to the laser pulses starts when the laser pulse arrives at the laser focus. In order to catch and count ionized molecules, an electric field $\vec{E}$ accelerates all ions depending on their charge and mass towards the time-of-flight mass spectrometer (TOFMS). Note that the entrance of the TOFMS is just a small slit. This dramatically decreases the ion counts, but has the effect that the trajectories of the molecules are kept straight as a good approximation of flight distance d. The acceleration of ionized fragments of C_{60} is achieved by creating an electric field between two electrodes. The repeller electrode is positioned at the control paddle. The other negatively charged electrode is located in the TOFMS. Their electric potential is about $-200\,\mathrm{V}$. A micro-channel plate (MCP) is mounted at the end of the spectrometer, which acts as an amplifier by registering the time of each impinging ion. Each ion hit causes an electron avalanche within the MCP channel, which is registered as count of an ionized fragment. The time span between these two times is referred to as *time-of-flight* (TOF, t_{TOF}).

The time-of-flight spectroscopy allows one to measure the fraction between mass and charge of molecules as shown in Eq. 3.1.

$$E_{\mathrm{pot}} \overset{!}{=} E_{\mathrm{kin}} \quad \Leftrightarrow qU \overset{!}{=} \frac{1}{2}mv^2 = \frac{1}{2}m\left(\frac{d}{t_{\mathrm{TOF}}}\right)^2 \quad \Leftrightarrow t_{\mathrm{TOF}}^2 = \underbrace{\frac{1}{2U}d^2}_{k^2:=}\frac{m}{q} \quad \Rightarrow t_{\mathrm{TOF}} = k\sqrt{\frac{m}{q}} \tag{3.1}$$

All ions are accelerated at a distance d by the same electric potential U. After passing the spectrometer, they are no longer accelerated. Thus, the potential energy has been converted into kinetic energy (Eq. 3.1).

4 Simulation

To get insights into the processes one has to imitate the behavior of fullerene under the conditions of the experiment, using simulations. Note that although simulations are conducted in this work, it is focusing on the visualization thater than the simulation. It would be too challenging to obtain a simulation in this context that can compete with state of the art methods like for example the approach by Murphy et al. [MOJ+14]. In the beginning of this chapter, a collection of methods to simulate molecules are presented (Section 4.1). The fundamentals of the chosen molecule dynamics simulation method for this work are given in Section 4.2, followed by Section 4.3, focuses on the forces in a molecule. Finally, the approach to model the experiment is given in Section 4.4.

4.1 Molecule Simulation Methods

An outlook of the physical theory, solution approaches, possible simulation frameworks, and a short motivation for the chosen simulation framework follows here.

4.1.1 Simulation-Based Theory

The choice of a simulation method is trading off accuracy and speed. There are two fundamental theories to describe systems of atoms or particles. In the classical view of physics, a particle with index i of mass m_i at a certain position $\vec{x}_i(t)$ moves with a velocity $\vec{v}_i(t)$ over time t. It can be accelerated (with $\vec{a}_i(t)$) by forces $\vec{F}_i(t)$ due to Newton's law $\vec{F}(t) = m \cdot \vec{a}(t)$. Thus, the fundamental approach is to solve a system of ordinary differential equations that consists of all acting forces.

$$\frac{\partial \vec{v}_i(t)}{\partial t} = \frac{1}{m_i}\left(\vec{F}(\vec{x}_i) + \sum_j \vec{F}(\vec{x}_i, \vec{x}_j) + ...\right), \qquad \frac{\partial \vec{x}_i(t)}{\partial t} = \vec{v}_i(t). \tag{4.1}$$

All considered force terms are listed in Section 4.3.1. The theory of classical physics is, however, not able to explain phenomena observed in experiments at smaller scales, like the scale of atoms. This is where quantum theory comes into play. It has been shown that either the position or the momentum $\vec{p} = m \cdot \vec{v}$ of a particle can be observed at a time. This is known as Heisenberg's uncertainty principle. Thus, the position in combination with the velocity of a particle can only be given with a certain probability in quantum theory. Such probabilities can be obtained by solving the time-dependent Schrödinger equations, determining the resulting complex wave function $\psi(\vec{x}, t)$, and computing its squared modulus $|\psi(\vec{x}, t)|^2$.

4.1.2 Justification for the Chosen Simulation Framework

Naturally, one would like to use the more accurate quantum theory. However, the choice in this work is to use a molecular dynamics framework based on the classical view. This is mainly due to performance reasons, as also discussed at the end of this section. Leimkuhler and Matthews [LM15] suggested some frameworks that implement molecule dynamic code and emphasized the LAMMPS framework [Pli95] to be developer-friendly since it offers easy integration of customized code. A short investigation of the framework pointed out its suitability for this work.

To name frameworks that implement quantum dynamics code, there are for example a CPU-based implementation GAMESS [GBD$^+$05] and a GPU-based commercial quantum chemistry software package named TeraChem [UM09]. Wang et al. [WTM$^+$14] performed ab initio molecular dynamic simulations with TeraChem in order to research chemical reaction processes. They used the Hartree-Fock method (e.g., see [PKvdBS16]), which approximates the solution of the Schrödinger equation. It appears to be popular for quantum based calculations.

In between there are also hybrid methods that use the advantages of both types of approaches. For example, the state-of-the-art simulation of fullerene [MOJ$^+$14], also mentioned in the beginning of Section 4, uses this hybrid approach. Quantum dynamics code is only used as long as there are critical situations, in which molecular dynamics code gives too coarse approximations. According to Kozlíková et al. [KKF$^+$16], such situations are for instance systems that depend on sensitively fine-tuned energy values or the breaking of covalent bonds. Unfortunately, the latter also happens during the process in our experiment.

To give a short comparison of compute times, the website of TeraChem shows a performance comparison between GAMESS and TeraChem for simulating the fullerene. The GAMESS CPU code took 22.6 min, the TeraChem GPU code 19.72 s

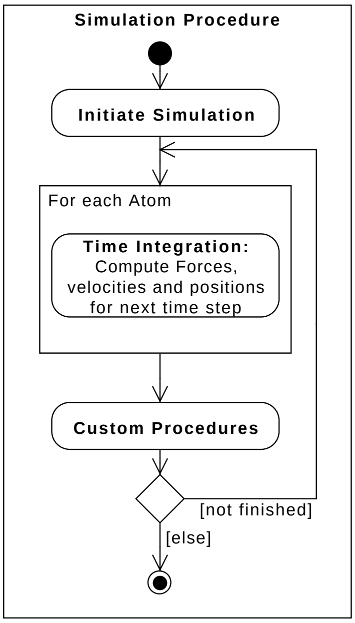


Figure 4.1: A rough sequence diagram of the simulation process.

for one Hartree-Fock iteration [Ter]. Assuming that one iteration corresponds to one time step of size 0.5 fs, the simulation of 1 ps with the CPU code would have taken 753 h, and with the GPU code 11 h. Wang et al. [WTM+14] simulated systems that contained 156 and 228 atoms, which took 74 h and 102 h, respectively. In comparison, the molecule dynamic based LAMMPS framework took 0.1 s with a single CPU for computing the simulation at the same time resolution.

For this work it is important to have such a fast code. The analysis relies on a large quantity of simulation runs to obtain dependable statistics. The need of experimenting with the simulation to tune the simulation results as near to the results of the experiments as possible makes the use of a slow method inappropriate, regarding the restricted time for this work.

4.2 Simulation Procedure

As motivated in Section 4.1.2 the framework LAMMPS is used for the molecule simulations in this work. It is freely distributed under the terms of the GNU Public License, and written in C++. There is a variety of models and methods between one can choose to adapt the simulation for a certain purpose.

The LAMMPS framework is designed for both large-scale and small simulations.

It is able to run simulations on single CPUs, OpenMP, MPI, or GPUs (either CUDA or OpenCL) application. Since describing the framework in detail would go beyond the scope of this thesis, we refer to Plimpton [Pli95] for further information.

For each LAMMPS simulation run, a configuration file must be passed as command line argument. defining the simulation run. Figure 4.1 illustrates roughly the procedure of one simulation run. The first step is the initialization. In the following, the most important settings are listed with the corresponding choice for this work in brackets. General settings like unit convention and dimensionality (3D) are set. The type of model is used to determine the data to store (molecular model, which stores atom IDs, molecule IDs, position, velocity, total force, charge per atom). The active forces with their parameters (molecular and pairwise, see Sections 4.3.1 and 4.3.2) and the atoms with their positions and other properties (the configuration of one fullerene molecule with its bonds, see also Section 3.2) are defined.

Next, time integration is performed for each atom. This incorporates the computation of the previously defined forces, the velocities, and the positions. This is done by a so-called *fix*. Generally, fixes are modules of the LAMMPS framework that can influence or analyze the simulation. For the time integration in each time step, a time stepping scheme must be chosen. This work uses the nvt-fix [SSM04], a time integrator, which besides solving Eq. 4.1 also includes thermodynamic effects. The chosen starting and end temperatures are both 550°C. The damping constant was set to 1000, which does not cause a damping of the velocities of all atoms, but kept their velocity constant (as long as no other forces interfere). 0.05 fs has been chosen as time step size. There has also been used a SHAKE fix [RCB77] to increase the stability of the molecular forces with respect to the time step size. The used parameters are 0.0001 for the SHAKE solution accuracy, and the maximum number of iterations in each SHAKE solution was set to 10. We further use a fix to randomly rotate the molecule depending on a global seed. Each fix that uses random numbers is fed with the same seed such that whole simulations can be reproduced. This comes in handy if certain analysis fixes were turned off to save computation time and need to be run for certain simulation results.

It is also possible to write and include customized code by implementing a fix. Various fixes have been implemented during this work. One implements the laser pulses and their ionization effects (Sections 4.4.1 and 5.3.4). The bond-breaking fix has been adapted to break bonds of charged atom pairs (Section 4.4.2). Fixes have also been used to export visualization data like the diffraction images (Section 5.3.5),

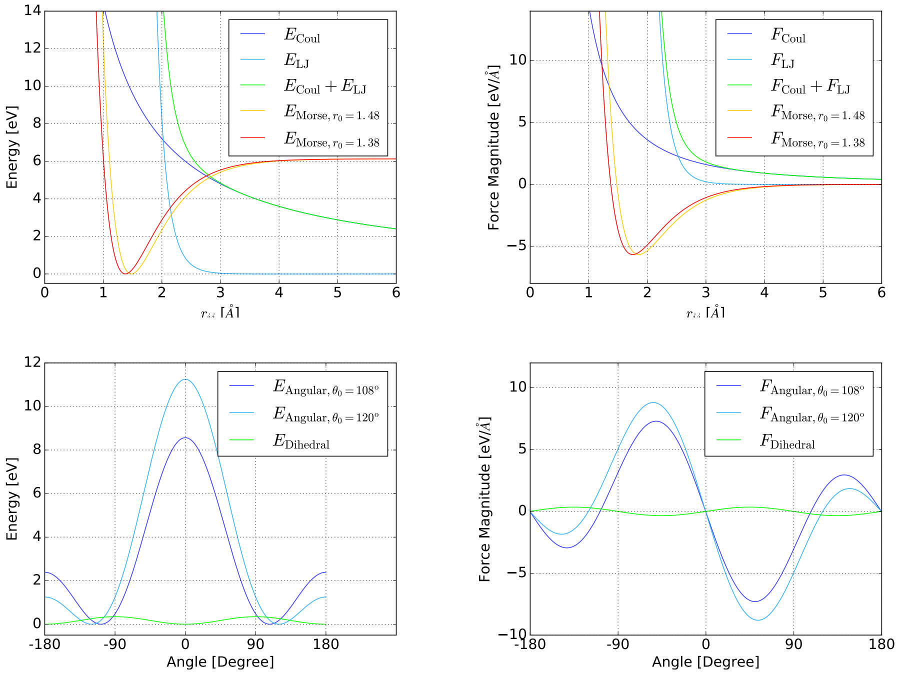


Figure 4.2: On the left, the energy potential curves, and on the right, their resulting forces are shown. Positive force magnitudes repel, negative ones attract atoms to each other. On the top there are the radially symmetric potentials including the Coulomb (Coul), Lennard-Jones (LJ), and the Morse potentials for two different equilibrium distances r_0 with respect to r_{ij}, the pairwise distance of two atoms. The bottom row shows the potentials with respect to the angles, depicted in Figure 4.3. The angular model is shown for two angles at rest θ_0. Note that the Morse and angular potential have their minimum at rest. As a result, the force in the equilibrium is zero.

atoms and their bonds (Section 5.3.1), atom trajectories (Section 5.3.2), and the trajectories of decayed fullerene fragments (Section 5.3.3). Once all computations for one time step are done, the next time integration is started and the process is repeated as long as the wanted number of time steps is not reached.

4.3 Force Field of Fullerene

This section details the force field of fullerene, used in our simulation. The force field is composed of different kinds of forces, which are modeled as functions. For each model function and for each kind of bond, parameters have to be determined. These parameter values are usually derived by quantum calculations. Berkai2015 et

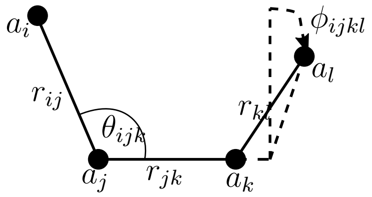


Figure 4.3: A list of atoms a_i to a_l, and exemplary distances r_{ij}, r_{jk}, r_{kl}, angle θ_{ijk}, and dihedral angle ϕ_{ijkl}. The vectors between the atoms a_i, a_j and a_j, a_k span a plane. ϕ_{ijkl} is the angle between this plane and the atom a_l. For each bond, a pair, for each angle, a triplet and for each dihedral angle, a quartet of indices must be defined to simulate the respective bond, angle, or dihedral angle potential.

al. [BdMB15] explored parameters for a set of models to simulate fullerene. Luckily, they have been implemented similarly in the LAMMPS framework, such that they could be used almost directly. This is because the formulas only differ sightly from each other. The notation of the LAMMPS framework was preferred for this section. Some variables have been renamed for the sake of uniqueness of variables, however.

Each model comes with a potential, the energy of the model. The derivative of the potential serves as the force. All models with their energies and forces are shown in Figure 4.2. Principally, there are molecular forces and pairwise forces. Pairwise forces act between every two atoms, which are not directly bonded to each other. Molecular forces act only in molecules due to covalent bonds, the sharing of electrons between atoms that keeps the molecule together.

4.3.1 Molecular Forces

This section describes which potentials have been used to model bonds. Each one needs a sequence of atom indices in order to know which energy belongs to what set of bonds and atoms, see Figure 4.3. All parameter values were taken from Berkai2015 et al. [BdMB15] and adapted for the LAMMPS framework.

Bond Potential

The binding or stretching potential describes the stability of a bond between a pair of atoms. For this purpose, the Morse potential has been used:

$$E_{\text{Morse}}(r_{ij}) = D\left(1 - e^{-\alpha(r_{ij}-r_0)}\right)^2 . \tag{4.2}$$

Here, r_{ij} is the distance between two atoms with indices i and j, $D = 6.13220\,\mathrm{eV}$ determines the depth of the potential well, r_0 is either 1.48 Å or 1.38 Å, depending on the particular equilibrium bond distance, and $\alpha = 1.8502$ is a stiffness parameter. Note that Berkai2015 et al. [BdMB15] used the same potential, but with a additional constant $-D$, which is lacking in the LAMMPS framework. However, the result of the simulation is not influenced since the constant is dropped when deriving the potential to obtain the driving force.

Figure 4.2 illustrates that the potential acts repelling for $r_{ij} < r_0$, attracting for $r_{ij} > r_0$ and is zero for $r_{ij} = r_0$. It loses influence and tends to zero for r_{ij} being higher than 4, which has been used as maximum limit of the bond length, for details see Section 4.4.2.

Angle Potential

The angle or bending potential keeps the shape of the pentagonal and hexagonal structure by forces with respect to an angle. The following model has been used:

$$E_{\text{angle}}(\theta_{ijk}) = k_\theta(\cos(\theta_{ijk}) - \cos(\theta_0))^2. \tag{4.3}$$

θ_{ijk} is sketched in Figure 4.3. θ_0 is either 108° in pentagons or 120° in hexagons, and represents the equilibrium angle. Finally, $k_\theta = 5\,\mathrm{eV}$ is an energy constant.

Consider Figure 4.2. Similar to the binding potential, the angle potential has its energy minima in the equilibrium, so for angle values of $\pm\theta_0$. The force plot further reveals that the shape is rather kept in the equilibrium for sharper angles ($|\theta| < \theta_0$) than for obtuser angles ($|\theta| > \theta_0$).

Dihedral Angle Potential

The dihedral angle, or torsion potential, also cause forces with respect to an angle, just like the angle potential, as illustrated in Figure 4.3. The potential has the following form:

$$E_{\text{dihedral}}(\phi_{ijkl}) = k_\phi(1 - \cos(n\phi_{ijkl})). \tag{4.4}$$

$k_\phi = 0.173\,\mathrm{eV}$ is an energy constant and $n = 2$. The minima of the potential are at 0 and ±180°. For a quartet of atoms, belonging to one ring (pentagon or hexagon) in the fullerene molecule, $\phi_{ijkl} = 0$ means the ring is planar. Thus, the equilibrium shape possesses planar ring structures. Note that this is a weak potential compared to the other molecular potentials.

4.3.2 Pairwise Forces

In the force field, pairwise forces act between atoms, which are not directly connected by bonds. Forces between atom pairs that are farther away than a cut-off distance $r_{\max}$ are neglected. For example, in Figure 4.3 pairwise interactions are taking place between atom pairs $(a_i, a_k), (a_i, a_l), (a_j, a_l)$, but not between $(a_i, a_j), (a_j, a_k)$ and (a_k, a_l). Note that atoms do not need to be connected like they are in this example.

Coulomb Force

The Coulomb force is caused by electric charges. For the modeling of the experiment, it is a very important force because it is responsible for the decay of fullerene. The potential is given by

$$E_{\text{Coul}}(r_{ij}) = \frac{C q_i q_j}{\varepsilon r_{ij}},\ r_{ij} < r_{\max} \text{ with } C = \frac{1}{4\pi\varepsilon_0} = 14.399645\,\frac{\text{eVÅ}}{e^2}. \tag{4.5}$$

In this case, C is a unit conversion constant with ε_0 being the electric constant, q_i and q_j are the charges of the atoms with indices i and j in units of the charge e of one electron. Moreover, $\varepsilon = 1$ is a scaling factor and $r_{\max} = 30\,\text{Å}$ is the cut-off distance. The forces between atom pairs, which are farther away than $r_{\max}$ are neglected. The corresponding minimum force, being considered accordingly is $||\vec{F}(r_{\max})||_2 = 0.016\,\text{eV}\,\text{Å}^{-1}$ when assuming $q_i = q_j = 1\,e$.

Van der Waals Force

For van der Waals forces, the Lennard-Jones (LJ) potential is used that weakly stabilizes the distance between second neighbor atom pairs. It has the following form:

$$E_{\text{LJ}}(r_{ij}) = 4\epsilon\left(\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right). \tag{4.6}$$

$\epsilon = 0.0115\,\text{eV}$ is the depth of its minimum, $\sigma = 3.4681\,\text{Å}$ the minimum energy distance. It is so weak that its potential can hardly be seen in Figure 4.2, but it has a energy minimum at $3.889\,\text{Å}$ with $E_{\text{LJ}}(3.889\,\text{Å}) = -0.00287\,\text{eV}$. Note that it has a

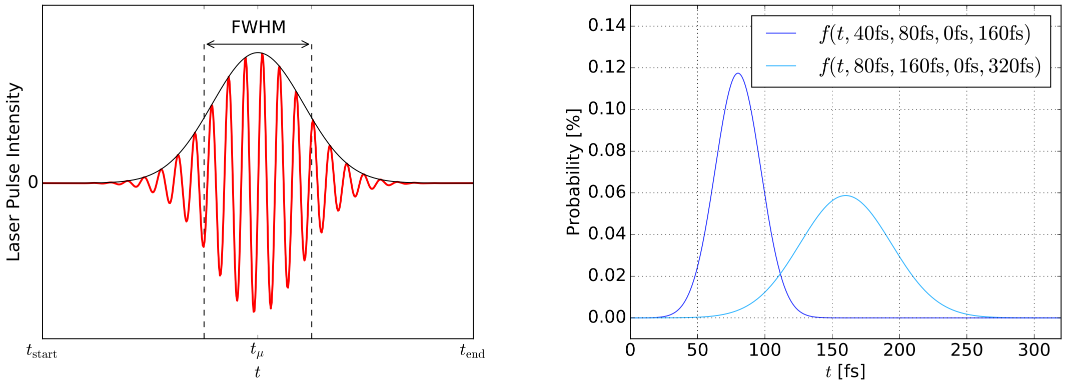


Figure 4.4: **Left:** Schematic illustration of a laser pulse (red) and its intensity distribution (black) with respect to the simulation time. **Right:** Plots of the truncated normal distribution for two different laser pulse lengths of 40 fs and 80 fs. Both distributions are normalized, such that a charge of $q_{\text{mean}} = 1$ in C_{60} is ionized on average.

strongly repelling character for distances smaller than 2 Å. The equilibrium bond length, however, is smaller. Thus, if the molecule is simulated with this potential, it is only stable because of the fact that pairwise atom interaction is disabled between atoms that share a bond. Nevertheless, this strong repulsion must be considered when breaking bonds due to charge, see Section 4.4.2. It induces very abrupt and strong repulsion, which we assume to be undesired.

4.4 Modeling of the Experiment

The concrete modeling of the experiment is described in this section. Most of the functionalities were implemented during this work. The modeling of the experiment includes the modeling of laser pulses and their effects in Section 4.4.1. The deviation of the results between simulation and experiment was fine-tuned by bond-breaking approaches, which are described in Section 4.4.2. There are also effects which have not been modeled, listed in Section 4.4.3.

4.4.1 Laser Pulses and Ionization

In the experiment, infrared femtosecond laser pulses are shot on fullerene molecules. The atoms partially absorb the energy of these laser beams, excites them into higher energy states. Once the beam is too intensive, it starts to ionize atoms in the molecule.

The problem with these processes is that the detailed effects are unknown. Note that the X-ray laser pulses of the experiment for recording the diffraction images are neglected (see Section 3.3.2).

Regarding our simulation, this simulation a laser beam is depicted in Figure 4.4 on the left and modeled in the following way. The laser beam is considered to hit the molecule in the time span $T_{\text{Laser}} = [t_{\text{start}}; t_{\text{end}}]$. During this time, the energy of the laser pulses is considered to be Gaussian distributed with a certain full width half maximum (FWHM) value, for details to this model see Demtröder [Dem13]. In the experiment, the FWHM value was equal to 40 fs. For Gaussian distributions, the FWHM has an equivalent standard deviation value [Wei]:

$$\text{FWHM} = 2\sqrt{2\ln(2)}\sigma \qquad \Leftrightarrow \sigma = \frac{\text{FWHM}}{2\sqrt{2\ln(2)}}. \tag{4.7}$$

It is assumed that the probability to ionize an atom in the molecule is proportional to the energy distribution. Here, the probability is assumed not to be depending on the position of the atom. In other words, the probability is equal for all atoms, but varies in time. This is because in the experiment, the area that is pumped (ionized) is larger than the scattering region for obtaining diffraction images (Section 3.3.1).

A distribution that fulfills these modeling requirements is the truncated normal distribution, e.g., see Johnson et al. [JKB94]. For $t \in T_{\text{Laser}}$, its probability density function is given in Eq. 4.8 being zero if $t \notin T_{\text{Laser}}$. Figure 4.4 shows the function for two different FWHM values:

$$f(t; \sigma, t_\mu, t_{\text{start}}, t_{\text{end}}) = \frac{1}{\sigma} \cdot \frac{\phi(\frac{t-t_\mu}{\sigma})}{\Phi(\frac{t_{\text{end}}-t_\mu}{\sigma}) - \Phi(\frac{t_{\text{start}}-t_\mu}{\sigma})} \tag{4.8}$$

$$\phi(x) = \frac{1}{2\pi\sqrt{2}} \exp\left(-\frac{1}{2}x^2\right) \tag{4.9}$$

$$\Phi(x) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right). \tag{4.10}$$

Here, $t_\mu = (t_{\text{end}} - t_{\text{start}})/2$ is the time of the laser pulse intensity maximum, $\phi(x)$ is the standard normal distribution, $\Phi(x)$ its cumulative distribution function, and $\operatorname{erf}(x)$ is the error function. A property of the truncated normal distribution is that it is a normal distribution ($p(T_{\text{Laser}}) = 1$). This makes it easy to control the mean of the number of electrons q_{mean} that is ionized in the complete C_{60} by one laser pulse.

In the simulation code, q_{mean} is used as a simulation parameter to regulate the intensity of the laser pulse. First, an uniform random number $x \in$unif$(0;1)$ between 0 and 1 is generated. An ionization event takes place if $x < f(t;\sigma,t_\mu,t_{\text{start}},t_{\text{end}}) \cdot q_{\text{mean}}$. The probability to ionize is truncated at $t_\mu \pm 2 \cdot \text{FWHM}$, so the ionization duration becomes $t_{\text{end}} - t_{\text{start}} = 4 \cdot \text{FWHM} = 160\,\text{fs}$.

The uniform random numbers were generated by the LAMMPS framework, which implements the Marsaglia random number generator [Mar95].

4.4.2 Bond Breaking

Bonds have a large contribution to the stability of molecules. There are some cases, however, in which it makes sense to break particular bonds. In this work, two classes have been considered. As mentioned in Section 4.3.1, the Morse potential is used to model bonds. Considering its force graph in Figure 4.2, one notices that for a pair distance of more than 4 Å the force is practically zero. Therefore, we break bonds if the distance of two atoms is longer than 4 Å. The other condition to break bonds considers the charge of atoms. If atoms in bond pairs are ionized, bonds will disappear due to a lack of electrons. Some concrete conditions to break bonds are presented in this section. Their behavior is tested in Section 6.2.2.

No-Charged-Bonds: Do not break bonds due to charge.

Individual-Bonds: For single covalent bond, break if the respective charges are at least $1\,e$. For double covalent bonds, break if both atoms have a charge of at least $1\,e$ and their sum is at least $4\,e$.

Single-Bonds: All bonds are treated as single covalent bond. Thus, we break if both atoms are charged by at least $1\,e$.

Immediately-1-Bond: First apply Individual-Bonds condition and break bonds. If an atom is ionized, make sure that at least one of its bonds is broken.

Immediately-2-Bonds: First apply Individual-Bonds condition and break bonds. If an atom is ionized, make sure that at least two of its bonds is broken.

We provide a discussion on how to break bonds for the best fit to the experimental data in Sections 6.3.1 and 6.3.2. It turned out that the approach Immediately-2-Bonds is the best condition. Bond-breaking due to distances was already provided by the LAMMPS framework and has been extended for break bonds due to charge.

4.4.3 Unconsidered Effects

Recall that there are two kind of lasers in the experiment. We model the the infrared laser by randomly ionize atoms of the molecule, which itself is an approximation. The other laser is the source for the X-ray laser pulses of the experiment for recording the image. The effect of the second laser can be neglected as good approximation because of its low influence.

In Murphy et al. [MOJ$^+$14] there have been considered two more effects, which due to their complexity have not been implemented in this work. The first one is a secondary ionization process. An electron of an atom that was hit by a laser pulse is also able to ionize other atoms. This process is depending on the energy of the electron. This effect turned out to have a strong impact on the simulation results.

The other one is called Auger effect. If electrons in lower shells are thrown out of an atom, the created hole will certainly be filled by an electron of an upper shell. During this process energy is emitted, which may be transfered to another electron, leaving the atom as well. The additional consideration of this effect further improved their results a little.

5 Visualization

Visualization is a powerful tool to analyze complex data and this their underlying processes. We start with a description of a direct visualization representation for molecular data. We provide an overview of existing techniques for molecular visualization in Section 5.2,leading to several techniques developed in this work, to depict the ionization and the fragmentation of fullerene, in Section 5.3.

5.1 Representation of a Molecule

To visualize molecules, we need a representation that on the one hand takes into account the data of the molecule, and on the other hand can be mapped to a renderable representation. A molecule is composed of atoms and bonds. The straightforward approach is to depict these directly as a graph, described in Section 5.1.1. The renderable representations are described in Section 5.1.2, together with possible information encodings in Section 5.1.3.

5.1.1 Modeling of a Molecule

Atoms can be represented as vertices $v_i \in V$ with an ID i. The number of atoms in the fullerene is 60. Each atom possesses some properties, thus we have vertex-based data. Since all atoms are of type carbon, we do not need to distinguish between different sorts of atoms. Each has a position $\vec{x}_i(t)$, $\vec{x}_i : \mathbb{R} \to \mathbb{R}^3$, which in molecular dynamics is depending on time t. The velocity $\vec{v}_i(t)$, describes the change of the position in time, whereas the change of the velocity is given by the acceleration $\vec{a}_i(t)$, $\vec{a}_i : \mathbb{R} \to \mathbb{R}^3$, which reflects the force $\vec{F}_i(t)$, $\vec{F}_i : \mathbb{R} \to \mathbb{R}^3$ in classical physics by Newton's law $\vec{F}_i(t) = m_i \cdot \vec{a}_i(t)$, where m_i is the mass of the atom. For the sake of simplicity, we neglect the time dependency in the notation if it is not needed.

Molecules are more than just sets of placed atoms with some properties. The stability of the structure depends on energies, described by potentials. Their derivatives are forces that accelerate atoms within the force field of a molecule. The main potential that contributes to the stability of the molecule is the binding potential, which models the bonds.

Bonds of molecules connect multiple atoms with each other. They are represented as edges, pairs of vertices $e_i \in E,\ e_i = \{v_j, v_k\}$. Here, E is the set of edges of the whole molecule. The fullerene possesses 90 bonds. There are also properties that belong to bonds. The most obvious property is the length of a bond, given by the Euclidean distance of the positions $\vec{x}_j$, $\vec{x}_k$ of its vertices v_j, v_k: $r_{ij} = ||\vec{x}_j - \vec{x}_k||_2$.

The above-mentioned potentials provide the energies. We distinct between kinetic energy E_{kin} that represents motion, and potential energy E_{pot} that is present inside the atom. The kinetic energy in classical physics of an atom is given as

$$E_{\text{kin},i} = \frac{1}{2} m_i ||\vec{v}_i||_2^2, \tag{5.1}$$

with the total kinetic energy being the sum of the kinetic energy of all atoms $E_{\text{kin},i}$

$$E_{\text{kin, tot}} = \sum_{v_i \in V} E_{\text{kin},i} = \frac{1}{2} m_{\text{C}} ||\vec{v}_i||_2^2 \tag{5.2}$$

with $m_{\text{C}} = 12.011\,\text{u}$ being the mass of one carbon atom.

The potential energy is composed of all potentials used in the model (see Section 4.3). Therefore, we consider the following sets. The set of bonds contains vertex pairs $\mathcal{V}_{\text{bond}} \subseteq V \times V$. For angular potentials we, need to consider the set of triplets of vertices $(v_i, v_j, v_k) \in \mathcal{V}_{\text{angle}}$ and their angle $\theta_{ijk} \in \Theta$, so in total $\mathcal{V}_{\text{angle}} \subseteq V^3,\ \Theta \subset \mathbb{R}$. Finally, dihedral potentials are given by the set of quartets of vertices $(v_i, v_j, v_k, v_l) \in \mathcal{V}_{\text{dihedral}}$ and their dihedral angle $\phi_{ijkl} \in \Phi$ with $\mathcal{V}_{\text{dihedral}} \subseteq V^4,\ \Phi \subset \mathbb{R}$. These sets belong to the molecular forces. There additionally are the pairwise potentials such as the Coulomb and Lennard-Jones potentials $E_{\text{Coul}}(r_{ij}),\ E_{\text{LJ}}(r_{ij})$. They act between pairs of atoms that are not directly connected by bonds. The total potential energy is then given as the sum of all potentials

$$\begin{aligned} E_{\text{pot, tot}} = &\sum_{(v_i,v_j) \in \mathcal{V}_{\text{bond}}} E_{\text{Morse}}(||\vec{x}_i - \vec{x}_j||_2) + \sum_{(v_i,v_j,v_k) \in \mathcal{V}_{\text{angle}}, \theta_{ijk} \in \Theta} E_{\text{angle}}(\theta_{ijk}) \\ + &\sum_{(v_i,v_j,v_k,v_l) \in \mathcal{V}_{\text{dihedral}}, \phi_{ijkl} \in \Phi} E_{\text{dihedral}}(\phi_{ijkl}) + \sum_{(v_i,v_j) \in V^2 \setminus \mathcal{V}_{\text{bond}}} E_{\text{Coul}}(||\vec{x}_i - \vec{x}_j||_2) \\ + &\sum_{(v_i,v_j) \in V^2 \setminus \mathcal{V}_{\text{bond}}} E_{\text{LJ}}(||\vec{x}_i - \vec{x}_j||_2). \end{aligned} \tag{5.3}$$

The sets of vertices and edges build a graph $G = (V, E)$. A molecule is considered to be an interconnected graph. This means that there is always a path P that connects two arbitrary but distinct vertices $v_i, v_j \in V,\ v_i \neq v_j$ with each other. A path is a subset of the set of edges of the molecule $P \subseteq E$. The connected components algorithm [HT73] is a common method to obtain such a component.

5.1.2 Renderable Representations

Vertex Representations

For the direct visualization, we have to assign a renderable representation for vertices. The simplest choice would be to assign them to pixels in the rendering image plane. This might be easy and fast, but visually not aesthetic and hard to perceive. Other rendering primitives are textures, e.g., textures of spheres. This approach is also fast to render and is visually more preferable. Nevertheless, this primitive still lacks spatial perception. The best and probably also the most popular primitive is a sphere as glyph representation for depicting atoms also use in our approaches.

Edge Representation

Edges connect vertices as a representation of bonds between atoms. As a visual representation, a pixel-based line could be chosen as a simple approach. It would have the same advantages and disadvantages as the pixel representation, discussed above. A more appropriate primitive is the cylinder. Shading effects give us clues in understanding shapes. Related to cylinders tubes are another representation. They basically look like cylinders, but they can be curved. An additional feature is that they are able to vary their radius at distinct segments of it. In our approaches, we primary make use of tubes.

5.1.3 Information Encoding

Each technique discussed below must be able to illustrate some information in a certain way. In the following, we provide an overview how information can be encoded, often in combination with renderable representations.

Domain

Let us start with the domain. Many techniques use the spatial range as the domain for intuitive representation. Positions of the considered data can directly be represented. As long as the spatial data is of interest, this information encoding is often preferred. The spatial mapping of other properties, such as time, leads to more abstract models. One must be aware of occluding effects, however. Therefore, the adequateness of such mappings depends on the sparsity of the data.

Colors

A popular approach to encode information is to use colors. They can be used to represent data values within a given data range by a specified color map. A color map $c : X \to [0; 1]^3$ maps a data value $x \in X$ of a data range $X = [x_{\text{low}}; x_{\text{high}}]$ to a RGB value. Each color channel here is considered to be between 0 and 1. Color maps have to define (RGB) values for the starting value x_{low} and analogously also an ending color for the value x_{high}. For values within the data range $x_{\text{low}} \leq x \leq x_{\text{high}}$ the color is interpolated linearly.

One has to take into account that similar color can be difficult to distinguish by the human eye. For that reason, color maps often include additional colors within the data range to increase the number of distinguishable colors and their respective data values. One special sort of color maps is the diverging one. They are designed such that the region in the middle of the color map represents some kind of neutral or equilibrium state in the data. Usually, the color of this state is shown white.

Figure 5.2 gives an example. The color map for the charge q has been chosen such that each possible charge has its non-mixed color when considering a maximum charge of $3\,e$ (e is the elementary charge). For the bond length, a diverging color map was used in oder to show the equilibrium distance.

The color mapping can of course be applied to lines, surfaces, volumes, but also to renderable primitives. This way, color maps help us to encode additional information.

Transparency and Opacity

Opacity is often used together with color. Especially in complex scenes, events may be missed due to occlusion. In addition to color, opacity can help visualize hidden objects. The transparency value α defines the fraction of the light that passes through an artificial volume. The opposite of transparency is opacity β, that can be derived by $\beta = 1 - \alpha$. Opaque objects ($\beta = 1$) will occlude the objects behind them. Transparency has been used, for example, in Figure 5.3. Due to the amount and length of the trajectories occlusions occur without the usage of transparency. The latest time steps can be emphasized by a higher opacity while former steps are faded out by increased transparency.

Connectivity

Sometimes it is desired to connect certain objects. In the simplest example, scattered points in a diagram can be connected by lines. Since human perception is focusing on contrast, continuous representations are often preferred over discrete ones. We have already discussed topological connection in Section 5.1.1. The topology of molecules is discussed detailed in Sections 5.3.3 and 5.3.6.

Degrees of Freedom of Renderable Primitives

In Section 5.1.2, we already learned about renderable primitives. Now, we have a look at the degrees of freedom that are available to encode information into their visualization. Generally, each representation discussed above can be color-coded.

For vertex representations, we introduced plain pixels, sphere textures and glyphs. Except for the case of pixels, the radius can also be varied. However, varying the size of, e.g., spheres clashes with perspective projection (the farther away, the smaller their projections are), which could interfere without additional information.

The introduced edge representations are lines, cylinders, and tubes. Unlike vertices, which are points, edges and their representatives have an extent. This extend implicitly serves as a rough distance information.

The width of edges is one degree of freedom that can be used to map data on their representation, since edges are 1-simplices of dimension one. Possible data that can be mapped onto edges are the two values of their vertices. Considering two vertices v_i, v_j one of their properties can be linearly interpolated along the edge. Additionaly, tubes offer the possibility to vary their radius.

5.2 Molecular Visualization Taxonomy

In a recent state of the art report [KKF$^+$16], a taxonomy of biomolecular structure visualization was given. We briefly discuss this taxonomy in this section, to provide

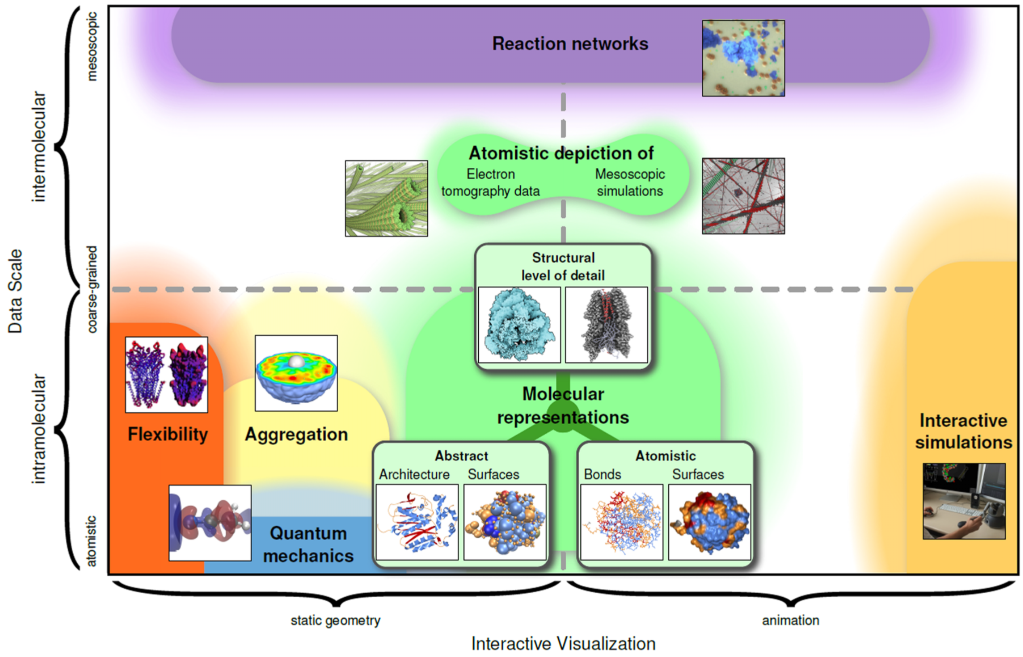


Figure 5.1: Illustration of the taxonomy of visualization techniques, taken from [KKF+16]. The x-axis shows the alterability of the method, the y-axis the molecule data scale, increasing to the right and top, respectively.

an overview on existing visualization techniques. The focus is on the techniques relevant for the topic of this thesis.

Figure 5.1 provides an illustration of Kozlíková et al.'s taxonomy. The x-axis shows the type of the visualization technique with respect to its alterability. Static geometry techniques, like atom trajectories, would not alter its geometry for a given time interval. In contrast, animation techniques like the direct depiction of atoms may show atoms at completely different positions for different times in the same time interval, which the atom trajectories depicted. Vertically, the visualization methods are ordered by the amount and complexity of molecular data. The data scale can further be subdivided. The intramolecular scale ranges from the atomic up to the coarse-grained molecular representations. Intermolecular visualization techniques range up to the mesoscopic data regime where molecules are already considered to be the smallest unit.

In the following, the visualization categories are described as detailed as relevant for this work. Concrete realizations belonging to the respective categories are also listed in the related work, Chapter 2.

5.2.1 Molecular Representations

The most important data sources for visualization techniques that directly depict atoms and molecules are experimental measurements (Figure 5.1, atomistic depiction) and simulations (Figure 5.1, molecular representation). Since this work relies on simulations, let us focus on the illustration of simulation data.

Molecular representations can be divided into three subcategories. Atomistic representations directly depict atoms, either by depicting atoms and bonds explicitly (bond-centric representations) or as a surface of the whole molecule. The simplest bond-centric representations are the *line* and the *stick* representation, which just show the bonds without atoms, usually as simple lines or as a cylindric surface, respectively. The *ball-and-stick* representation additionally depicts atoms (usually as spheres) and is one of the most often used bond-centric representations. Surface representations can take advantage of *kernel filter convolutions* to also show quantities of occluded atoms. A radial kernel is used as a weighting function for a property of the surrounding atoms on the surface of the molecule which is convoluted on the whole molecular surface.

Illustrating large macro molecules that consist of millions of atoms is a challenging field (Figure 5.1, structural level of detail). The main challenge is the rendering of their surface at interactive rates. Other issues are overcoming occlusion effects and the indication of depth on coarse molecular surfaces.

Abstract representations also have two subcategories. In large molecules, certain structures may appear multiple times. These substructures can alternatively be shown as own graphical appearances, e.g., straight cylinders for helices. Abstract surface representations map a measure on a surface. These representations can be used in combination with kernel filter convolutions, mentioned above. Another approach is to map quantities on a bounding sphere.

These direct representations are a quite simple, but also intuitive to understand. They are useful to illustrate processes with low dynamics, where the components are located compactly and the number of vertices is not too high. They are also used in this work.

5.2.2 Flexibility

The category flexibility (Figure 5.1) points out molecular dynamics visualizations that go beyond showing the geometry at different times. The category covers trajectories, graphs, and uncertainty illustration techniques. They can make use of aggregation techniques to point out properties of groups of molecules, which are considered next.

Methods of this category are very useful to show motions. Later, we present two techniques of this category that significantly help realize the motion of fragments.

5.2.3 Aggregation

Aggregation or grouping can be used to reduce the dimensionality of data. An important fact for temporal aggregation is that the center of mass is usually not moving or can be centered by changing the frame of reference. The grouping of atoms or molecules by their properties is also possible. Sometimes, it may be desired to use visualization techniques from different fields. They usually need a continuous volume representation, which can also be extracted from the scattered point data as shown in Section 5.2.4.

5.2.4 Volumetric Representations

Instead of analyzing the point-wise properties that may be saved in atoms or molecules, it is also possible to convert information into a continuous volume representation. To convert unstructured point representation, the nodes can be filled by the sum of the properties of atoms, weighted, and merged by radial basis functions within a certain radius around the nodes, into a continuous representation. The data can then also be analyzed by visualization techniques designed for other fields. This includes direct visualization techniques like volume rendering for density representations, or techniques used for representing motion like arrows or streamlines.

Our molecule is rather small, so volume reconstruction might result in little information. Nevertheless, the volume representation becomes dense and increases in size. We therefore avoided the use of volumetric representations in this work.

5.2.5 Reaction Networks

The illustration of chemical reactions and large systems of them is a rather specific field. However, the techniques of this category take advantage of more general techniques of other fields. Directed graph representations are used to show which reactants are combined to form new products. Animation techniques often use molecular representations in combination with certain kinds of highlighting or focusing

Section	Technique	Category	Visualization Type
5.3.1	Glyph (Spheres)	Molecular, Atomistic	Animation
5.3.1	Ball-and-Stick	Molecular, Bond-Centric	Animation
5.3.2	Atom Trajectory	Flexibility	Static
5.3.3	Fragment Tree	Flexibility / Reaction	Static
5.3.5	Diffraction Image	Aggregation	Static

Table 5.1: Categorization of the realized visualization techniques according to the taxonomy of [KKF$^+$16] that is illustrated in Figure 5.1.

on ongoing reactions, which may be easily overseen in huge and dense crowds of molecules. Opacity of the most important reactants may further emphasize processes.

At first glance, the category of reaction networks may seem rather unrelated to the topic of this work since the data scale is completely different and there are no molecular reactions. Actually, there are many similar problems between the decay of fullerene and reaction networks. The decay of C_{60} can be seen as a special case of molecular reaction, where C_{60} is the only reactant and the products are many distinct fragments of C_{60}. Another challenge in visualizing chemical reactions is that there is an individual yield of reactants and products, which additionally varies over time. In this sense, the decay of fullerenes comes with the same challenges.

5.2.6 Quantum Mechanics

Studying certain molecular processes requires a detailed understanding of quantum effects. The main focus in quantum mechanical visualization is the illustration of orbitals. Nevertheless, the simulation in this work cannot provide any quantum mechanical information, so we close the discussion of the visualization categories and start with the molecule representation in the following section.

5.3 Realized Techniques

After the overview, given in Section 5.2 this section presents the realized visualization techniques that have been developed during this work. The techniques are classified in Table 5.1 according to the described taxonomy of Section 5.2. Further techniques are the illustration of laser pulses (Section 5.3.4) and an abstract stability measure called fragment holes (Section 5.3.6). In the end of each respective technique, advantages and disadvantages are discussed in order to find suiting visualization applications.

A key role plays the behavior of the molecule. The behavior can be roughly classified into three energy regimes. In the low energy regime the molecule remains stable and starts oscillating. In the medium energy regime the oscillations of the molecule becomes stronger and it can happen that the molecule is tearing into fragments. Last but not least, in the high energy regime the molecule certainly decays within a Coulomb explosion causing high velocities of the fragments. They need to be considered to find an adequate representation of the processes of the fragmentation dynamics of fullerene.

In this work we use ParaView [AGL05], [Aya15] as a visualization toolkit to illustrates our results. This toolkit already provides many functionalities and prevented us from necessary work load. The visualization techniques work as a submodule (fix, see Section 4.2) of the LAMMPS framework. This enables us to develop in situ visualization techniques, which we also make use of. In situ visualizations tackle the problem of processing, e.g., simulation data while the simulation is running to produce visualization data. This is a whole field of research [RCMS12]. Generally, the visualization techniques developed during this work are in situ techniques, exporting files for visualization, whereas our analysis techniques are working on these representations. While the analysis techniques are shown in the result section, the visualization techniques are presented in the following.

5.3.1 Atom and Bond Representation

Let us start with a direct representation of the molecule. In a direct representation, the elements for the visualization that were introduced in Section 5.1 are directly depicted. These elements are the vertices and the edges. The visualization domain is chosen to be the domain of the positions of the vertices. Like atoms, which are placed in space, the vertices are also placed in space as their representatives.

Edges connect vertices as a representation of bonds between atoms. As a visual representation, a pixel-based line could be chosen as a simple approach. This is considered to be the *lines* representation, which, like the pixel representation before, is neither visually aesthetic nor provides spatial perception. For a more appropriate representation, cylinders leads to the *licorice* or *stick* representation.

The representation depicting atoms and bonds concurrently is called *ball-and-stick* representation. The preferred primitives are sphere glyphs for atoms and cylinders for bonds. In Figure 5.2, all these representations are illustrated. Note that depicting bonds as shaded sticks gives us hints about the structure of C_{60}.

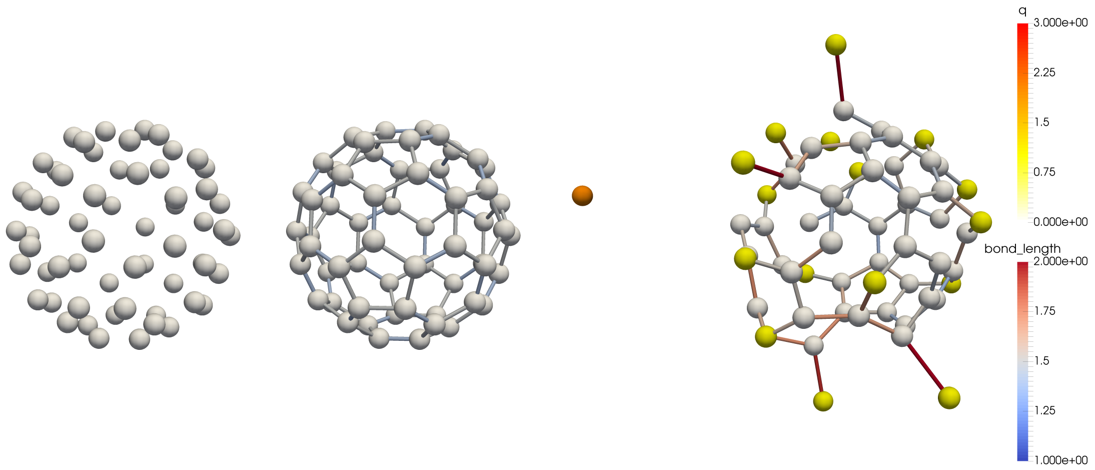


Figure 5.2: C_{60} as ball representation (left) and as ball-and-stick representation (middle, right). Preferred color indicators are the charge (q in elementary charge units) to indicate the repelling Coulomb forces of the atoms. Bonds may be colored by their lengths (in Å) for a better prediction of potential bond-breaking due to too high distance.

Color encoding can give us further insights into the process. For atoms, the preferred shown property is the charge of the atom. Indicating the charge informs us about the Coulomb forces, one of the most driving forces in simulation and experiment. It tells us which atoms repel each other and how strongly. For example, in Figure 5.2 on the right, the orange-colored single atom was obviously repelled because of its high charge.

An appropriate coloring for bonds is to depict their lengths. Again, consider the right image in Figure 5.2. The red indicated lines can potentially break within the next time steps.

A drawback of this representation is that it is not able to encode time information. Humans have problems to remind detailed information about images within an animation. An example for such a case is given in Figure 5.3. Considering the image in the top right, the atoms themselves cannot give any information about former time steps. Therefore, we want to encode hints about the former time steps. A representation that tackles this problem are atom trajectories, that also have been used in this image. Their description follows next.

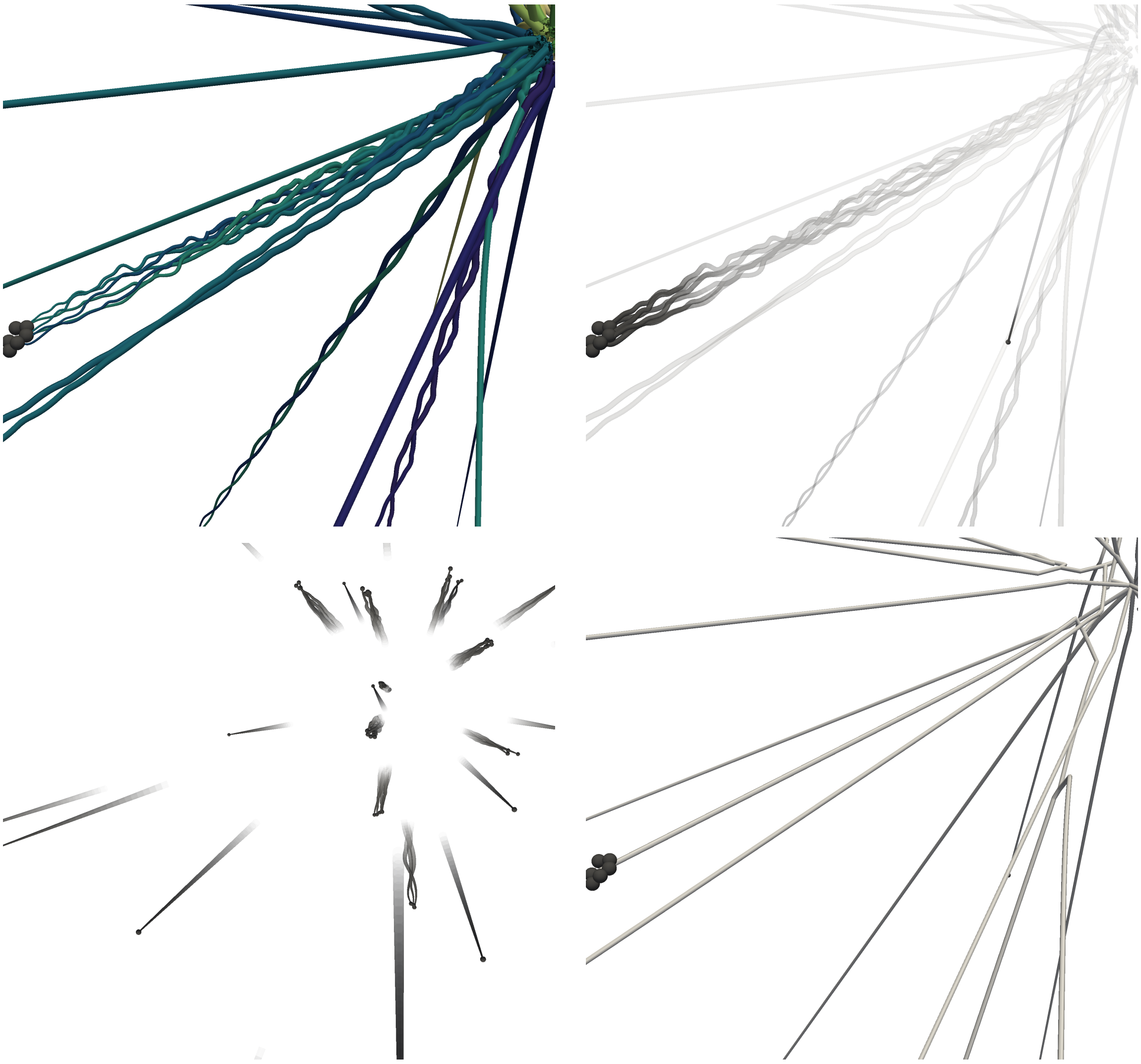


Figure 5.3: Examples for atom trajectories in comparison with the fragment tree. **Top Left:** The trajectories are colored by atom IDs and their tube radius scaled inversely with time. **Top Right:** A sparser trajectory representation with opacity proportional to time. **Bottom Left:** A zoomed out view in which not the full length of the trajectories are shown. **Bottom Right:** The fragment tree as alternative representation that tracks the center of mass of each fragment, see Section 5.3.3.

5.3.2 Atom Trajectory

An atom trajectory depicts the position of an atom over time. The atom trajectories of a molecule contain all positions $\vec{x}_i(t)$ of their vertices v_i at time interval $T \subseteq \mathbb{R}$.

Since our simulation has to do a time discretization into time steps, and the representation is usually not given as function of time, but as a data structure, we have to adapt the representation. The time discretization is done by defining $T_m = \{t_1, t_2, ..., t_m\} \subseteq \mathbb{R},\ t_k < t_{k+1}\ \forall\ k = 1, ..., m-1$ with $m = |T_m|$ being the number of time steps. The discretized trajectory of all vertices $\mathcal{X} \subseteq V \times V \times T_m$ can be given as set of tuples, containing the vertex pairs to connect with a certain time.

The connections between vertices in the trajectory are again achieved by edges. We consider the position of each vertex $\vec{x}_i(t_k),\ i = 1, ..., n$ at time step $t_k \in T_m$. The edge pairs that are built simply contain the two positions $\vec{x}_i(t_k)$ and $\vec{x}_i(t_{k+1})$ with $t_k, t_{k+1} \in T_m$. Note that for depictions of a time step $t_k \in T_m$, we usually consider only trajectories with the time discretization $T_k = \{t_1, t_2, ..., t_k\}$. In other words, only former time steps are shown.

From these mathematical fundamentals, we come back to the visualization aspect. For the depiction of the trajectories, tubes are preferably used. Figure 5.3 shows some example visualizations for atom trajectories. Note the helix-like trajectories, which indicate rotations around the axis nearly parallel to the direction of moving fragments. In the upper left image, trajectories are colored by the atom ID of the corresponding atom. This indication further emphasizes the helix-like motion. Information about the time is encoded as varying radius of the tubes. Note that the method has no real use when considering low energy regimes, i.e., when the fullerene does not break up. The trajectories only oscillate around their starting position in these cases, and are rather in the way instead of providing information.

In the top right of Figure 5.3, the trajectories are shown in a semi-transparent view. The transparency map has been chosen such that the latest time steps are shown more opaque then the previous ones. Since the map depends on time, the length of the opaque tail of the atoms also gives hints about the velocity of the atoms. This is nicely illustrated in the lower-left corner. The original center of the molecule at the top right in the image cannot only be recognized by the orientation of the trajectories. Fragments that are farther apart also have longer trajectories. This reminds of vector fields that are often represented by length-scaled arrows.

To conclude this section, let us reflect the features of this techniques. The method shows the trajectories of each atom. It was motivated by a lack of temporal information in direct representations. Adding spatial information for multiple time steps makes the method belonging to the category flexibility (Figure 5.1). Such methods are considered to be static methods. Preliminary results during time integration can also emphasize the velocities and directions of atoms.

Recall that the method has no practical use without fragmentation. Therefore, the method is rather suited for medium up to high energy regimes. Another problem is that overlaps without the use of transparency might be confusing. To motivate the next technique, let us consider Figure 5.3 on the bottom right. This technique provides a sparse representation of the atom trajectories by means of centers of mass of the fragments. On the one hand, the information about rotational behavior is lost, but on the other hand it can depict the fragmentation by branching trajectories.

5.3.3 Fragment Tree

An interesting part of the decay of fullerene are its fragments and their motion. The atom bond representation is able to show that parts are no longer connected to the original molecule and the atom trajectories only illustrated the movement of individual atoms without any fragmentation information. The method introduced here is called *fragment tree*. It combines these two features by showing the trajectories of the center of mass of each fragment.

Mathematical Formulation

First of all, let us define what the frequently used word fragment means. In simple words, a fragment of C_{60}, in short just fragment, is a part of the original molecule where its carbon atoms are interconnected by their chemical bonds. We are speaking of equal fragments if two fragments have the same number of (carbon) atoms.

We can provide a more precise definition by using the visual representation. Let $G_{\mathrm{C}_{60}} = (V_{\mathrm{C}_{60}}, E_{\mathrm{C}_{60}})$ be the graph of the C_{60} molecule. A fragment, given by its graph $G = (V, E)$ must necessarily fulfill the following conditions:

$$V \subseteq V_{\mathrm{C}_{60}} \qquad\qquad E \subseteq E_{\mathrm{C}_{60}}. \tag{5.4}$$

Moreover, all of its vertices $v_i \in V$ must be interconnected by its edges $e_j \in E$. The properties of vertices (including the position) and edges are explicitly excluded from these requirements. Two fragments G_1, G_2 are referred to be equal iff $|V_1| = |V_2|$.

Note that by this definition, a complete C_{60} molecule can also be considered as fragment. The most degenerate case of a fragment is a single carbon atom.

Next, we describe what a fragment tree is and what it shows. An illustration of the following description is given in Figure 5.4. We consider the graph $G_{C_{60}}$ of the C_{60} molecule as initial state of our simulation and visualization $G_0^{(t_0)} = G_{C_{60}}$, where the graph in time step t_k is denoted as $G_i^{(t_k)} = (V_i^{(t_k)}, E_i^{(t_k)})$, with an assigned fragment ID i. The start of the shown fragment tree trajectory $\vec{x}_0^{(t_0)}$ is located at the center of mass of the initial graph. Generally, the center of mass of a fragment with ID i is given in as:

$$\vec{x}_i^{(t_k)} = \frac{1}{|V_i^{(t_k)}|} \sum_{v_j \in V_i^{(t_k)}} \vec{x}_j. \tag{5.5}$$

The trajectory connects these positions by edges $e_i^{(t_k)} = (\vec{x}_i^{(t_k)}, \vec{x}_i^{(t_{k+1})})$ between two time steps t_k, t_{k+1}.

During the simulation, the graph changes. In most cases, the properties of vertices and bonds are affected, especially due to changes in position. The breaking of bonds in between two time steps $t_k,\ t_{k+1}$ causes a graph $G_i^{(t_k)}$ to lose edges. Note that the simulation is set up such that atoms and thus also vertices are never lost. In a case where an edge in $G_i^{(t_k)}$ is lost, we check its connectivity.

Data Structure and Implementation

If the graph has seperated into subgraphs, each of them must be saved in a data structure. A tree structure comes in handy here. We denote this tree as fragment tree. Each node of the tree saves one fragment in its node, referred to as `FragmentNode`. This `FragmentNode` stores all properties of the fragment, including its graph. Has a split been detected in time step t_k, the `FragmentNode` of the broken graph $G_i^{(t_k)}$ adds each separated fragment with graphs $G_{i+1}^{(t_k)}, ..., G_{i+p}^{(t_k)}$ as new `FragmentNode`. The parent node with graph $G_i^{(t_k)}$ is replaced by the graphs of the children $G_{i+1}^{(t_k)}, ..., G_{i+p}^{(t_k)}$. We have to connect the latest position of the last time step $\vec{x}_i^{(t_{k-1})}$ with each center of mass $\vec{x}_{i+1}^{(t_k)}$, ..., $\vec{x}_{i+p}^{(t_k)}$ of the new fragments by edges $e_j = (\vec{x}_i^{(t_{k-1})}, \vec{x}_{i+1}^{(t_k)}), ..., e_{j+p-1} = (\vec{x}_i^{(t_{k-1})}, \vec{x}_{i+p}^{(t_k)})$. This process is repeated until the simulation ends. The whole procedure for updating the fragment tree data structure is given in Algorithm 1.

The simulation runs for a certain number of time steps. We frequently write visualization and fragment data files, produced by calling `updateFragmentTree()`.

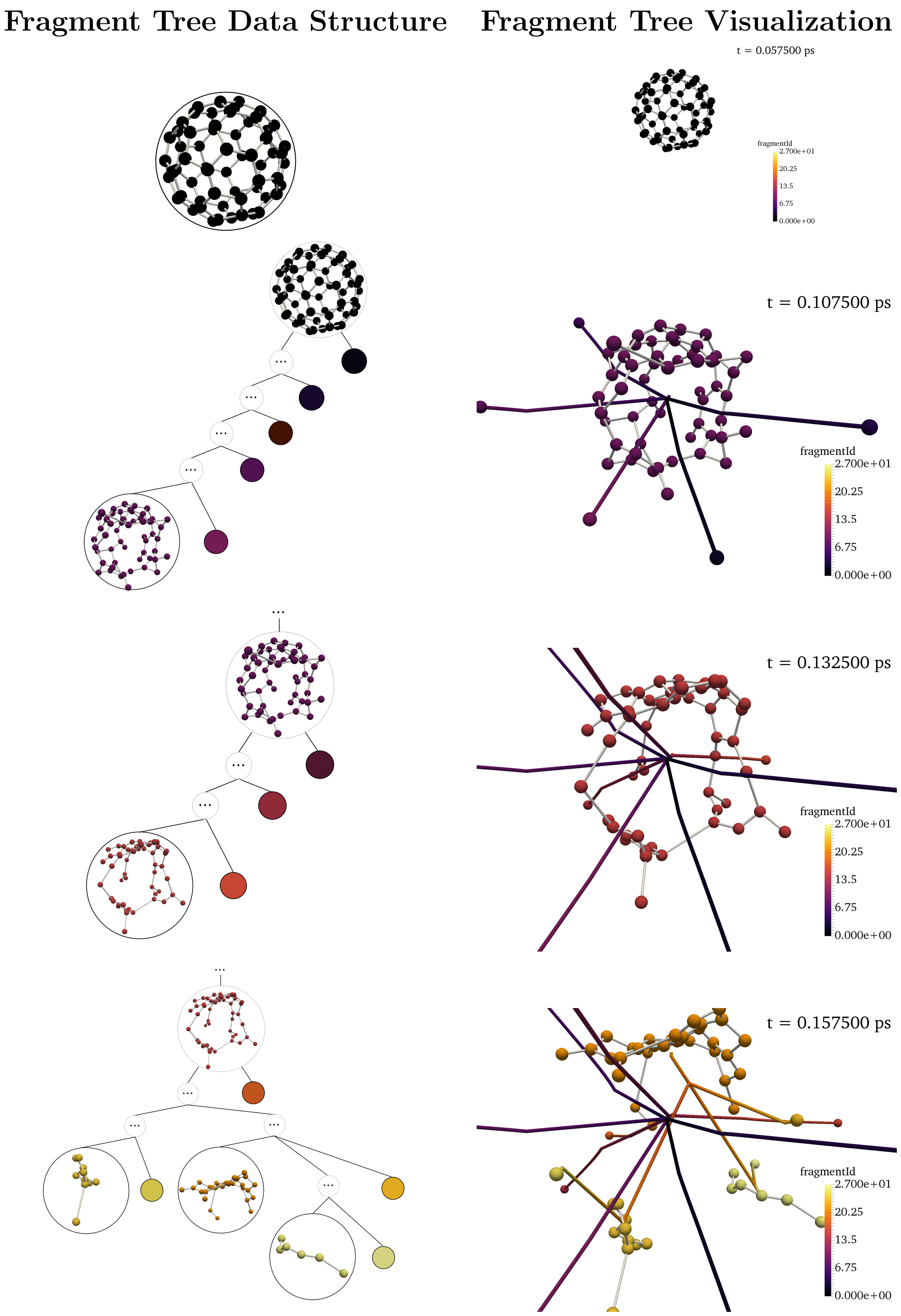


Figure 5.4: **Left:** The fragment tree as tree data structure. The enframed fragment graphs are its nodes. They are black if they are leaf nodes, gray otherwise. Three dots within a frame mark placeholders for temporary fragments. Some parent nodes in the third and fourth row were left out. Single colored circles represent single atoms. **Right:** The fragment tree as visualization technique. The tubed lines indicate the trajectories of whole fragments. Atoms and the fragment tree are colored by fragment ID.

Algorithm 1 Algorithm to update the fragment tree data structure.

```
Input: The fragment tree's root FragmentNode root, a FragmentNode fn
procedure onSimulationUpdate()                      ▷ Called after simulation time steps.
    updateFragmentTree(root)
end procedure

procedure updateFragmentTree(fn)                    ▷ Update whole fragment tree.
    if fn is a leaf node then                       ▷ Leaf nodes are fragments in the current step.
        updateFragmentNode(fn)
    else                                            ▷ Non-leaf nodes have already decayed.
        for each childNode of fn do
            updateFragmentTree(childNode)           ▷ Search for further leafs.
        end for
    end if
end procedure

procedure updateFragmentNode(fn)                    ▷ Update a single FragmentNode.
    Update the FragmentGraph of fn depending on the simulation results.
    if bond count in fn did not change then
        return                                      ▷ No topological change, stop.
    end if
    Get a set $\mathcal{G}$ of connected component graphs $G \in \mathcal{G},\ G \subseteq G_{\mathrm{C}_{60}}$ for fn.
    if $|\mathcal{G}| < 2$ then
        updatePolygons(fn)                          ▷ Update for hole detection, see Algorithm 2.
        return                                      ▷ Stop since the fragment fn did not split.
    end if
    for each $G \in \mathcal{G}$ do
        Initialize new FragmentNode fnc with $G$.
        Copy Polygon map and edge list from fn into fnc.
        updatePolygons(fnc)                         ▷ Update for hole detection, see Algorithm 2.
        Add fnc as a child node to fn.
    end for
end procedure
```

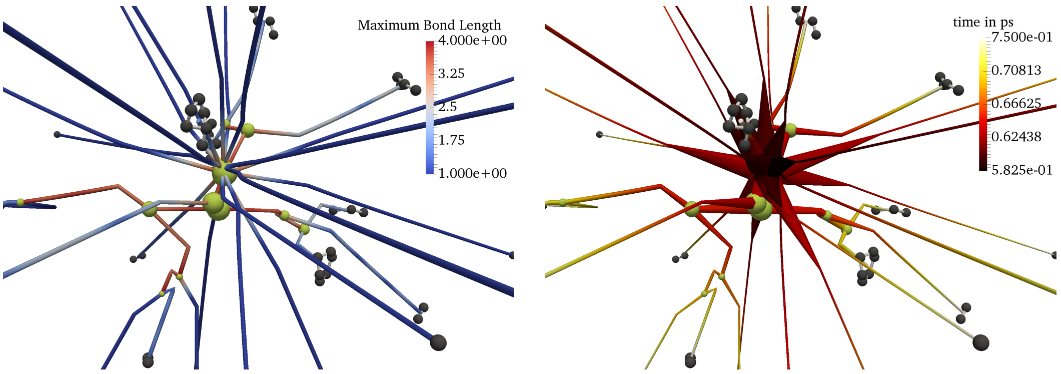


Figure 5.5: **Left:** Fragment tree with focus on fragmentation. A split of a fragment is indicated by a yellow-greenish colored sphere. Its size is proportional to the radius of a bounding sphere of the fragment. **Right:** Same fragment tree, colored by simulation time. The time ranges from the time of the first decay until the currently shown simulation step, depicted by atoms. The bond length is given in Å.

The root of the fragment tree, which is also a `FragmentNode`, is used as input to this method, and afterwards it calls `updateFragmentNode()` on each leaf node of the data structure. Leaf nodes correspond to the fragments that have not decayed, yet. On the left of Figure 5.4, they are shown by black frames. The method `updateFragmentNode()` basically manages the split of a fragment by creating new `FragmentNodes` and updating the properties of each changed fragment. For the split detection, the connected component algorithm of the boost library [Sch11] is used.

Examples

Let us focus on the usage of the fragment tree by considering some examples. In the left image of Figure 5.5, the focus is set on showing the fragmentation of the molecule. So far, splits of the molecule have been shown indirectly with direct visualization techniques that hided covalent bonds, or by atom trajectories that were drifting apart. The fragment tree is able to show which parts of the molecule once were connected and at which position they have separated. Such separations are indicated by the yellow-greenish spheres. They show the position at which a fragment was connected by bonds the last time until it split. Often, the connection from this point to the new center of masses of the individual new fragments is orthogonal to the direction of the velocity. Except for low fragment velocities (for example, at the origin of the molecule) this also makes a split visible even without the spheres as indicators. As an additional split indicator, the fragment tree trajectories are colored by the maximum bond length in a fragment. Thus, red indicates approaching splits since bonds are broken for distances greater than 4 Å (recall Section 4.4.2).

The fact that each fragment has the same origin also gives us hints about the time of the split. The nearer the position of the fragment to the origin, the earlier in the simulation the split happens. The time-position ratio is depending on the velocity. In Section 3.3.3, we learned that the time-of-flight of a fragment, accelerated in an electric field depends on its mass and charge. This principle can be at least qualitatively applied on the Coulomb force acceleration. The higher the charge and the smaller the mass, the higher is the resulting acceleration and thus also the velocity. In the right image of Figure 5.5, we can see basically the same fragment tree as in the left. This time, the coloring of the fragment tree is changed to the simulation time. Furthermore, the radius of the tubes of the fragment tree is varied by the number of atoms of a fragment. Note that slim trajectories mostly have a single color. They correspond to ionized atoms. We can conclude that these ionized atoms have a high velocity with respect to other, larger fragments.

Introducing the variation of the radii causes the appearance of spikes from the origin. They are especially visible for small radii. They come from the linear interpolation between two points on the fragment tree trajectories. The points on the trajectories are sampled every n time steps. If an atom moves very fast, the distance between two points becomes large. Due to the abruptly changing radius of a fragment, the radius is interpolated on a clearly visible distance which appears as spikes. The only way to avoid them is to sample finer in time. In this and almost all other examples, a sampling rate of 2.5 fs was used.

In the former data set a single and short laser pulse caused the fragmentation of the molecule. As a consequence, most fragmentations at the origin of the molecule are due to the charge whereas apart from the origin the fragmentation is caused by bond-breaking due to the bond distance. In the next fragment tree example in Figure 5.6, the fragment tree is applied to a data set in which the molecule is hit by multiple laser pulses at a time. In such cases, it might be desired to see the ionization events on the trajectories of the fragment tree. We introduce an indicator for ionization events by showing a purple sphere. Its radius is scaled by the changed charge of a fragment per ionization event. The usage of additional colors is avoided, since the changes usually do not reach high values as well as additional color bars may

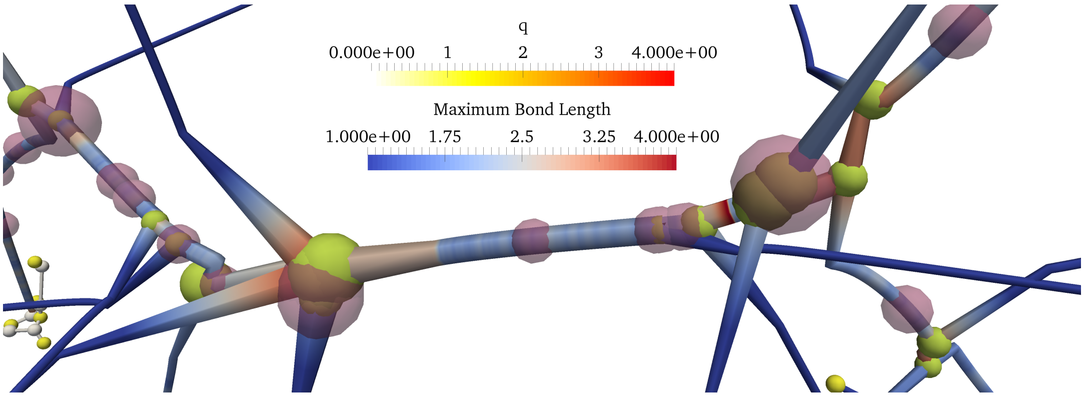


Figure 5.6: Fragment tree applied on a data set with multiple laser pulses. In such data sets ionization events are spatially spread and are indicated by purple spheres. Due to collisions with yellow-greenish split events they are shown semi-transparent and further are scaled by the changed charge. q is given in elementary charge units, bond length in Å.

interfere with each other. Ionization events are shown semi-transparent since they often collide with fragmentation events (yellow-greenish spheres). A collision of both events can be interpreted as immediate consequence of the ionization. Frequently, the fragmentation happens after a short time.

The atom trajectories are able to show rotational dynamics of decaying fragments, e.g., see the top left image of Figure 5.3. This is not possible with the fragment tree data structure. However, it is able do show some other dynamics, namely oscillations within fragments. Figure 5.7 illustrates two variants of fragment trees, focusing on showing the dynamics of fragments. Since both look quite familiar, let us start by considering what both trees have in common. Like in Figure 5.6, the fragment tree is colored by the maximum bond length in a fragment, however within a shorter range. On the one hand, approaching fragmentations are no longer clearly distinguishable from medium sized-bonds. On the other hand, focusing on this range has the advantage of emphasizing the oscillations within a fragment. Ideally, oscillations can be identified on the tubes by alternating colors, corresponding to the low and high values of the diverging color map where white represents the equilibrium distance. Practically, there are multiple equilibrium distances within the molecule, so the white color only shows an approximation for the equilibrium distance. Nevertheless, oscillations become clearly visible. Note that fragments with two atoms tend to have periodic oscillations. In contrast, fragments having just one more atom rather

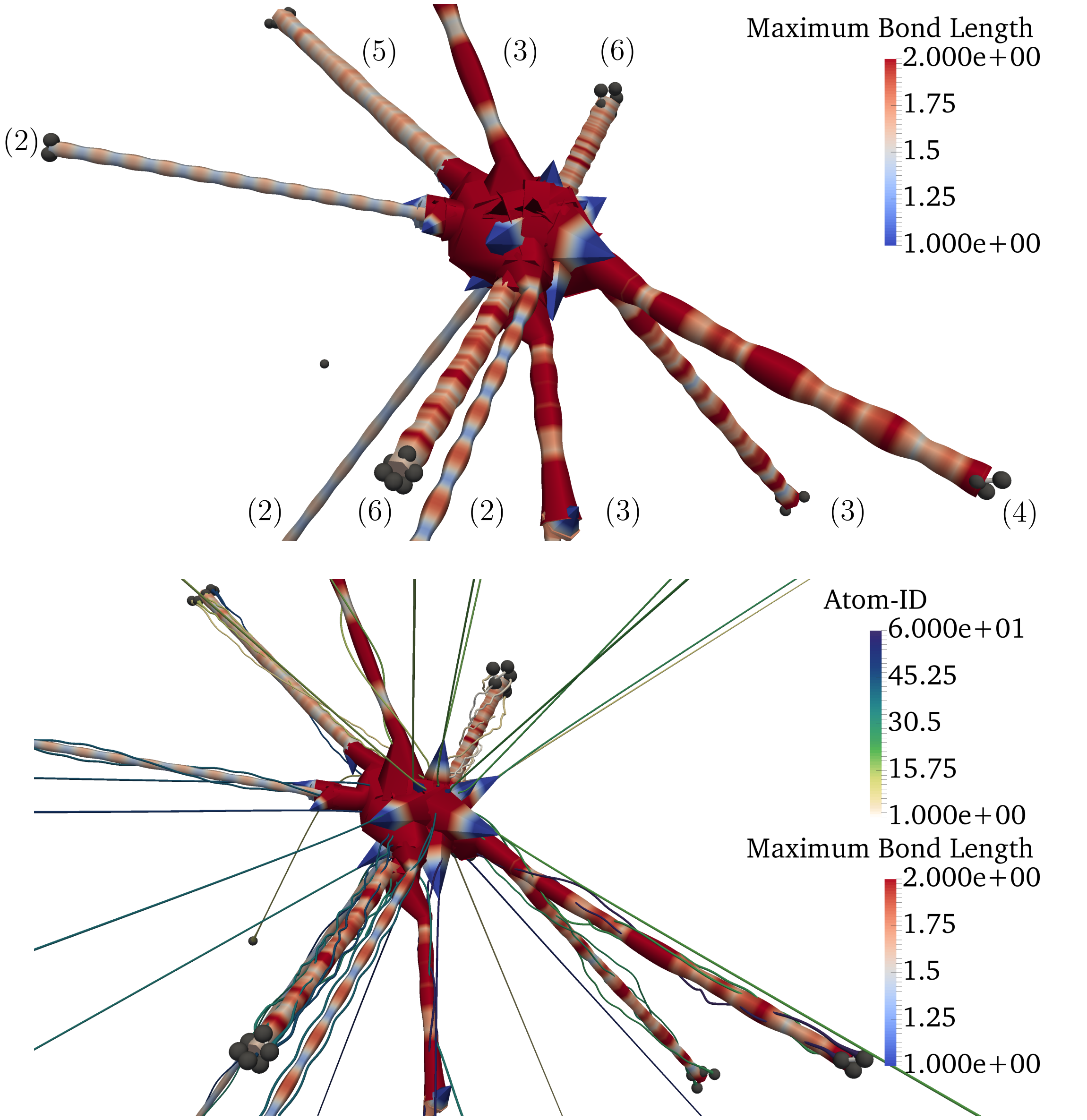


Figure 5.7: Fragment trees with focus on fragment dynamics. **Top:** The radius of the tubes of the fragment tree is set to the average distance to its center of mass. The radius of single atoms is set to zero. The numbers in brackets show the number of atoms of the individual fragments. **Bottom:** The radius of the tubes are additionally scaled a constant factor. Added atom trajectories from Figure 5.3 indicate rotations of the fragments. Both fragment trees are colored by the maximum bond length in a fragment. The bond length is given in Å.

seem to have an extremely long oscillation period if they even have any. Even larger fragments show a wild oscillation behavior.

The tubes are not only colored, but their radii are also varied. In this example, not the number of atoms is used, but the average distance to the center of mass between each atom in the fragment is mapped to the radius. We denote this measure as *average fragment radius* for a shorter notation. The fragment tree in Figure 5.7 on the top emphasizes the shrinkage and growing of a fragment. We also see correlations between the maximum bond length within a fragment and the average fragment radius. Especially fragments with two atoms have a high correlation, since there is only one degree of freedom in both measures. The fragment tree on the top serves as a sparse representation. The trajectories of single atoms are hidden and only indicated by the spikes, coming out of the origin of the molecule.

Also note that near the spikes some corner-like artifacts appeared. They are caused by overemphasizing the average fragment radius. We can remove them by simply scaling the radii of the tubes by a constant factor. The bottom image in Figure 5.7 shows a version in which these artifacts are nearly suppressed. However, the varying radius of tubes that belong to single fragments is hardly visible in this approach. Instead, we included the atom trajectories from Figure 5.3 of the upper left image. Too large radii of the tubes of the fragment tree would completely occlude the atom trajectories. This way, we are able to maximize the information content of our fragment dynamics visualization.

The atom trajectories provide some additional insights into the process. One of the fragments with two atoms shows no rotational motion, whereas both other fragments are rotating around an axis that is nearly parallel to the orientation of their mutual velocity. Interestingly, the oscillation patterns in both cases are different. The period in which the bond is rather long dominates for the fragment without rotation, whereas the rotating one rather seems to have a balanced long bond and short bond period. Maybe the rotation is not responsible for this behavior, but the difference of the oscillation amplitude. Considering the bond potential (see Figure 4.2) shorter bonds are easier repelled than long bonds, which are held together.

Classification and Limitation

The introduced fragment tree has two functions. On the one hand, its data structure transforms simulation data to fragment data, which serves as an in situ component. On the other hand, it exports the trajectories of all fragments, which serves a variety of information and visualization modes. It is a flexible method with static geometry, since it is able to encode information over time in a static representation. The fragment tree is only useful if there is fragmentation. It can be used for medium high dynamic processes, but works best for highly charged molecule dynamics since for low velocities, events and information collide on identical positions. We observed this especially in the center of the molecule.

The last example in Figure 5.6 illustrates ionization events on the fragment tree. The following section describes a method that directly depicts laser pulses for an intuitive prediction of ionization events in animation-based visualizations.

5.3.4 Laser Pulse

In this section, we introduce an animation technique to depict laser pulses. If we want to show an animation of the decay of a molecule, we can directly show the ionized atoms, e.g., by coloring them. The problem is that these events might take place very abruptly because the laser pulse is very sharp. The observer might be surprised by that and wondering why this is suddenly happening. So far, animations do not provide information about the laser pulse, neither the varying intensity of the laser pulse, nor its duration, nor its polarization. It is not even clear that the ionization is caused by a laser pulse at all.

Putting in as much information about the laser pulse as possible, we directly depict the laser pulse by an animation technique. Laser beams are electromagnetic waves, i.e., light. An electromagnetic wave is given as an orthogonal system, consisting of an electric field $\vec{E}$, a magnetic field $\vec{B}$, and the propagation velocity $\vec{v}$. They are transversal waves which means that they oscillate in the direction of the polarization which is orthogonal to the direction of the propagation. The polarization of an electromagnetic wave is denoted as the polarization of its electric field. Laser beams are usually depicted by sinusoidal curves which represent their electric fields. Such beams can be bundled to form laser pulses.

Shaping the laser pulse is performed by several steps that are illustrated in Figure 5.8. In each image, we see a representation of the laser pulse before and after performing modifications. There are two vectors, the horizontal vector, denoted as $\vec{n}_{\text{propagation}}$ serves the propagation direction, and the vertical one $\vec{n}_{\text{polarization}}$ provides the polarization orientation of the laser pulse. The origin of both vectors is the center of the laser pulse $\vec{x}_{\text{pulse}}(t)$ which is derived later. The starting point is an arbitrary sinusoidal function (left most image). This function is scaled by the

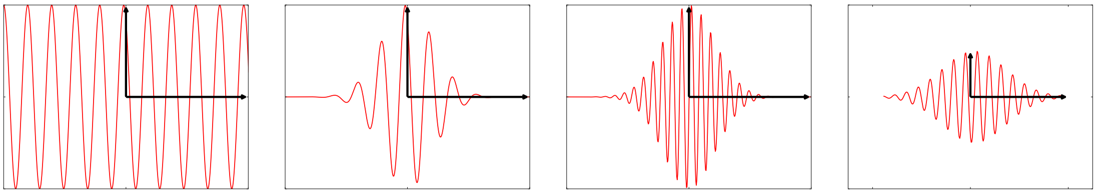


Figure 5.8: Steps for shaping the laser pulses. 1. weight sin/cos function by the probability distribution. 2. adapt frequency 3. Crop laser pulse in propagation direction, scale to desired size. The horizontal vectors point into propagation, the vertical into polarization direction.

probability distribution of Eq. 4.8 to obtain the middle left image. In the next step, the frequency of the pulse is adapted by a scaling factor f. To obtain the final result in the rightmost image, we crop the function of the laser pulse to cut off the straight left and right most parts of the pulse. Finally, we scale the pulse in $\vec{n}_{\text{propagation}}$ and $\vec{n}_{\text{polarization}}$ direction to the the respective lengths.

The described technique was mainly introduced to provide information of the ionization process. Thereby, we have to trade off between showing real values for information encoding, and practicability with respect to animation. Figure 5.9 shows a resulting illustration of a laser pulse with some preferred parameters. Let us start with the size of the laser pulse. As mentioned in Section 3.3.1, the focal spot of the laser ranges in the order of micrometers. The molecule, however, is approximately four orders of magnitude smaller and therefore the laser pulse would fill the whole scene. The essence is to show that each atom is hit by a laser pulse, so we adapt the amplitude $||\vec{n}_{\text{polarization}}||_2$ such that it is a little larger than the size of the molecule. Adapting the propagation $\vec{n}_{\text{propagation}}$ and the frequency factor f would theoretically be also possible, but for similar reasons, they are adapted to the size of the amplitude for a better illustration. The important information is the ionization strength depending on time.

As mentioned before, we need a position for the center of the pulse $\vec{x}_{\text{pulse}}(t)$. It depends on the simulation time t, the ionization intensity at this time, and the position of the laser focus $\vec{x}_{\text{focus}}$. At the time $t = t_{\text{max}}$ at which the ionization probability is maximal, the position of the center of the laser pulse $\vec{x}_{\text{pulse}}(t_{\text{max}}) = \vec{x}_{\text{focus}}$. The position of the laser focus $\vec{x}_{\text{focus}}$ is set to the center of the molecule. The propagation velocity is depending on the ionization probability, given by Eq. 4.8. The length of the laser pulse in direction of the propagation $2||\vec{n}_{\text{propagation}}||_2$ corresponds to the ionization duration t_{duartion}. This determines the propagation velocity of the laser

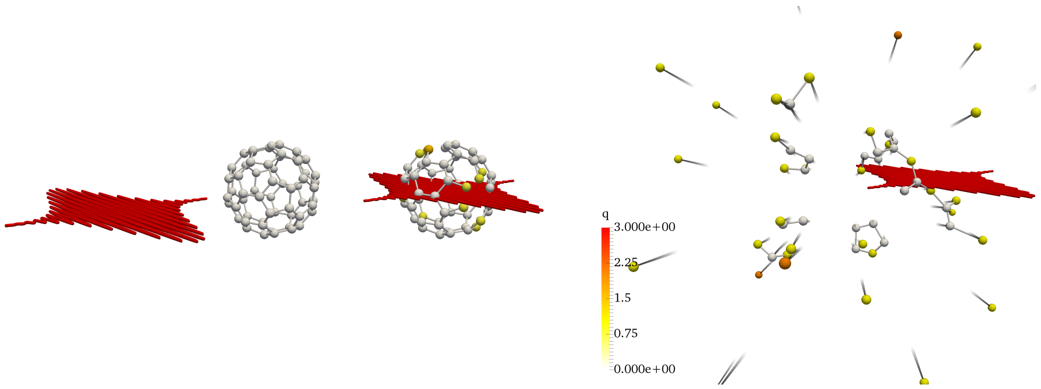


Figure 5.9: Illustration of the laser pulse before approaching the molecule (left), at maximum intensity (middle), and after ionization (right). The charge q is given in elementary charge units.

pulse $\vec{v}_{\text{pulse}} = 2\vec{n}_{\text{propagation}}/t_{\text{duartion}}$. By means of the velocity, the trajectory of the center of the laser pulse is be given as:

$$\vec{x}_{\text{pulse}}(t) = \vec{x}_{\text{focus}} + (t - t_{\text{max}}) \cdot \vec{v}_{\text{pulse}}. \tag{5.6}$$

This representation provides information about time and strength of the ionization in an intuitive and illustrative way. Note that this method is not suited for analyzing the dynamics and should rather not be used in this context.

Section 3.3.2 reported the use of X-ray pulses to record diffraction images in the experiment. The following section shows how these images can be generated virtually from the simulation data.

5.3.5 Virtual Diffraction Images

Diffraction images are recorded in the described experiments, whereas we compute them from the simulations. They are interesting means for the comparison between experiment and simulation. Thus, similar images may indicate similar states of the molecule. While the experiment is completely restricted to the analysis of the images themselves, simulations can provide further insights, e.g., by the computed fragment data or other statistics.

Consider Figure 5.10, the image plane of a photon detector is positioned by a support vector $\vec{r}_{\text{detector}}$ $[L]^3$ ($[L]$: in length unit L, $[L]^3$: a 3D vector with in length unit L). It is pointing to the lower left corner of the detector. Additionally, we need two orthonormal span vectors $\vec{w}_x$, $\vec{w}_y$ $\in \mathbb{R}^3$, $||\vec{w}_x||_2 = ||\vec{w}_y||_2 = 1$, pointing in x- and

y-direction in image coordinates, respectively. For an image with extends s_x, s_y $[L]$ and resolution $n_x,\ n_y$ in px, the node-based position of a sample $\vec{r}_{\text{img}}(i_x, i_y)$ $[L]^3$ in physical space with positive integer indices i_x, i_y in px is:

$$\vec{r}_{\text{img}}(i_x, i_y) = \frac{s_x i_x}{n_x - 1\,\text{px}} \vec{w}_x + \frac{s_y i_y}{n_y - 1\,\text{px}} \vec{w}_y + \vec{r}_{\text{detector}}. \tag{5.7}$$

It holds $0 \leq i_x \leq n_x - 1\,\text{px}$ and $0 \leq i_y \leq n_y - 1\,\text{px}$. The distance between molecule and detector in the experiment was 7 cm. The extents of the recorded diffraction images correspond to the sizes of 10.35 cm in horizontal and 8.34 cm in vertical direction. We introduce a planar wave front of one laser pulse that is parallel to the image plane, accordingly with direction $\vec{n}_{\text{laser}} = -(\vec{w}_x \times \vec{w}_y)/||\vec{w}_x \times \vec{w}_y||_2$. Now let $\vec{r}'_j$ $[L]^3$ be the atom positions and define $l_j(\vec{r}) = ||\vec{r}||_2 + ||\vec{d}_j||_2$ $[L]^3$ where $1 \leq j \leq N_{\text{atoms}}$, N_{atoms} is the number of atoms, and $\vec{d}_j$ $[L]^3$ is $\vec{n}_{\text{laser}}$ weighted by the distance from each atom to the wave front. Such a distance for one atom j is given by the projection of $\vec{r}'_j$ onto $\vec{n}_{\text{laser}}$. It can be derived by a simple scalar product when assuming the wave front plane normal to be placed at the origin and $\vec{n}_{\text{laser}}$ to be normalized:

$$||\vec{d}_j||_2 = |\vec{r}'_j \cdot \vec{n}_{\text{laser}}|. \tag{5.8}$$

Note that this formulation assumes no diffraction between the wave front and the positions of each atom. This approximation is fine here, since the distance to the detector is of much larger orders of magnitude. Thus, the inner molecule diffraction can be neglected. The diffraction image represents $|E(i_x, i_y)|^2$ per image pixel, $E(i_x, i_y)$ being the electric field of the X-ray pulse. The electric field of an electromagnetic wave is modeled as a wavefront, scattered by the atoms of the molecule and described as follows:

$$E(i_x, i_y) = \sum_{j=1}^{N_{\text{atoms}}} E_0 \exp(\mathrm{i}k l_j(\vec{r}_{\text{img}}(i_x, i_y) - \vec{r}'_j) - \omega t),\ k = \frac{2\pi}{\lambda}. \tag{5.9}$$

We choose a certain phase by setting $\omega t = 0$. Additionally, we set the amplitude to $E_0 = 1$. λ denotes the wavelength of the laser pulse. In our case, we used a laser pulses of energy $E_p = 1.8\,\text{keV}$:

$$E = h \cdot f, \qquad f = \frac{c}{\lambda} \qquad \Rightarrow \lambda_p = \frac{h \cdot c}{E_p} = 6.8880109659\,\text{Å}. \tag{5.10}$$

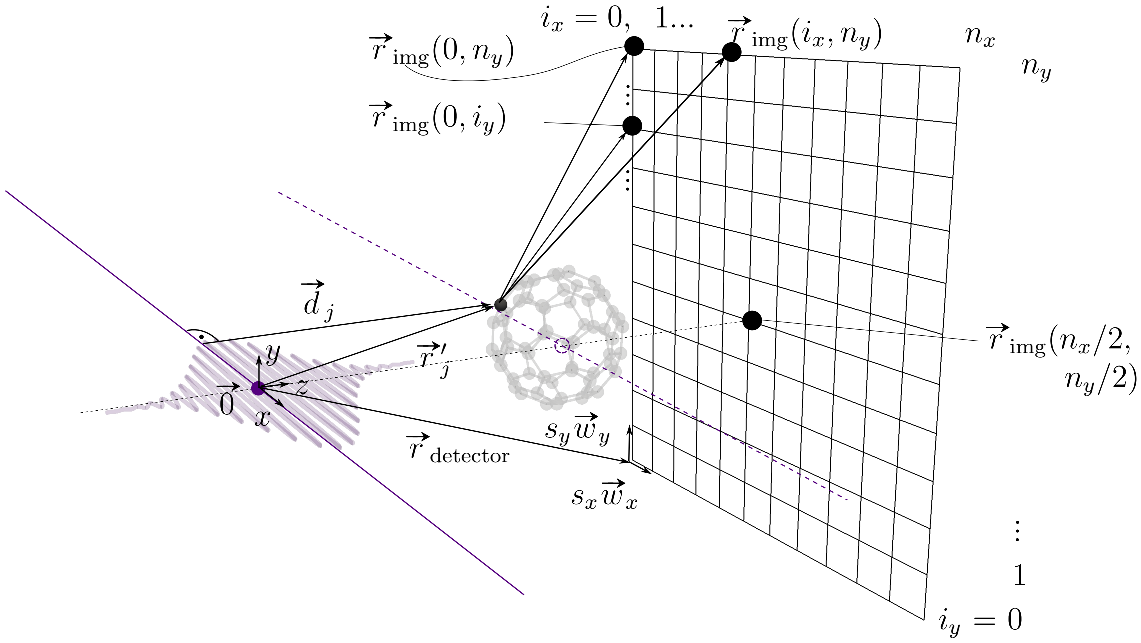

$i_x = 0,\ 1...$
$\vec{r}_{\text{img}}(0, n_y)$
$\vec{r}_{\text{img}}(i_x, n_y)$
n_x
n_y
$\vec{r}_{\text{img}}(0, i_y)$
$\vec{d}_j$
$\vec{r}'_j$
y
z
$\vec{0}$
x
$\vec{r}_{\text{img}}(n_x/2, n_y/2)$
$s_y\vec{w}_y$
$\vec{r}_{\text{detector}}$
$s_x\vec{w}_x$
$\vdots$
1
$i_y = 0$

Figure 5.10: Sketch of the reconstruction of diffraction images. The origin $\vec{0}$ can be placed arbitrarily. In the sketch, the origin is located at the center of the X-ray laser pulse that is shown in purple (not visible light), and pointing to the center of the image plane. The pulse is considered to be a wavefront for the derivation. It is shown as straight purple line, the dashed lines indicate the location used in the simulation. Points on the wave front are scattered exemplary by one carbon atom, emphasized in black, onto all image nodes $\vec{r}_{\text{img}}(i_x, i_y)$ of the regular grid.

As an optimization, we can save the computation of the square root in the Euclidean norm by a Taylor expansion of $l_j(\vec{r})$.

$$\begin{aligned}
\frac{\partial l_j(\vec{r})}{\partial r_x} &= \frac{1}{2} \cdot 2 \cdot r_x \cdot \frac{1}{\sqrt{r_x^2 + r_y^2 + r_z^2}} \\
\Rightarrow l_j(\vec{r} - \vec{r}') &= l_j(\vec{r}) - r'_x \frac{\partial l_j(\vec{r})}{\partial r_x} - r'_y \frac{\partial l_j(\vec{r})}{\partial r_y} - r'_z \frac{\partial l_j(\vec{r})}{\partial r_z} + \mathcal{O}(r'^2_x + r'^2_y + r'^2_z) \\
&= \sqrt{r_x^2 + r_y^2 + r_z^2} + ||\vec{d}_j||_2 + \frac{1}{\sqrt{r_x^2 + r_y^2 + r_z^2}} \cdot ((-r'_x)r_x + (-r'_y)r_y + (-r'_z)r_z) \\
&\quad + \mathcal{O}(r'^2_x + r'^2_y + r'^2_z) \\
&= ||\vec{r}||_2 + ||\vec{d}_j||_2 - \frac{\vec{r} \cdot \vec{r}'}{||\vec{r}||_2} + \mathcal{O}(r'^2_x + r'^2_y + r'^2_z).
\end{aligned} \tag{5.11}$$

The optimized formula for the energy value per pixel becomes

$$\begin{aligned} E(i_x, i_y) = \exp\left(\frac{2\pi \mathrm{i}}{\lambda} \cdot ||\vec{r}_{\mathrm{img}}(i_x, i_y)||_2\right) \\ \cdot \sum_{j=1}^{N_{\mathrm{atoms}}} \exp\left(\frac{2\pi \mathrm{i}}{\lambda}\left(||\vec{d}_j||_2 - \frac{\vec{r}_{\mathrm{img}}(i_x, i_y) \cdot \vec{r}\,'_j}{||\vec{r}_{\mathrm{img}}(i_x, i_y)||_2}\right)\right). \end{aligned} \tag{5.12}$$

Pre-factors are precomputed as well as $||\vec{d}_j||_2$ per simulation, $\vec{r}_{\mathrm{img}}(i_x, i_y)$ per time step, and $||\vec{r}_{\mathrm{img}}(i_x, i_y)||_2$ per pixel to speed up the computation. We do not show any diffraction images here. They are discussed in detail in Section 6.4.3.

The breaking of bonds causes changes of the structure of the molecule. Our previously presented techniques are able to visualize the dynamics after a decay. The changes before the fragmentation were only shown by direct representation techniques, which are not suited for comparisons. The following section introduces fragment holes for the description of ionized fullerenes.

5.3.6 Fragment Holes

The decay of fullerene is a highly dynamical process with a variety of resulting states. The analysis of multiple simulation results is easy to achieve with the previously discussed techniques. The properties of fullerene, e.g., the charge must always take their relative position into account, which makes motion predictions hard. It is even harder to compare these predictions to each other. Fullerene is a highly symmetric structure, which we want to make use of. It may also be desired to have a measure of the state to judge the current stability of the molecule. One indicator is the number of edges in a molecule. However, this is a rather rough measure since this it is not taking the connectivity into account. There might be configurations of the molecule which are more stable than other ones, although both might have the same number of edges. In this section, we introduce the *fragment hole* as an abstract measure for the stability of the molecule.

Mathematical Formulation

The special structure of fullerene provides us the possibility to study it from a topological point of view. The hollow sphere-like structure does not only interconnect all atoms with each other. Its topology can be seen as a sphere in 3D which is a 2D-manifold without border. This surface consists of polygons, initially 12 pentagons and 20 hexagons. The breaking of bonds causes changes of this structure which influences the stability of the whole molecule. The removal of an edge causes a union of two polygons. In this thesis, we denote a *fragment hole*, in short only *hole*, as a polygon that consists of at least seven atoms and seven edges of the fullerene molecule, where all of its bonds and atoms build one cycle. This requires the union of at least two initial polygons (the pentagons or hexagons which are not counted as holes).

Let us work out a more precise definition. We consider the molecule as graph representation $G_{\mathrm{C}_{60}} = (V_{\mathrm{C}_{60}}, E_{\mathrm{C}_{60}})$. Let the set of polygons be given as $\mathcal{P} = \{P_1, ..., P_{n_{\mathrm{poly}}}\}$. The number of polygons for fullerene is $n_{\mathrm{poly}} = 12 + 20 = 32$. One polygon P_i is a subset of the fullerene graph $P_i = (V_i, E_i) \subseteq G_{\mathrm{C}_{60}}$. For each polygon, the number of vertices is equal to the number of edges $n_i = |V_i| = |E_i|$. Furthermore, the polygon builds a simple cycle. A simple cycle is a path $P = (v_1, ..., v_n)$ with identical starting and end vertex $v_1 = v_n$, that are consecutively connected by edges. The initial set of polygons consists of all pentagons ($n_i = 5$) and hexagons ($n_i = 6$) with $5 \leq n_i \leq 6 \ \forall \ i = 1, ..., n_{\mathrm{poly}}$.

Creation of Fragment Holes

If the connectivity of a graph changes due to bond-breaking, we have to update the set of polygons. Let us introduce fragment holes by means of an example. In Figure 5.11 (1), we can see a section of the fullerene graph, shown with one pentagon and two hexagons. All are polygons P_i as an element of the polygon set $P_i \in \mathcal{P}$. In the figure, the index i is indicated by color. Let us suppose that due to bond-breaking two bonds in the simulation data structure vanished from one step to another. In Figure 5.11, we marked them by a crossing red line. In this case, we update the data structure of our visualization representation (details see below). We simplify the problem to remove all marked edges by their stepwise removal. The sequence of removing the edges is assumed not to matter, at least we did not find cases in which the results differed (except for the hole IDs) during the implementation. An edge e_j to remove that is shared by two polygons $e_j \in P_a, \ e_j \in P_b$, is processed by uniting the polygons to a new fragment $P_{n_{\mathrm{poly}}+1}$, without adding the edge e_j:

$$V'_{n_{\mathrm{poly}}+1} = V_a \cup V_b, \quad E'_{n_{\mathrm{poly}}+1} = E_a \cup E_b \setminus e_j, \quad P'_{n_{\mathrm{poly}}+1} = (V'_{n_{\mathrm{poly}}+1}, E'_{n_{\mathrm{poly}}+1}). \quad (5.13)$$

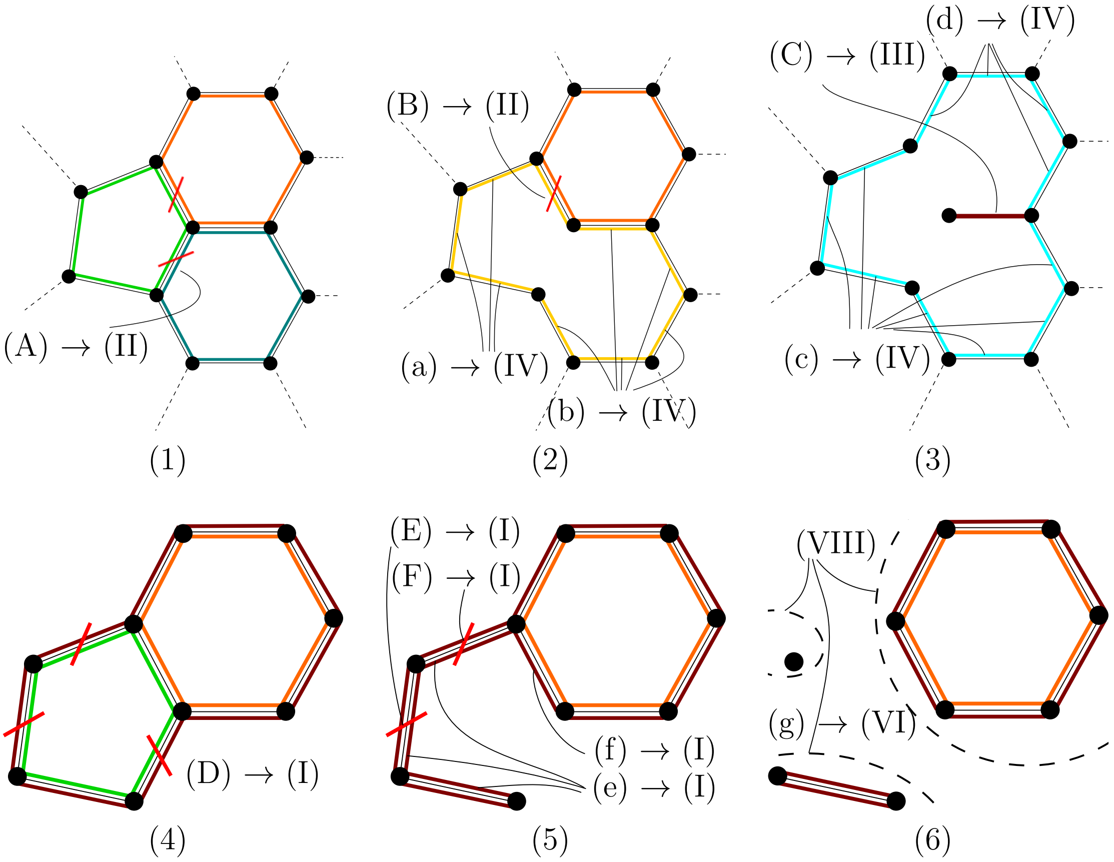


Figure 5.11: The process of the creation of holes by the pairwise union of polygons. (A)–(F): Edge removals, (a)–(g): edge ID updates, (I)–(VIII): respective locations in Algorithms 2 and 3, (1)–(3), (4)–(6): examples for polygon data structure update. (1): Initial fullerene graph configuration. The edges of each polygon are depicted by individual colors, so here we have three polygons. Assume that two edges must be removed, indicated by the red lines. (2): We remove edges stepwise. The polygons which share an edge to remove are united (both set of vertices and edges), except the edge to remove, which is not added to the united polygon. United polygons are referred as holes. (3): Edges that do not have to be removed and are shared by both polygons become hanging edges, marked in dark red. They are not added to the newly united polygon. (4): In another example, there are two isolated polygons, marked by the dark red outside. (5): Breaking isolated edges destroys the polygon. (6): A step is skipped to show that products of the procedure can belong to new fragments, marked by the dashed lines. Thus, vertices, edges, and polygons not belonging to the processed fragment must be removed.

This will serve us the configuration of Figure 5.11 (2). If we apply this procedure once more, we obtain the graph in Figure 5.11 (3). In this configuration, an irregular edge appears, marked in dark red. We consider them as *hanging edge*. They have not been considered in Eq. 5.13, which is why we set a prime for the definitions of the new polygon sets. We provide an algorithm to obtain the correct sets later on. The reason for the hanging edges can be seen in the former step (Figure 5.11 (2)). We see that both polygons share two edges unlike in the case before. Only one of these edges needs to be removed. The other one remains as a part of the fragment, but is no longer part of the polygon to fulfill the requirement to form a simple cycle.

Definition of a Fragment Hole

Let us first define the size of a polygon. The size of a polygon P is given by its edge count $|E|$. Due to the property of a simple cycle of a polygon, the edge count $|E|$ is equal to the vertex count $|V|$. A *fragment hole*, in short only *hole*, is a polygon P that possesses a polygon size of at least nine ($|E| \geq 9$).

Let us give a short explanation for the definition of a fragment hole. The original graph is unique and its topology is not changing during the simulation. In contrast, united polygons are a part of a modified structure with a different stability. In the case of fullerene, the smallest possible polygon merge is between a pentagon and hexagon (e.g., step from (1) to (2)) resulting in a polygon size of nine. The polygon size nine is smaller than all other original polygons. This is why we chose $|E| < 9$ as discrimination criterion between original polygons and holes.

Data Structure and Implementation

We start by describing the data structure. For a given edge, which can potentially be removed, we need to know the polygons that share this edge. Thus, for each edge of the graph of fullerene $G_{\mathrm{C}_{60}}$, we save a `PolygonEdge` that contains two vertex IDs and two Polygon IDs. A `Polygon` contains a vector of `PolygonEdges`. The set of polygons is implemented as a map with a polygon ID as key and a `Polygon` as value. We have to save such a map for each `FragmentNode` in the fragment tree data structure. There, we additionally save a vector of all `PolygonEdges`. These rather simple data structures are sufficient for the aimed implementation.

In the following, we present Algorithms 2 and 3 to update the polygon map for the current molecule, while implicitly detecting holes in it. If bonds are broken, the `FragmentGraph` G is updated. The polygon data structure is updated by

Algorithm 2 Algorithm to update the polygon data structure.

Input: A `PolygonEdge` e_{rem}, a leaf `FragmentNode fn` of the fragment tree with respective `FragmentGraph` G, polygon map `pm`, and edge vector `ev`.

```
procedure UPDATEPOLYGONS(fn)                                ▷ Called by Algorithm 1.
    Get a set of edges $E_{\text{rem}}$, removed from $G$, but still present in pm and ev.
    for each $e_{\text{rem}} \in E_{\text{rem}}$ do          ▷ Stepwise update of polygon data structure.
        mergePolygonsAtEdge(pm, ev, $e_{\text{rem}}$)        ▷ See Algorithm 3.
    end for
    Get a set of vertex IDs $I_v$, removed from $G$, but still present in pm and ev.
    Get a set $E$, containing all PolygonEdges $(v_{j,1}, v_{j,2}, p_{j,1}, p_{j,2}) \in$ ev with
      $\forall\ 1 \leq j \leq |\text{ev}|\ \forall\ i_v \in I_v : (v_{j,1} = i_v) \vee (v_{j,2} = i_v)$.
    Get a set of all polygon IDs $I_p$ of $E$.
    for each $i_p \in I_p$ do                                ▷ (VIII) → Figure 5.11 (6)
        Replace each polygon ID entry equal to $i_p$ in ev and pm by -1.
        Erase all polygons with ID equal to $i_p$ in pm.
    end for
    Find and erase all edges $e \in E$ in ev.
end procedure
```

executing Algorithm 2. Edges that have been removed from G also need to be removed in the polygon data structure. The algorithm first updates the IDs and stepwise removes the edges by calling `mergePolygonsAtEdge()`. Vertices, which do no more belong to the processed fragment, must be removed, so all edges and polygons containing these vertices are removed as well (VIII).

The functionality of the method `mergePolygonsAtEdge()` has been already discussed by outlining the principle of the technique. However, there are some more cases to consider. Starting from Figure 5.11 (4), the processing of a fragment is shown. The two polygons are isolated so they posses an invalid fragment hole ID of -1, indicated by the dark red color in the figure. The removal of edges containing at least one hole ID of -1 must be handled differently, since the second polygon to merge is already removed. Instead, all IDs of the remaining (green) polygon are set to -1, since it is cracked by removing one of its border edges (Figure 5.11 (5)).

Algorithm 3 Algorithm to remove an edge and merge two polygons on demand.

Input: A `PolygonEdge` $e_{\text{rem}} = (v_1, v_2, p_1, p_2)$ with v_1, v_2 being vertex IDs and p_1, p_2 being polygon IDs, polygon map `pm`, and edge vector `ev`.

```
procedure MERGEPOLYGONSATEDGE(pm, ev, $e_{\text{rem}}$)                ▷ See Figure 5.11.
    if $(p_1 = -1) \vee (p_2 = -1)$ then                              ▷ $-1 \hat{=}$ dark red in Figure 5.11.
        for $i \in \{1, 2\}$ do
            if $p_i \neq -1$ then                                    ▷ (I) → Figure 5.11 (D)–(F), (e), (f)
                Replace each vertex ID entry equal to $p_i$ in ev and pm by -1.
                Remove polygon $p_i$ from pm.
            end if
        end for
        Remove $e_{\text{rem}}$ from ev.
        return                          ▷ Removing a border edge, no need to merge polygons.
    end if
    Get polygons $P_{p_1}, P_{p_2}$ containing edge sets $E_{p_1}, E_{p_2}$ by IDs $p_1, p_2$.
    Allocate merged polygon $P_{i_{\text{merged}}}$.
    for each $e_j \in E_{p_1}$ do                  ▷ Add all unshared edges from $E_{p_2}$ to $P_{i_{\text{merged}}}$.
        if $e_j$ is a shared edge of $P_{p_1}, P_{p_2}$ then
            if $e_j$ is the edge to remove $e_{\text{rem}}$ then
                Remove $e_{\text{rem}}$ from ev.                     ▷ (II) → Figure 5.11 (A), (B)
            else                                                   ▷ (III) → Figure 5.11 (C)
                Find $e_j$ in ev and set both polygon IDs to -1.
            end if
        else                                     ▷ $e_j$ not a shared edge, unite polygon step.
            Add $e_j$ to $P_{i_{\text{merged}}}$.                        ▷ (IV) → Figure 5.11 (a)–(d)
        end if
    end for
    for each $e_j \in E_{p_2}$ do                  ▷ Add all unshared edges from $E_{p_2}$ to $P_{i_{\text{merged}}}$.
        if $e_j$ is not a shared edge of $P_{p_1}, P_{p_2}$ then
            Add $e_j$ to $P_{n_{\text{poly}}+1}$.                        ▷ (IV) → Figure 5.11 (a)–(d)
        end if          ▷ No else branch, hanging edges were handled in iteration above.
    end for
    Erase polygons $P_{p_1}, P_{p_2}$ from pm.
    if $|E_{i_{\text{merged}}}| > 0$ then                              ▷ (V) → Figure 5.11 (a)–(d)
        $i_{\text{merged}} \leftarrow n_{\text{poly}} + 1$
        Add $(i_{\text{merged}}, P_{i_{\text{merged}}})$ to pm.
    else
        $i_{\text{merged}} \leftarrow -1$                                  ▷ (VI)→ Figure 5.11 (g)
    end if
    Replace the old polygon IDs $p_1, p_2$ by $i_{\text{merged}}$ in ev and pm.
    if $P_{i_{\text{merged}}}$ is isolated then                            ▷ (VII) → Figure 6.13
        Set the higher polygon ID of $P_{i_{\text{merged}}}$ to -1 for each $e_j \in E_{i_{\text{merged}}}$
    end if      ▷ The higher ID corresponds to thnote polygon that was inserted later.
end procedure
```

6 Results and Discussion

In this chapter, we start with an overview of the performed simulations and their analysis, together with details of the computational effort in Section 6.1. We further analyze the behavior of the simulation in Section 6.2. The plausibility validation of the simulation models are given in Section 6.3. The simulation models are tested by comparing them with the experimental results. The chapter concludes with an analysis of the fragment dynamics in Section 6.4.

6.1 Simulation and Computational Cost

In the beginning, we want to outline which computational effort was necessary to produce our results. Table 6.1 shows the computed data sets on the top whereas the number of different simulation parameters is shown below, and at the bottom, the effort per data set is shown. Let us first describe the data sets. Data set TOF1 contains data of the fragments per simulation for a combination of all parameters. These data are used to produce Figures 6.4 and 6.5. We exemplary checked also the bond-breaking approach individual-bond with Lennard-Jones potential with data set TOF2. Note that these two data sets ran faster than data set TOF3 since they have been simulated with many low intensities, which results in a lower dynamics an thus in lower computational cost to compute the data of the fragments. Data set TOF3 was computed only for the bond-breaking method immediately-2-bond for the hole analysis, see Figure 6.14. Note that the analysis over all time steps took nearly as long as producing the data itself. The writing of the files was a bottleneck for these data sets.

We further computed data sets with diffraction images. These are computationally more expensive. This is the reason for the speedup when using multiple threads. We started computing data set D1 with the usual time resolution of 1001 time steps per simulation, which turned out to be few. We kept this data for creating animations, but reduced the time resolution. We then computed data sets D2 and D3, which we

Data Sets	TOF1	TOF2	TOF3	D1	D2	D3
Bond-Breaking	5	1	1	1	1	1
$\#q_{\text{mean}}$	31	31	5	5	5	5
Simulations	250	50	1900	190	460	850
Time Steps	1001	1001	1001	1001	21	21
pixels x	-	-	-	345	345	345
pixels y	-	-	-	278	278	278
Files	$3.9 \cdot 10^7$	$1.5 \cdot 10^6$	$9.5 \cdot 10^6$	$9.5 \cdot 10^5$	$4.8 \cdot 10^4$	$8.9 \cdot 10^4$
Threads	5	1	1	5	1	1
Computation t [h]	32	6.5	33.5	56.5	22.5	46

Table 6.1: Overview over data sets, obtained by simulations.

used to create Figures 6.11 and 6.12.

6.2 Simulation

After the decision to use the LAMMPS framework, we performed exploratory simulations with examples of similar molecular systems. The next step was to run simulations with C_{60}. The first data file of C_{60} that we used was provided by Berman et al. [BWF+00]. We were able to load and show the C_{60} molecule, but the force field was missing to move the atoms. In search for a force field, we found the work of Berkai et al. [BdMB15] providing one. However, most of the potentials needed the bond, angle, and dihedral angle lists for the simulation (see Section 4.3.1), which are not provided by the data file, we found. Further investigation brought us to the work of Monticelli [Mon12], providing the necessary lists. Adapting the files information for the LAMMPS framework, we collected all information for running simulations.

To get familiar with this simulation setup , we explored its behavior for several configurations. Various simulations were performed to obtain the final configurations while the simulation and visualization tools were developed. The configurations have been adapted from time to time. We briefly explained the configurations that are the same in each simulation run in Section 4.2. The most interesting configurations are explored in detail in the following sections.

6.2.1 Force Field

This section reports the behavior of the simulation for a collection of force field models that have been introduced in Section 4.3. Figure 6.1 gives an illustration of

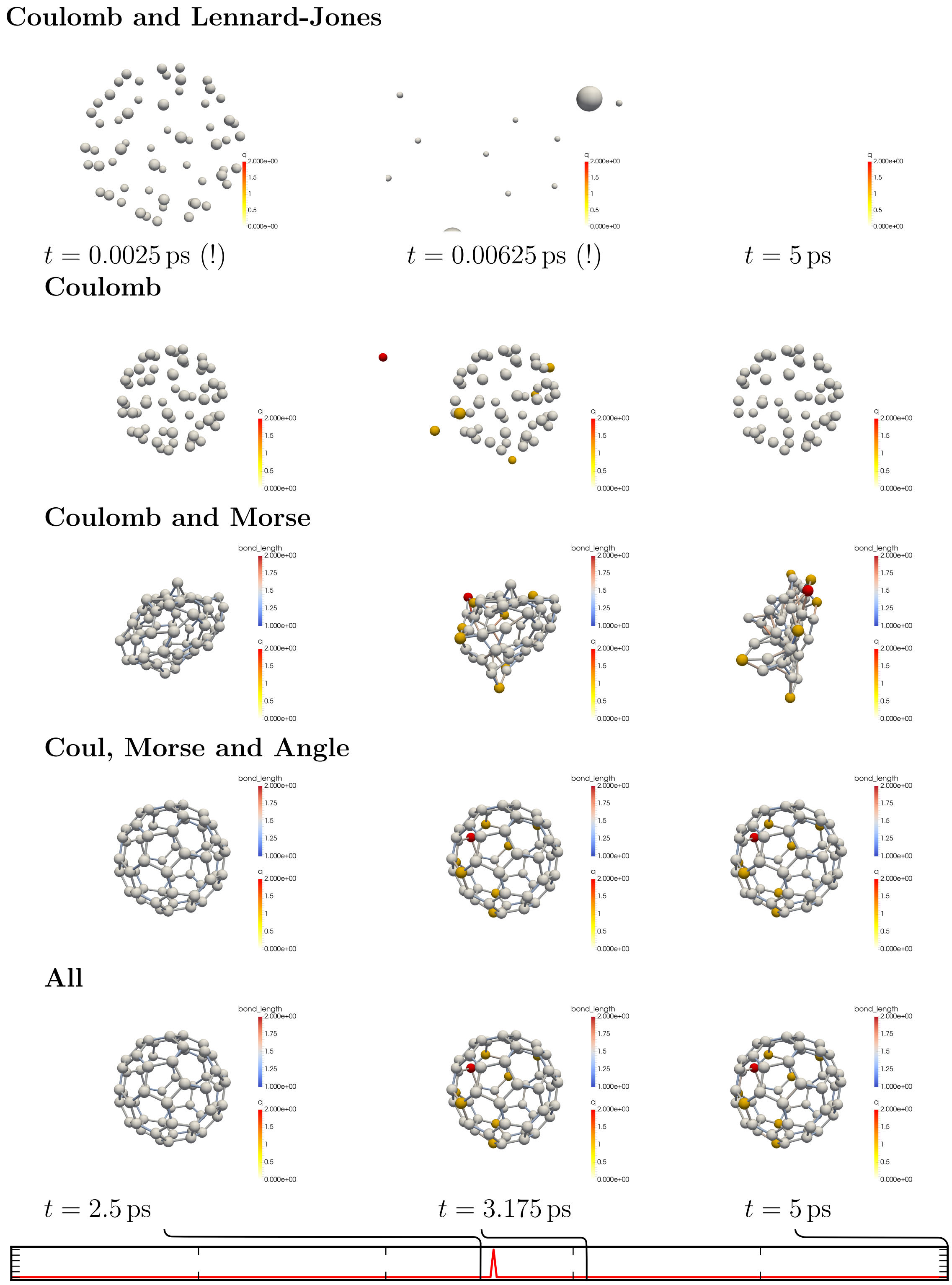


Figure 6.1: Overview of different force field models with the ionization probability on the bottom. Note the different times for the first row. Bond length given in Å, charge q in elementary charge units.

simulations with an equal ionization pattern but distinct simulation models. Note that no bonds are broken during the simulations.

Let us start describing the first row. There, only pairwise models are used, namely the Coulomb and Lennard-Jones (LJ) potentials. Note that the times of the snapshots of the simulation is different to the other rows. The reason is that the simulation is completely instable. After 2.5 fs the buckyball system approximately doubled its radius and is nearly vanished after 6.25 fs. In contrast, the LJ potential is also used in the last row, which is completely stable. What happened in the simulation of the first row? If we consider Figure 4.2, we see that the LJ potential repels atoms with a lower distance of about 3 Å. The equilibrium distance however is at approximately 1.5 Å, so actually the simulation result is correct.

In another example, the charge of two neighboring atoms in a simulation with Morse potential has been explicitly set to really high values, but the atoms did not repel each other. We consulted the article of the force filed values [BdMB15] once again. Then we noted that the LJ potential is meant to only act between atoms that are not directly connected by bonds with each other. The distance of second neighbors is about 2.8 Å where the potential has hardly an influence. The LAMMPS framework behaved the same way for the pairwise forces (coulomb and LJ), so this seems to be the reason for the explosion in the first row.

For the other simulation results, we show snapshots at simulation times 2.5 ps, 3.175 ps, and 5 ps. The first column illustrates the simulation without Coulomb forces since all atoms are not ionized. In the second column, the simulation is shown shortly after atoms are ionized. The last state is shown at $t = 5$ ps.

In the results of the Coulomb simulation only charged atoms are repelling each other, whereas neutral atoms are not moving. The neutral atoms remain in their starting position. Adding bonds to the force field, the molecule starts moves shaky with slight oscillations between bond partner. However, slightest perturbations causes the molecule to deform, even without ionization. The ionized atoms in the molecule make the behavior even wilder until it looks completely deformed in the final state. The next row shows that the angular potential is essential to keep the structure of the fullerene molecule. Finally, the last row shows the results of all used potential simulations. This includes the LJ, Coulomb, Morse, angle, and dihedral potentials.

Note that in the last two rows that even the ionization of nine electrons still results in a stable molecule. Due to the time-of-flight measurements, this is probably not the case in reality. Let us consider Figure 6.3, which shows the experimentally detected

C_{60} fragments. The trend is that already four times charged C_{60}^{4+} molecules have been barely detected regarding the yield of the less charged ions C_{60}^{3+}, C_{60}^{2+}, and C_{60}^{+}. This fact leads us to Section 6.2.2 in which we try to destabilize the molecule by breaking bonds with different approaches.

6.2.2 Bond-Breaking Approaches

We introduced the bond-breaking approaches in Section 4.4.2 in order to tune the quality of the simulation results with respect to the experimentally determined time-of-flight histograms. Figure 6.2 shows an overview of simulation results for different bond-breaking approaches. The graph on the bottom shows the ionization probability of a laser pulse with respect to the simulation time. Note that the graph is truncated from $t = t_{\text{start}}$ to $t = t_{\text{end}}$ as it is realized in the simulation.

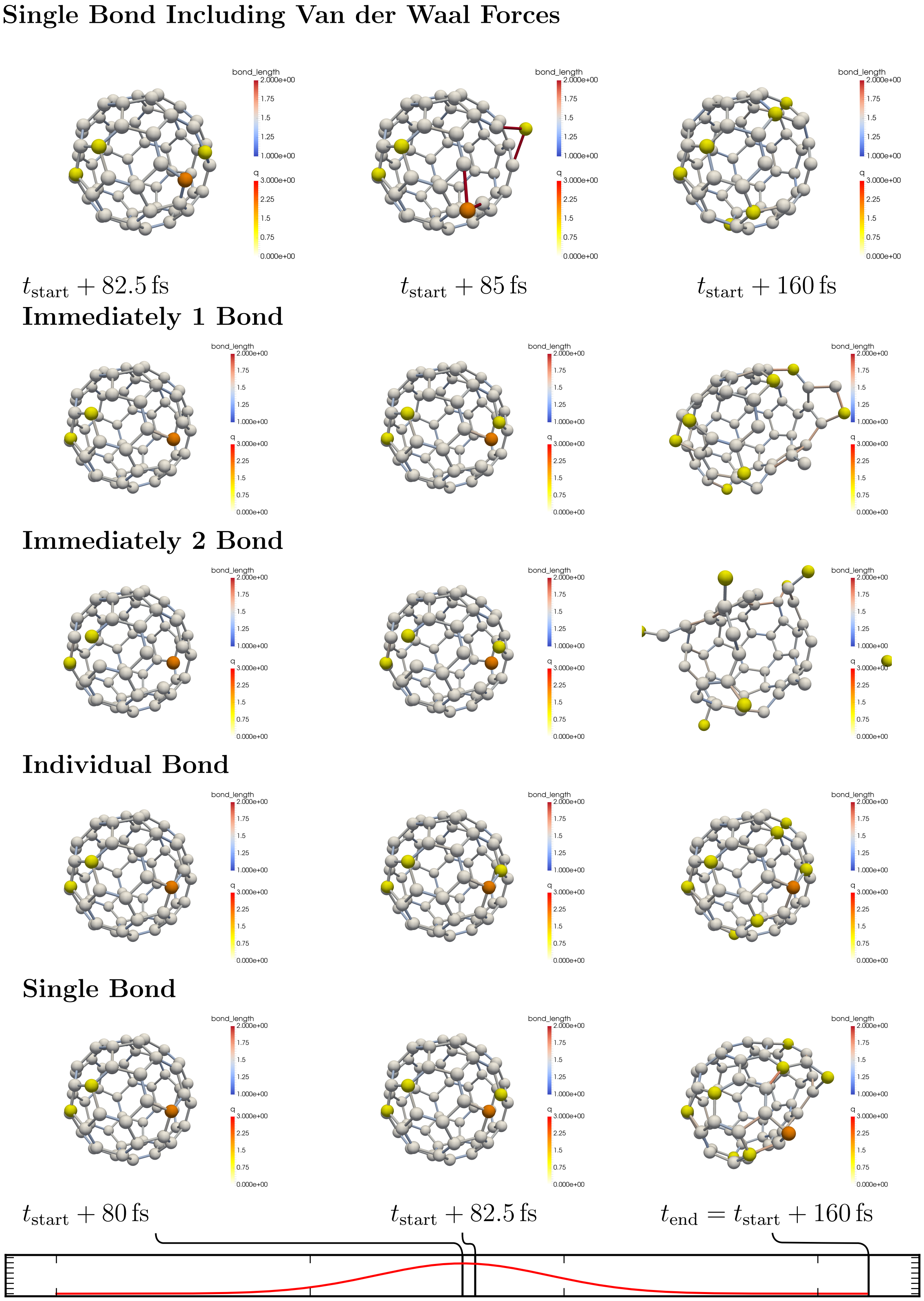
The overview shows all implemented bond-breaking approaches that have been simulated with all potentials except the Lennard-Jones potential. Exemplary, a simulation with the Lennard-Jones potential is performed with the bond-breaking approach Single-Bonds in the first row. Another possibility to adapt the stability of the molecule is to vary the constant D of Eq. 4.2 in the Morse potential. We did not do that because the constant was determined and validated by Berkai et al. [BdMB15] so changes might introduce additional inconsistencies.

6.3 Plausibility of Simulation Results

In this chapter, we want to judge how trustworthy the simulation results are. This can be done by comparison with the time-of-flight measurements of the experiments that contain information about masses and charges of ions microseconds after the laser pulses. We compare results of the simulation and the experiment in two ways. Section 6.3.1 shows a direct comparison of time-of-flight histograms where in Section 6.3.2 the ratio of the most common fragments is compared.

6.3.1 Comparison of Time-of-Flight Histograms

The time-of-flight experiment gives information about the fraction between mass and charge of ions. A weakness of this method is the ambiguity of the time-of-flight.



Single Bond Including Van der Waal Forces

Immediately 1 Bond

Immediately 2 Bond

Individual Bond

Single Bond

bond_length
2.000e+00
1.75
1.5
1.25
1.000e+00
q
3.000e+00
2.25
1.5
0.75
0.000e+00
$t_{\text{start}} + 82.5\,\text{fs}$
$t_{\text{start}} + 85\,\text{fs}$
$t_{\text{start}} + 160\,\text{fs}$
$t_{\text{start}} + 80\,\text{fs}$
$t_{\text{start}} + 82.5\,\text{fs}$
$t_{\text{end}} = t_{\text{start}} + 160\,\text{fs}$

Figure 6.2: Overview of bond-breaking approaches with the ionization probability on the bottom. Note the different times for the first row. The bond length is given in Å, the charge q in elementary charge units.

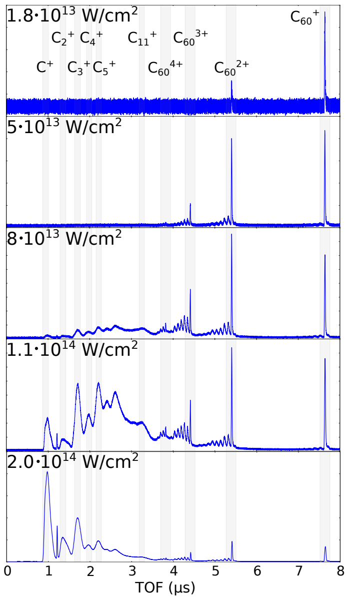


Figure 6.3: Experimental time-of-flight measurements for different laser pulse intensities. The y-axis shows the fragment yield. Peaks in the histogram correspond to multiple ions. The ions with the highest correspondence are shown in the top plot. They can be found in the highlighted areas.

For example the time-of-flight for two ions C_{30}^{+} and C_{60}^{2+} are identical, see Eq. 6.1.

$$t_{\mathrm{TOF}}(\mathrm{C}_{30}^{+}) = k\sqrt{\frac{m}{q}} = k\sqrt{\frac{12.011\,\mathrm{u}\cdot 30}{1\,e}} = k\sqrt{\frac{12.011\,\mathrm{u}\cdot 60}{2\,e}} = t_{\mathrm{TOF}}(\mathrm{C}_{60}^{2+}) \tag{6.1}$$

Since the simulation provides full information about mass and charge of ions the ambiguity can be avoided by calculating the time-of-flight values form the simulation results. Therefore, the constant k must be calculated. This can be easily done by means of the time-of-flight measurements of the experiments, see Figure 6.3. The analysis of the experiment showed the correspondence between certain ions and peaks in the time-of-flight histograms. Eq. 6.2 shows that k can be determined by the peak that corresponds to a single ionized fullerene molecule C_{60}^{+} at $t_{\mathrm{TOF}} = 7.782\,\mu\mathrm{s}$.

$$k = t_{\mathrm{TOF}} \cdot \sqrt{\frac{q}{m}} = 7.782\,\mu\mathrm{s} \cdot \sqrt{\frac{1\,e}{60 \cdot 12.011\,\mathrm{u}}} = 0.29\,\mu\mathrm{s} \cdot \sqrt{\frac{e}{\mathrm{u}}} \tag{6.2}$$

This constant holds for all other ions and this way the time-of-flight can be extracted form the simulation data. Next, we want to compare the bond-breaking techniques and the experiment with each other. A mapping from the intensity values in the experiment to the parameter q_{mean} in the simulation would be useful.

Experiment in $\mathrm{W\,cm^{-2}}$	Simulation q_{mean}
$5 \cdot 10^{13}$	0.5
$8 \cdot 10^{13}$	2.5
$1.1 \cdot 10^{14}$	10
$2 \cdot 10^{14}$	60

Table 6.2: Laser Intensity Assignment of Experiment and Simulation

The mapping is done visually by looking at time-of-flight histograms for multiple q_{mean} values and finding the highest similarities between simulations and experiments. The intensity assignments are presented in Table 6.2.

The extracted time-of-flight histograms show not a continuous line, but a graph with really sharp peaks. Due to this fact one-dimensional plots of the graphs would occlude each other. Therefore, the time-of-flight yields are shown as 2D-plots in Figure 6.4 for an easier comparison. The figure shows scatter plots for multiple mappings of laser pulse intensities. Each plot is with respect to the bond-breaking approach, the corresponding experimental intensity and the time-of-flight.

Results of the simulation and experiment roughly agree in the lowest intensity. Starting from intensity $8 \cdot 10^{13}\,\mathrm{W\,cm^{-2}}$ the results differ tremendously. Noticeably, the oscillations left of the C_{60} ion peaks do never appear in the simulation results. They correspond to C_{59}^{i}, C_{58}^{i} and so on, with the same charge of the original C_{60}^{i} with $i \in \{1, 2, 3, 4\}$. In the plot with the intensity of $1.1 \cdot 10^{14}\,\mathrm{W\,cm^{-2}}$ the experimental results show a high yield for $t_{\text{TOF}} \in [1; 3]\,\mu\text{s}$ whereas in the results of the simulations the yields in this region are rather sparse. At the intensity $2 \cdot 10^{14}\,\mathrm{W\,cm^{-2}}$ the simulations produce also fragments with small mass and high charge ($t_{\text{TOF}} < 1\,\mu\text{s}$).

6.3.2 Comparison of Selected C_{60} Fragment Yields

From extracted time-of-flight histograms of Section 6.3.1 judging the quality of the bond-breaking approaches is hard. They all show a similar pattern. If we really want to have a ranking of the bond-breaking approaches, we need a more substantial method with comparable results. The comparison of the results of simulations and experiments in this section uses the time-of-flight yield ratio of the classified peaks in Figure 6.3. These particular ions were extracted and counted from the simulation results. For the comparison each of the 10 ion yields was normalized by the total yield in the experiment and simulation, respectively such that we compare yield

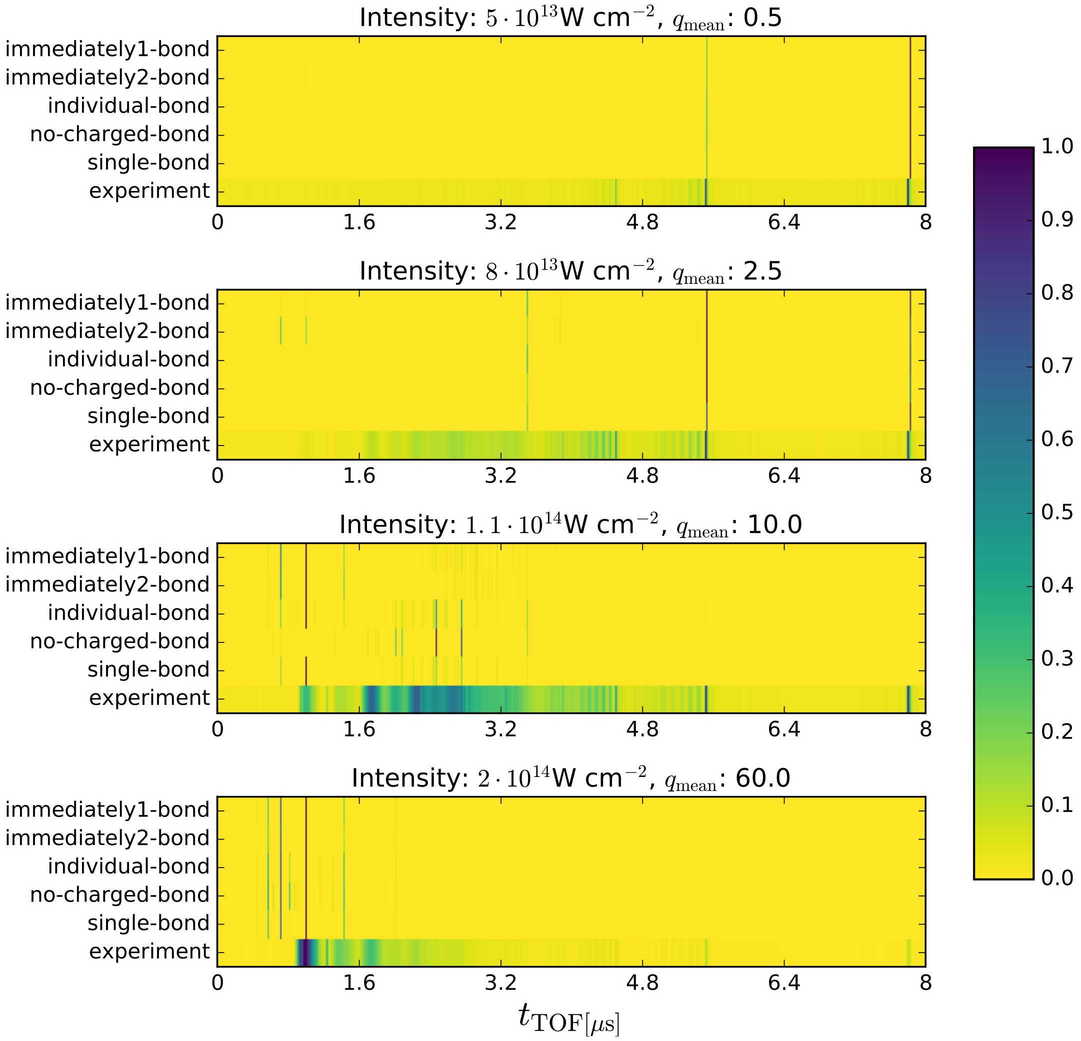


Figure 6.4: Time-of-flight histograms comparing between results of experiment and simulation for multiple laser pulse intensities. The color bar indicates the fragment yield.

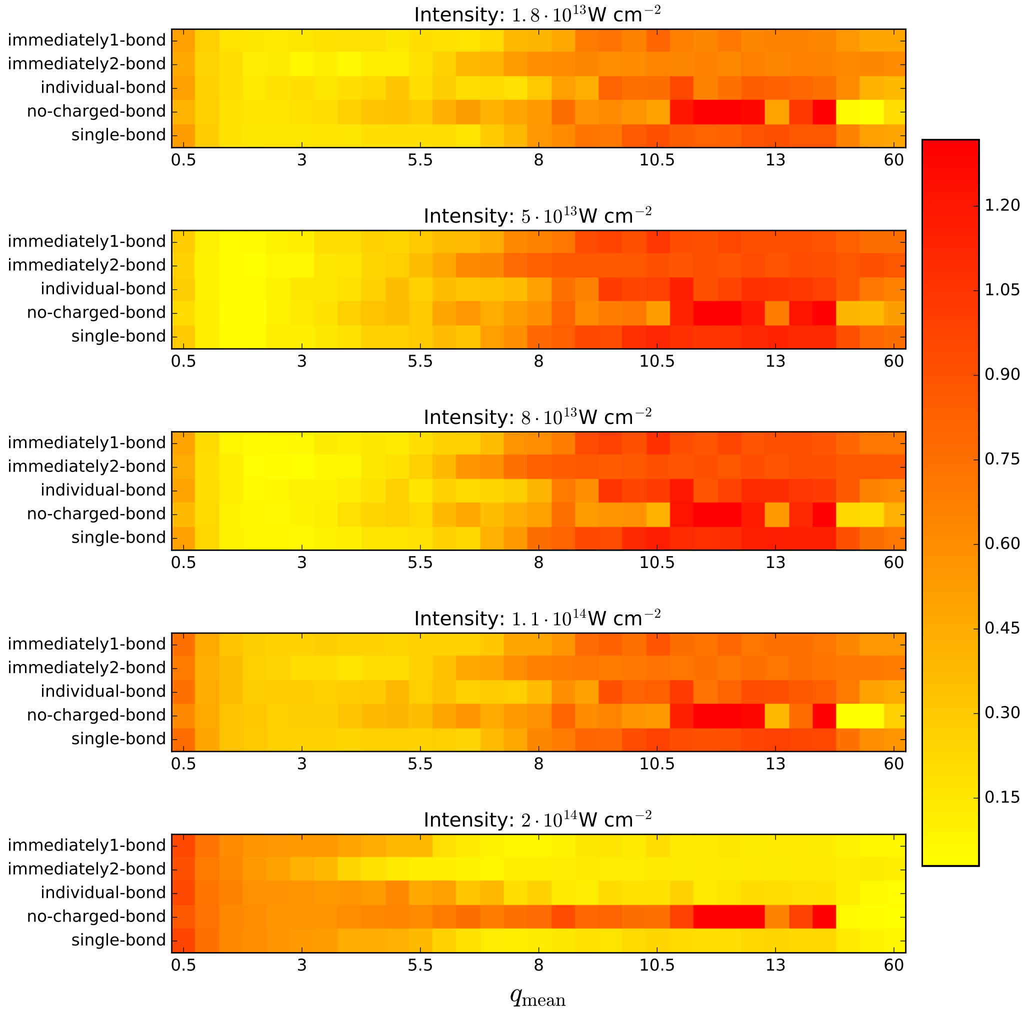


Figure 6.5: Sum of the squared differences of the ion yield ratio of experiment and simulation. The lower the sum of the squared differences, the higher the correspondence between simulation and experiment. Laser pulse intensity values of the experiment are given above the plots. Each is shown with respect to the distinct bond-breaking approaches and the laser intensity parameter q_{mean} of the simulation. Note that the last values of q_{mean} are not equidistant, $q_{\text{mean}} = (..., 13.5, 14, 20, 30, 60)$.

Bond-Breaking Approach	Sum of Squared Differences
immediately1-bond	7.16
immediately2-bond	6.91
individual-bond	7.33
no-charged-bond	9.35
single-bond	7.95

Table 6.3: The ranking results of the bond-breaking approaches.

rations.

Due to the ambiguity of the time-of-flight histograms, we assume that the peaks mainly correspond to the assigned ions. For obtaining the ion yields in the experiment two approaches are chosen. Since the measurements come with a certain inaccuracy the ion yield can be accumulated within a small region as also highlighted in Figure 6.3, or simply use the time-of-flight values at the yield maxima within the same area.

After having collected the yield ratios of simulations and the experiment the values were compared by summing over the squared difference of the selected ten ions. Figure 6.5 shows in each plot the results for one intensity used in the experiment. Each individual plot is given with respect to the bond-breaking method and the simulated laser intensity, given as parameter q_{mean}.

Recall the mapping of the laser pulse intensity in Table 6.2. Let us first consider the low intensities and look at the region of the corresponding values of q_{mean}. Intensities up to $8 \cdot 10^{13}\,\text{W}\,\text{cm}^{-2}$ correspond to values $q_{\text{mean}} < 3$. The trend in this region has a rather good agreement with the experiment. This is also the case for the highest intensity regime considered here. The only intensity that seems problematic is $8 \cdot 10^{13}\,\text{W}\,\text{cm}^{-2}$. This regime was already problematic for the mapping of the intensities of simulation and experiment because the differences are high. Note that the approach not to break bonds due to charge behaves a little outstanding compared to the other approaches. This might be due to the high stability of the molecule.

To do the ranking, we want to composite the results into one number per bond-breaking approach. We again use the mapping of Table 6.2 to get q_{mean} for each intensity. We sum up the squared difference values within the neigboring three entries per bond-breaking approach. These temporary results are afterwards summed to one value. The result of this calculation is shown in Table 6.3. The best approach regarding this ranking is the one that immediately breaks two bonds if an atom was ionized. Contrary, the approach not to break bonds due to charge becomes the worst.

6.4 Analysis of Fragment Dynamics

In former sections, we chose the methods that suits the experimental data best. In this section, we mostly use the best ranked bond-breaking approach immediately2-bond for further analysis of the fragmentation dynamics of C_{60}. Here, we forgo presenting ordinary molecule properties like the charge or the velocity of fragments but rather focus on pointing out the molecule dynamics. Keep in mind that we are analyzing simulation results that did not agree well with the experimental data as pointed out in Section 6.3. However, the analysis approach is independent of the data and may also be applied on different data sets.

6.4.1 Simulation Results for Different Intensities

In this section, we want to overview the simulation for several laser intensity parameters q_{mean} over time in Figure 6.6. This serves us insights into typical scenarios for the intensity regimes, we have listed in Table 6.2.

For the first two parameter values $q_{\text{mean}} = 0.5, 2.5$, we cannot see a big difference of the molecule. The simulation result for $q_{\text{mean}} = 10$ shows that two single atoms are repelled from the molecule. The rest of the atoms remained in an rather large fragment, even if many bonds are broken. With increasing intensity the dynamics increases dramatically. For $q_{\text{mean}} = 20$ fragments of all kind of sizes are repelled through space. A quite large fragment still remains intact. In contrast, the C_{60} molecule completely scatters in a Coulomb explosion. Most of the fragments are single carbon atoms and only few smaller fragments survive.

In general, we observe that the mean velocity of the fragments is increasing. This is indicated by the vanished atoms at the same frame of reference for each time step. The number of fragments is clearly increasing as well. Also note that the corresponding diffraction images have also been computed for the snap shots in Figure 6.6. They are presented in Figure 6.10.

6.4.2 Modes

In the physical background (Section 3.2) some vibrational modes are mentioned that have been observed in experiments. While experimenting with different sorts of ion constellations, we were able to reconstruct approximate oscillation patterns that fit to the description of the vibrational modes. The performed simulations were set up

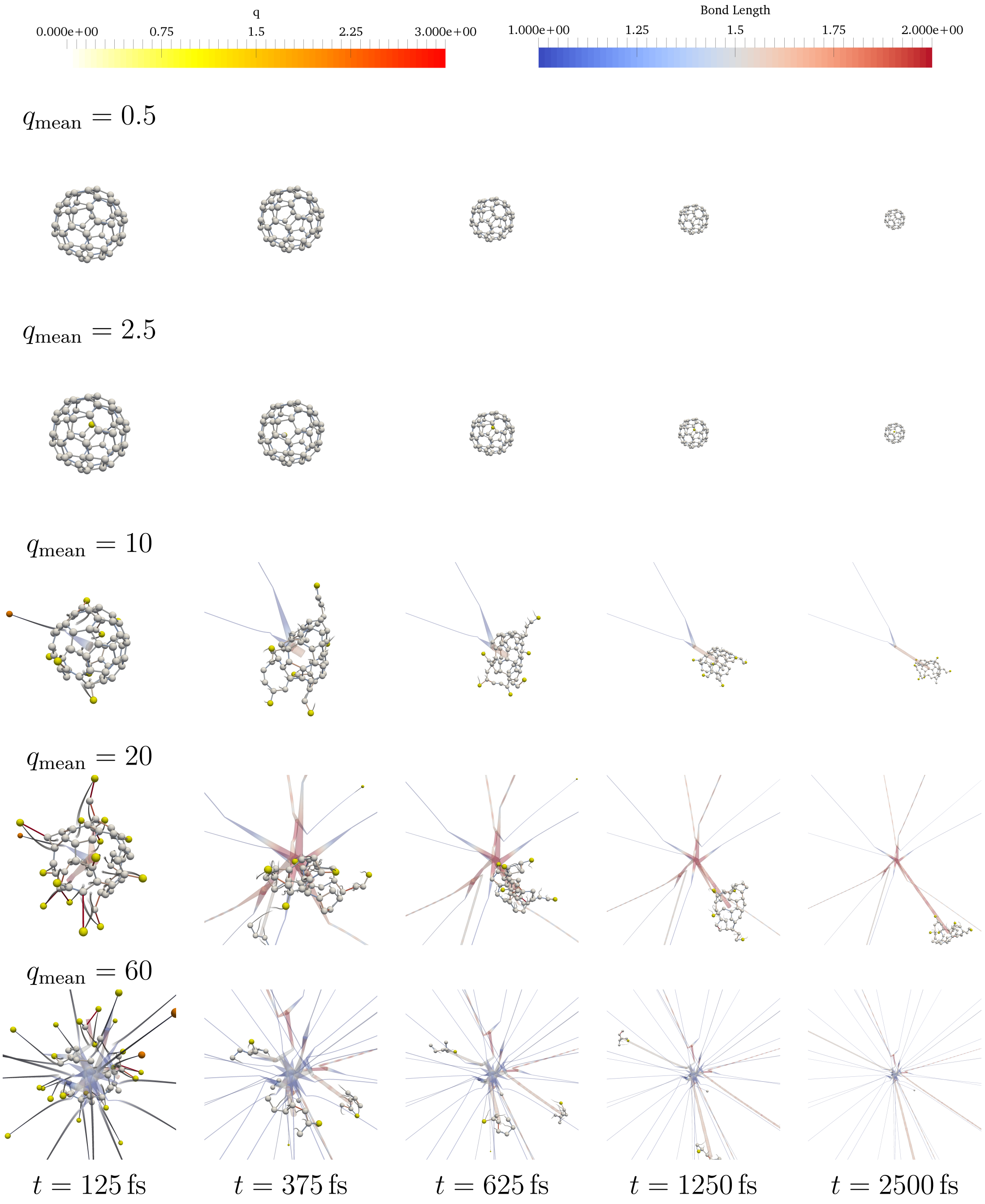


Figure 6.6: Exemplary simulation results for multiple intensity parameters q_{mean} with zoomed-out view over time. Atoms are colored by charge, bonds by length, and the semitransparent fragment tree colored by the maximum bond length of the fragment. The radii of the tubes of the fragment tree are additionally varied depending on the corresponding fragment radius. The bond length is given in Å, the charge q in elementary charge units.

Oscillation Period [fs]	Simulation	[DDE96]	Abs. Error	Rel. Error [%]
Breathing Mode	85	67	18	26.87
Prolate/Oblate Mode	170	122	48	39.34

Table 6.4: Oscillation period comparison between simulation results and theoretical calculations [DDE96] with absolute (abs.) and relative (rel.) errors.

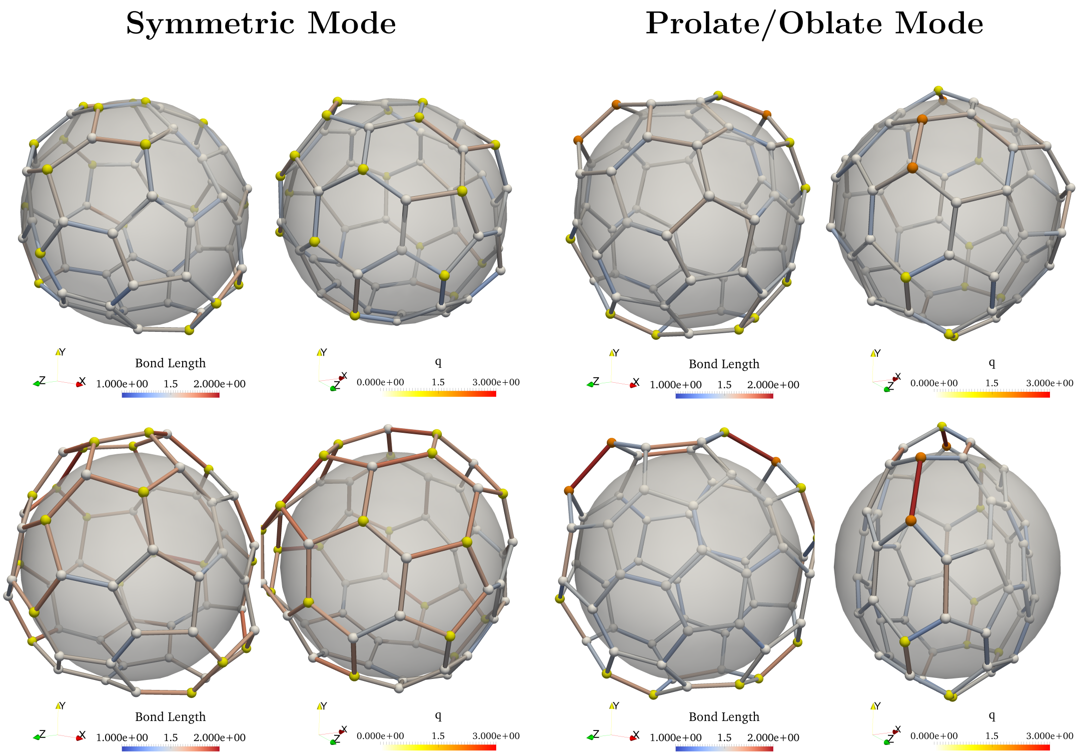
without bond-breaking due to charge. We thereby demonstrate examples for the undesired high stability stability of the molecule and use it as an advantage to show how the Coulomb forces are able to deform C_{60} molecules that posses all of its bonds.

Consider Figure 6.7 to see the vibrational modes. Both simulations started from the atom constellation in rest. The ionization charges are set immediately in the beginning for all selected atoms. The ionization charge has been chosen such that an additional charge would cause bond-breaking due to too long bond lengths. The Coulomb forces causes the molecule to deform in the beginning of the simulation to the bottom states of the figure. Afterwards, the binding forces drive the molecule back together to the top state of the figure. Starting from this state at time t_1, we measure the next times to reach the bottom state at t_2 and the following top state at t_3. From theses times, we determined the oscillation period $t_3 - t_1$. t_2 has additionally be given to show possibly occurring unsymmetrical time spawns distributions within an oscillation period. Note that the oscillations are not perfectly symmetrical and the atom constellation varies in between multiple oscillation periods. Even the individual atom constellations in Figure 6.7 are only approximately symmetric.

We can use the oscillation period to compare the simulation results with the results of the theoretical calculations of Dresselhaus et al. [DDE96], see Table 6.4. We can also relatively compare the intensity for both modes. The energy needed for the breathing mode according to Dresselhaus et al. [DDE96] is $498\,\mathrm{cm}^{-1}$, the one for the prolate/oblate mode is $273\,\mathrm{cm}^{-1}$. Since the intensity correlates with the ionized charge, we can use it as a measure for the intensity in our simulation. Their relative difference is shown in Eq. 6.3.

$$\frac{498\,\mathrm{cm}^{-1}}{273\,\mathrm{cm}^{-1}} = 1.82, \quad \frac{22\,e}{15\,e} = 1.47, \quad \text{rel. error: } \frac{|1.82 - 1.47| \cdot 100}{1.82} = 19.41\% \tag{6.3}$$

Despite medium high errors the simulation is able to qualitatively reproduce the periods of the vibrational modes, observed in experiments. This is noticeable since the agreement with the time-of-flight measurements is far worse. The statement that



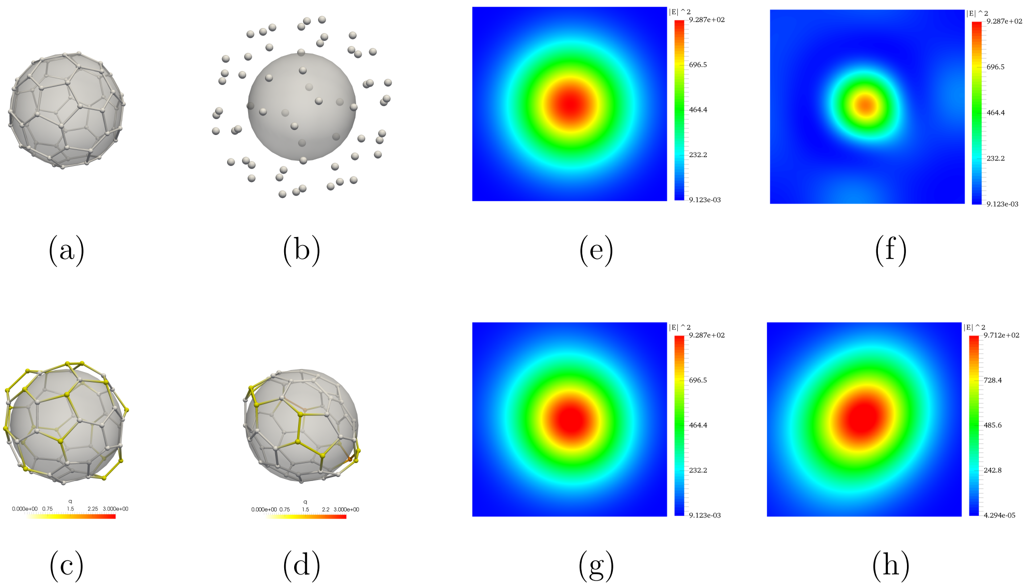
Symmetric Mode

Prolate/Oblate Mode

C_{60}^{22+}, oscillation period: 85 fs
$t_1 = 170\,\text{fs},\ t_2 = 217.5\,\text{fs},\ t_3 = 255\,\text{fs}$
$t_2 - t_1 = 47.5\,\text{fs},\ t_3 - t_2 = 37.5\,\text{fs}$

C_{60}^{15+}, oscillation period: 170 fs
$t_1 = 165\,\text{fs},\ t_2 = 250\,\text{fs},\ t_3 = 355\,\text{fs}$
$t_2 - t_1 = 85\,\text{fs},\ t_3 - t_2 = 105\,\text{fs}$

Figure 6.7: Ion configurations in C_{60} to mimic experimentally observed vibrational modes. The molecule in rest is positioned on the semitransparent sphere which provides an orientation (also see Figure 6.8). Each pair of a column shows the same molecule but at different views (note the coordinate system). Top and bottom views show the states the molecules is oscillating in between. **Left:** Symmetric breathing mode. Electrons are tried to be placed uniformly around the molecule. **Right:** Oblate mode on the top, prolate mode at the bottom. The electrons are placed in a ring, resulting in this vibrational modes. The bond length is given in Å, the charge q in elementary charge units.



(a) (b) (e) (f)

(c) (d) (g) (h)

Figure 6.8: C_{60} modes and their diffraction images. **Left:** Atom constellations of C_{60}. (a) The molecule in rest where a semitransparent sphere is fitted to the molecule, also shown for the other constellations. (b) Clearly larger molecule radius. (c) Expanded molecule in breathing mode. (d) Molecule in prolate mode. **Right:** (e)–(h) Diffraction images of the respective atom constellations (a)–(d). q is given in elementary charge units.

molecule dynamic simulations are a good approximation if no bonds are broken is strengthen by this observation (consider the discussion in Section 4.1.2).

We want to use these vibration mode examples as another verifying comparison, namely to judge the qualitative correctness of the diffraction images. Figure 6.8 shows atom constellations and their corresponding diffraction images.

Let us go on by comparing these results with the ones of the physicists that performed the experiments at LCLS, given in Figure 3.2. Since the methods to compute the diffraction images differ and different measures are shown, we only want to qualitatively compare the results. Initially, we have to consider the simulated experimental setup. This includes the photon energy, the distance between fullerene molecules and detector, and the extends of the detector. The physicists used an energy of 2 keV, a distance of 69 mm, and detector extends of 76.8 mm. Our computed images use the setup that really was used in the experiment. As also mentioned in Section 5.3.5, we use an energy of 1.8 keV and a distance of 140 mm. For the symmetric images detector extends of 140 mm are chosen. The computed diffraction images are equal for our computing method when multiplying the detector extends and distance by the same factor. Thus, we have an approximately equal setup.

The diffraction images for the atom constellation in rest looks like a symmetric for both images. For the breathing mode constellation (Figure 6.8 (c)), we get an image (Figure 6.8 (g)) that is very similar to Figure 6.8 (e), corresponding to the molecule in rest (Figure 6.8 (a)). In order to show the changes in the diffraction images for clearly larger radii, we used the simulation results of Section 6.2 (see Figure 6.1, $t = 0.0025\,\mathrm{ps}$). Figure 6.8 demonstrates the impact of the larger molecule radius. The values in the center are clearly reduced and the symmetric structure shrined. Finally, the diffraction image of the prolate mode is shown in Figure 6.8 (h). The structure of the image seems to be horizontally mirrored relative to the center of the molecule in Figure 6.8 (a). Note that we left out the oblate mode since its atom constellation is very close to the one in rest. It looks similar to the result of Figure 6.8 (g). Our method shows the same behavior like the method of the physicists and therefore showed its qualitative correct functioning.

6.4.3 Diffraction Images

Diffraction images are an interesting source for the comparison between experiment and simulation. In this work, we are not comparing images form experiments and simulation. Nevertheless, we provide the data for it. Another motivation to look at averaged diffraction images is to see if we can observe a certain trend in them.

Some facts should be taken into account for the comparison of the diffraction images of the simulation and the experiments. In our simulations, we reconstruct the diffraction image for one molecule per simulation run. Contrary, in the experiment a whole ensemble of molecules within a focal spot size of $60\,\mu\mathrm{m}$ is recorded. To imitate an ensemble of molecules, we want to accumulate and average over multiple diffraction images, even if their computation is costly. This step is necessary since the scattered laser pulses superimpose each other, which also happens in an ensemble of molecules in the experiment.

Diffraction Image Resolution

We have chosen the resolution of the diffraction images ($345\,\mathrm{px} \times 278\,\mathrm{px}$) as a quarter of the resolution of post processed images (added margin, see below), taken in the experiment $(n_{x,\,\mathrm{adapt}}, n_{y,\,\mathrm{adapt}}) = (1380\,\mathrm{px} \times 1112\,\mathrm{px})$. This makes pixel based comparison easy.

An exemplary image from the experiments is shown in Figure 3.4 with an original image resolution of $(n_{x,\,\mathrm{org}}, n_{y,\,\mathrm{org}}) = (1302\,\mathrm{px}, 1036\,\mathrm{px})$. Note that the laser was not perfectly aligned such that the scattering center for experimental diffraction images is not the center of the image. A 2D-Gaussian fit (white circles in Figure 3.4) of the data serves a prediction of the actual scattering center at image position $(c_{x,\,\mathrm{org}}, c_{y,\,\mathrm{org}}) = (612\,\mathrm{px}, 480\,\mathrm{px})$. The scattering center of the diffraction images form simulations are designed to also be the center of the images. We do not want to lose this implicit information, so for comparison, we have to add a margin to center the focal spot into the center of the image. Therefore, we determine the longest extend from the scattering center to the bounds of the image, see Eq. 6.4.

$$\begin{aligned} c_{x,\,\mathrm{adapt}} &= \max(|0 - c_{x,\,\mathrm{org}}|, |n_{x,\,\mathrm{org}} - c_{x,\,\mathrm{org}}|) = \max(612, |1302 - 612|)\,\mathrm{px} = 690\,\mathrm{px} \\ c_{y,\,\mathrm{adapt}} &= \max(|0 - c_{y,\,\mathrm{org}}|, |n_{y,\,\mathrm{org}} - c_{y,\,\mathrm{org}}|) = \max(480, |1036 - 480|)\,\mathrm{px} = 556\,\mathrm{px} \end{aligned} \tag{6.4}$$

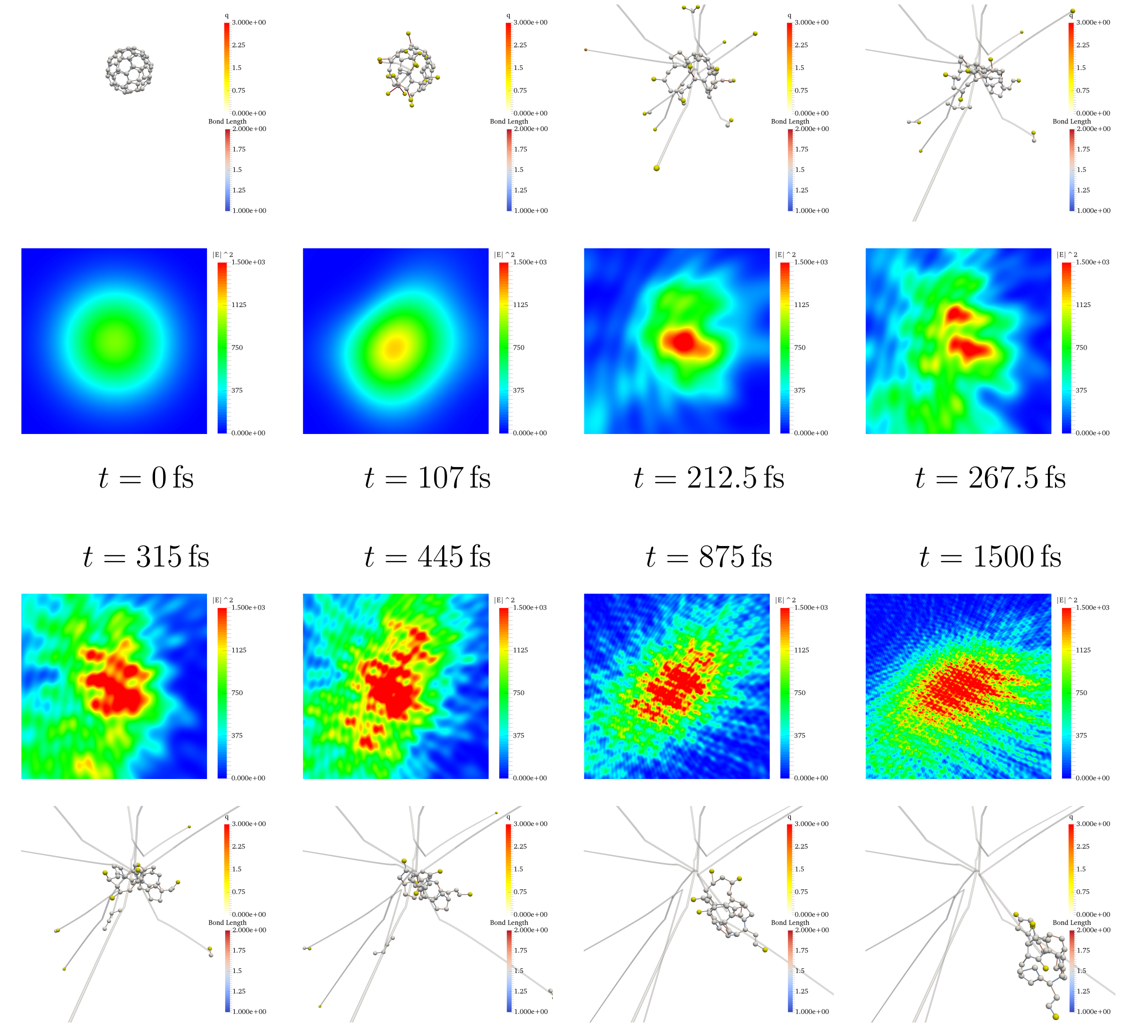
With these extends, we obtain a new image resolution where the scattering center lies also in the center of the image, see Eq. 6.5.

$$n_{x,\,\mathrm{adapt}} = 2 \cdot c_{x,\,\mathrm{adapt}} = 1380, \quad n_{y,\,\mathrm{adapt}} = 2 \cdot c_{y,\,\mathrm{adapt}} = 1112 \tag{6.5}$$

Compared to the other visualization techniques the computation of the diffraction images is computationally rather expensive. The complexity depends on the number of computed pixels. The computation time for a single diffraction image of resolution 1380 × 1112 takes approximately 13 s whereas the computation for an image of the quartered resolution 345 px × 278 px takes approximately 0.8 s (speedup of approximately $4^2 = 16$). Considering the various parameters in a data set, we have to trade off between resolution and number of simulation runs. We set the resolution to 345 px × 278 px which still serves an agreeable resolution.

Time Dependency

Let us start looking at results by considering the diffraction images of a single simulation over time. Figure 6.9 shows the molecule and their diffraction image during the ionization process of one simulation at different times. In the beginning, we see a circular structure with an increasing intensity towards the center. First ionizations cause atoms to leave the fullerene molecule which results in a deformation of this circular structure ($t = 107\,\mathrm{fs}$). Starting from time $t = 212.5\,\mathrm{fs}$, we can see



$t = 0\,\mathrm{fs}$ $t = 107\,\mathrm{fs}$ $t = 212.5\,\mathrm{fs}$ $t = 267.5\,\mathrm{fs}$

$t = 315\,\mathrm{fs}$ $t = 445\,\mathrm{fs}$ $t = 875\,\mathrm{fs}$ $t = 1500\,\mathrm{fs}$

Figure 6.9: Diffraction images and atom constellations for one simulation run ($q_{\mathrm{mean}} = 20$).

wave front like sections of circles, spread all over the image. They may be caused by the small fragments that are drifted apart from the rest of the molecule by now. Also note that the region of high intensity spreads to a larger extend between $t = 212.5\,\mathrm{fs}$ and $t = 267.5\,\mathrm{fs}$. Generally, we observed that the size of this region is fluctuating by animations with a higher time resolution. In the bottom row the region of the highest intensity is changing its shape depending on the atom constellation. More interestingly, the structure punctually becomes more intense where the punctual radii become smaller and smaller. Meanwhile, the largest molecule fragment is drifting towards to the observer. Regarding the times in the bottom row, we see that the steps are getting coarser. The dynamic of the image becomes fewer over time. The last image of the figure is shown for time $t = 1500\,\mathrm{fs}$. The last computed image at time $t = 2500\,\mathrm{fs}$ is of the same character like the on at time $t = 1500\,\mathrm{fs}$.

Dependence of Intensity in Single Simulations

In this section, we want to overview some diffraction image results for multiple intensity parameters q_{mean} with respect to the simulation time. The diffraction images correspond to the results, presented in Figure 6.6.

Let us start with the top row. In the simulation there has been ionized no atom and the diffraction images look identical. In the row below with parameter $q_{\mathrm{mean}} = 2.5$ where 1 atom was ionized by one electron charge, we can see slight changes in the second image from left and in the right most image. Further investigation of the data showed that the molecule that lost two of its bonds indeed started oscillating in the breathing mode. The maximum radius configuration of the circular structure in the diffraction image remained unchanged until the time where the molecule is ionized, namely $t = 87.5\,\mathrm{fs}$. The minimum of the radius in the diffraction image (and the maximum of the molecule radius) is observed at time $t = 400\,\mathrm{fs}$ and again the maximum radius in the diffraction image at time $t = 737.5\,\mathrm{fs}$. This makes a oscillation period of $\Delta t = 650\,\mathrm{fs}$. Note that this is an approximate estimate since the times were determined by eye.

In the next row ($q_{\mathrm{mean}} = 10$), we come to the regime of fragmentation. First, the circular structure becomes deformed while building an intensity peak (yellow). At the time of the second column two atoms already left the molecule due to Coulomb forces. In addition, the molecule broke up and increased the projected area onto the image plane. Contrary, the structure in the diffraction image becomes smaller and elongated. In consideration of all images in the row it seems like the more sphere

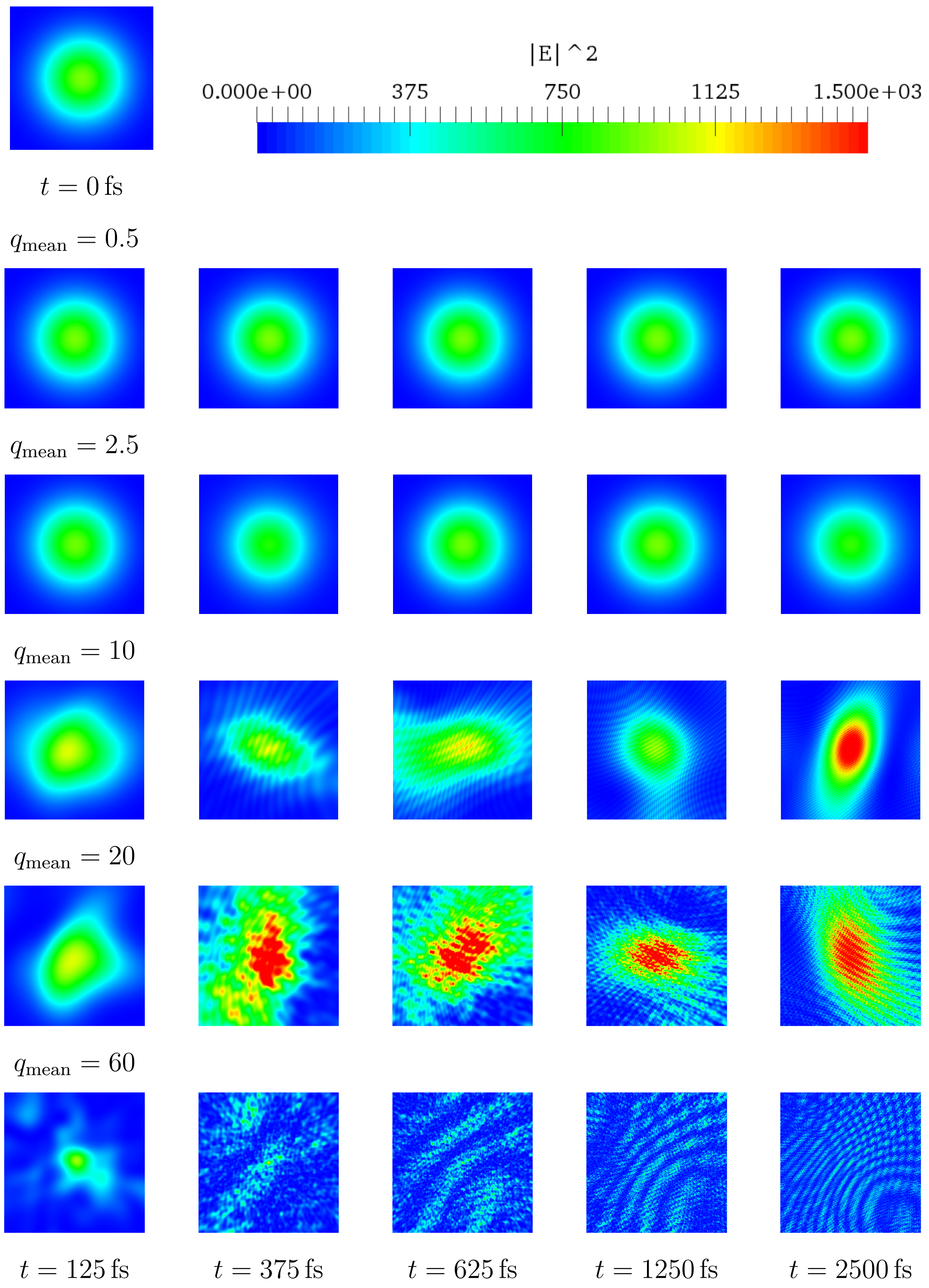


Figure 6.10: Diffraction images of the simulations in Figure 6.6 for multiple intensity parameter q_{mean} over time. At the top the diffraction image in the beginning is shown as reference.

like the shape of the molecule, the more the circular pattern in the diffraction image appears. Considering the right most image, we remark that the intensity spot in the center is wide and high. We are not sure why this is the case for this particular molecule configuration which seems to be similar to the other configurations. There is also some wave like pattern around the area of the molecule which becomes more peaky during time.

The results for the simulation with parameter $q_{\text{mean}} = 20$ looks similar to the one before. Here, we note that the high intensity peaks appear earlier and are spread on a wider area. The noise pattern around the area of the high intensity starts becoming more peaky and is also earlier visible.

In the last row the intensity becomes so high that it starts exploding. This is also visible in a clearly different pattern on the diffraction image. At first, the area we already observed in the former rows becomes significantly smaller while the distance of each atom to the center becomes larger. At the second time a very tiny peak is still visible. The rest of the image looks very random except a diagonal from lower left to the center of the image. In the next image, however, we see a coarse wave-like structure, which looks more distant the further time passes. The last image reminds of two circular expanding waves that superimpose each other with some background noise.

Dependence of Intensity for Multiple Simulations

Now that we have learned a bit about the connection between atom constellations and diffraction images, we consider averaged diffraction images. We first want to learn how many simulation runs are necessary to obtain a converging average of diffraction images if we are even able to find any. Therefore, we computed two data sets D2, D3 of diffraction images. Figure 6.11 overviews diffraction images for multiple ionization intensities of the simulation q_{mean} vertically and multiple number of averaged images horizontally. Note that the four left most columns are computed by simulation results of the data set D2 whereas the right most column is computed from the data set D3. This serves an exemplary example of the difference of the results of two data sets.

Let us begin with the top row. For really low intensities (here $q_{\text{mean}} = 0.5$), we hardly see any difference between the multiple numbers of averaged simulation runs. The results for $q_{\text{mean}} = 2.5$ look similar to the ones in the first row, however, we see slight differences, especially considering the last two columns.

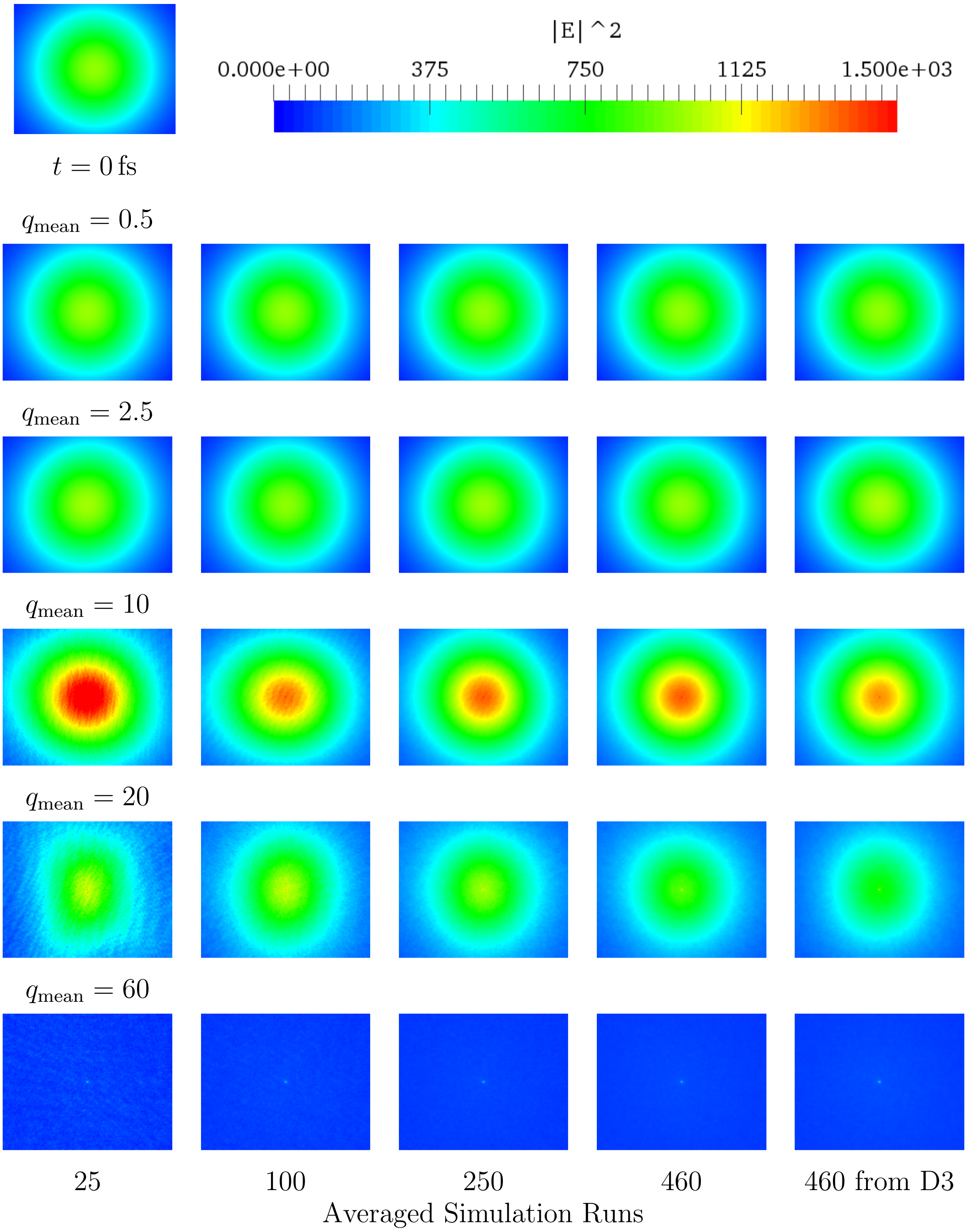


Figure 6.11: Diffraction images for multiple intensities, averaged over multiple numbers of images (simulation runs), showing the result of the last time step ($t = 2.5\,\text{ps}$). The diffraction image at the beginning is shown as reference. The left most columns are calculated by the data set D2, the right most column by the data of D3 (see Section 6.1). Note the highest difference for $q_{\text{mean}} = 10, 20$ when comparing the two right most columns with the highest number of averaged images.

Starting from $q_{\mathrm{mean}} = 10$, we see significant changes in the behavior. The circular structure appears to be slightly larger than the previously discussed ones for all results in this row. From the mode section above, we learned that this indicates a shrinkage of the radius of the molecule. Let us fist consider the image with only 25 averaged images. First of all note that we see a wave like noise all over the image. This indicates that the molecule might have teared into small pieces in some simulations, where a larger molecule is mostly intact. We observed the same behavior in Figure 6.9, $t = 212.5\,\mathrm{fs}$). There are also high intensity peaks in the center of the image in both images. Maybe an atom constellation as shown in Figure 6.9, $t = 212.5\,\mathrm{fs}$ is common. For additional simulation runs the structure of the images changes. On the one hand, the low frequent noise partly vanishes for example in the regions with low intensity. This might be explained by the fact that small fragments which probably cause this noise fly in different directions in each simulation and if their position does not superimpose positions in other simulations their noise contribution is diluted by the averaging. On the other hand, we can see higher frequent noise in the center of the images. The frequency increases with the number of simulations.

Let us continue with the following row, corresponding to the intensity parameter $q_{\mathrm{mean}} = 20$. We can see a clearly a more deformed pattern in the image for few simulation runs. More simulations show the trend that the circle becomes uniformer again. Note that the differences between the two data sets D2 and D3 are the highest in this row.

In the bottom row, we again see noisy images which is reduced with increasing number of diffraction images while an intensity peak is located in the center. Noticeablely, these peaks can also be seen for $q_{\mathrm{mean}} = 20$. There they are very sharp. We can also slightly see them for $q_{\mathrm{mean}} = 10$ with many simulation runs. Regarding the simulation results for high intensities in Section 6.4.1, we can be pretty sure that the molecule decayed in a strong Coulomb explosion. This point thus indicates the intensity of the laser pulse itself, which is no longer scattered by the molecule, at least at the final time $t = 2.5\,\mathrm{ps}$.

In the following section, we average all diffraction images from both data sets D2 and D3 together. There are slightly visible changes for $q_{\mathrm{mean}} = 20$ and $q_{\mathrm{mean}} = 10$.

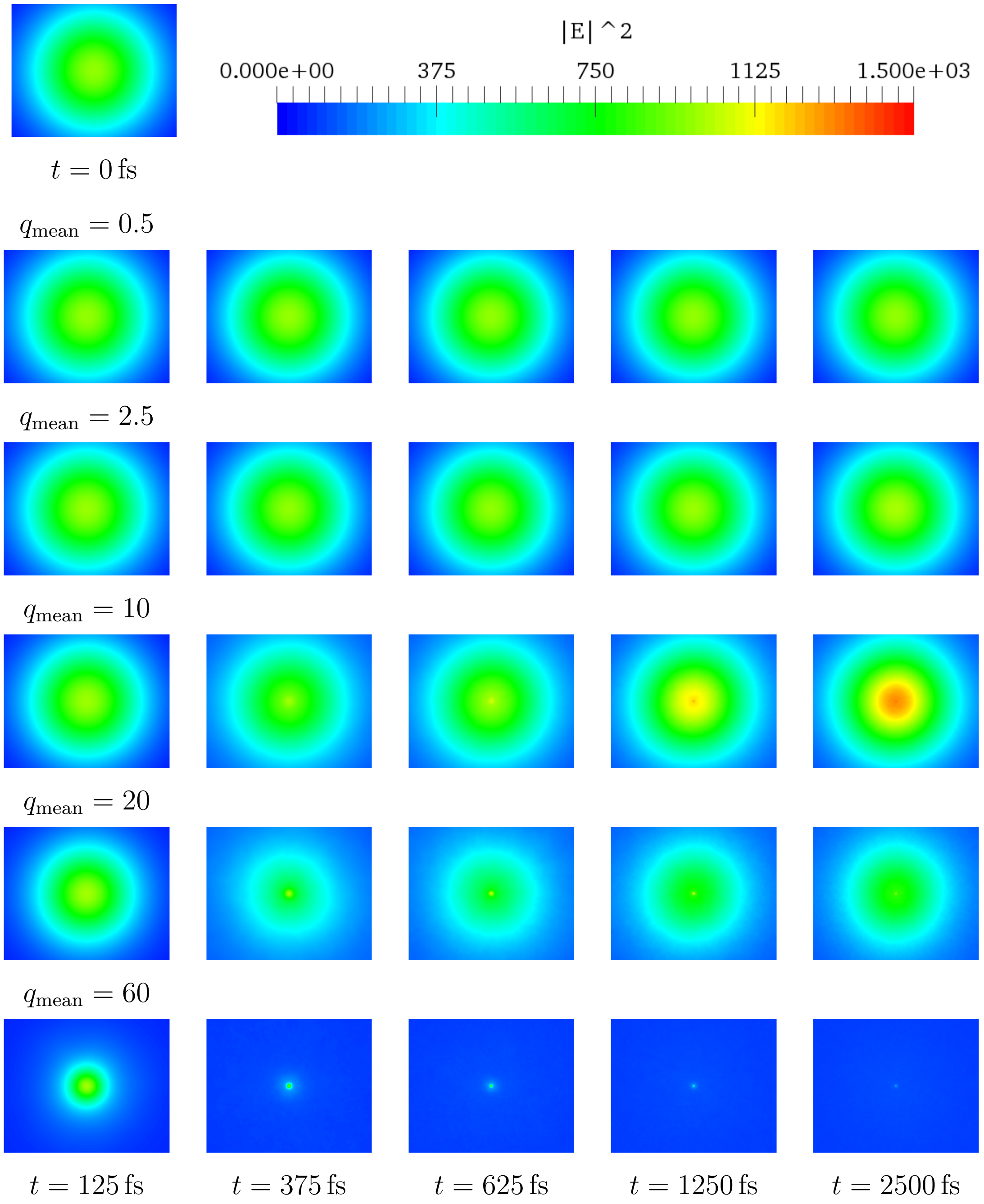


Figure 6.12: Diffraction images for multiple intensity parameter q_{mean}, averaged over data sets D2 and D3 (see Section 6.1). At the top the diffraction image in the beginning is shown as reference.

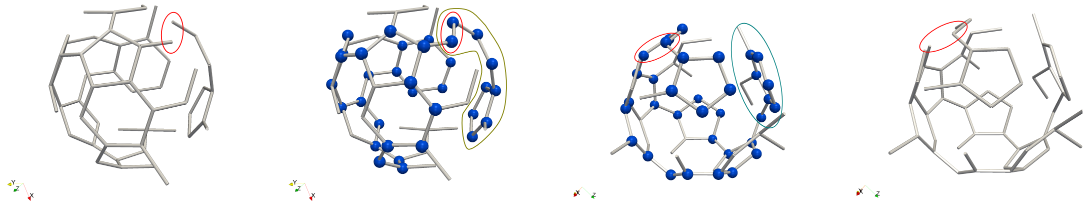


Figure 6.13: Example for a big fragment hole of size 29. The two outer and inner images show the same hole from a different viewing directions. The red ellipses show which edge is removed from the tow images in the middle. **Middle:** The blue spheres depict the vertices belonging to the hole. There are two stubs, one shown on both sides. **Outside:** A removed edge made the hole disappear.

Results for Highest Number of Simulation Runs

Figure 6.12 shows the final averaging results, which we present for multiple intensities q_{mean} at multiple times t. The two rows at the top hardly show differences during time. For $q_{\text{mean}} = 10$, we see a different behavior. The blurry intensity peak in the center becomes stronger during time. Moreover, a sharp peak builds in the center. As discussed above this indicates that the molecule already decayed in some simulations at that time. The row corresponding to $q_{\text{mean}} = 20$ shows an interesting intensity distribution. For $t = 125\,\text{fs}$ the behavior is similar to the one for $q_{\text{mean}} = 10$. The peak described in the previous row is built earlier. The size of the circular structure however first shrinks and grows afterwards. This might also be the case for earlier discussed rows, but was not visible before. The bottom row is again easy to interpret. The structure in the image continuously becomes smaller over time. This is due to a Coulomb explosion. Note that the noise in the image is relatively high for $t = 375\,\text{fs}$ and decreases over time. The noise is probably caused by fragments that are located near the center. There, an amplification of their caused waves is more probable since their positions are closer together.

6.4.4 Fullerene Hole Analysis

We introduced fragment holes in Section 5.3.6 as a measure of the state or stability of a fragment. The method works well in most cases and the count of the holes is probably correct. The complexity of the code and the variety of possibilities makes the validation of the method hard. Unfortunately, during the validation of the method, we encountered an example where our method did not work as intended.

In the context of this work, we were not able to fix this mistake.

We focused on a special case, in which a big hole of a fragment was created and vanished after some steps. Figure 6.13 shows a fragment with a fragment hole of size 29. The top row shows the molecule before, the bottom row after the hole vanishes from two views (left and right). All vertices belonging to the hole are shown as blue spheres. Note that the method also counts stubs that contain polygons (in conflict to definition!). In Figure 6.13 we see two of theses stubs, marked in two different colors. The removal of the edge, marked in red, disconnected one stubs. This causes the hole to disappear instead of an expected resize of it. We therefore cannot guarantee that all holes are detected correctly. As far as we know, the hole size is detected correctly.

Since we have invested so much effort into this method, we want to outline the idea for the analysis of the fragment holes. Therefore, we consider Figure 6.14. Here, we present the averaged hole size yield per simulation. In the corresponding simulations the ionization started at $t = 0$ and ended $t = 160\,\mathrm{fs}$. We provide the results for a single simulation on the top. We can see that at first small holes are built. Even for a single simulation the combination of hole sizes and length are complex to understand. Noticeable is that certain hole sizes are skipped, e.g., 15. This may be due to the fact that especially initially, the possible configurations for first holes is limited. This trend might accumulated during time. After $t = 100\,\mathrm{fs}$ The molecule remains in a configuration with a small hole.

Considering the results for multiple simulation runs, we can see that with increasing intensities the yield of the hole sizes moves to earlier times. Generally, large hole sizes are rather rare in comparison to the smaller ones. In the case of the highest intensity almost all holes vanish after the pulse hit the molecule.

We again want to remark that these results have not been validated completely and might contain wrong results.

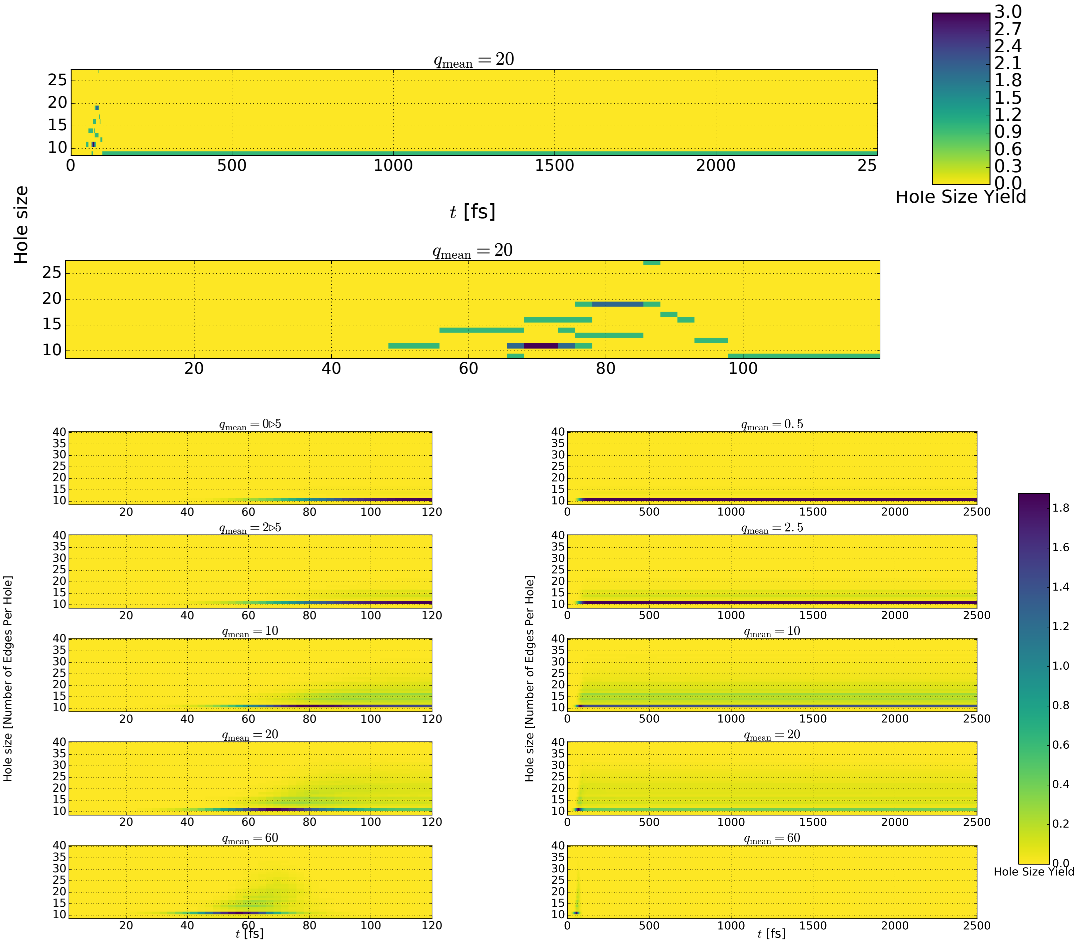


Figure 6.14: The hole size yield (see color bar) with respect to the hole size of fragments (y-axis) against time (x-axis). **Top:** Results for a single simulation with whole time interval in the upper and a zoom in the lower part. **Bottom:** Hole size yield averaged over 1000 simulation runs (left: zoom, right: hole interval). Note that these are not completely validated results.

7 Conclusion

In this work, we performed simulations and visualizations of the fullerene. In the simulation part, we used the molecule dynamic framework LAMMPS to simulate experiments, in which laser pulses interact the with fullerene. We modeled the fullerene by a force field that included bonds which are driven by a binding potential, and moreover the bending, dihedral, and Coulomb potentials. These models showed a stable behavior without the ionization of the molecule. The intensity of the laser pulses was assumed to have a Gaussian distribution. We further assumed that the fullerene-laser interaction causes the atoms of the molecule to be ionized. More precisely, we modeled this process by ionizing an atom with a probability that is assumed to be proportional to the laser pulse intensity. The used probability function is a truncated normal distribution with an intensity parameter q_{mean} of the simulation that determines how many atoms are ionized in fullerene for one laser pulse.

Fullerenes become unstable with increasing ionization charge. Once these charges are too high, the molecule starts to decay into fragments. For the separation of fragments from the original molecule, we broke too long bonds and identified separated fragments by the connected component algorithm. This lead to our first analysis.

In order to verify the plausibility of the simulation results, we wanted to compare the simulation results with the experiments of the physicists. In their experiments, they performed time-of-flight measurements. These measurements serve the information of the fraction between charge and mass of an ion, which is used to identify ionized molecule fragments, the products of the fullerene fragmentation process. For the comparison, we performed an analysis, in which we reconstructed time-of-flight data from our simulation data. The results in the low and partly also the high laser intensity regime approximately agreed with the experiments. Unfortunately, our simulation results strongly deviated from the measurements of the physicists in the medium laser intensity regime. We also looked for a mapping of the intensity between the simulation parameter q_{mean} and the true laser intensity of the experimental results. We determined a rough mapping by finding similarities in the time-of-flight histograms.

To improve the quality of the simulation results, we developed and investigated bond-breaking approaches due to charge during this work to adapt the simulation results to the physical experiments. Five approaches have been tested and compared with the result that the roughest approach, which immediately breaks two neighboring bonds if an atom was ionized. We used this approach for further analysis.

In the visualization part, we first introduced the representation of a molecule in the visualization, followed by discussing a taxonomy of possible molecule visualizations. This brought us to the visualization contribution of this work. We presented direct techniques that simply depict the atoms and bonds of a molecule as they are. Methods suited for the dynamic analysis are the atom trajectories and the fragment tree. They depict the trajectories of single atoms and interconnected fragments, respectively. The fragment tree has been filled with various information to also illustrate the properties of the fragments for further insights. We then introduced the depiction of a laser pulse for the illustration of the process of the experiment and for an animating demonstration.

For the analysis of dynamics, we first investigated modes of the fullerene molecule. The simulation was able to show oscillation modes that have been observed in experiments. Even the oscillation periods and the relative intensity agreed qualitatively with theoretical calculations. The relative error with about 25–40% was moderate. We then came to the analysis of the fragmentation process. Here, our diffraction images, computed from the simulations, have been investigated in detail. They are interesting in the way that they can be and also were measured in the physical experiments. We showed the results for multiple scenarios, for single simulations over time, for multiple intensities, and also for diffraction images, averaged over multiple simulation runs.

Our hole tree analysis would have been an interesting measure to investigate, which however, we were not able to complete due to the lack of validation. We hope that we can provide some results in future work.

Bibliography

[ABE$^+$13] Logan S. Ahlstrom, Joseph L. Baker, Kent Ehrlich, Zachary T. Campbell, Sunita Patel, Ivan I. Vorontsov, Florence Tama, and Osamu Miyashita. Network visualization of conformational sampling during molecular dynamics simulation. *Journal of Molecular Graphics and Modelling*, 46:140–149, 2013. doi:10.1016/j.jmgm.2013.10.003.

[AGL05] James Ahrens, Berk Geveci, and Charles Law. Paraview: An end-user tool for large data visualization. 01 2005.

[Aya15] Utkarsh Ayachit. *The ParaView Guide: A Parallel Visualization Application*. Kitware, Inc., USA, 2015.

[BCR03] V. R. Bhardwaj, P. B. Corkum, and D. M. Rayner. Internal laser-induced dipole force at work in c_{60} molecule. *Phys. Rev. Lett.*, 91:203004, Nov 2003. doi:10.1103/PhysRevLett.91.203004.

[BdMB15] Z. Berkai, M. daoudi, N. Mendil, and A. Belghachi. Theoretical study of fullerene (C_{60}) force field at room temperature. *Energy Procedia*, 74:59–64, 2015. doi:10.1016/j.egypro.2015.07.521.

[BWF$^+$00] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235, 2000. URL: www.rcsb.org/ligand/60C, doi:10.1093/nar/28.1.235.

[Car05] François Cardinali. *Synthèse de nouveaux dérivés du C_{60} et de complexes de métaux de transition à bande de transfert de charge*. PhD thesis, Université Louis Pasteur, France, 2005.

[CRG$^+$14] Matthieu Chavent, Tyler Reddy, Joseph Goose, Anna C. E. Dahl, John E. Stone, Bruno Jobard, and Mark S. P. Sansom. Methodologies for the analysis of instantaneous lipid diffusion in MD simulations of large membrane systems. *Faraday discussions*, 169:455–475, 2014.

[CW98] Jianshu Cao and Kent R. Wilson. Ultrafast x-ray diffraction theory. *The Journal of Physical Chemistry A*, 102(47):9523–9530, 1998. doi:10.1021/jp982054p.

[DDE96] Mildred S. Dresselhaus, Gene Dresselhaus, and Peter C. Eklund. *Science of fullerenes and carbon nanotubes: their properties and applications.* Academic press, 1996.

[Dem13] Wolfgang Demtröder. *Laser Spectroscopy - Basic Concepts and Instrumentation.* Springer Science & Business Media, Berlin Heidelberg, 2013. doi:10.1007/978-3-662-08257-7.

[dHCKMK13] Pablo de Heras Ciechomski, Michael Klann, Robin Mange, and Heinz Koeppl. From biochemical reaction networks to 3d dynamics in the cell: The zigcell3d modeling, simulation and visualisation framework. In *Biological Data Visualization (BioVis), 2013 IEEE Symposium on*, pages 41–48. IEEE, 2013. doi:10.1109/BioVis.2013.6664345.

[DMS+93] S. L. Dexheimer, D. M. Mittleman, R. W. Schoenlein, W. Vareka, X. D. Xiang, A. Zettl, and C. V. Shank. *In Ultrafast Phenomena VIII.* Springer-Verlag, Berlin, 1993.

[EKK+14] Thomas Ertl, Michael Krone, Stefan Kesselheim, Katrin Scharnowski, Guido Reina, and Christian Holm. Visual analysis for space–time aggregation of biomolecular simulations. *Faraday discussions*, 169:167–178, 2014. doi:10.1039/C3FD00156C.

[FKRE10] Martin Falk, Michael Klann, Matthias Reuss, and Thomas Ertl. 3D visualization of concentrations from stochastic agent-based signal transduction simulations. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1301–1304, 2010. doi:10.1109/ISBI.2010.5490235.

[FMKT03] Akira Funahashi, Mineo Morohashi, Hiroaki Kitano, and Naoki Tanimura. CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. Biosilico, 1(5):159–162, 2003. doi:10.1016/S1478-5382(03)02370-9.

[FSTR13] Matthew Fioravante, Adam Shook, Ian Thorpe, and Penny Rheingans. Visualizing motional correlations in molecular dynamics using geometric deformations. *Computer Graphics Forum*, 32(3):311–320, 2013. doi:10.1111/cgf.12118.

[GBD+05] Martyn F. Guest, Ian J. Bush, Huub J. J. Van Dam, Paul Sherwood, Jens M. H. Thomas, Joop H. Van Lenthe, Remco W. A. Havenith, and John Kendrick. The GAMESS-UK electronic structure package: algorithms, developments and applications. *Molecular Physics*, 103(6-8):719–747, 2005. doi:10.1080/00268970512331340592.

[HCF+94] H. Hohmann, C. Callegari, S. Furrer, D. Grosenick, E. E. B. Campbell, and I. V. Hertel. Photoionization and fragmentation dynamics of c_{60}. *Phys. Rev. Lett.*, 73:1919–1922, Oct 1994. doi:10.1103/PhysRevLett.73.1919.

[HDS96] William Humphrey, Andrew Dalke, and Klaus Schulten. VMD – Visual Molecular Dynamics. *Journal of Molecular Graphics*, 14:33–38, 1996. doi:10.1016/0263-7855(96)00018-5.

[HHB+91] Kenneth Hedberg, Lise Hedberg, Donald S. Bethune, C. A. Brown, H. C. Dorn, Robert D. Johnson, and M. De Vries. Bond lengths in free molecules of buckminsterfullerene, C_{60}, from gas-phase electron diffraction. *Science*, 254(5030):410–412, 1991. doi:10.1126/science.254.5030.410.

[HLS05] I. V. Hertel, T. Laarmann, and C. P. Schulz. Ultrafast excitation, ionization, and fragmentation of C_{60}. *Advances in Atomic, Molecular, and Optical Physics*, 50:219–286, 2005. doi:http://dx.doi.org/10.1016/S1049-250X(05)80010-2.

[HT73] John Hopcroft and Robert Tarjan. Algorithm 447: Efficient algorithms for graph manipulation. *Commun. ACM*, 16(6):372–378, June 1973. URL: http://doi.acm.org/10.1145/362248.362272, doi:10.1145/362248.362272.

[JBS+17] Andreas C. Jakowetz, Marcus L. Böhm, Aditya Sadhanala, Sven Huettner, Akshay Rao, and Richard H. Friend. Visualizing excitations at buried heterojunctions in organic semiconductor blends. *Nature Materials*, 2017. `doi:10.1038/nmat4865`.

[JKB94] Norman L. Johnson, Samuel Kotz, and N. Balakrishnan. *Continuous univariate distributions*. Wiley, New York, 2. edition, 1994.

[KFS+17] Michael Krone, Florian Frieß, Katrin Scharnowski, Guido Reina, Silvia Fademrecht, Tobias Kulschewski, Jürgen Pleiss, and Thomas Ertl. Molecular surface maps. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):701–710, Jan 2017. `doi:10.1109/TVCG.2016.2598824`.

[KHO+85] Harold W. Kroto, James R. Heath, Sean C. O'Brien, Robert F. Curl, Richard E. Smalley, et al. C_{60}: buckminsterfullerene. *Nature*, 318(6042):162–163, 1985. `doi:10.1038/318162a0`.

[KKF+16] B. Kozlíková, M. Krone, M. Falk, N. Lindow, M. Baaden, D. Baum, I. Viola, J. Parulek, and H.-C. Hege. Visualization of biomolecular structures: State of the art revisited. *Computer Graphics Forum*, 2016. `doi:10.1111/cgf.13072`.

[LM15] Benedict Leimkuhler and Charles Matthews. *Molecular Dynamics: With Deterministic and Stochastic Numerical Methods*. Interdisciplinary Applied Mathematics. Springer, 5 2015. `doi:10.1007/978-3-319-16375-8`.

[LMWPV15] Mathieu Le Muzic, Manuela Waldner, Julius Parulek, and Ivan Viola. Illustrative timelapse: A technique for illustrative visualization of particle-based simulations. In *Visualization Symposium (PacificVis), 2015 IEEE Pacific*, pages 247–254. IEEE, 2015. `doi:10.1109/PACIFICVIS.2015.7156384`.

[LV02] Chang H. Lee and Amitabh Varshney. Representing thermal vibrations and uncertainty in molecular surfaces. In *Electronic Imaging 2002*, pages 80–90. International Society for Optics and Photonics, 2002. `doi:10.1117/12.458813`.

[Mar95] George Marsaglia. The Marsaglia random number cdrom with the diehard battery of tests of randomness. 1995.

[MOJ+14] B. F. Murphy, T. Osipov, Z. Jurek, L. Fang, S. K. Son, M. Mucke, J. H D Eland, V. Zhaunerchyk, R. Feifel, L. Avaldi, P. Bolognesi, C. Bostedt, J. D. Bozek, J. Grilj, M. Guehr, L. J. Frasinski, J. Glownia, D. T. Ha, K. Hoffmann, E. Kukk, B. K. McFarland, C. Miron, E. Sistrunk, R. J. Squibb, K. Ueda, R. Santra, and N. Berrah. Femtosecond x-ray-induced explosion of C_{60} at extreme intensity. *Nature Communications*, 5, 6 2014. doi:10.1038/ncomms5281.

[Mon12] Luca Monticelli. On atomistic and coarse-grained models for C_{60} fullerene. *Journal of Chemical Theory and Computation*, 8(4):1370–1378, 2012. File last accessed 2017-03-02. URL: www.dsimb.inserm.fr/~luca/wp-content/files/fullerene/C60.pdb, doi:10.1021/ct3000102.

[MPSV14] Mathieu Le Muzic, Julius Parulek, Anne-Kristin Stavrum, and Ivan Viola. Illustrative visualization of molecular reactions using omniscient intelligence and passive agents. *Computer Graphics Forum*, 33(3):141–150, June 2014. Article first published online: 12 JUL 2014.

[NKI+12] Naoyuki Niitsu, Miyu Kikuchi, Hayato Ikeda, Kaoru Yamazaki, Manabu Kanno, Hirohiko Kono, Koichiro Mitsuke, Mikito Toda, and Katsunori Nakai. Nanosecond simulations of the dynamics of C_{60} excited by intense near-infrared laser pulses: Impulsive raman excitation, rearrangement, and fragmentation. *The Journal of chemical physics*, 136(16):164304, 2012. doi:10.1063/1.4704896.

[NWvdS+00] Richard Neutze, Remco Wouts, David van der Spoel, Edgar Weckert, and Janos Hajdu. Potential for biomolecular imaging with femtosecond X-ray pulses. *Nature*, 406(6797):752–757, 2000. doi:10.1038/35021099.

[ÖOD96] Yngve Öhrn, Juan Oreiro, and Erik Deumens. Bond making and bond breaking in molecular dynamics. *International Journal of Quantum Chemistry*, 58(6):583–591, 1996. doi:10.1002/(SICI)1097-461X(1996)58:6<583::AID-QUA4>3.0.CO;2-U.

[PKvdBS16] Eva Pavarini, Erik Koch, Jeroen van den Brink, and George Sawatzky. *Quantum Materials: Experiments and Theory*, volume 6 of *Modeling and Simulation*. Forschungszentrum Jülich, Jülich, Sep 2016. URL: http://juser.fz-juelich.de/record/819465.

[Pli95] Steve Plimpton. Fast parallel algorithms for short-range molecular dynamics. *Journal of Computational Physics*, 117(1):1–19, 1995. URL: http://lammps.sandia.gov, doi:10.1006/jcph.1995.1039.

[RBT14] Dmitri Rozmanov, Svetlana Baoukina, and D. P. Tieleman. Density based visualization for molecular simulation. *Faraday discussions*, 169:225–243, 2014. doi:10.1039/C3FD00124E.

[RCB77] Jean-Paul Ryckaert, Giovanni Ciccotti, and Herman J. C. Berendsen. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *Journal of Computational Physics*, 23(3):327–341, 1977. doi:10.1016/0021-9991(77)90098-5.

[RCMS12] Marzia Rivi, Luigi Calori, Giuseppa Muscianisi, and Vladimir Slavnic. In-situ visualization: State-of-the-art and some use cases. *PRACE White Paper*, pages 1–18, 2012.

[RJ99] Penny Rheingans and Shrikant Joshi. *Visualization of Molecules with Positional Uncertainty*, pages 299–306. Springer Vienna, Vienna, 1999. doi:10.1007/978-3-7091-6803-5_30.

[Sch11] Boris Schling. *The Boost C++ Libraries*. XML Press, 2011.

[Sch15] Schrödinger, LLC. The PyMOL molecular graphics system, version 1.8. November 2015.

[SEH05] Johannes Schmidt-Ehrenberg and Hans-Christian Hege. Visual analysis of molecular conformations by means of a dynamic density mixture model. In *International Symposium on Computational Life Science*, pages 229–240. Springer, 2005. doi:10.1007/11560500_21.

[SSM04] Wataru Shinoda, Motoyuki Shiga, and Masuhiro Mikami. Rapid estimation of elastic constants by molecular dynamics simulation under constant stress. *Physical Review B*, 69(13):134103, 2004. doi:10.1103/PhysRevB.69.134103.

[Ter] Quantum code performance comparision. Last accessed 2017-03-12. URL: http://www.petachem.com/performance.html.

[UM09] Ivan S. Ufimtsev and Todd J. Martinez. Quantum chemistry on graphical processing units. 3. analytical energy gradients, geometry optimization, and first principles molecular dynamics. *Journal of Chemical Theory and Computation*, 5(10):2619–2628, 2009. doi:10.1021/ct9003004.

[VBPK14] Dimitrios Vlachakis, Elena Bencurova, Nikitas Papangelopoulos, and Sophia Kossida. Chapter Seven – Current State-of-the-Art Molecular Dynamics Methods and Applications. volume 94 of *Advances in Protein Chemistry and Structural Biology*, pages 269–313. Academic Press, 2014. doi:10.1016/B978-0-12-800168-4.00007-X.

[Wei] Eric W. Weisstein. Full width at half maximum. From MathWorld–A Wolfram Web Resource. Last accessed 2017-03-13. URL: http://mathworld.wolfram.com/FullWidthatHalfMaximum.html.

[WTM+14] Lee-Ping Wang, Alexey Titov, Robert McGibbon, Fang Liu, Vijay S. Pande, and Todd J. Martínez. Discovering chemistry with an ab initio nanoreactor. *Nature chemistry*, 6(12):1044–1048, December 2014. doi:10.1038/nchem.2099.

[YNN+14] Kaoru Yamazaki, Takashi Nakamura, Naoyuki Niitsu, Manabu Kanno, Kiyoshi Ueda, and Hirohiko Kono. Two-step explosion processes of highly charged fullerene cations C_{60}^{q+} ($q = 20$–60). *19th International Conference on Ultrafast Phenomena*, page 07.Mon.P1.21, 2014. doi:10.1364/UP.2014.07.Mon.P1.21.